# Contact Methods in Finite Element Simulations

Netherlands Institute
*for* Metals Research

This research was carried out under project number ME97032 in the framework of the Strategic Research programme of the Netherlands Institute for Metals Research in the Netherlands (www.nimr.nl)

Samenstelling van de promotiecommissie:

| | |
|---|---|
| *voorzitter en secretaris:* | |
| Prof. dr. ir. H.J. Grootenboer | Universiteit Twente |
| *promotor:* | |
| Prof. dr. ir. J. Huétink | Universiteit Twente |
| *leden:* | |
| Dr. R.M.J. van Damme | Universiteit Twente |
| Prof. dr. ir. J.B. Jonker | Universiteit Twente |
| Prof. dr. ir. A. van Keulen | Technische Universiteit Delft |
| Prof. dr. J. Molenaar | Technische Universiteit Eindhoven &<br>Universiteit Twente |
| Prof. dr. ir. D.J. Schipper | Universiteit Twente |

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Kloosterman, Gertjan

Contact Methods in Finite Element Simulations
Gertjan Kloosterman
Proefschrift Enschede. -Met lit. opg. -Met
samenvatting in het Nederlands
ISBN 90-77172-04-1
Trefw.: contact modellen, wrijving, eindige elementen

Printed by Ponsen & Looijen, Wageningen

# CONTACT METHODS IN FINITE ELEMENT METHODS

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. F.A. van Vught,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op vrijdag 20 december 2002 te 15.00 uur.

door

Gertjan Kloosterman

geboren op 5 maart 1973
te Leeuwarden

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. J. Huétink

# Contents

# SUMMARY

The application of finite element methods in forming processes has gained a lot of ground in recent years, especially due to the spectacular advances in the chip industry. Simulating the process with the finite element method offers a producer the advantage of spotting possible production problems early in the development cycle, the ability to optimise the forming process for superior final results and an increased insight into the shaping process in general.

A typical forming process employs one or more tools with which a work piece is deformed into a desired shape, from which we can conclude that there is interaction between different solid components within the process. To model the interactions between the various components, a contact algorithm is required. It is this algorithm which forms the topic of this thesis.

The popularity of the method has sparked a flurry of activity within the finite element community to develop increasingly accurate material models and perform simulations with ever decreasing granularities. The rate of convergence of the simulations has been erratic, which has been often blamed on the specifics of the contact algorithm employed. This brings about the aim of this work: Study the contact problem applied to metal forming simulations, see where it originates, formulate the equations and constraints, discuss the various ways in which to enforce those constraints, and finally how to incorporate it all into a finite element framework. By scrutinising the complete path of development of the equations to the implementation, we intend to identify all the possible pitfalls occurring when contact methods are applied.

The first part of this thesis studies the purely theoretical part, and develops the contact equations in forming, as well as discusses most popular methods to solve it with a sufficient degree of accuracy, stability and efficiency. By retaining a high level of abstractness it is shown that all the popular methods arise from a common form.

The second part of this thesis deals with the practicalities of the methods. This involves computing distances accurately and efficiently and using those to perform the integrations that are required in the contact methods developed in the first part. As it turns out it is this part which is most responsible for the ill behaviour of the algorithm, whereas most of the developments in the literature have addressed the discussion in the first part.

To overcome these difficulties, different integration algorithms are discussed as well as the possibility of overcoming the convergence difficulties introduced by the discretisation by smoothing methods. Additionally a highly efficient projection algorithm is introduced to further contribute to the applicability of contact algorithms. Finally the theory and technicalities are tested by performing a set of numerical experiments.

# SAMENVATTING

De eindige elementen methode in vormgevingsprocessen heeft vele toepassingen gevonden in recente jaren, in het bijzonder dankzij de spectaculaire vooruitgang in de chip-industrie. Het simuleren van een proces met de eindige elementen methode biedt een producent een aantal voordelen: het is mogelijk om product problemen vroeg in de ontwerpfase te herkennen, het is mogelijk het vormgevingsproces te optimaliseren voor superieure eindresultaten en het biedt extra inzicht in het proces in het algemeen.

Een typisch vormgevingsproces gebruikt een of meerdere gereedschappen met welke een werkstuk wordt vervormd tot een gewenste vorm. Hieruit kunnen we concluderen dat er interactie bestaat tussen verschillende componenten binnen het proces. Ten einde de interacties tussen de componenten te simuleren is een contact algoritme nodig. Dit algoritme is het onderwerp van dit proefschrift.

De populariteit van de methode heeft een enorme activiteit binnen de eindige elementen onderzoeksgemeenschap ontketend om steeds nauwkeuriger materiaalmodellen te ontwikkelen en simulaties te draaien met steeds kleiner wordende elementgroottes. De convergentiesnelheid van de simulaties zijn tot nu toe wisselend, wat vaak aan de werking van het contact algoritme werd geweten. Dit brengt ons bij het doel van dit proefschrift: Onderzoek het contactprobleem zoals het wordt toegepast in metaalomvormsimulaties, formuleer de vergelijkingen en beperkingen, bediscussieer de verschillende manieren waarop de beperkingen kunnen worden afgedwongen, en tenslotte onderzoek hoe het geheel opgenomen kan worden in een eindig elementen pakket. Door het gehele pad van afleiden van de vergelijkingen tot de implementatie te onderzoeken stellen we ons tot doel alle mogelijke problemen te identificeren die kunnen optreden als contactmethodes worden gebruikt.

Het eerste deel van dit proefschrift bestudeert het puur theoretisch gedeelte. In dit deel worden de contactvergelijkingen voor vormgevingsprocessen afgeleid en worden de meest populaire methodes om deze vergelijking op te lossen besproken. Belangrijke punten hierbij zijn nauwkeurigheid, stabiliteit en efficiëntie. Door een hoog niveau van abstractie te volgen kan worden aangetoond dat alle populaire methodes een identieke structuur hebben.

Het tweede deel van dit proefschrift behandelt de praktische kant van de methodes. Dit omhelst het nauwkeurig en efficiënt uitrekenen van afstanden om deze vervolgens te gebruiken om de integralen uit te rekenen die nodig zijn om de contact methodes te gebruiken die zijn ontwikkeld in het eerste deel van dit werk. Wat blijkt is dat dit deel vooral

verantwoordelijk is voor het slechte convergentie gedrag van het algoritme, en dat terwijl het meeste werk in de literatuur wordt besteed aan de discussie in het eerste deel.

Om deze problemen de baas te worden, worden verschillende integratie algoritmes beschouwd alswel de mogelijkheid om de convergentieproblemen op te lossen door het gladmaken van de contactoppervlakken. Bovendien ontwikkelen we een zeer efficiënt projectie algoritme om de toepasbaarheid van contact algoritmes verder te vergroten. Tenslotte worden de theorie en de implementatie geïllustreerd aan de hand van een verzameling numerieke experimenten.

# Chapter 1

# INTRODUCTION

## 1.1 Contact in forming

In forming processes, a piece of material called a work piece is forged into a specific shape by employing tools. The formed material is called the work piece, and the manner in which it is supported determines the type of the forming process. To illustrate the idea, let us mention several different methods that are widely used in industry: deep drawing, stretch forming, forging, extrusion and rolling.

The principle of deep drawing is illustrated in Figure 1.1. In the process an initially

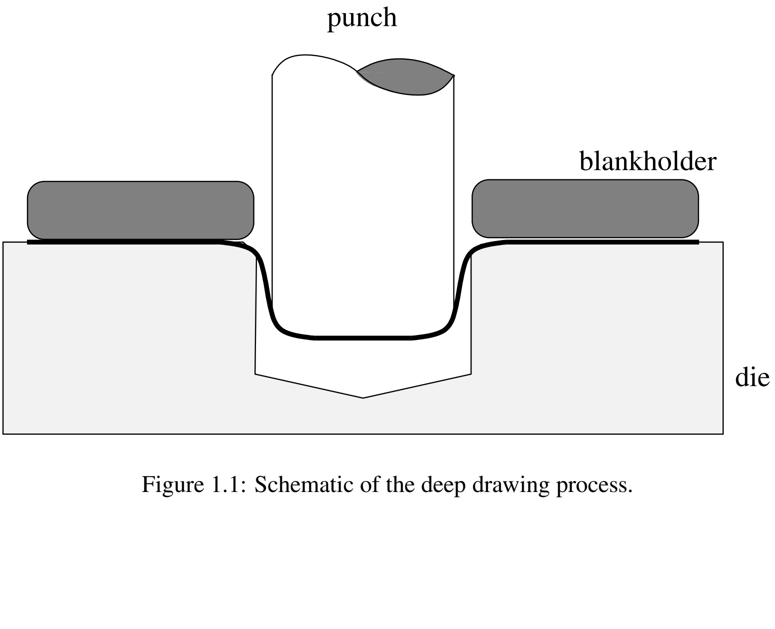


Figure 1.1: Schematic of the deep drawing process.

flat or pre-shaped sheet material called the blank is placed between a blank holder and a die. The blank holder is loaded with a force to prevent wrinkling and to control the amount of material that flows into the die. After that the punch is moved downwards, pushing into the blank, which then partly stretches, and partly pulls material from under the blank holder into the die. The result of the process is a product which contains both the shape of the punch and that of the die. The flow of the material into the die is essential for the formability of the product and its properties.

When the blank holder force is very high, the material cannot draw in from under the blank holder into the die, and consequently the blank only stretches. When the process is applied thus, it is called stretch forming.

Another process is rolling. The schematic for this process is illustrated in Figure 1.2. In the rolling process a slab of material, is deformed between two rolls which deform the slab. Rolling is used to create sheet material, which can subsequently be used in a deep drawing process. Alternative uses are to give the rolls a specific shape across their width. Using the rolling process in this way, is called profile rolling.

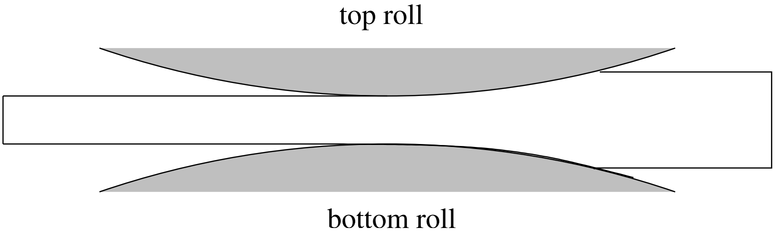


Figure 1.2: Schematic of the rolling process.

All the three aforementioned methods have one thing in common: the forming is achieved trough the interaction between the work piece and the tools. The interaction forces are then the contact forces which consist of normal components which prevent the objects from interpenetrating and tangential components, which are also known as the frictional components. If the forming process hinges on the transfer of these forces from the work piece to the tool, then we need to determine these forces as accurately as possible if we wish to simulate the problem correctly.

## 1.2 The finite element method

Since the advent of the finite element method, its popularity has gradually risen. This is greatly due to advances in the chip industry, which has made it possible to simulate problems with ever increasing complexity. Originally the method was employed for linear problems in strength analyses and safety. Currently, the method has also found a place in industry for cutting costs in the design and development cycles of products.

In a finite element simulation, the workpiece is modelled using a mesh. A mesh consists of elements, which share nodes with one another. The behaviour of the elements can be computed if the locations of its nodes are known. Through this method, a complex object is modelled by dividing it into more simple elementary parts. The resulting model is loaded on the boundary of the mesh, and due to this loading a certain response is observed from the model. Part of the response is the deformation of the mesh. Due to the deformation, parts of the workpiece may come into contact with the tool, or conversely may loose contact. This change of contact status, changes the loading that is applied on the boundary of the mesh. The method that tracks the locations of contact of the workpiece with the tools during the simulation, and accounts for the change in boundary conditions is called the contact algorithm.

## 1.3 Aim of this thesis

The contact problem is present in nearly all forming simulations. Many different methods have been proposed to solve it. A monograph studying the problem from the variational view point is by Kikuchi and Oden (1988). The theory presented there however, is not immediately applicable for implementation in non linear finite element codes.

The topic of how to handle the problem in that setting has been the topic of many articles even in recent years: Givoli and Doukhovni (1996); Bathe and Bouzinov (1997); BittenCourt and Creus (1998); Chabrand et al. (1998); Chawla and Laursen (1998); Chenot and Fourment (1998); Christensen et al. (1998); Chabrand et al. (2001); Czekanski et al. (2001); Farahani et al. (2001) are just a few. One of the first extensive discussions on the topic is by Laursen and Simo (1992, 1993). The accuracy of the contact problem is also under scrutiny, which is clear from Coorevits et al. (2000); Crisfield (2000); El-Abbasi and Bathe (2001); Rieger and Wriggers (2001).

The aim of this thesis is to study the contact problem applied to metal forming simulations. To see where it originates, formulate the equations and constraints, discuss the various ways in which to enforce those constraints, and finally how to incorporate it all into a finite element framework. By scrutinising the complete path of development of the equations to the implementation, we intend to identify all the possible pitfalls occurring when contact methods are applied.

Furthermore, by identifying the possible problems, it becomes possible to avoid them. An additional advantage is that the different methods which are discussed in literature are brought into one unifying theory. Solutions that seem so different from one another at a first glance, turn out to have more in common than would be suspected.

## 1.4 Outline of this thesis

The outline of this thesis follows the path which was sketched above. We start in Chapter 2 with the development of the contact equations and constraints. In Chapter 3 several methods are introduced in the semi-discrete setting to enforce the contact constraints. At this point it can be seen that a lot of the methods proposed in literature come from a common mould. In Chapter 4 the final theoretical part is presented, which is the computation of the discretised contact integral. This all hinges on the computation of distances. As it turns out, most of the problems that can occur in contact pop up at this point. In Chapter 5 several examples and applications are shown for the method. Finally, in Chapter 6 we end the discussion with presenting the conclusions.

# Chapter 2

# CONTACT MECHANICS

## 2.1 Introduction

In this chapter, the equations which characterise the contact problem in forming simulations are deduced. Upon consultation of the literature, one finds that most authors prefer to skip the derivation. This is perfectly reasonable, since the derivation is quite involved and can obscure the ideas that are being proposed. However, in a thesis dedicated to the solution of contact problems in finite element simulations it can not be omitted.

The derivation presented in this chapter is rather rigorous, in the spirit of Laursen and Simo (1993)[1]. The rigor is necessary to underline the validity of the methods employed to solve the contact problem. Apart from correctness of methods, something else is gained by following the rigorous derivation: We can see the common ground in all the ideas which are proposed in the literature to solve the problem. Moreover, several seemingly different ideas can be cast in a single guise by retaining sufficient abstractness.

The purpose of this chapter is to find structure in the description of contact. This structure turns out to follow from work conjugate pair associations. Let us clarify this somewhat. On the one hand there are the spatial quantities, such as displacements, penetration, and sliding distance. Whereas on the other hand there are the force quantities such as stresses and tractions. Typical work conjugate pairs as found in structural mechanics are the different stress-strain pairs. The Cauchy stress tensor is for example work conjugate to the rate of deformation tensor, and equivalently so to the Euler-Almansi strain tensor. The work conjugate pairs are multiplied in the equations to give the dimension of virtual work (or virtual power), hence the phrase ‘work conjugate’.

Through this relationship, the force quantity is characterised by its work conjugate spatial quantity. The characterisation is often called a constitutive relationship. As an example: the stress tensor is assumed to be a function of the rate of deformation. By

[1] Unfortunately the article contains an error, which is discussed in Appendix A.

applying the same ideas to contact, we intend to pair up the contact normal traction with the penetration distance as well as pair up the frictional traction with the slip velocity. In the next chapter the contact tractions are assumed to be a function of their work-conjugate spatial quantities. This approach can be seen as one which regularises the discontinuous nature of the problem.

The derivation requires a background knowledge in continuum mechanics, see for example Malvern (1969); Bonet and Wood (1997). Notational conventions differ among authors. For clarity, we set them at the beginning of the chapter. Apart from continuum mechanics, some differential geometry is employed. The concepts used, are mostly straightforward. For more background information, refer to do Carmo (1976). An important topic is the computation of variational derivatives. For the sake of completeness we discuss the manner in which these computations are performed.

The contact equations are derived twice: once using the principle of virtual work, and once using the principle of virtual power. It turns out that the interpretation of the weighing functions results in different derivation paths, but the results are equivalent. This should obviously hold since the starting point for both derivations is equilibrium of forces. Although only one derivation is sufficient to be able to read the remainder of this thesis, both of the derivations are presented for sake of completeness.

The layout of this chapter follows the line of reasoning above. Before anything can be discussed a notational convention is required, as well as a number of mathematical preliminaries, this is done in Section 2.2. In Section 2.3 the contact kinematics are discussed. Concepts that are introduced comprise the normal distance and interface velocity. In Section 2.4 the contact problem in elasticity is discussed, as a simplified but illustrative example of the general contact problem. In Section 2.5 the characterising constraints on the contact tractions and frictional responses are discussed. Finally, in Section 2.6 the weak form is derived and manipulated to obtain the form in which the force quantities are paired up with their work conjugate spatial quantities. Once with virtual displacements, and once with virtual velocities. In the next chapters algorithms are proposed to solve the resulting set of equations. Finally, in Section 2.7 we present the conclusions.

## 2.2 Notational conventions

In this section the notational conventions are introduced that are used throughout the remainder of this chapter. The notational conventions in this chapter concern the portrayal of properties in the reference configuration and in the current configuration.

The layout of this section is accordingly as follows: In Section 2.2.1, the reference configuration is introduced with its associated property notations. Next in Section 2.2.2, this is repeated for the current configuration. Finally, some additional terminology on differentiation is presented in Section 2.2.3.

### 2.2.1 The reference configuration

The problem under consideration involves the contacting properties of two deformable bodies, whose initial configuration is known. The initial configuration is called the reference configuration (see Figure 2.1). Properties in this configuration are generally written using

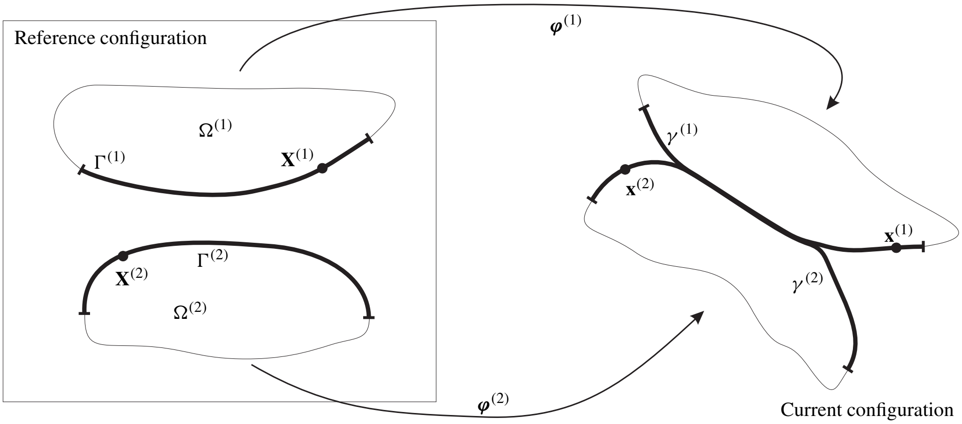


Figure 2.1: Bodies, boundaries and points

capital Roman and Greek letters. In the reference configuration the set of points that make up the $i$-th body is denoted by $\Omega^{(i)}$. The points are elements of $\mathbb{R}^{n_d}$, where $n_d$ denotes the number of space dimensions, which is either 2 or 3.

A typical element of $\Omega^{(i)}$ is denoted by $\mathbf{X}^{(i)}$. Objects containing multiple components, such as vectors and tensors are printed bold, whereas scalar quantities, and the associated length of a vector are not. As an example if **a** is a vector, then “a” is the length of that vector (in the 2-norm), and $a$ is a scalar, which has got nothing do to with the vector **a**.

The boundary of $\Omega^{(i)}$ is denoted by $\partial\Omega^{(i)}$. The portions of $\partial\Omega^{(1)}$ and $\partial\Omega^{(2)}$ where contact may occur are assigned to the sets $\Gamma^{(1)}$ and $\Gamma^{(2)}$ respectively.

### 2.2.2 The current configuration

Properties in the current configuration are generally written with lowercase Roman and Greek letters. A point in the current configuration is denoted by $\mathbf{x}^{(i)}$. It is assumed that there exists an invertible map $\boldsymbol{\varphi}^{(i)}$ that maps each point in the reference configuration onto a point in the current configuration. By assigning a subscript $t$ to these functions, a mapping for each time can be obtained. Or equivalently, we could extend the function $\boldsymbol{\varphi}^{(i)}$ to take an additional argument. Hence, $\boldsymbol{\varphi}_t^{(1)}(\mathbf{X}^{(1)})$ for $t \in [0, T]$ gives the trajectory of a material point $\mathbf{X}^{(1)}$ in $\Omega^{(1)}$. For a particular time $t$, the notations $\boldsymbol{\varphi}_t^{(i)}(\mathbf{X}^{(i)})$, $\boldsymbol{\varphi}^{(i)}(\mathbf{X}^{(i)}, t)$ and $\boldsymbol{\varphi}^{(i)}(\mathbf{X}^{(i)})$ can be used to denote the same thing: The current position of a particular material point. Here it is assumed that the time is fixed and known. In an identical manner, the boundary of either body in the current configuration can be found by applying $\boldsymbol{\varphi}^{(i)}$ to $\Gamma^{(i)}$. The boundary in the current configuration is denoted by $\gamma_t^{(i)} = \boldsymbol{\varphi}_t^{(i)}(\Gamma^{(i)})$. In the previous notation, the subscript $t$ can again be omitted if the time is known.

### 2.2.3 Differentiation and variations

Differentiation with respect to time is denoted by putting a superimposed dot on the corresponding function. Hence, the material velocity of a point can be denoted by

$$\mathbf{v}_t^{(i)}(\mathbf{X}^{(i)}) = \dot{\boldsymbol{\varphi}}_t^{(i)}(\mathbf{X}^{(i)}) = \frac{\mathrm{d}}{\mathrm{d}t}\left[\boldsymbol{\varphi}_t^{(i)}(\mathbf{X}^{(i)})\right]. \tag{2.1}$$

Partial derivatives of a function $f$ are denoted by $f_{,i}$ or $f_{,\alpha}$. The former notation using Roman letters indicates that $i = 1, \dots, n_d$, whereas the latter notation, using Greek letters, indicates that $\alpha = 1, \dots, n_d - 1$. As an example, the contact surface in 3D is a function of two independent parameters. Consequently, each coordinate function of that surface has only two partial derivatives, one less than the number of space dimensions.

This type of indexing is also assumed on indexing of other functions. Thus, writing down $X_i$, using a Roman letter indicates that $i$ runs over $1, \dots, n_d$. Whereas, writing $\xi^\alpha$, assumes that $\alpha = 1, \dots, n_d - 1$. Also note that the superscript does not indicate taking $\xi$ to the power $\alpha$, but merely the $\alpha$-th component in the (contravariant) vector $\boldsymbol{\xi}$.

A functional is a function that takes functions as an argument. As an example of a functional consider the potential energy due to gravitation of a body $\Omega$:

$$P(\boldsymbol{\varphi}) = \int_\Omega \rho g \boldsymbol{\varphi}(\mathbf{X}) \cdot \mathbf{n}_z \, \mathrm{d}\Omega, \tag{2.2}$$

where $\mathbf{n}_z$ is the unit normal pointing upward. The variational derivative of a functional is a directional derivative. It is computed by adding a small variation to the argument of the functional and then taking the derivative (hence, the name variational derivative). As an example, consider a fixed function $\boldsymbol{\varphi}$ and an arbitrary function $\hat{\boldsymbol{\varphi}}$. A perturbation of $\boldsymbol{\varphi}$ in the direction of $\hat{\boldsymbol{\varphi}}$ is denoted by:

$$\boldsymbol{\varphi}_\epsilon = \boldsymbol{\varphi} + \epsilon\hat{\boldsymbol{\varphi}}. \tag{2.3}$$

Entering the perturbed function into a functional $f$ yields a functional dependent on $\epsilon$. Differentiating this functional with respect to $\epsilon$ results in the variational derivative in the direction of $\hat{\boldsymbol{\varphi}}$ at $\boldsymbol{\varphi}$.

Let $D_{\mathbf{a}}[g(\mathbf{x})]$ denote the derivative of g in the direction of $\mathbf{a}$ at the point $\mathbf{x}$. Then taking the variational derivative of a functional $f$, can be denoted by:

$$D_{\hat{\boldsymbol{\varphi}}}[f(\boldsymbol{\varphi})] = \left.\frac{\mathrm{d}}{\mathrm{d}\epsilon}\right|_{\epsilon=0}\left[f(\boldsymbol{\varphi}_\epsilon)\right] = \left.\frac{\mathrm{d}}{\mathrm{d}\epsilon}\right|_{\epsilon=0}\left[f(\boldsymbol{\varphi} + \epsilon\hat{\boldsymbol{\varphi}})\right]. \tag{2.4}$$

As can be concluded from the above equation the variational derivative of a functional is again a functional, this time taking two functions as arguments. In an equivalent manner it is possible to take the directional derivative of a function itself:

$$D_{\hat{\boldsymbol{\varphi}}}[\boldsymbol{\varphi}] = \left.\frac{\mathrm{d}}{\mathrm{d}\epsilon}\right|_{\epsilon=0}\left[\boldsymbol{\varphi}_\epsilon\right] = \left.\frac{\mathrm{d}}{\mathrm{d}\epsilon}\right|_{\epsilon=0}\left[\boldsymbol{\varphi} + \epsilon\hat{\boldsymbol{\varphi}}\right] = \hat{\boldsymbol{\varphi}}. \tag{2.5}$$

Further property notations are introduced in the places where they are required. In the next section a start is made with the description of properties on the (potential) contact surface.

## 2.3 Kinematics of the contact surface

To achieve the characterisation we discussed in the introduction of this chapter, a discussion of the kinematics of contact surfaces is required. To describe the kinematics of the contacting surfaces a coordinate system on the surface is desired. This coordinate system is known as a parametrisation. Consecutive properties are represented with respect to this parametrisation.

The layout of this section is as follows: In Section 2.3.1 parametrisations are discussed. The parametrisations give rise to a local basis, which is the topic of discussion in Section 2.3.2. Finally in Section 2.3.3 the surface quantities we are after are introduced: The normal distance and the sliding velocity.

### 2.3.1 Parametrisation of the contact surface

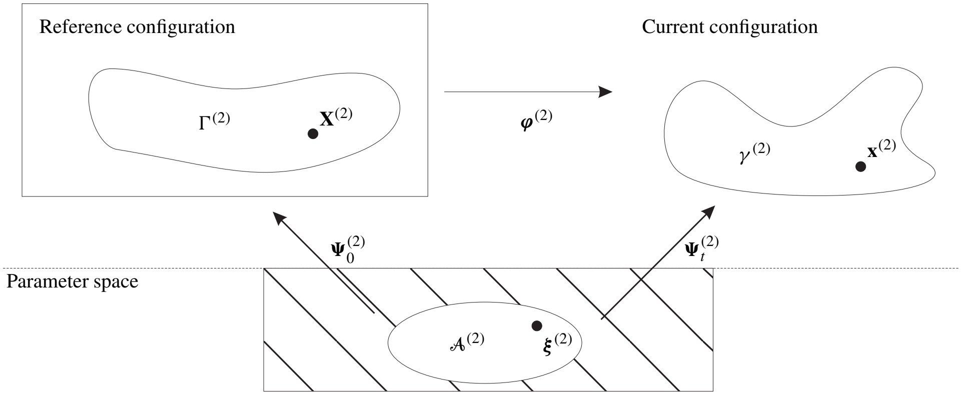


Figure 2.2: Parametrisation of the surface

The dimension of the contact surface $\Gamma^{(i)}$ is one lower than the number of space dimensions. Thus, for a 2D problem, $\Gamma^{(i)}$ is a curve, and for a 3D problem $\Gamma^{(i)}$ is a surface. Furthermore, without loss of generality we assume that $\Gamma^{(i)}$ is simply connected. As a consequence of the connectivity, all the points in $\Gamma^{(i)}$ can be continuously mapped onto a unique point in $\mathbb{R}^{n_d-1}$ and vice versa. The mapping in itself is not unique.

To cast the above reasoning in a formula: call $\mathcal{A}^{(i)}$ the connected set of points mapped to in $\mathbb{R}^{n_d-1}$. Using this notation there exists a continuous and invertible map

$$\boldsymbol{\Psi}_0^{(i)} : \mathcal{A}^{(i)} \subset \mathbb{R}^{n_d-1} \longrightarrow \mathbb{R}^{n_d}, \tag{2.6}$$

such that

$$\Gamma^{(i)} = \boldsymbol{\Psi}_0^{(i)}(\mathcal{A}^{(i)}). \tag{2.7}$$

Furthermore, if $\Gamma^{(i)}$ is sufficiently smooth, $\boldsymbol{\Psi}_0^{(i)}$ can be chosen differentiable. We denote a typical element of $\mathcal{A}$ by $\boldsymbol{\xi}$.

The concepts are illustrated in Figure 2.2. The mapping discussed is known as a *parametrisation* of the contact surface, where $\boldsymbol{\xi}$ is the parameter.

A parametrisation of the contact surface at any later time, can be obtained through applying $\boldsymbol{\varphi}_t^{(i)}$ after $\boldsymbol{\Psi}_0^{(i)}$. This composition of the two functions is given a new name, $\boldsymbol{\Psi}_t^{(i)}$. Thus

$$\begin{aligned}\gamma^{(i)} = \boldsymbol{\varphi}_t^{(i)}(\Gamma^{(i)}) &= \boldsymbol{\varphi}_t^{(i)}\left(\boldsymbol{\Psi}_0^{(i)}(\mathcal{A}^{(i)})\right) \\ &= \left(\boldsymbol{\varphi}_t^{(i)} \circ \boldsymbol{\Psi}_0^{(i)}\right)(\mathcal{A}^{(i)}) = \boldsymbol{\Psi}_t^{(i)}(\mathcal{A}^{(i)}). \qquad (2.8)\end{aligned}$$

Consequently for an arbitrary point on the boundary, we find:

$$\mathbf{X}^{(i)} = \boldsymbol{\Psi}_0^{(i)}(\boldsymbol{\xi}^{(i)}), \qquad (2.9)$$

$$\mathbf{x}^{(i)} = \boldsymbol{\Psi}_t^{(i)}(\boldsymbol{\xi}^{(i)}). \qquad (2.10)$$

## 2.3.2 The local basis

To be able to describe the normal distance function, and express the friction law, a local basis is required. As the first vector in this local basis we take the unit normal vector to the surface. To complete the description of the local basis, $n_d - 1$ additional vectors are required, all of which should be tangential to the surface.

Let us again start in the reference configuration. In this case, vectors tangential to the surface are found by taking the partial derivatives with respect to $\boldsymbol{\xi}$ of $\boldsymbol{\Psi}_0^{(i)}$. In our shorthand notation we obtain $n_d - 1$ tangential vectors, denoted by $\mathbf{T}_\alpha^{(i)}$:

$$\mathbf{T}_\alpha^{(i)} = \boldsymbol{\Psi}_{0,\alpha}^{(i)}(\boldsymbol{\xi}^{(i)}). \qquad (2.11)$$

A tangential vector in the current configuration can be found in an identical manner:

$$\boldsymbol{\tau}_\alpha^{(i)} = \boldsymbol{\Psi}_{t,\alpha}^{(i)}(\boldsymbol{\xi}^{(i)}). \qquad (2.12)$$

We note that $\boldsymbol{\Psi}_t^{(i)}$ is the composition of $\boldsymbol{\varphi}_t^{(i)}$ and $\boldsymbol{\Psi}_0^{(i)}$, upon substituting this in (2.12 and using the chain rule we find:

$$\begin{aligned}\frac{\partial \boldsymbol{\Psi}_t^{(i)}(\boldsymbol{\xi}^{(i)})}{\partial \xi_\alpha^{(i)}} &= \frac{\partial\left(\boldsymbol{\varphi}_t^{(i)} \circ \boldsymbol{\Psi}_0^{(i)}\right)(\boldsymbol{\xi}^{(i)})}{\partial \xi_\alpha^{(i)}} && (2.13a) \\ &= \left.\frac{\partial \boldsymbol{\varphi}^{(i)}(\mathbf{X}^{(i)})}{\partial \mathbf{X}^{(i)}}\right|_{\mathbf{X}^{(i)}=\boldsymbol{\Psi}_0^{(i)}(\boldsymbol{\xi}^{(i)})} \cdot \frac{\partial \boldsymbol{\Psi}_0^{(i)}(\boldsymbol{\xi}^{(i)})}{\partial \xi_\alpha^{(i)}} && (2.13b) \\ &= \mathbf{F}_t^{(i)} \cdot \mathbf{T}_\alpha^{(i)}. && (2.13c)\end{aligned}$$

From this it can be seen, that where $\boldsymbol{\varphi}_t^{(i)}$ maps points from the reference configuration to the current configuration, $\mathbf{F}_t^{(i)}$ maps vectors from the reference configuration to the current configuration. The tensor $\mathbf{F}_t^{(i)}$ is known as the *deformation gradient*.

A remark has to be made about the vectors $\mathbf{T}_\alpha^{(i)}$ and $\boldsymbol{\tau}_\alpha^{(i)}$: They are not necessarily orthogonal with respect to one another, nor will they in general have unit length. Hence, care has to be taken in how quantities are expressed with regard to the local basis.

As an additional remark, if $n_d = 3$, the unit normal vector can be computed by:

$$\mathbf{n}^{(i)} = \frac{\boldsymbol{\tau}_1^{(i)} \times \boldsymbol{\tau}_2^{(i)}}{\|\boldsymbol{\tau}_1^{(i)} \times \boldsymbol{\tau}_2^{(i)}\|}. \tag{2.14}$$

Note that the outer product is dependent on the order of the vectors that appear in it, since $\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}$. As a result, we have to take care in the selection of the orientation of the parametrisation, so that the normal is always pointing outward.

### 2.3.3 Normal distance and interface velocity

Using the local basis defined above, the concepts of normal distance and sliding velocity can be introduced. A last concept that is required is that of the nearest point to a point on the opposing surface. This is known as a projection. The vector from a point to its projection gives the normal distance. The time derivative of the projection point gives a measure for the slip velocity.

To commence, choose $\Omega^{(1)}$ as the slave body and $\Omega^{(2)}$ as the master body. The selection in the continuum setting is completely arbitrary, only when making a discretisation the selection is of relevance. The following treatise is fully equivalent when 1 and 2 are simply interchanged.

The contact surface of the slave body in the current configuration is fully parametrised through the mapping $\boldsymbol{\Psi}_t^{(1)}$ and parameters $\boldsymbol{\xi}^{(1)}$ from the domain $\mathcal{A}^{(i)}$ (see Section 2.3.1). For each value of $\boldsymbol{\xi}^{(1)}$ a point $\mathbf{x}^{(1)}$ on the contact surface of the slave body can be identified.

The projection of $\mathbf{x}^{(1)}$ on $\gamma^{(2)}$ is defined as that point on $\gamma^{(2)}$ which minimises the distance between the point and the boundary. The minimum distance is expressed by:

$$\min_{\mathbf{x}^{(2)} \in \gamma^{(2)}} \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|. \tag{2.15}$$

The point of projection is thus a function of the point to be projected, and is denoted by $\overline{\mathbf{x}}^{(2)}$. Hence, $\overline{\mathbf{x}}^{(2)} \in \gamma^{(2)}$ is that point, which gives the minimum distance

$$\overline{\mathbf{x}}^{(2)} = \arg \min_{\mathbf{x}^{(2)} \in \gamma^{(2)}} \left\| \mathbf{x}^{(1)} - \mathbf{x}^{(2)} \right\|. \tag{2.16}$$

Unfortunately, the point is not necessarily unique. If that is the case, the distance between the bodies is necessarily non zero. We are then free to choose one of the possible projection points.

From now on quantities defined on the master body defined through the slave body are overlined. For a clarification of concepts, refer to Figure 2.3.

The location of $\overline{\mathbf{x}}^{(2)}$ is the map of a point $\boldsymbol{\xi}^{(2)} \in \mathcal{A}^{(2)}$. This point then depends on the deformations of both bodies and $\boldsymbol{\xi}^{(1)}$. Sticking to the notation, where properties on the master boundary that depend on the slave boundary are overlined, we introduce:

$$\overline{\boldsymbol{\xi}}^{(2)} = \overline{\boldsymbol{\xi}}^{(2)}(\boldsymbol{\xi}^{(1)}, \boldsymbol{\varphi}_t^{(1)}, \boldsymbol{\varphi}_t^{(2)}). \tag{2.17}$$

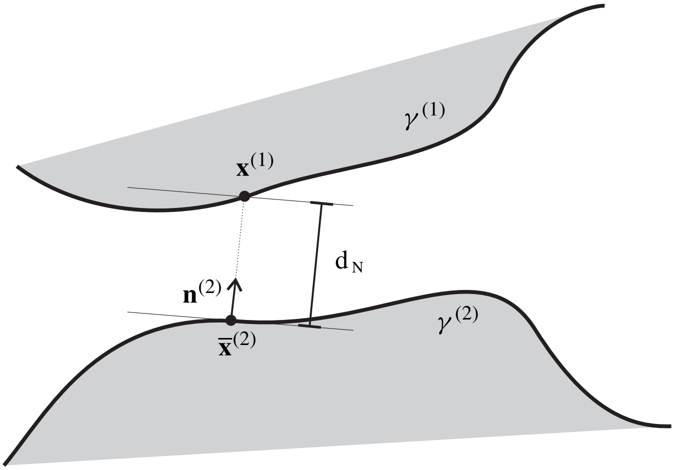


Figure 2.3: Illustration of distance properties.

Thus, we can say that the contact surface of interest is completely parametrised through $\boldsymbol{\xi}^{(1)}$. Parameters $\boldsymbol{\xi}^{(2)}$ that are not addressed through the previous projection mapping can never be in contact with $\partial\Omega^{(1)}$, and as a result can be ignored for this master-slave pair.

The normal distance vector, is expressed using this mapping:

$$\mathbf{d}_\mathrm{N}^{(1)}(\mathbf{x}^{(1)}) = \mathbf{x}^{(1)} - \overline{\mathbf{x}}^{(2)}, \tag{2.18}$$

where

$$\overline{\mathbf{x}}^{(2)} = \boldsymbol{\Psi}_t^{(2)}\left(\overline{\boldsymbol{\xi}}^{(2)}\right). \tag{2.19}$$

The signed normal distance is found by projecting this vector along the outward unit normal vector of the master body:

$$\mathrm{d}_\mathrm{N}^{(1)} = \overline{\mathbf{n}}^{(2)} \cdot \left[\mathbf{x}^{(1)} - \overline{\mathbf{x}}^{(2)}\right]. \tag{2.20}$$

Using this definition, a description for the slip velocity can be found. First it is noted that there can only be a slip velocity at a certain point $\boldsymbol{\xi}^{(1)}$ if the normal distance remains zero, which is expressed through the relationship:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\mathbf{x}^{(1)} - \overline{\mathbf{x}}^{(2)}\right] = \mathbf{0}. \tag{2.21}$$

Let us analyse this expression somewhat further. First, the total time derivative of $\mathbf{x}^{(1)}$ is:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\mathbf{x}^{(1)}\right] = \frac{\mathrm{d}}{\mathrm{d}t}\left[\boldsymbol{\varphi}_t^{(1)}\right] = \mathbf{v}_t^{(1)}. \tag{2.22}$$

This result we already encountered in Section 2.2.3. It follows from the fact that the argument of $\boldsymbol{\varphi}_t^{(1)}$ which is $\mathbf{X}^{(1)}$ or equivalently $\boldsymbol{\Psi}_0^{(1)}(\boldsymbol{\xi}^{(1)})$ does not depend on time.

Secondly, the total time derivative of $\overline{\mathbf{x}}^{(2)}$ is to be computed, which is somewhat more complex:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\left[\overline{\mathbf{x}}^{(2)}\right] &= \frac{\mathrm{d}}{\mathrm{d}t}\left[\boldsymbol{\varphi}_t^{(2)}(\boldsymbol{\Psi}_0^{(2)}(\overline{\boldsymbol{\xi}}^{(2)}))\right] \\
&= \overline{\mathbf{v}}_t^{(2)} + \frac{\partial \boldsymbol{\varphi}^{(2)}}{\partial \mathbf{X}^{(2)}} \cdot \frac{\mathrm{d}}{\mathrm{d}t}\left[\boldsymbol{\Psi}_0^{(2)}(\overline{\xi}^{(2)})\right] \\
&= \overline{\mathbf{v}}_t^{(2)} + \overline{\mathbf{F}}_t^{(2)} \cdot \overline{\boldsymbol{\Psi}}_{0,\alpha}^{(2)} \dot{\xi}^{(2)\alpha} \\
&= \overline{\mathbf{v}}_t^{(2)} + \overline{\boldsymbol{\tau}}_\alpha^{(2)} \dot{\xi}^{(2)\alpha}.
\end{aligned} \tag{2.23}$$

Using the previous result in (2.21), we find for the tangential slip velocity which is the difference in velocity between the contacting points:

$$\mathbf{v}_{\mathrm{T}}^{(1)} = \mathbf{v}_t^{(1)} - \overline{\mathbf{v}}_t^{(2)} = \dot{\xi}^{(2)\alpha}\, \overline{\boldsymbol{\tau}}_\alpha^{(2)}, \tag{2.24}$$

where the bar on the properties indicate that their properties are with respect to the master body.

To simplify the introduction of the contact problem in general deformation problems, we first present a simpler case: Contact in elasticity.

## 2.4 Contact in elasticity

Quite a few problems in physics are characterised as being the solution of the minimisation of a potential function. As an example, we refer back to (2.2), which expresses the potential energy due to gravity. The trajectory $\boldsymbol{\varphi}$ of the solution of (2.2) is such that in the solution it minimises the potential energy with respect to the spatial constraints.

Another example of a problem that is completely characterised through the minimisation of a potential function is that of elastically deforming bodies. In this section we assume that the bodies $\Omega^{(1)}$ and $\Omega^{(2)}$ are strictly hyper-elastic. This means that there exists a potential function which characterises the state of the body. Just as in the case of the gravitational problem, there are geometric constraints on the final solution of the problem. In this case, we only consider prescribed displacements on the final solution and the satisfaction of the impenetrability constraint, which is more fully discussed in the next section.

Let $P^{(i)}(\boldsymbol{\varphi}^{(i)})$ denote the elastic energy in $\Omega^{(i)}$. In this case the problem is fully characterised by the following equations:

$$\min_{\boldsymbol{\varphi}^{(1)},\boldsymbol{\varphi}^{(2)}} \quad P^{(1)}\left(\boldsymbol{\varphi}^{(1)}\right) + P^{(2)}\left(\boldsymbol{\varphi}^{(2)}\right), \tag{2.25a}$$

$$\text{s.t.} \quad \mathrm{d}_{\mathrm{N}}(\mathbf{X}^{(1)}) \geq 0 \text{ on } \Gamma^{(1)}, \tag{2.25b}$$

$$\boldsymbol{\varphi}^{(1)} = \widetilde{\boldsymbol{\varphi}}^{(1)} \text{ on } \Gamma_\varphi^{(1)}, \tag{2.25c}$$

$$\boldsymbol{\varphi}^{(2)} = \widetilde{\boldsymbol{\varphi}}^{(2)} \text{ on } \Gamma_\varphi^{(2)}. \tag{2.25d}$$

In the previous equation $\Gamma_\varphi^{(i)}$ is that part of the boundary where we prescribe displacements. There is a third boundary set apart from $\Gamma^{(i)}$ and $\Gamma_\varphi^{(i)}$ which makes up the total boundary

$\partial\Omega^{(i)}$ on which it is possible to prescribe tractions (forces per area). For now, we assume that this set is empty. We note that the distances only need to be checked for the master body, the other boundary is completely characterised through points on the master body.

For brevity of notation, and to simplify the problem somewhat, we assume that all functions $\boldsymbol{\varphi}^{(i)}$, which are considered satisfy the prescribed boundary conditions. Furthermore, we introduce the composition of functions:

$$\boldsymbol{\varphi}(\mathbf{X}) = \begin{cases} \boldsymbol{\varphi}^{(1)}(\mathbf{X}) & \text{if } X \in \Omega^{(1)} \\ \boldsymbol{\varphi}^{(2)}(\mathbf{X}) & \text{if } X \in \Omega^{(2)} \end{cases}. \tag{2.26}$$

In this case, $P^{(1)}\left(\boldsymbol{\varphi}^{(1)}\right) = P^{(1)}(\boldsymbol{\varphi})$, since the dependency is now implicit, the same holds for $P^{(2)}$. Writing $P = P^{(1)} + P^{(2)}$, we arrive at the following simplified form:

$$\min_{\boldsymbol{\varphi}} \quad P(\boldsymbol{\varphi}) \tag{2.27a}$$

$$\text{s.t.} \quad \mathrm{d}_{\mathrm{N}}(\boldsymbol{\varphi}) \geq 0 \text{ on } \Gamma. \tag{2.27b}$$

In this equation we dropped the superscript from the contact boundary. The problem expressed in (2.27a) has the following Lagrangian:

$$\mathcal{L}(\boldsymbol{\varphi}, \lambda_{\mathrm{N}}) = P(\boldsymbol{\varphi}) + \int_{\Gamma} \lambda_{\mathrm{N}} \mathrm{d}_{\mathrm{N}}(\boldsymbol{\varphi})\ \mathrm{d}\Gamma. \tag{2.28}$$

The solution of (2.27a) is a saddle point of $\mathcal{L}$. The conditions for this saddle point are that:

$$D_{\delta\boldsymbol{\varphi}}[\mathcal{L}(\boldsymbol{\varphi}, \lambda_{\mathrm{N}})] = 0 \quad \text{for all } \delta\boldsymbol{\varphi}, \tag{2.29a}$$

$$\lambda_{\mathrm{N}} \mathrm{d}_{\mathrm{N}} = 0, \tag{2.29b}$$

$$\lambda_{\mathrm{N}} \leq 0, \tag{2.29c}$$

$$\mathrm{d}_{\mathrm{N}} \geq 0. \tag{2.29d}$$

The first equation (2.29a) is telling us that the final solution needs to be a critical point with respect to $\boldsymbol{\varphi}$. Thus that all directional derivatives are zero. The conditions (2.29b)–(2.29d) are known as the Karush-Kuhn-Tucker conditions. They follow from the minimisation problem directly. It is now interesting to note that we *obtain* the KKT conditions from the optimisation problem (2.25a-2.25d), whereas later, we have to put them in explicitly to get to the variational problem. The meaning of these constraints are discussed in Section 2.5.

Expanding (2.29a) leads to the following equation:

$$D_{\delta\boldsymbol{\varphi}}[P(\boldsymbol{\varphi})] + \int_{\Gamma} \lambda_{\mathrm{N}} D_{\delta\boldsymbol{\varphi}}[\mathrm{d}_{\mathrm{N}}(\boldsymbol{\varphi})]\ \mathrm{d}\Gamma = 0, \tag{2.30}$$

or equivalently, rewriting the integral to the current configuration:

$$D_{\delta\boldsymbol{\varphi}}[P(\boldsymbol{\varphi})] + \int_{\gamma} \widehat{\lambda}_{\mathrm{N}} D_{\delta\boldsymbol{\varphi}}[\mathrm{d}_{\mathrm{N}}(\boldsymbol{\varphi})]\ \mathrm{d}\gamma = 0. \tag{2.31}$$

Here the determinant of the Jacobian, which appears when the integration parameter is changed, is multiplied directly into $\lambda_{\mathrm{N}}$ which results in a (different) Lagrange multiplier $\widehat{\lambda}_{\mathrm{N}}$.

In the next section the contact constraints are introduced, which are inserted into the general equations in Section 2.6. From these general equations, we then attempt to regain some of the structure which is evident in (2.31).

## 2.5 The contact constraints

In this section, the contact constraints are discussed. This involves the constraints for impenetrability and friction. These constraints pair the traction in normal and tangential direction with the normal distance and tangential slip respectively. The stress tensor which is used later in this chapter is the Cauchy stress tensor. This stress tensor is sometimes also called the true stress tensor.

The discussion in this section proceeds as follows: First in Section 2.5.1, we introduce the Cauchy stress tensor. Next in Section 2.5.2 the impenetrability constraint is cast into a form using tractions. Finally, in Section 2.5.3 the friction constraints are introduced. The resulting formulations are employed in Section 2.6 for deriving the weak formulation of contact.

### 2.5.1 Stresses and tractions

In this section the concept of stresses and tractions that are required for the contact description are introduced.

**The Cauchy stress tensor**

We start by giving the definition of the “natural” traction. Consider a small piece of surface $\Delta\mathbf{A} \subset \gamma$ (in the current configuration), around a central point $\mathbf{x}$. The area of this section, we denote by $\Delta A$. The sum of forces acting on the surface section equals $\Delta\mathbf{f}$. Then the traction vector (or stress vector) in $\mathbf{x}$ is defined as:

$$\mathbf{t} = \lim_{\Delta A \to 0} \frac{\Delta\mathbf{f}}{\Delta A} = \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}A}. \tag{2.32}$$

A unique tensor $\boldsymbol{\sigma}$ exists, that relates the normal of a surface element to the traction vector for that surface element:

$$\mathbf{t} = \boldsymbol{\sigma} \cdot \mathbf{n}. \tag{2.33}$$

The tensor $\boldsymbol{\sigma}$ is called the natural or Cauchy stress tensor. It is fully defined on the current configuration, and is therefore physically meaningful.

**The traction decomposition**

Let us now define the decomposition of the traction. We repeat the definition of traction at a boundary point $\mathbf{X}^{(i)}$ through the Cauchy stress tensor, $\boldsymbol{\sigma}^{(i)}$:

$$\mathbf{t}^{(i)} = \boldsymbol{\sigma}^{(i)} \cdot \mathbf{n}^{(i)}, \tag{2.34}$$

where $\mathbf{n}^{(i)}$ is the outward normal to the current boundary at $\mathbf{x}^{(i)}$.

The traction is now decomposed in a component normal to the boundary surface and a component tangential to it. The decomposition can be made by using the projection tensor:

$$\mathbb{P}^{(i)} = \mathbf{I} - \mathbf{n}^{(i)} \otimes \mathbf{n}^{(i)}. \tag{2.35}$$

The decomposition is written as:

$$\mathbf{t}^{(i)} = \mathbf{t}^{(i)}_{\mathrm{N}} + \mathbf{t}^{(i)}_{\mathrm{T}}, \tag{2.36}$$

where

$$\begin{aligned}\mathbf{t}^{(i)}_{\mathrm{N}} &= \left(\mathbf{t}^{(i)} \cdot \mathbf{n}^{(i)}\right) \mathbf{n}^{(i)}, \\ \mathbf{t}^{(i)}_{\mathrm{T}} &= \mathbb{P}^{(i)} \cdot \mathbf{t}^{(i)}.\end{aligned}$$

The magnitudes of the respective vectors are denoted by the same symbols, but by using a regular character, such as $\mathrm{t}^{(i)}_{\mathrm{N}}$ and $\mathrm{v}_{\mathrm{N}}$, instead of bold.

## 2.5.2 The impenetrability constraint

In a continuum model, it is not allowed that two points occupy the same location in space. For the interior points within an object, this is taken care of by choosing appropriate candidate functions for the solution of the problem. For multiple bodies this problem reduces to stating that no boundary point of the first body may penetrate the other.

In fact, the whole contact problem can be reduced to a boundary based problem, by demanding that the signed distance of any point on the first body is non-negative with respect to the other, where the signed distance between a point and a boundary is defined as in the previous section.

The normal contact constraints read:

$$\mathrm{d}^{(i)}_{\mathrm{N}} \geq 0, \tag{2.37a}$$

$$\mathrm{t}^{(i)}_{\mathrm{N}} \leq 0, \tag{2.37b}$$

$$\mathrm{t}^{(i)}_{\mathrm{N}} \cdot \mathrm{d}^{(i)}_{\mathrm{N}} = 0. \tag{2.37c}$$

The first condition (2.37a) states that no penetration may occur. Hence, this is the form in which the impenetrability constraint is cast. Using this, the normal traction can be characterised. The second condition (2.37b) states that the contact normal traction should be compressive. Finally, the third condition (2.37c) states a complementarity condition. If there is no contact, then no compressive tractions can occur. Alternatively: If there are no compressive stresses, then the distance must be positive.

These conditions are known as the Karush-Kuhn-Tucker conditions for optimality. For the general case, they are required to be introduced into the weak form of equilibrium. However, if the equilibrium conditions can be derived from a minimum energy principle (as in elasticity), then we get the contact constraints from just demanding impenetrability.

As with the Karush-Kuhn-Tucker constraints for plasticity one additional equation can be added to the previous set, which is the persistency condition, see Simo and Hughes (1998),. If the point remains in contact, then:

$$\mathrm{t}^{(i)}_{\mathrm{N}} \cdot \dot{\mathrm{d}}^{(i)}_{\mathrm{N}} = 0. \tag{2.38}$$

### 2.5.3 Frictional constraints

Without further discussion we continue by presenting the friction constraints:

$$\begin{aligned}
\Phi^{(i)} := \|\mathbf{t}_{\mathrm{T}}^{(i)}\| - \mu\|\mathbf{t}_{\mathrm{N}}^{(i)}\| &\leq 0, && \text{(2.39a)}\\
\mathbf{v}_{\mathrm{T}}^{(i)} + \zeta^{(i)}\mathbf{t}_{\mathrm{T}}^{(i)} &= 0, && \text{(2.39b)}\\
\zeta^{(i)} &\geq 0, && \text{(2.39c)}\\
\Phi^{(i)} \cdot \zeta^{(i)} &= 0. && \text{(2.39d)}
\end{aligned}$$

The first condition (2.39a) states the (Coulomb) friction condition. In the case that $\mu$ is allowed to be a function of velocity and/or pressure, more general frictional laws can be introduced. The second condition (2.39b), together with the third condition (2.39c) constrains the tangential traction to work opposite to the direction of slip. Finally the fourth condition states another complementarity condition: There is no slip if the tangential traction has not reached its (local) maximum. And if there is slip, then it has reached its maximum.

It can be seen here that the friction tractions are complementary to the slip *velocity*. Most finite element methods are concerned with displacement based methods. How this can be dealt with is discussed in the next section.

## 2.6 The weak form of contact

The weak formulation of contact is obtained from the strong formulation of contact by applying the principle of virtual work. The strong form is given by the constraints (2.37a–2.39d) and an equilibrium model. The equilibrium model presented here is a quasi-static one, although no additional difficulty is to be encountered if a dynamic model would be used.

The model is commonly given as:

$$\left\{\begin{aligned}
\nabla \cdot \boldsymbol{\sigma}^{(i)} + \mathbf{f}^{(i)} &= \mathbf{0} && \text{in } \omega,\\
\boldsymbol{\sigma}^{(i)} \cdot \mathbf{n}^{(i)} &= \tilde{\mathbf{t}}^{(i)} && \text{on } \gamma_\sigma^{(i)},\\
\boldsymbol{\varphi}^{(i)} &= \tilde{\boldsymbol{\varphi}}^{(i)} && \text{on } \gamma_\varphi^{(i)}.
\end{aligned}\right. \tag{2.40}$$

In the equilibrium equation $\boldsymbol{\sigma}^{(i)}$ is the Cauchy stress tensor in body $i$ and $\mathbf{f}^{(i)}$ represents the body forces acting on body $i$. There are furthermore boundary conditions prescribed on $\gamma_\sigma^{(i)}$, which is that part of the boundary on which tractions are prescribed. The prescribed tractions are $\tilde{\mathbf{t}}^{(i)}$. Also boundary conditions are prescribed on $\gamma_\varphi^{(i)}$, which is that part of the boundary on which displacements are prescribed. The prescribed displacements are given by $\tilde{\boldsymbol{\varphi}}^{(i)}$.

### 2.6.1 Forming the contact integrals

If the equilibrium equation holds, then upon taking the inner-product with some arbitrary weighing function $\mathbf{w}^{(i)}$, it follows that:

$$\left[\nabla \cdot \boldsymbol{\sigma}^{(i)} + \mathbf{f}^{(i)}\right] \cdot \mathbf{w}^{(i)} = 0. \tag{2.41}$$

To see that this is true, choose for $\mathbf{w}^{(i)}$ consecutively $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ to regain the equilibrium equations.

Integrating the above equation yields:

$$\int_{\omega}\left[\nabla\cdot\boldsymbol{\sigma}^{(i)}+\mathbf{f}^{(i)}\right]\cdot\mathbf{w}^{(i)}\,\mathrm{d}\omega=0. \tag{2.42}$$

Integrating the previous equation by parts by applying the divergence theorem, yields:

$$\begin{aligned}\int_{\omega}\left[\boldsymbol{\sigma}^{(i)}:\nabla\mathbf{w}^{(i)}-\mathbf{f}^{(i)}\cdot\mathbf{w}^{(i)}\right]\mathrm{d}\omega-\int_{\gamma_{\sigma}^{(i)}}\tilde{\mathbf{t}}^{(i)}\cdot\mathbf{w}^{(i)}\,\mathrm{d}\gamma&\\ -\int_{\gamma_{\varphi}^{(i)}}\left[\boldsymbol{\sigma}^{(i)}\cdot\mathbf{n}^{(i)}\right]\cdot\mathbf{w}^{(i)}\,\mathrm{d}\gamma-\int_{\gamma^{(i)}}\mathbf{t}^{(i)}\cdot\mathbf{w}^{(i)}\,\mathrm{d}\gamma&=0.\end{aligned} \tag{2.43}$$

Introducing the compatibility condition $\mathbf{w}^{(i)}=0$ on $\gamma_{\varphi}^{(i)}$, does not change the solution of the original problem. It will however cause the integral over the prescribed displacement boundary to vanish. By using the following shorthand:

$$\begin{aligned}G^{(i)}\left(\boldsymbol{\varphi}^{(i)},\mathbf{w}^{(i)}\right)&=\int_{\omega}\left[\boldsymbol{\sigma}^{(i)}:\nabla\mathbf{w}^{(i)}-\mathbf{f}^{(i)}\cdot\mathbf{w}^{(i)}\right]\mathrm{d}\omega-\int_{\gamma_{\sigma}^{(i)}}\tilde{\mathbf{t}}^{(i)}\cdot\mathbf{w}^{(i)}\,\mathrm{d}\gamma,\\ G_c^{(i)}\left(\boldsymbol{\varphi}^{(i)},\mathbf{w}^{(i)}\right)&=-\int_{\gamma^{(i)}}\mathbf{t}^{(i)}\cdot\mathbf{w}^{(i)}\,\mathrm{d}\gamma,\end{aligned}$$

we can also write

$$G^{(i)}\left(\boldsymbol{\varphi}^{(i)},\mathbf{w}^{(i)}\right)+G_c^{(i)}\left(\boldsymbol{\varphi}^{(i)},\mathbf{w}^{(i)}\right)=0. \tag{2.44}$$

Up until this point, nothing was really done to obtain the work conjugate pairs as was suggested in the introduction. To obtain these, an *interpretation* is to be made on the meaning of $\mathbf{w}^{(i)}$. On the one hand, $\mathbf{w}^{(i)}$ are often interpreted as virtual displacements. If this approach is selected, the work of Laursen and Simo (1993) is followed. The interpretation of the above weak equations is then that of virtual work. On the other hand $\mathbf{w}^{(i)}$ can be interpreted as virtual velocities. In that case one arrives at the principle of virtual power, discussed in Bonet and Wood (1997)[2]. To show the effects on the resulting derivation, both interpretations are used.

In the next section, we discuss the equilibrium of tractions and join the contact integrals. In the two sections following that, the derivation is finalised using first virtual displacements, and then virtual velocities.

### 2.6.2 Equilibrium of tractions

So far, the equilibrium descriptions for both master and slave body were performed separately. In this section they are going to be coupled by an additional piece of information: the equilibrium of tractions. This allows us to have only one integral of the contact boundary, which is more similar in structure to (2.31).

[2] Although the resulting quantity is power, the principle is still called virtual work in the book.

Consider a small area of contact surface d**A** in the current configuration. On such a piece of surface differential contact forces are working: $\mathrm{d}\mathbf{f}^{(i)}$, such that

$$\mathrm{d}\mathbf{f}^{(1)} = -\,\mathrm{d}\bar{\mathbf{f}}^{(2)}. \tag{2.45}$$

This means that

$$\mathbf{t}^{(1)}\,\mathrm{d}\gamma^{(1)} = -\bar{\mathbf{t}}^{(2)}\,\mathrm{d}\overline{\gamma}^{(2)}. \tag{2.46}$$

We are now ready to join the two contact integrals. By doing this a simpler form of the contact constraint is achieved. Currently we have a variational equation for each $i$. What we want is a single equation. If we add all the equations, the following form arises:

$$\sum_i \left[G^{(i)}\left(\boldsymbol{\varphi}^{(i)}, \mathbf{w}^{(i)}\right) + G_c^{(i)}\left(\boldsymbol{\varphi}^{(i)}, \mathbf{w}^{(i)}\right)\right] = 0. \tag{2.47}$$

Or, upon collecting similarly named terms:

$$\sum_i \left[G^{(i)}\left(\boldsymbol{\varphi}^{(i)}, \mathbf{w}^{(i)}\right)\right] + \sum_i \left[G_c^{(i)}\left(\boldsymbol{\varphi}^{(i)}, \mathbf{w}^{(i)}\right)\right] = 0. \tag{2.48}$$

As it turns out, this one equation is just as powerful as all the original equations. This is due to the fact that $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$ can be chosen independently. To see this, filling in $\mathbf{w}^{(2)} = 0$ as a test function yields the equation for $i = 1$. An identical results holds if $\mathbf{w}^{(1)} = 0$ is substituted, in which case the equation for $i = 2$ reappears.

The last equation is rather lengthy, so as in Section 2.4, we introduce a composition of functions to state the same more briefly. The composition is made in the following way:

$$\mathbf{w}(\mathbf{x}) = \begin{cases} \mathbf{w}^{(1)}(\mathbf{x}) & \text{if } x \in \omega^{(1)}, \\ \mathbf{w}^{(2)}(\mathbf{x}) & \text{if } x \in \omega^{(2)}. \end{cases} \tag{2.49}$$

In an identical matter, $\boldsymbol{\varphi}$ is defined as the composition of $\boldsymbol{\varphi}^{(1)}$ and $\boldsymbol{\varphi}^{(2)}$.

In this case, we can write $G^{(i)}(\boldsymbol{\varphi}, \mathbf{w}) = G^{(i)}(\boldsymbol{\varphi}^{(i)}, \mathbf{w}^{(i)})$. Furthermore, we define

$$G(\boldsymbol{\varphi}, \mathbf{w}) = \sum_i G^{(i)}(\boldsymbol{\varphi}, \mathbf{w}), \tag{2.50}$$

$$G_c(\boldsymbol{\varphi}, \mathbf{w}) = \sum_i G_c^{(i)}(\boldsymbol{\varphi}, \mathbf{w}). \tag{2.51}$$

It is the second term in the previous equation which holds our interest. Expanding this term for $i = 1, 2$ leads to a new form for the combined contact integral $G_c$:

$$\begin{aligned} G_c\left(\boldsymbol{\varphi}, \mathbf{w}\right) &= -\int_{\gamma^{(1)}} \mathbf{t}^{(1)} \cdot \mathbf{w}^{(1)}\,\mathrm{d}\gamma - \int_{\gamma^{(2)}} \mathbf{t}^{(2)} \cdot \mathbf{w}^{(2)}\,\mathrm{d}\gamma \\ &= -\int_{\gamma^{(1)}} \mathbf{t}^{(1)} \cdot \left[\mathbf{w}^{(1)} - \overline{\mathbf{w}}^{(2)}\right] \mathrm{d}\gamma. \end{aligned} \tag{2.52}$$

In the combination of the two integrals we used (2.46) As before $\overline{\mathbf{w}}^{(2)}$ is the value of $\mathbf{w}^{(2)}$ at the projection point of $\mathbf{x}^{(1)}$.

In the following two sections, an interpretation is made for the weighing functions $\mathbf{w}$. With an interpretation, we can find some additional structure in the problem. Depending on which interpretation is chosen, different derivations appear. As it turns out, we can recreate the same form for the contact normal distances paired with the contact normal tractions as it appeared in (2.31).

### 2.6.3 Weak form with virtual displacements

In this section, we interpret the functions $\mathbf{w}^{(i)}$ as variations on the solution displacement fields $\boldsymbol{\varphi}^{(i)}$. To exemplify this interpretation, we set:

$$\mathbf{w}^{(i)} = \delta\boldsymbol{\varphi}^{(i)}. \tag{2.53}$$

Hence, a perturbation of $\boldsymbol{\varphi}^{(i)}$ is written as:

$$\boldsymbol{\varphi}_{\epsilon}^{(i)} = \boldsymbol{\varphi}^{(i)} + \epsilon\,\delta\boldsymbol{\varphi}^{(i)}. \tag{2.54}$$

The generality follows from the fact that the perturbations are arbitrary. In particular, for problems that do not include contact we have that:

$$D_{\delta\boldsymbol{\varphi}^{(1)}}[\boldsymbol{\varphi}^{(2)}] = 0 \text{ and } D_{\delta\boldsymbol{\varphi}^{(2)}}[\boldsymbol{\varphi}^{(1)}] = 0, \tag{2.55}$$

which states that the fields $\boldsymbol{\varphi}^{(2)}$ and $\boldsymbol{\varphi}^{(1)}$ are independent.

However, upon using contact, things are different. Most notably, $\boldsymbol{\varphi}^{(2)}$ is dependent on $\boldsymbol{\varphi}^{(1)}$ on the contact boundary. The variations of both functions need to be coupled as a result of this dependency.

#### From derivative of distance to derivative of distance vector

Let us now return to the result in (2.31). In this equation we note the appearance of the term $D_{\delta\boldsymbol{\varphi}}[\mathrm{d}_{\mathrm{N}}]$. It is this term, which we would like to see reappear upon rewriting (2.52), as well as a term involving sliding. The distance function used is the one for the slave body. To free ourselves of tedious superscripting, the notational convenience is made that $\mathrm{d}_{\mathrm{N}} = \mathrm{d}_{\mathrm{N}}^{(1)}$, and in an identical fashion that $\mathbf{d}_{\mathrm{N}} = \mathbf{d}_{\mathrm{N}}^{(1)}$.

Employing the definition of $\mathrm{d}_{\mathrm{N}}$, such as given in (2.20), we arrive at:

$$D_{\delta\boldsymbol{\varphi}}[\mathrm{d}_{\mathrm{N}}] = D_{\delta\boldsymbol{\varphi}}[\overline{\mathbf{n}}^{(2)} \cdot \mathbf{d}_{\mathrm{N}}]. \tag{2.56}$$

Using the product rule on the right hand side, (plus using the the symmetry of the inner-product) leads to:

$$D_{\delta\boldsymbol{\varphi}}[\mathrm{d}_{\mathrm{N}}] = \mathbf{d}_{\mathrm{N}} \cdot D_{\delta\boldsymbol{\varphi}}[\overline{\mathbf{n}}^{(2)}] + \overline{\mathbf{n}}^{(2)} \cdot D_{\delta\boldsymbol{\varphi}}[\mathbf{d}_{\mathrm{N}}]. \tag{2.57}$$

By definition of normals we have that $\overline{\mathbf{n}}^{(2)} \cdot \overline{\mathbf{n}}^{(2)} = 1$. Taking derivatives left and right of this identity, and again using symmetry of the inner-product leads to:

$$\overline{\mathbf{n}}^{(2)} \cdot D_{\delta\boldsymbol{\varphi}}[\overline{\mathbf{n}}^{(2)}] = 0. \tag{2.58}$$

Now expanding $\mathbf{d}_{\mathrm{N}} = \mathrm{d}_{\mathrm{N}}\overline{\mathbf{n}}^{(2)}$ in (2.57), and applying (2.58) leads to the following derivation for the directional derivative of the normal distance:

$$\begin{aligned} D_{\delta\boldsymbol{\varphi}}[\mathrm{d}_{\mathrm{N}}] &= \left(\mathrm{d}_{\mathrm{N}}\overline{\mathbf{n}}^{(2)}\right) \cdot D_{\delta\boldsymbol{\varphi}}[\overline{\mathbf{n}}^{(2)}] + \overline{\mathbf{n}}^{(2)} \cdot D_{\delta\boldsymbol{\varphi}}[\mathbf{d}_{\mathrm{N}}] \\ &= \mathrm{d}_{\mathrm{N}}\left(\overline{\mathbf{n}}^{(2)} \cdot D_{\delta\boldsymbol{\varphi}}[\overline{\mathbf{n}}^{(2)}]\right) + \overline{\mathbf{n}}^{(2)} \cdot D_{\delta\boldsymbol{\varphi}}[\mathbf{d}_{\mathrm{N}}] \\ &= \overline{\mathbf{n}}^{(2)} \cdot D_{\delta\boldsymbol{\varphi}}[\mathbf{d}_{\mathrm{N}}]. \end{aligned} \tag{2.59}$$

### Computing the derivative of the distance vector

The next task we face is to compute $D_{\delta\boldsymbol{\varphi}}[\mathbf{d}_{\mathrm{N}}]$. By definition we have that from (2.20):

$$D_{\delta\boldsymbol{\varphi}}[\mathbf{d}_{\mathrm{N}}] = D_{\delta\boldsymbol{\varphi}}[\mathbf{x}^{(1)} - \overline{\mathbf{x}}^{(2)}]. \tag{2.60}$$

Since, $\mathbf{x}^{(1)}$ is a boundary point, there exists a $\boldsymbol{\xi}^{(1)}$, such that $\mathbf{x}^{(1)} = \boldsymbol{\Psi}_t^{(1)}(\boldsymbol{\xi}^{(1)})$. The point $\overline{\mathbf{x}}^{(2)}$ is then known through the mapping $\boldsymbol{\Psi}_t^{(2)}(\overline{\boldsymbol{\xi}}^{(2)})$. If we interpret $\mathbf{d}_{\mathrm{N}}$ as a function, which gives for each boundary point the distance to the master then this function can also be written as:

$$D_{\delta\boldsymbol{\varphi}}[\mathbf{d}_{\mathrm{N}}] = D_{\delta\boldsymbol{\varphi}}[\boldsymbol{\varphi}^{(1)}] - D_{\delta\boldsymbol{\varphi}}[\overline{\boldsymbol{\varphi}}^{(2)}]. \tag{2.61}$$

The first partial derivative is easy:

$$D_{\delta\boldsymbol{\varphi}}[\boldsymbol{\varphi}^{(1)}] = \delta\boldsymbol{\varphi}^{(1)}. \tag{2.62}$$

This follows straight from the Definition (2.5).

The computation of the second partial derivative is somewhat more involved, and we refer back to the aforementioned definition to compute it.

$$\begin{aligned} D_{\delta\boldsymbol{\varphi}}\left[\overline{\boldsymbol{\varphi}}^{(2)}\right] &= D_{\delta\boldsymbol{\varphi}}\left[\boldsymbol{\varphi}^{(2)}\left(\overline{\mathbf{X}}^{(2)}\right)\right] \\ &= \left.\frac{\mathrm{d}}{\mathrm{d}\epsilon}\right|_{\epsilon=0}\left\{\boldsymbol{\varphi}^{(2)}\left(\overline{\mathbf{X}}^{(2)}\right) + \epsilon\delta\boldsymbol{\varphi}^{(2)}\left(\overline{\mathbf{X}}^{(2)}\right)\right\} \\ &= \overline{\mathbf{F}}^{(2)} \cdot D_{\delta\boldsymbol{\varphi}}[\overline{\mathbf{X}}^{(2)}] + \delta\overline{\boldsymbol{\varphi}}^{(2)}. \end{aligned} \tag{2.63}$$

In the above we used for the directional derivative of $\varphi^{(2)}(\overline{\mathbf{X}}^{(2)})$ the same reasoning as in (2.12) and (2.23).

Remains the computation of $\overline{\mathbf{X}}^{(2)}$, this follows again straight from the definition of $\overline{\mathbf{X}}^{(2)}$:

$$\begin{aligned} D_{\delta\boldsymbol{\varphi}}[\overline{\mathbf{X}}^{(2)}] &= D_{\delta\boldsymbol{\varphi}}[\boldsymbol{\Psi}_0^{(2)}\left(\overline{\boldsymbol{\xi}}^{(2)}\right)] \\ &= \overline{\mathbf{T}}_\alpha^{(2)} D_{\delta\boldsymbol{\varphi}}[\overline{\xi}^{(2)\alpha}]. \end{aligned} \tag{2.64}$$

Combining (2.63) and (2.64) results in:

$$D_{\delta\boldsymbol{\varphi}}\left[\overline{\boldsymbol{\varphi}}^{(2)}\right] = \overline{\boldsymbol{\tau}}_\alpha^{(2)} D_{\delta\boldsymbol{\varphi}}[\overline{\xi}^{(2)\alpha}] + \delta\overline{\boldsymbol{\varphi}}^{(2)}. \tag{2.65}$$

Substituting (2.62) and (2.65) in (2.61) yields the following expression for the directional derivative of the normal distance vector:

$$D_{\delta\boldsymbol{\varphi}}[\mathbf{d}_{\mathrm{N}}] = \delta\boldsymbol{\varphi}^{(1)} - \delta\overline{\boldsymbol{\varphi}}^{(2)} - \overline{\boldsymbol{\tau}}_\alpha^{(2)} D_{\delta\boldsymbol{\varphi}}[\overline{\xi}^{(2)\alpha}]. \tag{2.66}$$

### The contact integral for virtual displacements

We now have to insert the result that was found for the normal distance vector back into the contact integral. Let us first again state (2.52) for the case where $\mathbf{w}$ is interpreted as virtual displacements:

$$G_c\left(\boldsymbol{\varphi}, \delta\boldsymbol{\varphi}\right) = -\int_{\gamma^{(1)}} \mathbf{t}^{(1)} \cdot \left[\delta\boldsymbol{\varphi}^{(1)} - \delta\overline{\boldsymbol{\varphi}}^{(2)}\right] \mathrm{d}\gamma. \tag{2.67}$$

In it we recognise immediately the term $\delta\boldsymbol{\varphi}^{(1)} - \delta\overline{\boldsymbol{\varphi}}^{(2)}$, which is also present in the derivative of the normal distance vector. Making the appropriate substitution of (2.66) in (2.67) results in:

$$G_c\left(\boldsymbol{\varphi}, \delta\boldsymbol{\varphi}\right) = -\int_{\gamma^{(1)}} \mathbf{t}^{(1)} \cdot \left[D_{\delta\boldsymbol{\varphi}}[\mathbf{d}_{\mathrm{N}}] + \overline{\boldsymbol{\tau}}_{\alpha}^{(2)} D_{\delta\boldsymbol{\varphi}}[\overline{\xi}^{(2)\alpha}]\right] \mathrm{d}\gamma. \tag{2.68}$$

Let us split $\mathbf{t}^{(1)}$ into its normal and tangential components:

$$\begin{aligned} \mathbf{t}^{(1)} &= \mathbf{t}_{\mathrm{N}}^{(1)} + \mathbf{t}_{\mathrm{T}}^{(1)} \\ &= \mathrm{t}_{\mathrm{N}}^{(1)}\mathbf{n}^{(1)} + \mathbf{t}_{\mathrm{T}}^{(1)} \\ &= -\mathrm{t}_{\mathrm{N}}^{(1)}\overline{\mathbf{n}}^{(2)} + \mathbf{t}_{\mathrm{T}}^{(1)}. \end{aligned} \tag{2.69}$$

The last step in this equation holds, because only when the bodies are in contact is $\mathrm{t}_{\mathrm{N}}$ non-zero, and do the normals have opposite directions. Substituting the result (2.69) into (2.68) and using (2.59) gives:

$$\begin{aligned} G_c\left(\boldsymbol{\varphi}, \delta\boldsymbol{\varphi}\right) &= -\int_{\gamma^{(1)}} \left[-\mathrm{t}_{\mathrm{N}}^{(1)}\overline{\mathbf{n}}^{(2)} + \mathbf{t}_{\mathrm{T}}^{(1)}\right] \cdot \left[\overline{\mathbf{n}}^{(2)} D_{\delta\boldsymbol{\varphi}}[\mathrm{d}_{\mathrm{N}}] + \overline{\boldsymbol{\tau}}_{\alpha}^{(2)} D_{\delta\boldsymbol{\varphi}}[\overline{\xi}^{(2)\alpha}]\right] \mathrm{d}\gamma \\ &= \int_{\gamma^{(1)}} \mathrm{t}_{\mathrm{N}}^{(1)} D_{\delta\boldsymbol{\varphi}}[\mathrm{d}_{\mathrm{N}}] - \mathbf{t}_{\mathrm{T}}^{(1)} \cdot \overline{\boldsymbol{\tau}}_{\alpha}^{(2)} D_{\delta\boldsymbol{\varphi}}[\overline{\xi}^{(2)\alpha}]\, \mathrm{d}\gamma. \end{aligned} \tag{2.70}$$

In which we used the orthogonality of the tangential vectors and normal vectors. Defining $\mathrm{t}_{\mathrm{T},alpha}^{(1)} = -\mathbf{t}_{\mathrm{T}}^{(1)} \cdot \overline{\boldsymbol{\tau}}_{\alpha}^{(2)}$, finally results in:

$$G_c\left(\boldsymbol{\varphi}, \delta\boldsymbol{\varphi}\right) = \int_{\gamma^{(1)}} \mathrm{t}_{\mathrm{N}}^{(1)} D_{\delta\boldsymbol{\varphi}}[\mathrm{d}_{\mathrm{N}}]\, \mathrm{d}\gamma + \int_{\gamma^{(1)}} \mathrm{t}_{\mathrm{T},\alpha}^{(1)} D_{\delta\boldsymbol{\varphi}}[\overline{\xi}^{(2)\alpha}]\, \mathrm{d}\gamma. \tag{2.71}$$

Or dropping the superscripts and introducing $\delta\mathrm{d}_{\mathrm{N}} = D_{\delta\boldsymbol{\varphi}}[\mathrm{d}_{\mathrm{N}}]$ and for the sliding parameters $\delta\overline{\xi}^{(2)\alpha} = D_{\delta\boldsymbol{\varphi}}[\overline{\xi}^{(2)\alpha}]$, we obtain the more familiar:

$$G_c\left(\boldsymbol{\varphi}, \delta\boldsymbol{\varphi}\right) = \int_{\gamma^{(1)}} \mathrm{t}_{\mathrm{N}}\delta\mathrm{d}_{\mathrm{N}}\, \mathrm{d}\gamma + \int_{\gamma^{(1)}} \mathrm{t}_{\mathrm{T},\alpha}\delta\overline{\xi}^{(2)\alpha}\, \mathrm{d}\gamma. \tag{2.72}$$

Notice the equivalence of the above contact integral for the normal displacements with the result we had for the elastic potential in Section 2.4.

### 2.6.4 Weak form with virtual velocities

In this section, it is assumed that the functions $\mathbf{w}^{(i)}$ are variations on the solution velocity fields. As in the previous section, this distinction is made clear by renaming the weighing functions. Here, we set:

$$\mathbf{w}^{(i)} = \delta \mathbf{v}^{(i)}, \tag{2.73}$$

where $\mathbf{v}^{(i)}$ is the velocity field at the solution point. The velocity field is in the quasi-static problem not a variable.

As was the case with the tractions, the velocity can be decomposed in a normal and a tangential direction, yielding:

$$\mathbf{v}^{(1)} - \overline{\mathbf{v}}^{(2)} = \mathbf{v}_{\mathrm{N}} + \mathbf{v}_{\mathrm{T}}. \tag{2.74}$$

Applying (2.24) and the definition of normal velocity, the previous equation can also be written as:

$$\mathbf{v}_{\mathrm{N}} + \mathbf{v}_{\mathrm{T}} = -\mathrm{v}_{\mathrm{N}} \overline{\mathbf{n}}^{(2)} + \dot{\xi}^{(2)\alpha}\, \overline{\boldsymbol{\tau}}^{(2)}_{\alpha}. \tag{2.75}$$

Taking the directional derivative in the direction of $\delta\mathbf{v}$ of the previous results in:

$$D_{\delta\mathbf{v}}[\mathbf{v}_{\mathrm{N}} + \mathbf{v}_{\mathrm{T}}] = -D_{\delta\mathbf{v}}[\mathrm{v}_{\mathrm{N}}] \overline{\mathbf{n}}^{(2)} + D_{\delta\mathbf{v}}[\dot{\xi}^{(2)\alpha}]\, \overline{\boldsymbol{\tau}}^{(2)}_{\alpha}. \tag{2.76}$$

The derivatives with respect to the velocity for the normal and tangential vectors are $\mathbf{0}$, since they do not depend on the velocity.

Differentiating (2.74) in an identical manner yields

$$D_{\delta\mathbf{v}}[\mathbf{v}_{\mathrm{N}} + \mathbf{v}_{\mathrm{T}}] = \delta\mathbf{v}^{(1)} - \delta\overline{\mathbf{v}}^{(2)}. \tag{2.77}$$

Equating the previous two equations gives:

$$\delta\mathbf{v}^{(1)} - \delta\overline{\mathbf{v}}^{(2)} = -D_{\delta\mathbf{v}}[\mathrm{v}_{\mathrm{N}}] \overline{\mathbf{n}}^{(2)} + D_{\delta\mathbf{v}}[\dot{\xi}^{(2)\alpha}]\, \overline{\boldsymbol{\tau}}^{(2)}_{\alpha}. \tag{2.78}$$

Using (2.69) in (2.52) results in the following contact integral part for the virtual velocities case:

$$G_c\,(\boldsymbol{\varphi}, \delta\mathbf{v}) = \int_{\gamma} \mathrm{t}_{\mathrm{N}}\, D_{\delta\mathbf{v}}[\mathrm{v}_{\mathrm{N}}]\, \mathrm{d}\gamma + \int_{\gamma} \mathrm{t}_{\mathrm{T},\alpha}\, D_{\delta\mathbf{v}}[\dot{\xi}^{(2)\alpha}]\, \mathrm{d}\gamma, \tag{2.79}$$

By doing some renaming:

$$\begin{aligned} \delta\mathrm{v}_{\mathrm{N}} &= D_{\delta\mathbf{v}}[\mathrm{v}_{\mathrm{N}}], & (2.80)\\ \delta\mathbf{v}_{\mathrm{T}} &= D_{\delta\mathbf{v}}[\dot{\xi}^{(2)\alpha}]\, \overline{\boldsymbol{\tau}}^{(2)}_{\alpha}. & (2.81) \end{aligned}$$

We can also write the integral as

$$G_c\,(\boldsymbol{\varphi}, \delta\mathbf{v}) = \int_{\gamma} \mathrm{t}_{\mathrm{N}}\, \delta\mathrm{v}_{\mathrm{N}}\, \mathrm{d}\gamma + \int_{\gamma} \mathbf{t}_{\mathrm{T}} \delta\mathbf{v}_{\mathrm{T}}\, \mathrm{d}\gamma. \tag{2.82}$$

The discussion of the remainder of this thesis is limited to the method based on the virtual displacement. The reasons we have given the virtual velocity based discussion are twofold:

- In a lot of articles, authors prefer the use of virtual velocity based schemes. Therefore it is useful to show that the same type of structure arises even when using virtual velocity schemes.
- To illustrate the impact of interpretation on a mathematical structure. Obviously the complete derivation of the structure is different even though the original equations were completely equivalent. This realisation is important when dealing with interpretation of symbols in any type theory forming.

## 2.7 Conclusions

In this chapter, the derivation was presented for the weak form of equilibrium including contact. The equivalence of the necessary conditions of the contact problem in elasticity and the final one for the virtual displacement case is shown. From this it can be seen that there is a strong coupling between variational problems and the minimisation of functionals. However, the incorporation of Coulomb frictional constraints into the potential minimisation based scheme is theoretically not possible, since minimisation problems always yield a symmetric variational operator. The Coulomb frictional constraints depend on the normal traction, whereas the normal traction does not dependent on the result of the friction. It is for this reason the derivations for the contact problem need to be made in the variational notation.

An advantage of the insight gained by the fact that there exists an association between the variation and minimisation problems, is that we can use methods that solve the contact problem for the minimisation problem and use it for the variation problem. There is a rich body of literature for the former problem, and several algorithms are discussed in the following section.

To solve the contact equations, we need to solve the following variational problem:

$$G(\boldsymbol{\varphi}, \mathbf{w}) + G_c(\boldsymbol{\varphi}, \mathbf{w}) = 0. \tag{2.83}$$

The first functional in this equation can be computed in the usual sense by a finite element method. The second functional contains the contribution of contact. This latter functional can be expanded by interpreting **w** either as virtual displacements or as virtual velocities. For virtual displacements, the result is:

$$G_c\left(\boldsymbol{\varphi}, \delta\boldsymbol{\varphi}\right) = \int_{\gamma^{(1)}} \mathrm{t}_{\mathrm{N}} \delta \mathrm{d}_{\mathrm{N}} \, \mathrm{d}\gamma + \int_{\gamma^{(1)}} \mathrm{t}_{\mathrm{T},\alpha} \delta \bar{\xi}^{(2)\alpha} \, \mathrm{d}\gamma. \tag{2.84}$$

For virtual velocities, the result is:

$$G_c\left(\boldsymbol{\varphi}, \delta\mathbf{v}\right) = \int_{\gamma} \mathrm{t}_{\mathrm{N}} \, \delta \mathrm{v}_{\mathrm{N}} \, \mathrm{d}\gamma + \int_{\gamma} \mathbf{t}_{\mathrm{T}} \delta \mathbf{v}_{\mathrm{T}} \, \mathrm{d}\gamma. \tag{2.85}$$

It is these two integrals that need to be computed to again have a fully defined finite element procedure. In these tow equations, only the values of the tractions are unknown. The computation of these tractions is the topic of the next Chapter.

# Chapter 3

# REGULARISATION METHODS

## 3.1 Introduction

In this chapter, various solution methods are proposed and discussed that can be used to enforce the impenetrability constraint in the finite element framework. We note that the impenetrability constraint is an *inequality constraint*: the signed normal distance between the two contacting boundaries must remain non-negative throughout the simulation.

Most methods employed in the simulation of contact are based upon enforcing *equality constraints*, in which the contacting points are constrained to have exactly zero distance. Since the contact constraints are inequality constraints, a method is required which selects those constraints that are *active*. An inequality constraint is said to be active, if it is on its bound in the solution of the problem.

Unfortunately, there is no simple procedure which is known to always make the correct selection of which constraints are active for a general inequality constrained problem. An active set selection strategy can portend extremely long simulation times if no care is taken, and possibly incorrect final results otherwise. What may go wrong is discussed in more detail later in this chapter.

Though the active set selection problem can be cumbersome at some point in the finite element simulation, in general it is somewhat mollified by taking small time increments. The latter procedure ensures that no large changes in contact status occur during an increment. This in turn means that the active set of constraints will not change too much within an increment.

After application of an active set selection scheme, we obtain an equality constrained problem which can be solved directly by using a mixed method or a constraint elimination method. An alternative procedure is to hide the explicit active set selection process by attempting to estimate the value of the normal traction. In this approach the normal traction, which in fact is a Lagrange multiplier, is assumed to be a function of the value of the signed normal distance. The latter approach is based upon the strong coupling between optimisation problems and variational problems. This type of approach is known as a regularisation, and methods that employ it are discussed in greater detail in this chapter.

This chapter is organised as follows: First the discretisation of smooth contact problems is given. This discretisation is required to explain some of the methods, and also to give an insight as into where problems might occur. In Section 3.3 a short overview is given of the different classes of methods that are currently available to solve the contact problem. Next in Section 3.4, the framework of the regularisation methods is discussed. Subsequently in Sections 3.4.1 till 3.4.3, the penalty method, the method of augmented Lagrangians and the modified barrier method are discussed. Next in Section 3.5 the regularisation of friction is discussed. Finally, in Section 3.6, we give a summary and present some conclusions.

## 3.2 Discretisation

In this section the discretisation is discussed of a single slave body with a non-discretised undeformable master body having a smooth boundary description. The more general case of a discretised slave body with a discretised and possibly deformable master body is postponed until Chapter 4. The reason for introducing this discretisation now, is that some of the advantages and disadvantages of several methods are best discussed in the discrete setting. This is especially true for the mixed and constraint methods. The latter method is even only presented in the discretised setting. The fact that we do not immediately pose the most general discrete form is that it unnecessarily draws the attention to problems and details which are not related to the solution method employed. The result of choosing the discrete problem in the proposed setting, is that the distance functions have the same smoothness as in the continuum setting. In the next chapter we see what happens when this is not the case. As it turns out, this may introduce a bigger problem than the selection of a constraint satisfaction method.

### 3.2.1 Restricting the solution space

In order to consider the discretisation of the contact problem, we require the discretisation of the complete problem. So let us return to the original problem at hand. First, let $\mathcal{S}$ denote the Sobolev space of functions $\boldsymbol{\varphi}$ that contains the solution to (2.83). Finding the solution from the space $\mathcal{S}$ is practically infeasible, and thus we have to be satisfied with an approximation to the actual solution. The approximation to the actual solution is selected from a finite dimensional subspace $\mathcal{S}^h \subset \mathcal{S}$. The superscript gives a measure of the accuracy of the finite dimensional subspace. The smaller the number, the larger the space, the more accurate our approximation will be.

The usual approach in selecting the structure of $\mathcal{S}^h$ is to first mesh the reference configuration $\Omega$ into a collection of *elements* which form an approximation $\Omega^h$ to the original configuration. The superscript $h$ is taken to be the typical size of the elements used in the mesh. An illustration of meshing is given in Figure 3.1.

It is now further assumed that the shape of each of the individual elements in the mesh is completely determined through a finite number of *nodes*. In the example depicted in Figure 3.1, the elements are quadrilaterals, whose shape is completely determined through

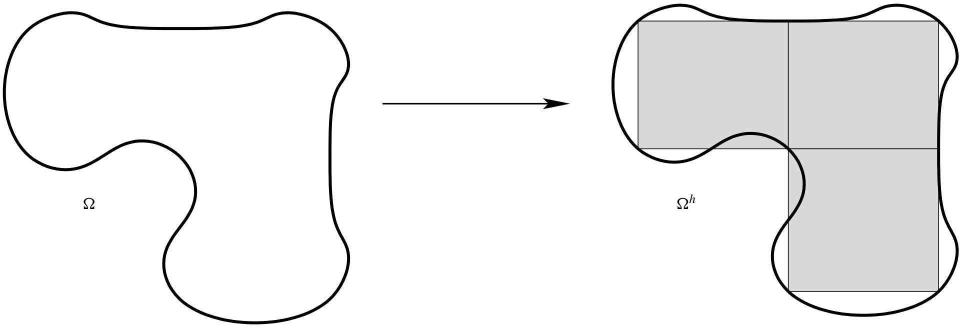


Figure 3.1: Illustration of meshing

the location of its corner points. These corner points are consequently the nodes for the quadrilaterals. Typically, neighbouring elements share nodes, so that their boundary shapes are compatible.

In the mesh the boundary is no longer tracked exactly. It is in itself approximated through the discretised boundary $\Gamma^h$. Hence, not all original reference points $\mathbf{X}$ can be addressed, instead we will refer to points in the approximated reference configuration $\Omega^h$ as $\mathbf{X}^h$.

The members $\boldsymbol{\varphi}^h$ of $\mathcal{S}^h$ are assumed to take the form:

$$\boldsymbol{\varphi}^h(\mathbf{X}^h, t) = \sum_{i=1}^{m} \mathbf{x}_i(t) N_i(\mathbf{X}^h), \tag{3.1}$$

where $\mathbf{x}_i(t)$ is the nodal displacement of node $i$ at time $t$, $m$ is the number of nodes in the discretisation and $N_i$ is the composition of element shape functions (or interpolation functions) for node $i$. For a more complete explanation refer to Hughes (1987); Belytschko et al. (2000). From this equation, we see that the functions $\boldsymbol{\varphi}_t^h$ tracks the evolution of discretised material point $\mathbf{X}^h$, just as $\boldsymbol{\varphi}$ did this for $\mathbf{X}$. Moreover, it is assumed that $\boldsymbol{\varphi}^h(\mathbf{X}^h, 0) = \mathbf{X}^h$.

The functions $\boldsymbol{\varphi}^h$ can also be written in the more usual displacement based formulation as[1]:

$$\begin{aligned}
\boldsymbol{\varphi}^h(\mathbf{X}^h, t) &= \sum_{i=1}^{m} \mathbf{x}_i(t) N_i(\mathbf{X}^h) \\
&= \sum_{i=1}^{m} \mathbf{x}_i(0) N_i(\mathbf{X}^h) + \sum_{i=1}^{m} \mathbf{u}_i(t) N_i(\mathbf{X}^h) \\
&= \mathbf{X}^h + \sum_{i=1}^{m} \mathbf{u}_i(t) N_i(\mathbf{X}^h). \qquad (3.2)
\end{aligned}$$

[1] The notation here is employed for iso-parametric elements

Now introduce the column vector $x$, by stacking the individual points $\mathbf{x}_i$:

$$x = [\mathbf{x}_1 \;\; \mathbf{x}_2 \ldots \mathbf{x}_\mathrm{m}]^\mathrm{T} . \tag{3.3}$$

Each $\boldsymbol{\varphi}^h$ is completely defined through this column vector. Creating an identical type of stacking for the interpolation functions $N_i$ into a matrix $N$ and omitting arguments, we can write:

$$\mathbf{x}^h = \varphi^h(\mathbf{X}^h) = Nx. \tag{3.4}$$

in which

$$N = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & & N_m & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 & \ldots & 0 & N_m & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 & & 0 & 0 & N_m \end{bmatrix}. \tag{3.5}$$

### 3.2.2 The discrete elasticity problem

We now return our attention to (2.27a), which was stated as

$$\begin{aligned} \min_{\boldsymbol{\varphi}} \quad & P(\boldsymbol{\varphi}) \\ \text{s.t.} \quad & \mathrm{d}_\mathrm{N}(\boldsymbol{\varphi}) \geq 0 \text{ on } \Gamma. \end{aligned} \tag{3.6}$$

By inserting the discretisation $\boldsymbol{\varphi}^h$ into the above equation, and using the fact that $x$ is now the real independent argument, we arrive at:

$$\begin{aligned} \min_{x} \quad & P^h(x) \\ \text{s.t.} \quad & \mathrm{d}_\mathrm{N}(x) \geq 0 \text{ on } \Gamma^h. \end{aligned} \tag{3.7}$$

In the above, we also changed the impenetrability condition as a constraint over the original boundary to one over the discretised boundary. There are only a finite number of nodes lying on the boundary, so we can replace the infinite number of constraints with a finite subset of them. The idea being, that if the constraints are satisfied at a number of sufficiently densely distributed points, that they are satisfied (sufficiently accurate) everywhere. The latter property follows from the dependency of the complete boundary on only a finite number of nodes. Each of the selected constraints is given a label $i$, and all labels are collected into a set $I$. Which constraints are chosen is decided by the selection of a boundary integration algorithm: Each integration point corresponds to a constraint. Integration points as constraints is more extensively discussed in Chapter 4. For now, it suffices to say that the number of constraints is reduced to a finite number. Using this observation with (3.7), we arrive at the discretised version of our initial optimisation problem.

$$\begin{aligned} \min_{x} \quad & P^h(x) \\ \text{s.t.} \quad & \mathrm{d}_{\mathrm{N},i}(x) \geq 0 \text{ for } i \in I. \end{aligned} \tag{3.8}$$

This is the same equation as (3.7), with the slight difference that there are now only a finite number of constraints. The minimum in (3.8) is characterised through the following first order conditions, that are also known as the Karush-Kuhn-Tucker or KKT-conditions:

$$\begin{aligned} \nabla_x P^h(x) + \sum_{i \in I} \lambda_{\mathrm{N},i} \nabla_x \mathrm{d}_{\mathrm{N},i} &= 0, & \text{(3.9a)} \\ \lambda_{\mathrm{N},i} \mathrm{d}_{\mathrm{N},i} &= 0, & \text{(3.9b)} \\ \lambda_{\mathrm{N},i} &\leq 0, & \text{(3.9c)} \\ \mathrm{d}_{\mathrm{N},i} &\geq 0. & \text{(3.9d)} \end{aligned}$$

The above conditions specify a saddle-point in the $(x, \lambda_{\mathrm{N}})$ space. For a more extensive survey of these properties the reader is referred to Luenberger (1973) or Bazaraa and Shetty (1979).

We note that the fact that one of the bodies was deformable or not was not used in obtaining the first order conditions in (3.9). Hence, the manner of discretisation is valid for an arbitrary (elastic) problem.

The only influence that the non-deformability has on the formulation as it is stated in (3.7) is on the properties of the distance functions $\mathrm{d}_{\mathrm{N},i}$. If the non-discretised master boundary is sufficiently smooth, the radius of curvature is not too small and if the constrained node is not too far from the boundary, then the distance function is continuously differentiable. Continuous differentiable distance functions are required for the correct theoretical discussion of the methods which we review in this chapter. Later we can show that some relaxation of this assumption is possible.

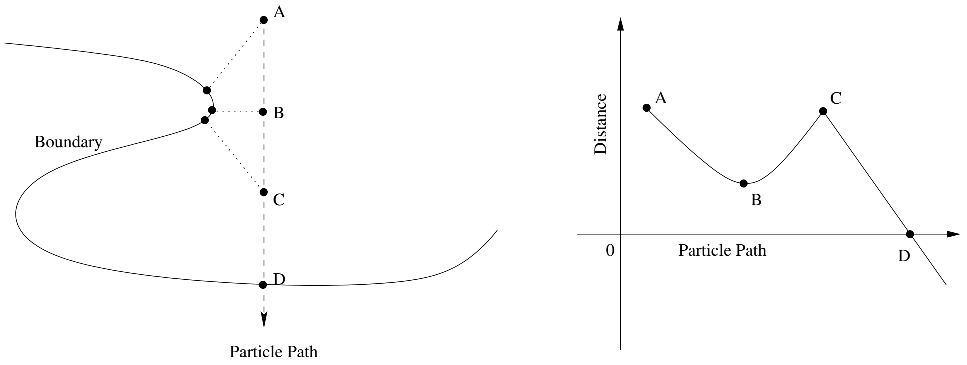


Figure 3.2: Example of a non-smooth distance function with smooth boundary.

An illustration of non-differentiability of a distance function, even when the boundary is smooth is presented in Figure 3.2. As can be seen, the sudden change of nearest projection point causes a non-differentiability of the distance function.

### 3.2.3 The variational problem

Discretising the more general variational problem from Section 2.6, is not really that different from discretising the elastic optimisation problem as was done in Section 3.2.2. It comes down to inserting the selected discretisation into our original problem (2.44). There is one additional point, that does not occur in the optimisation setting, but needs to be addressed in the variational setting is the selection of the test functions $\mathbf{w}$. In (2.43), we inserted the constraint that $\mathbf{w} = 0$ on $\gamma_\varphi$. This requirement needs to be satisfied in the discretised setting.

Note that if we demand

$$G(\boldsymbol{\varphi}^h, \mathbf{w}) + G_c(\boldsymbol{\varphi}^h, \mathbf{w}) = 0, \tag{3.10}$$

for all possible selections $\mathbf{w} \in \mathcal{V}$, that we end up with an overconstrained problem. It is unlikely that $\mathcal{S}^h$ contains the solution of (2.43), at best it contains an accurate approximation.

Consequently, we need to restrict the space $\mathcal{V}$ to a finite dimensional one, so that (3.10) has a unique solution. What is used in most finite element methods is that the discretised space of test-functions $\mathcal{V}^h$ is selected as a subspace of the discretised space of candidate solutions $\mathcal{S}^h$. This results in the Petrov–Galerkin method. If the spaces are selected differently, we end up with the Bubkov–Galerkin method. We only consider the former method.

The only demand that still needs to be met is to find a subspace of $\mathcal{S}^h$ which yields a unique solution. This restriction of $\mathcal{S}^h$ to $\mathcal{V}^h$ is simply setting $w_i = 0$, if $x_i$ is a node that lies on the prescribed displacement boundary $\Gamma^h_\varphi$. The $w_i$ is a component in the stacked vector $w$ which is constructed in an identical fashion as $x$. The indices of the components that are not set to zero are put into the index set $I_{\text{free}}$.

As can be seen from (2.43), both $G$ and $G_c$ are linear in $\mathbf{w}$. Using this fact results in:

$$\begin{aligned} G\left(\boldsymbol{\varphi}^h, \mathbf{w}^h\right) &= G\left(\boldsymbol{\varphi}^h, Nw\right) && (3.11)\\ &= G\left(\boldsymbol{\varphi}^h, N(\sum_{i\in I_{\text{free}}} e_i w_i)\right) && (3.12)\\ &= \sum_{i\in I_{\text{free}}} w_i G\left(\boldsymbol{\varphi}^h, Ne_i\right). && (3.13)\end{aligned}$$

In this equation $e_i$ represents the vector of all zeros, except for a 1 at the $i$-th location. Furthermore, the sum runs over all $i$-s that are not lying on the prescribed displacement boundary. An identical result holds for $G_c$. Thus, we end up with the following equation:

$$\sum_{i\in I_{\text{free}}} w_i \left(G(\boldsymbol{\varphi}^h, Ne_i) + G_c(\boldsymbol{\varphi}^h, Ne_i)\right) = 0. \tag{3.14}$$

Since this is to hold for each $w_i$, we end up with the following vector equation:

$$F_i = G(\boldsymbol{\varphi}^h, Ne_i) + G_c(\boldsymbol{\varphi}^h, Ne_i) = 0, \text{ for } i \in I_{\text{free}}. \tag{3.15}$$

## 3.3 Non regularisation methods

In this section we discuss two of the three main classes of methods that are used in practice. These are the mixed method and the constraint method. In Section 3.3.1 the mixed method is discussed, and after that in Section 3.3.2 the constraint method. The third class of methods, the regularisation methods, is introduced in Section 3.4 and further.

### 3.3.1 The mixed method

The mixed method is in fact a method, which is quite inefficient for implementation in most finite element codes. We merely mention it here as it is the explanation as to why other methods are preferred. The method is more explicitly discussed by Papadopoulos and Solberg (1998). A recent publication which considers smoothing in mixed methods is by Jones and Papadopoulos (2001).

In the mixed method, it is assumed that the normal tractions are true variables. The reasoning is equivalent to that employed when considering incompressible or nearly incompressible problems (see e.g. Hughes, 1987). A Lagrange multiplier for the contact normal traction is introduced. The name mixed method is chosen, since both types of degrees of freedom, spatial and force-like, are used as variables in the problem.

Let us again revert to the elasticity problem, where $P$ denotes the integrated stored energy function over the domain. See for example: Simo and Hughes (1998); Bonet and Wood (1997) and for extensions to plasticity revert to Simo (1988a,b); Simo and Hughes (1998).

Upon introduction of the Lagrange multiplier in (3.8) we arrive at the following form for the Lagrangian of the discretised contact equations in elasticity, which is equivalent in form to (2.31) where the integral over the contact boundary is approximated by a sum:

$$\mathcal{L}(x, \lambda_{\mathrm{N}}) = P(x) + \sum_{i \in I} \lambda_{\mathrm{N}_i} \mathrm{d}_{\mathrm{N},i}(x). \tag{3.16}$$

Suppose we know the set of active constraints $I_a \subset I$. In that case, the solution $(x, \lambda_{\mathrm{N}})$ of this set of equations is a saddle point of the reduced Lagrangian $\mathcal{L}$. The reduction means in this case that the inactive constraints are omitted. The first order conditions for a saddle point are then:

$$\left\{ \begin{array}{rcl} \nabla_x \mathcal{L} & = & 0 \\ \nabla_{\lambda_{\mathrm{N}}^a} \mathcal{L} & = & 0 \end{array} \right. . \tag{3.17}$$

This results in the following equations for this particular situation:

$$\left\{ \begin{array}{rcl} \nabla_x P + \sum_{i \in I_a} \lambda_{\mathrm{N}_i} \nabla_x \mathrm{d}_{\mathrm{N},i}(x) & = & 0 \\ d_{\mathrm{N}_i}(x) & = & 0 \end{array} \right. . \tag{3.18}$$

From (3.18), one can see some structure arising which is typical of mixed methods: loss of positive definiteness; The result not surprising, because the solution to the original problem is a saddle point for the Lagrangian: the solution is located in such a point that it is (locally) a minimum for variations in the displacements, and (locally) a maximum for variations in the Lagrange multiplier. The latter property is retained after discretisation. When we apply a Newton scheme to solve the set of equations, we end up with a stiffness matrix that is in part positive definite and in part negative definite. This type of mixed matrix structure can cause stability problems upon inversion of a discretised step.

Another drawback is that the method introduces a large number of additional unknowns in the discretised setting. Especially in plate forming simulations, where the number of degrees of freedom nearly doubles upon employing a mixed scheme.

Moreover, since the method is based on active inequality constraints, an active set method is required. The selection of the active set can be troublesome and time-consuming, since it is not evident at the beginning of an increment which constraints are going to be active at the end of the increment.

The aforementioned three reasons indicate that employing a mixed method does not seem a wise choice if one wants to find the solution efficiently and in a stable manner.

### 3.3.2 The constraint method

In the constraint method, the contact problem is dealt with by attempting to satisfy the constraint a priori. This is achieved by restricting the field $\delta\boldsymbol{\varphi}^{(2)}$ such, that it is compatible with $\delta\boldsymbol{\varphi}^{(1)}$ in the normal direction on that part of the boundary at which contact occurs. The compatibility restrictions are usually solved by a quadratic programming method. Application of such methods can be found in Givoli and Doukhovni (1996); Klarbring and Björkman (1988); Christensen et al. (1998) and Chabrand et al. (2001).

In general, the fields $\mathbf{w}^{(1)}$ and $\overline{\mathbf{w}}^{(2)}$ such as defined in (2.52), are set to have identical values at those points of the boundary where contact is occurring. This in turn means that the contact integral as defined in (2.52) is identically 0. The constraints are now no longer enforced with a contribution from a Lagrange multiplier. Instead they are eliminated from the system by enforcing compatibility of the displacement fields.

Note again, that here we are only discussing contact normal tractions. If there is friction occurring between the two boundaries, things are more complex.

The constraint method is specified in the discretised setting. When using a constraint method, the first step is again to select those constraints which are active. Call this set $I_a$. Given a current approximation for the solution $x_k$, this solution can be improved to $x_{k+1} = x_k + \Delta x_k$, by solving the approximated system:

$$\begin{array}{ll} \min\limits_{\Delta x_k} & \frac{1}{2}[\Delta x_k]^T[K][\Delta x_k] + [\Delta x_k]^T[F] + P(x_k) \\ \text{s.t} & \mathrm{d}_{\mathrm{N},i}(x_k) + [\nabla \mathrm{d}_{\mathrm{N},i}(x_k)] \cdot \Delta x_k = 0 \text{ for } i \in I_a. \end{array}$$

Here $K$ is the Hessian of $P$ evaluated in $x_k$, and $F$ is the gradient of $P$ evaluated in $x_k$. For the equivalent problem in (2.83) $K$ is the tangential stiffness matrix and $F$ is the residual force vector without contact. In either case $P(x_k)$ can be omitted, since it is a constant for the iteration and has as such no influence on the location of the minimum. The above approximated system contains essentially a second order Taylor series of $P$ around $x_k$ and a first order Taylor series of the active constraints $\mathrm{d}_{\mathrm{N},i}$ around $x_k$.

The above problem is known as a quadratic programming problem (QP). As long as the first order conditions are not sufficiently accurately satisfied, the linearisation procedure needs to be repeated several times. This then is known as sequential quadratic programming (SQP). This approach is very efficient and accurate, for equality constrained problems. Remains the problem of the selecting the active set method, and how it interacts with friction problems.

The application of this method to the general form is completely equivalent to the above. But in that case the matrix $K$ is assumed to be the stiffness matrix, and the vector $F$ is the residual vector without contact.

The solution of the SQP problem can be achieved in various ways, one could employ pivoting methods such as Lemke's method, see Chabrand et al. (2001). Or the constraints can be immediately eliminated from the system to leave a reduced system. The latter is the method used in the finite element code MARC, also by Farahani et al. (2000, 2001).

After having dealt with the mixed method and the constraint method, as well as having glimpsed the connection between optimisation problems and variational problems yet again, we set out in the next sections with a discussion on regularisation methods.

## 3.4 The regularisation framework

The third type of method that we discuss is the regularisation method. In this method, we neither introduce the Lagrange multipliers explicitly, nor do we eliminate the constraints from the system explicitly, since we do not want to revert to an active set method. Instead we assume that the tractions are a function of their work-conjugate spatial quantity: the normal distance. Moreover, the methods can be presented naturally in the continuum formulation.

The result is a solution to the contact problem that allows (small) violations of the contact constraints in order to estimate the direction and magnitude of the actual tractions. The heuristic behind this approach is that larger penetrations require larger normal forces to separate the contacting bodies.

From a mathematical point of view, penalty and augmented Lagrangian method are optimisation techniques. As a consequence the most natural explanation occurs from the optimisation viewpoint. Following this observation, we introduce the methods for elasticity problems. For these type of problems, the first order conditions are identical to the variational equations. From the first order conditions of the elasticity problem, the terms involving contact can be obtained. By adding these terms to the general variational problem which does not necessarily have an associated potential we obtain a solution method for the contact problem in inelasticity.

Let us once again state the optimisation problem:

$$\begin{array}{rl} \min_{\boldsymbol{\varphi}} & P(\boldsymbol{\varphi}) \\ \text{s.t.} & \mathrm{d}_{\mathrm{N}}(\boldsymbol{\varphi}(\mathbf{X})) \geq 0 \text{ for all } \mathbf{X} \in \Gamma. \end{array} \tag{3.19}$$

If we would only have a problem of the form $\min f(\boldsymbol{\varphi})$, then (the discretised version) could be solved by employing an unconstrained local search strategy such as a Newton-Raphson or a quasi-Newton method. Under the assumption that we can solve unconstrained problems, we convert (3.19) into a sequence of unconstrained optimisation problems whose limit is the solution $\boldsymbol{\varphi}^*$ of the original problem.

The form we are seeking can be obtained by appending to the objective $P$ a function $\Xi$ which penalises the (near) violation of a constraint in a single point. Typically such a function employs a set of parameters $\mathbf{q}$ at each constrained point, which modifies the function in such a way as to obtain an exact penalisation. If an exact penalisation could be obtained, then the solution of the extended potential would yield the solution of (3.19). Exact interpretations of how these parameters are supposed to work seem quite abstract now, but each of the methods discussed in the following sections gives an example which will make things more clear.

The extended unconstrained problem now takes the form:

$$\min_{\boldsymbol{\varphi}} P\left(\boldsymbol{\varphi}\right) + \int_{\Gamma} \Xi\left(\mathrm{d}_{\mathrm{N}}(\boldsymbol{\varphi}(\mathbf{X}); \mathbf{q}(\mathbf{X})\right)\ \mathrm{d}\Gamma. \tag{3.20}$$

The first order constraints of this function are then:

$$D_{\delta\boldsymbol{\varphi}}[P(\boldsymbol{\varphi})] + \int_{\Gamma} \frac{\partial\,\Xi(\mathrm{d}_{\mathrm{N}}; q)}{\partial \mathrm{d}_{\mathrm{N}}} D_{\delta\boldsymbol{\varphi}}[\mathrm{d}_{\mathrm{N}}(\boldsymbol{\varphi})]\ \mathrm{d}\Gamma = 0 \ \ \forall\, \delta\boldsymbol{\varphi} \in \mathcal{V}. \tag{3.21}$$

The space $\mathcal{V}$ is the space of allowable variations, which is basically identical in structure to $\mathcal{S}$. The difference is that for each element $\mathbf{w} \in \mathcal{V}$ we demand that $\mathbf{w}(\mathbf{X}) = 0$ for $\mathbf{X} \in \Gamma_{\varphi}$. Thus, it equals 0 on the prescribed displacement boundary.

Comparing (3.21) with (3.18) we see that the partial derivative of $\Xi$ with respect to the distance corresponds to the Lagrange multiplier $\lambda_{\mathrm{N}}$. This fact is later exploited to find better penalisation techniques.

Moreover, comparing (3.21) with (2.83) we notice that the same partial derivative takes the place of the contact tractions. Hence, any method which employs a penalisation approach by an arbitrary function $\Xi$ makes estimates of the contact traction via the derivative of $\Xi$. In other words, we make the regularisation

$$\mathrm{t}_{\mathrm{N}}(\mathrm{d}_{\mathrm{N}}, \mathbf{q}) = \frac{\partial\,\Xi(\mathrm{d}_{\mathrm{N}}; \{q\})}{\partial \mathrm{d}_{\mathrm{N}}}, \tag{3.22}$$

where $\mathrm{t}_{\mathrm{N}}$ now no longer is a variable, but assumed to be functionally defined through its work-conjugate spatial quantity: the normal distance.

Plugging the result back into (2.83), and restricting us to the normal distance, results in the regularised contact contribution:

$$G_c\left(\boldsymbol{\varphi}, \delta\boldsymbol{\varphi}\right) = \int_{\Gamma} \mathrm{t}_{\mathrm{N}}(\mathrm{d}_{\mathrm{N}}, \mathbf{q})\, D_{\delta\boldsymbol{\varphi}}[\mathrm{d}_{\mathrm{N}}]\ \mathrm{d}\Gamma. \tag{3.23}$$

In the following sections, we introduce various methods from the optimisation literature which have found their way into applications in contact mechanics. The penalty method and method of augmented Lagrangians are the most widely applied. We propose the modified barrier and smooth penalty methods to overcome some of the problems that are occurring when equality constrained based methods such as the penalty method are used. Another method which was proposed in Zavarise et al. (1998), which also falls into this class of methods, is not further discussed, since no further applications of it are known in the literature, but it serves to illustrate that the framework can lead to a plethora of methods.

### 3.4.1 The penalty method

The easiest, and perhaps earliest method which was and still is employed in the solution of contact problems is the penalty method. The original penalty method is a method which is employed in the solution of equality constrained optimisation problems. An extensive discussion of this type of method can be found in Fiacco and McCormick (1968). An introduction to this type of method can be found in Bazaraa and Shetty (1979); McCormick (1983). Application of the method with respect to contact mechanics can be found in a wide variety of articles, comprising Chenot and Fourment (1998); Shimizu and Sano (1995).

The penalty method was originally intended for equality constrained problems, i.e. the constraint $\mathrm{d}_\mathrm{N} \geq 0$ needs to be converted to a form $f(\mathrm{d}_\mathrm{N}) = 0$, in a way that if $\mathrm{d}_\mathrm{N}$ is positive that $f$ is 0, and $f(\mathrm{d}_\mathrm{N})$ is non-zero otherwise. This is achieved by using the Macaulay bracket. The Macaulay bracket is a function that is defined as:

$$\langle x \rangle = \begin{cases} 0 & \text{if } x < 0, \\ x & \text{if } x \geq 0. \end{cases} \tag{3.24}$$

Using this definition of the constraint we have:

$$x \geq 0 \Longleftrightarrow \langle -x \rangle = 0. \tag{3.25}$$

The above conversion comes at a cost, however, since the Macaulay bracket is not differentiable at $x = 0$. The non-differentiability can have serious consequences for the convergence of a Newton process that is used to solve the discretised problem.

The function $\Xi$ for the penalty problem in contact mechanics can now be defined as:

$$\Xi(\mathrm{d}_\mathrm{N}; \{p\}) = \frac{p}{2} \langle -\mathrm{d}_\mathrm{N} \rangle^2, \tag{3.26}$$

in which the only parameter used is a penalty value $p$. This is supposed to be a large number, such that when the constraint value is violated it gives a significant contribution to the extended unconstrained problem. Consequently the larger the penalty value is, the smaller the constraint value ought to be for the solution of the problem (3.21). The derivative of $\Xi$ with respect to $\mathrm{d}_\mathrm{N}$ is:

$$\frac{\partial \Xi}{\partial \mathrm{d}_\mathrm{N}} = \frac{p}{2} 2 \langle -\mathrm{d}_\mathrm{N} \rangle H(-\mathrm{d}_\mathrm{N}) \cdot (-1) = -p \langle -\mathrm{d}_\mathrm{N} \rangle. \tag{3.27}$$

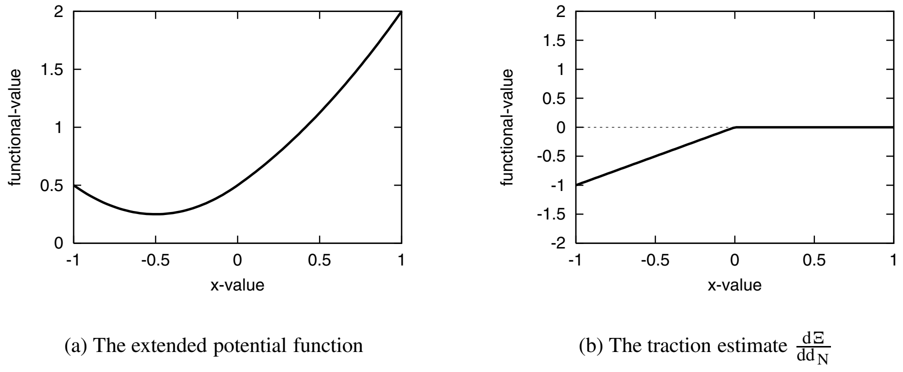
In the equation above $H(x)$ represents the Heaviside function, which is defined as:

$$H(x) = \begin{cases} 0 & \text{if } x < 0, \\ 1 & \text{if } x \geq 0. \end{cases} \tag{3.28}$$

The result now follows from using the chain rule and the fact that $H(x) \cdot \langle x \rangle = \langle x \rangle$.

The result for the normal traction in this case is:

$$\mathrm{t}_\mathrm{N}(\mathrm{d}_\mathrm{N}, p) = \frac{\partial \Xi(\mathrm{d}_\mathrm{N}; \{p\})}{\partial \mathrm{d}_\mathrm{N}} = -p \langle -\mathrm{d}_\mathrm{N} \rangle. \tag{3.29}$$



(a) The extended potential function

(b) The traction estimate $\frac{\mathrm{d}\Xi}{\mathrm{d}d_\mathrm{N}}$

Figure 3.3: Illustration of the penalty functional

For an example take a look at Figure 3.3. In this figure the penalty functional is given for the following problem:

$$\min_x \quad \frac{1}{2}(x+1)^2 \tag{3.30a}$$

$$\text{s.t.} \quad x \geq 0. \tag{3.30b}$$

The extended penalty functional for the above problem is:

$$\frac{1}{2}(x+1)^2 + \frac{p}{2}\langle -x\rangle^2. \tag{3.31}$$

In Figure 3.3 the penalty is chosen as 1. Also drawn in the figure is the traction estimate function, which is the derivative of $\Xi$ with respect to $d_\mathrm{N}$. The minimum can be seen to lie at $x = -\frac{1}{2}$, with an estimate for the Lagrange multiplier of $\lambda = -\frac{1}{2}$. In Table 3.4.1 a set of penalty values is presented versus the convergence of $x$ for the example.

| | p = 1 | p = 100 | p = 1000 | p = 10000 |
|---|---|---|---|---|
| $x$ | -0.50000000 | -0.90909090 | -0.00990099 | -0.00099900 |
| $\lambda$ | -0.50000000 | -0.90909090 | -0.99009900 | -0.99900099 |

Table 3.1: Convergence of the example problem with the penalty method.

Although the method is conceptually simple, there are a number of drawbacks to it. First of all there is the notable introduction of a non-differentiability upon contact. The second problem is that only for the penalty $p$ going to infinity can the constraint violations be eliminated. The final problem is that the penalty $p$ can not be too large because this worsens the condition number of the matrix that is obtained in the numerical solution of the problem. The last two problems counteract, and for a successful application some fiddling with the penalty parameter has to undertaken.

In the next section, the method of augmented Lagrangians is discussed, which attempts to solve the contact problem accurately, while keeping the penalty parameter small.

### 3.4.2 The method of augmented Lagrangians

To understand the method of augmented Lagrangians, we refer to the solution of (2.27a) which was characterised through the first order conditions:

$$D_{\delta\boldsymbol{\varphi}}[P(\boldsymbol{\varphi})] + \int_{\Gamma} \lambda_{\mathrm{N}} D_{\delta\boldsymbol{\varphi}}[\mathrm{d}_{\mathrm{N}}]\,\mathrm{d}\Gamma = 0. \tag{3.32}$$

The solution which satisfies (3.32) is denoted by $(\boldsymbol{\varphi}^*, \lambda^*_{\mathrm{N}})$. Here $\lambda^*_{\mathrm{N}}$ is the Lagrange multiplier that is equivalent to the normal traction at the solution point.

From the penalty method, employing (3.22), we already learned that $-p\langle -\mathrm{d}_{\mathrm{N}}\rangle$ at the solution of the regularised problem, gives an estimate for $\mathrm{t}_{\mathrm{N}}$ from (3.29). If we could somehow improve our estimate of the Lagrange multiplier using this previous value, then it would improve the accuracy in a subsequent minimisation problem.

This can be done by choosing $\Xi$ to contain part of the actual Lagrangian, but keeping the Lagrange multipliers fixed. The function $\Xi$ now takes the following form:

$$\Xi(\mathrm{d}_{\mathrm{N}}; \{p, \lambda_{\mathrm{N}}\}) = \frac{1}{2p}\left\langle -(\lambda_{\mathrm{N}} + p\mathrm{d}_{\mathrm{N}})\right\rangle^2, \tag{3.33}$$

where we stress that in this case the $\lambda_{\mathrm{N}}$ is not a variable, but a constant. The complexity of the form of the regularisation in (3.33) arises from the inequality-equality constraint conversion. However, it is not that different from the penalty functional, which could have been written as $\frac{p}{2}\langle -\mathrm{d}_{\mathrm{N}}\rangle^2 = \frac{1}{2p}\langle -p\mathrm{d}_{\mathrm{N}}\rangle^2$. From this equation we can see that in the penalty method the Lagrange multiplier is approximated as 0.

The resulting traction can be expressed as the derivative with respect to $\mathrm{d}_{\mathrm{N}}$ of the previous equation:

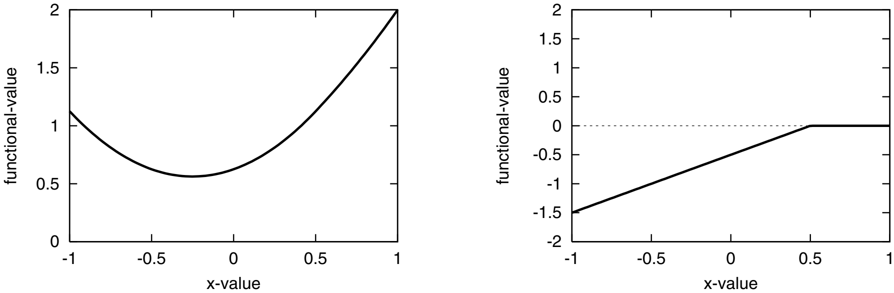
$$\mathrm{t}_{\mathrm{N}}(\mathrm{d}_{\mathrm{N}}(\mathbf{X}), \{p, \lambda_{\mathrm{N}}(\mathbf{X}\}) = -\left\langle -\left(\lambda_{\mathrm{N}}(\mathbf{X}) + p\mathrm{d}_{\mathrm{N}}(\mathbf{X})\right)\right\rangle, \tag{3.34}$$

Or dropping the arguments of all the functions, more clearly as:

$$\mathrm{t}_{\mathrm{N}} = -\left\langle -(\lambda_{\mathrm{N}} + p\mathrm{d}_{\mathrm{N}})\right\rangle. \tag{3.35}$$

If the estimate of $\lambda_{\mathrm{N}}$ would be the correct Lagrange multiplier $\lambda^*_{\mathrm{N}}$, then the result after minimisation of (3.20) would yield the solution of (3.6). However, in general we do not have the correct values of $\lambda_{\mathrm{N}}$. After solving (3.20) with the values of $\lambda_{\mathrm{N}}$ set to 0, will result in an estimate to the solution. Let us call this result $\boldsymbol{\varphi}^1$ (not to be confused with the field $\boldsymbol{\varphi}$ restricted to $\Omega^{(1)}$, which is denoted by $\boldsymbol{\varphi}^{(1)}$). After completion of the computation, we also have an approximation $\mathrm{t}_{\mathrm{N}}(\mathrm{d}_{\mathrm{N}}(\boldsymbol{\varphi}^1))$ to the correct multipliers $\lambda^*_{\mathrm{N}}$. Let us name this approximation $\lambda^1_{\mathrm{N}}$. So upon completion we have an approximation $(\boldsymbol{\varphi}^1, \lambda^1_{\mathrm{N}})$ to the correct solution of (3.6).

The values $\lambda^1_{\mathrm{N}}$ are better estimates to the optimal Lagrange multipliers than the values $\lambda^0_{\mathrm{N}}$, which were set to 0. That this is so, can be deduced as follows. If the distance for a particular point $\mathbf{X}$ is positive, then from (3.34), we can see that $\mathrm{t}_{\mathrm{N}}$ remains 0. Thus in



(a) The extended potential function

(b) The traction estimate $\frac{\mathrm{d}\Xi}{\mathrm{d}d_{\mathrm{N}}}$

Figure 3.4: Illustration of the augmented Lagrangian functional

that case, $\lambda^1_{\mathrm{N}} = \lambda^0_{\mathrm{N}} = 0$. However, if there is penetration then obviously the Lagrange multiplier is chosen too small. At the converged solution, the Lagrange multiplier is larger in magnitude, but still too small, since there will still be penetration. Hence, the result of restarting the minimisation process, only now with using $\lambda^1_{\mathrm{N}}$ as the estimating constants will yield an better approximation. Name the new results $(\boldsymbol{\varphi}^2, \lambda^2_{\mathrm{N}})$. After finishing this, $\lambda^2_{\mathrm{N}}$ would again form a better value, etc. The process of updating the $\lambda_{\mathrm{N}}$ with better estimates is called augmentation. We augment the estimation to the Lagrangian multipliers, which explains the name. Indeed for equality constrained problems, the procedure converges to the exact solution, see for example Bazaraa and Shetty (1979); Bertsekas (1982). There are many articles written on the application of the augmented Lagrangian method in the finite element method, among them are Laursen and Simo (1992, 1993); Simo and Laursen (1992); Zavarise et al. (1995); Refaat and Meguid (1997); Pietrzak and Curnier (1999).

An illustration of the process is given in Figure 3.4. Again the problem that is used for the illustration is given by:

$$\min_{x} \quad \frac{1}{2}(x+1)^2 \tag{3.36a}$$

$$\text{s.t.} \quad x \geq 0. \tag{3.36b}$$

The extended penalty functional for the above problem is:

$$\frac{1}{2}(x+1)^2 + \frac{1}{2p}\langle -(\lambda + px)\rangle^2. \tag{3.37}$$

Using again a penalty $p$ of 1, and the estimate for the Lagrange multiplier that was obtained from the penalty method as $-\frac{1}{2}$, the result as in Figure 3.4 is obtained as the functional to minimise. From the picture one can see that the location of the minimum of the extended potential is a better estimation of the actual minimum which is lying at 0 than the initial value of $-\frac{1}{2}$ that was obtained from the penalty method. The value that is obtained after one augmentation is $x = -\frac{1}{4}$. On the right in Figure 3.4 the estimate for the augmented Lagrangian method is shown for $\lambda = -\frac{1}{2}$. In Table 3.4.2, the convergence of the example problem is given for several different penalties. In each case, the initial value for the Lagrange multiplier $\lambda^{(0)}$ is $\frac{1}{2}$.

|  | p = 1 | p = 100 | p = 1000 | p = 10000 |
|---|---|---|---|---|
| $x^{(1)}$ | $-2.500 \cdot 10^{-1}$ | $-4.545 \cdot 10^{-2}$ | $-4.950 \cdot 10^{-3}$ | $-4.995 \cdot 10^{-4}$ |
| $x^{(2)}$ | $-1.250 \cdot 10^{-1}$ | $-4.132 \cdot 10^{-3}$ | $-4.901 \cdot 10^{-5}$ | $-4.990 \cdot 10^{-7}$ |
| $x^{(3)}$ | $-6.250 \cdot 10^{-2}$ | $-3.756 \cdot 10^{-4}$ | $-4.852 \cdot 10^{-7}$ | $-4.985 \cdot 10^{-10}$ |
| $\lambda^{(1)}$ | −0.75000000 | −0.95454545 | −0.99504950 | −0.99950049 |
| $\lambda^{(2)}$ | −0.87500000 | −0.99586776 | −0.99995098 | −0.99999950 |
| $\lambda^{(3)}$ | −0.93750000 | −0.99962434 | −0.99999951 | −0.99999999 |

Table 3.2: Convergence of the example problem with the method of augmented Lagrangians.

The method of augmented Lagrangians is designed for use with equality based problems. This means, that the non-differentiability introduced by the Macaulay bracket hinders the convergence of an augmented Lagrangian procedure in the same way as it hinders convergence in the penalty method. There are some advantages though: The penalties can be chosen a lot smaller than with the penalty method, since the augmentation procedure will help to converge to the correct solution. This helps the stability of the numerical method employed. Moreover, it is possible to get to the actual solution, which was not the case with the penalty method. An additional advantage is that if we increase the penalty parameter amidst augmentations, we gain superlinear convergence. A feat, which was also sought in Zavarise and Wriggers (1999) by an acceleration scheme.

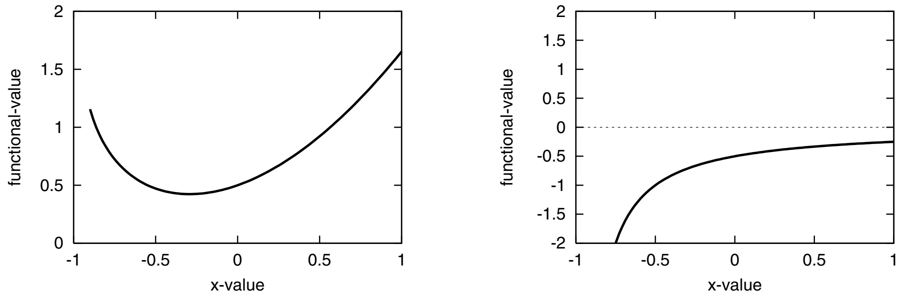
In the next section, we look at a method that attempts to overcome the problems of the non differentiability and tries to retain the advantageous properties for the method of augmented Lagrangians.

### 3.4.3 The modified barrier method

The previous two methods both converted the inequality constraints into equality constraints before an optimisation method was chosen to solve the resulting problem. This type of approach, however, introduced a non-differentiability in the constraints. It is well known fact that introduction of a non smoothness into a problem can causes bad convergence behaviour.

The method we propose to use in this case is the modified barrier method, see also Kloosterman et al. (2001). The convergence properties of the method are discussed by Polyak (1992). The method was discussed in Breitfeld and Shanno (1994, 1996); Franz et al. (1995), where it is tested on small but complex non-linear optimisation problems. Our interest lies more in the large scale optimisation problems. The method itself is an modification to the barrier method, originally proposed by Frisch (1955). The barrier method employs as its penalty function:

$$\Xi(\mathrm{d_N}; \{p\}) = -\frac{1}{p} \log \mathrm{d_N}. \tag{3.38}$$



(a) The extended potential function

(b) The traction estimate $\frac{\mathrm{d}\Xi}{\mathrm{d}\mathrm{d}_\mathrm{N}}$

Figure 3.5: Illustration of the modified barrier functional

The intent of this addition of a penalising function is clear: For large positive values of $\mathrm{d}_\mathrm{N}$, the contribution to the extended potential is negative. For small values near zero the contribution is big and positive. In effect, a constraint encounters a barrier in its range of allowable values.

There are some drawbacks to this approach, however. First of all, as it is the case with the penalty method, the barrier method is only accurate for large penalties, and generates poorly conditioned problems for penalties that are too large[2]. What can be considered even worse, is that the problem is only defined for feasible solutions. Hence, no constraint may be violated anywhere during the computation. This may pose a problem in some optimisation problems where finding a feasible initial configuration is sometimes already as hard as finding an optimal solution. This is not necessarily true for finite element simulations with contact, however, although small violations due to meshing may occur in the initial configuration.

To overcome the above problems, both an augmentation method is required, as well as a solution to the problem of the penalisation function Ξ's existence for negative values of the constraint functions. This is achieved by setting Ξ to the following functional form for feasible solutions:

$$\Xi(\mathrm{d}_\mathrm{N}; \{p, \lambda_\mathrm{N}\}) = \frac{1}{p}\lambda_\mathrm{N} \log\left(1 + p\,\mathrm{d}_\mathrm{N}\right). \tag{3.39}$$

In this equation, $\lambda_\mathrm{N}$ is a fixed estimate for the Lagrange multiplier, as it was in the augmented Lagrangian procedure. The inverse penalty $\frac{1}{p}$ is known as the barrier parameter.

The concept again can be explained by the following reasoning: As $\mathrm{d}_\mathrm{N}$ approaches $-\frac{1}{p}$, the logarithm adds a significant penalty to the the extended unconstrained potential as defined in (3.20). Whereas, for $\mathrm{d}_\mathrm{N}$ large, the function offers a negative contribution and is preferred by the global minimisation algorithm.

[2] For linear optimisation problems, there is a strong interest nowadays for barrier methods as an interior point approach. By using the barrier method, solutions to the linear optimisation problem can be found in polynomial time.

For an illustration of the procedure consider Figure 3.5. In it an arbitrary potential function was plotted, dependent solely on a variable $x$. The selected function is yet again $P(x) = \frac{1}{2}(x+1)^2$ under the constraint $x \geq 0$. The barrier term is then $\frac{1}{p}\lambda_{\mathrm{N}} \log(1 + p\,x)$, where $p$ is set to 1, and $\lambda_{\mathrm{N}} = -\frac{1}{2}$ as in the augmented Lagrangian example. The minimum for the constrained problem lies at $x = 0$. Looking at the picture one can see that the extended potential $\frac{1}{2}(x+1)^2 - \frac{1}{2}\log(1+x)$ is an approximation to the minimum. If we would have chosen the correct Lagrange multiplier for $\lambda_{\mathrm{N}}$, the minimum of the extended potential would lie exactly at the point 0. The approximate minimum lies at $x = -1 + \frac{1}{2}\sqrt{2} \approx -0.3$. In Table 3.4.3, the convergence of the example problem is given for several different penalties. In each case, the initial value for the Lagrange multiplier $\lambda^{(0)}$ is $\frac{1}{2}$.

| | p = 1 | p = 100 | p = 1000 | p = 10000 |
|---|---|---|---|---|
| $x^{(1)}$ | $-2.928 \cdot 10^{-1}$ | $-4.750 \cdot 10^{-2}$ | $-4.975 \cdot 10^{-3}$ | $-4.997 \cdot 10^{-4}$ |
| $x^{(2)}$ | $-1.159 \cdot 10^{-1}$ | $-4.335 \cdot 10^{-3}$ | $-4.925 \cdot 10^{-5}$ | $-4.992 \cdot 10^{-7}$ |
| $x^{(3)}$ | $-8.299 \cdot 10^{-2}$ | $-3.943 \cdot 10^{-4}$ | $-4.877 \cdot 10^{-7}$ | $-4.987 \cdot 10^{-10}$ |
| $\lambda^{(1)}$ | $-0.70710678$ | $-0.95249378$ | $-0.99502499$ | $-0.99950024$ |
| $\lambda^{(2)}$ | $-0.84089641$ | $-0.99566416$ | $-0.99995074$ | $-0.99999950$ |
| $\lambda^{(3)}$ | $-0.91700404$ | $-0.99960569$ | $-0.99999951$ | $-0.99999999$ |

Table 3.3: Convergence of the example problem with the modified barrier method.

The numerical approach that is followed is now roughly equivalent to the augmented Lagrangian approach for equality constrained problems. A value for $p$ is chosen, as well as a fixed value for the estimation of the Lagrange multiplier. The minimisation for the extended potential is then solved using these parameters. After that, using the minimiser $\boldsymbol{\varphi}$, the estimate for the Lagrange multipliers are improved, $p$ is increased and we start again. It was proven in Polyak (1992) that under certain conditions this approach converges linearly to a solution of (3.20).

### Modified barrier in elastic discretisation

In the remainder of this section, we concern ourselves with the discretised elastic problem to give an illustration of how the problem is to be solved. In this case, we have the following problem:

$$\begin{array}{rl} \min_x & P(x) \\ \text{s.t.} & \mathrm{d}_{\mathrm{N},i}(x) \geq 0 \text{ for } i \in I. \end{array} \tag{3.40}$$

The modified barrier method is then defined through the following function:

$$\min_x F(x, p, \lambda) = \min_x \left( P(x) + \frac{1}{p} \sum_{i \in I} \lambda_{\mathrm{N},i} \log\left(1 + p\,\mathrm{d}_{\mathrm{N},i}\right) \right). \tag{3.41}$$

To improve the rate of convergence, the constraints are scaled. This can be useful if for example the constraints appear with different orders of magnitude. The necessity of this is discussed in Breitfeld and Shanno (1994). If all constraints are of the same type scaling is not necessary, but for sake of completeness they are included in the discussion. To this end, a scaling factor $s_i$ for each of the constrains is introduced. The value of the constraint is then scaled to be $\mathrm{d}_{\mathrm{N},i}(x)/s_i$. Combining this with the unconstrained subproblem as in (3.41) results in:

$$\begin{aligned} F(x, p, \lambda_{\mathrm{N}}, s) &= P(x) + \frac{1}{p}\sum_{i\in I} \lambda_{\mathrm{N},i} \log\left(1 + p\,\frac{\mathrm{d}_{\mathrm{N},i}(x)}{s_i}\right) \\ &= P(x) + \frac{1}{p}\sum_{i\in I} \lambda_{\mathrm{N},i} \log\left(\frac{1}{s_i}\left(s_i + p\,\mathrm{d}_{\mathrm{N},i}(x)\right)\right) \qquad (3.42) \\ &= P(x) + \frac{1}{p}\sum_{i\in I} \lambda_{\mathrm{N},i} \log\left(s_i + p\,\mathrm{d}_{\mathrm{N},i}(x)\right) - \frac{1}{p}\sum_{i\in I} \lambda_{\mathrm{N},i} \log\left(s_i\right). \end{aligned}$$

Notice that the last term in (3.42) contains only terms in $p$, $\lambda_{\mathrm{N}}$ and $s$. Thus, if we take the minimum with respect to $x$ for set values of these parameters, the last term does not influence the value of the optimal $x$. Consequently, minimising (3.42) is equal to:

$$\min_x F(x, \mu, \lambda_{\mathrm{N}}, s) = \min_x \left(P(x) + \frac{1}{p}\sum_{i\in I} \lambda_{\mathrm{N},i} \log\left(s_i + p\,\mathrm{d}_{\mathrm{N},i}(x)\right)\right). \qquad (3.43)$$

An additional limitation that may occur, is that the function $F(x, p, \lambda, s)$ is not defined for all possible values for $\mathrm{d}_{\mathrm{N},i}(x)$. This is due to requiring that $\mathrm{d}_{\mathrm{N},i}(x) > -\frac{s_i}{p}$, since otherwise the logarithm is not defined. To overcome this limitation we introduce a quadratic continuation of the logarithm, from somewhere in the admissible region. This continuation location is defined through the parameter $\beta$, which is some relative point. If $\mathrm{d}_{\mathrm{N},i}(x) \geq -\frac{\beta s_i}{p}$, we use the logarithm. If $\mathrm{d}_{\mathrm{N},i}(x) < -\frac{\beta s_i}{p}$, we use the quadratic continuation. The parameter $\beta$ should lie within the range $(0, 1)$. The problem as it is defined in (3.43) becomes:

$$F(x, p, \lambda, s, \beta) = P(x) + \sum_{i\in I} \Xi(\mathrm{d}_{\mathrm{N},i}(x), \lambda_{\mathrm{N},i}, p, s_i, \beta). \qquad (3.44)$$

In the previous formula $\Xi$ is the combination of the logarithmic term and the quadratic extrapolation term. It is defined as:

$$\Xi(\mathrm{d}_{\mathrm{N},i}(x), \lambda_{\mathrm{N},i}, p, s_i, \beta) = \begin{cases} \frac{\lambda_{\mathrm{N},i}}{p} \log(s_i + p\,\mathrm{d}_{\mathrm{N},i}(x)) & \text{if } \mathrm{d}_{\mathrm{N},i}(x) \geq -\frac{\beta s_i}{p}, \\ \frac{1}{2} a_i \mathrm{d}_{\mathrm{N},i}(x)^2 + b_i \mathrm{d}_{\mathrm{N},i}(x) + c_i & \text{if } \mathrm{d}_{\mathrm{N},i}(x) < -\frac{\beta s_i}{p}. \end{cases} \qquad (3.45)$$

In this equation $a_i, b_i$ and $c_i$ are chosen such that the continuation and the logarithmic part attach twice continuously differentiable. From this conditions, it follows that they are defined as:

$$\begin{aligned} a_i^{(k)} &= \frac{-\left(p^{(k)}\right)^2}{(s_i^{(k)}(1-\beta))^2}, \\ b_i^{(k)} &= \frac{p^{(k)}\,(1-2\beta)}{s_i^{(k)}(1-\beta)^2}, \qquad (3.46) \\ c_i^{(k)} &= \frac{\beta(2-3\beta)}{2(1-\beta)^2} + \log(s_i^{(k)}(1-\beta)). \end{aligned}$$

The $\lambda_{\mathrm{N},i}$ are initialised to 1. Their correct value is computed through application of an augmentation scheme. Using this augmentation scheme and an updating strategy for the barrier parameter $\frac{1}{p}$, it was shown by Breitfeld and Shanno (1994) that a subsequence of points $(x^k, \lambda_{\mathrm{N}}^k)$ generated by the algorithm converges to a Karush–Kuhn–Tucker point of (3.40).

In the next section we give a complete presentation of the algorithm that is used to solve (3.40).

### The Algorithm

We now present a precise description of the barrier algorithm, such as employed in the non-linear optimisation framework for elastic problems only. The adaptation of this algorithm for use in non-linear finite element simulations is straightforward.

**Step 0. Start:** First we need to initialise all the parameters.

- Choose $x^{(0)} \in \mathbb{R}^N$. This is the starting point for the computation. In an incremental computation, $x^{(0)}$ is best chosen as the result of the previous increment.
- Choose $\tau > 0$, the outer loop termination criterion. This termination criterion is approximately equivalent to the unbalance termination criterion. $10^{-3}$ is a reasonable choice. It is a dimensionless parameter.
- Select a sequence of barrier parameters $\{p^{(k)}\}_{k\geq 0}$, which is ascending. In Breitfeld and Shanno (1994) $p^{(k+1)}$ is chosen as $10p^{(k)}$.
- Select a sequence $\{\epsilon^{(k)}\}_{k\geq 0}$, the inner loop termination criterion. A possible selection is to choose this sequence equal to the inverted $p$ sequence.
- Select $\lambda_{\mathrm{N},i}^{(0)} = 1$, for $i \in I$.
- Choose $0 \leq \beta \leq \beta_u < 1$, the relative extrapolation point.
- Choose the scaling terms $s_i^{(0)} = \min\{\max\{1, -\mathrm{d}_{\mathrm{N},i}(x^{(0)})\}, s_u\}$ for $i \in I$, where $s_u$ is some upper bound for the scaling terms.
- Compute the extrapolation coefficients for the $i$-th barrier function $a_i^{(0)}$, $b_i^{(0)}$ and $c_i^{(0)}$ by (3.46). These coefficients are completely determined by demanding $\Xi$ to be twice continuously differentiable.
- Set $k = 0$.

**Step 1. Unconstrained minimisation:** In this step we find an approximation $x^{(k+1)}$ of a local minimiser of the current regularised problem:

$$\min_x F(x, p^{(k)}, \lambda_{\mathrm{N}}^{(k)}, s^{(k)}, \beta) =$$

$$\min_x \left( P(x) + \sum_{i\in I} \Xi(\mathrm{d}_{\mathrm{N},i}(x), \lambda_{\mathrm{N},i}^{(k)} p^{(k)}, s_i^{(k)}, \beta) \right), \tag{3.47}$$

where the barrier function $\Xi$ is defined as:

$$\Xi(\mathrm{d}_{\mathrm{N},i}(x), \lambda_{\mathrm{N},i}, p, s_i, \beta) = \begin{cases} \frac{\lambda_{N,i}}{p} \log(s_i + p\,\mathrm{d}_{\mathrm{N},i}(x)) & \text{if } \mathrm{d}_{\mathrm{N},i}(x) \geq -\frac{\beta s_i}{p}, \\ \frac{1}{2} a_i \mathrm{d}_{\mathrm{N},i}(x)^2 + b_i \mathrm{d}_{\mathrm{N},i}(x) + c_i & \text{if } \mathrm{d}_{\mathrm{N},i}(x) < -\frac{\beta s_i}{p}. \end{cases} \tag{3.48}$$

Solving the unconstrained problem is the inner problem. This is the problem that actually involves taking Newton steps. To solve (3.47) we consider the first order condition for it:

$$\nabla_x F(x, p^{(k)}, \lambda_{\mathrm{N}}^{(k)}, s^{(k)}, \beta) = 0. \tag{3.49}$$

**A.** Set $y^{(0)} = x^{(k)}$, and $j = 0$.

**B.** Check the convergence:

$$w^{(j)} = \begin{cases} \frac{\|\nabla_x F(y^{(j)}, p^{(k)}, \lambda_{\mathrm{N}}^{(k)}, s^{(k)}, \beta)\|}{\nabla_x P(y^{(j)})} & \text{if } \|\nabla_x P(y^{(j)})\| > \epsilon^{(k)}, \\ \|\nabla_x F(y^{(j)}, p^{(k)}, \lambda_{\mathrm{N}}^{(k)}, s^{(k)}, \beta)\| & \text{otherwise.} \end{cases} \tag{3.50}$$

If $w^{(j)} < \epsilon^{(k)}$, set $x^{(k+1)} = y^{(j)}$ and stop.

**C.** Solve the linearisation of the first order condition (3.49) at $y^{(j)}$:

$$\delta y^{(j)} = -\left[\nabla_x^2 F(y^{(j)}, p^{(k)}, \lambda_{\mathrm{N}}^{(k)}, s^{(k)}, \beta)\right]^{-1} \nabla_x F(y^{(j)}, p^{(k)}, \lambda_{\mathrm{N}}^{(k)}, s^{(k)}, \beta) \tag{3.51}$$

**D.** Take $y^{(j+1)} = y^{(j)} + \alpha \delta y$, where $\alpha$ is chosen such that $F$ decreases. If we are sufficiently close to the minimum $\alpha = 1$ is a good choice, and corresponds to taking full Newton steps. The $\alpha$ is also known as a line search parameter.

**E.** Return to step B.

**Step 2. Check convergence:** We now test whether the convergence criterion is met. First we define $v^{(k+1)}$ according to:

$$v^{(k+1)} = \frac{|\sum_{i \in I} \lambda_{\mathrm{N},i}^{(k)} \mathrm{d}_{\mathrm{N},i}(x^{(k+1)})|}{\sqrt{\sum_{i \in I} \left(\lambda_{\mathrm{N},i}^{(k)}\right)^2}}. \tag{3.52}$$

If $v^{(k+1)} < \tau$ then stop. The parameter measures the satisfaction of the Kuhn–Tucker complementarity condition. The satisfaction of the unbalance criterion is already established at the end of Step 1.

**Step 3. Update parameters:** We now update the scaling parameters, by scaling with respect to the new point:

$$s_i^{(k+1)} = \min\{\max\{1, -\mathrm{d}_i(x^{(k+1)})\}, s_u\}. \tag{3.53}$$

And the Lagrange multipliers are updated according to:

$$\lambda_{\mathrm{N},i}^{(k+1)} = \Xi'(\mathrm{d}_{\mathrm{N},i}(x^{(k+1)}), \lambda_{\mathrm{N},i}^{(k)}, p^{(k)}, s_i^{(k)}, \beta), \tag{3.54}$$

where $\Xi'$ denotes the derivative of $\Xi$ with respect to the first variable only.

The motivation for choosing $\lambda_{\mathrm{N}}$ in this manner comes from (3.49). If we expand this equation we find:

$$\nabla P(x^{(k+1)}) + \sum_{i \in I} \frac{\mathrm{d}\Xi}{\mathrm{d}\mathrm{d}_{\mathrm{N},i}}(\mathrm{d}_{\mathrm{N},i}(x^{(k+1)}), \lambda_{\mathrm{N},i}^{(k)}, p^{(k)}, s_i^{(k)}, \beta)\nabla \mathrm{d}_{\mathrm{N},i}(x^{(k+1)}) = 0. \tag{3.55}$$

Thus using the solution of (3.49), we get that selecting $\lambda_i^{(k+1)}$ in the proposed way we have:

$$\nabla P(x^{(k+1)}) + \sum_{i \in I} \lambda_{\mathrm{N},i}^{(k+1)} \nabla \mathrm{d}_{\mathrm{N},i}(x^{(k+1)}) = 0. \tag{3.56}$$

In this we can recognize part of the conventional variational form of the equilibrium equations including contact. It is, however, not completely the same, since the $\lambda_{\mathrm{N},i}$ are held constant.

**Step 4. Update coefficients:** What remains to be done is to update the extrapolation coefficients used in the barrier functions $\Xi$. These coefficients are fully determined by enforcing $\Xi$ to be twice continuously differentiable, and are presented in (3.46).

**Step 5. Continuation** Set $k = k + 1$, and return to Step 1.

To achieve fast convergence some fine tuning is required for the barrier sequence $\frac{1}{p}$. Regardless of the fine tuning, Breitfeld and Shanno (1994) showed that the algorithm has at least a convergent subsequence for any such sequence. However, in this proof it was assumed that functions $P$ and $\mathrm{d}_{\mathrm{N},i}$ where twice differentiable functions. This is for the discretised finite element method not generally the case, but can be enforced if a good integration scheme is used. We return to this topic in Chapter 4.

## 3.5 Regularising Friction

Apart from creating a regularisation of the contact traction, the correct frictional tractions are also to be computed. Just as with the normal contact tractions the computation of the tangential tractions is done through a regularisation. Only this time we do not add the functional to the optimisation problem. Instead, the method is introduced immediately into the variational setting. The reason for this is the non-associativity of the frictional traction in the case of slip for Coulomb friction.

It is assumed that $\mathbf{t}_{\mathrm{T}}$ is a function of $\mathbf{d}_{\mathrm{T}}$ and $\mathrm{t}_{\mathrm{N}}$. A complication with (Coulomb) friction is the occurrence of stick and slip. Depending on the nature of the problem, determining which nodes are sticking and which are slipping can be quite difficult. In the case of stick, the frictional tractions behave associatively: There is no direct dependence of the tangential tractions on the contact normal traction. In this case any of the methods which can be applied for contact normal tractions can be applied for the determination of the contact tangential tractions. In slip however, the amount of force exerted by the frictional tractions depends on the normal traction. In the latter case, associativity is lost.

The way in which the tangential tractions are computed when slipping is by employing a regularisation method as with the normal contact tractions, and then apply a return-mapping scheme (See also Laursen and Simo, 1992; Giannakopoulos, 1989) In fact the return mapping is just a scaling of the frictional tractions to satisfy the Coulomb friction constraint.

The lay-out of this section is as follows: In Section 3.5.1, the penalty method in friction is introduced. Next in Section 3.5.2 the method of augmented Lagrangians in the case of friction is discussed.

### 3.5.1 Penalty approach

Let us begin by repeating the Coulomb friction constraints for contact. These are the same as (2.39a)–(2.39d):

$$\begin{aligned} \Phi := \|\mathbf{t}_{\mathrm{T}}\| - \mu|\mathrm{t}_{\mathrm{N}}| &\leq 0, &\quad (3.57\mathrm{a})\\ \mathbf{d}_{\mathrm{T}} + \Delta t \zeta \mathbf{t}_{\mathrm{T}} &= 0, &\quad (3.57\mathrm{b})\\ \zeta &\geq 0, &\quad (3.57\mathrm{c})\\ \Phi \cdot \zeta &= 0. &\quad (3.57\mathrm{d}) \end{aligned}$$

The first condition (3.57a) states the (Coulomb) friction condition. In the case that $\mu$ is allowed to be a function of velocity, and or pressure, more general frictional laws can be introduced. The second condition (3.57b), together with the third condition (3.57c) constrains the tangential traction to work opposite to the direction of slip. Finally the fourth condition (3.57d) states another complementarity condition: There is no slip if the tangential traction has not reached its (local) maximum. And if there is slip, then it has reached its maximum.

Regularisations in friction work slightly different from regularisation in contact. The difference lies in what constraint needs to be enforced. In contact, the constraint to enforce was that the distance between the two contacting bodies is non-negative. Hence, we have an inequality constraint on the displacements. If (3.57a) is satisfied, we have sticking friction. In that case to satisfy (3.57d) it is required that $\zeta = 0$. Substituting this result in (3.57b), we obtain:

$$\mathbf{d}_{\mathrm{T}} = \mathbf{0}. \tag{3.58}$$

The above stick condition is an equality constraint. Thus, in analogy to regularisation methods for normal contact, we can immediately apply the penalty method to the above (vector) equation to obtain an estimate for the contact normal tractions:

$$\mathbf{t}_{\mathrm{T}} = p\mathbf{d}_{\mathrm{T}}. \tag{3.59}$$

We note that $p$ is a generic penalty value, which does not necessarily have the same value as the penalty parameter for the normal contact methods, though some methods couple the values.

Vice-versa, one could use (3.59) to decide whether the node is sticking or sliding. If $\|p\mathbf{d}_{\mathrm{T}}\| \leq |\mathrm{t}_{\mathrm{N}}|$, then (3.57a) is satisfied and (3.59) gives an appropriate estimation to the frictional traction. However, if $\|p\mathbf{d}_{\mathrm{T}}\| > |\mathrm{t}_{\mathrm{N}}|$ then our assumption that $\zeta = 0$ does not hold, and (3.57b) is used to scale the estimation so that (3.57a) is satisfied.

First let us name $\Delta t\zeta = \Delta\zeta$. Also, we define $p\mathbf{d}_{\mathrm{T}}$ as the predictor of the frictional traction $\mathbf{t}_{\mathrm{T}}^{p}$. The frictional tractions can now be computed by:

$$\mathbf{t}_{\mathrm{T}} = \begin{cases} \mathbf{t}_{\mathrm{T}}^{p} & \text{if } \|\mathbf{t}_{\mathrm{T}}^{p}\| \leq |\mathrm{t}_{\mathrm{N}}|, \\ \mu|\mathrm{t}_{\mathrm{N}}|\frac{\mathbf{t}_{\mathrm{T}}^{p}}{\|\mathbf{t}_{\mathrm{T}}^{p}\|} & \text{otherwise.} \end{cases} \tag{3.60}$$

### 3.5.2 The method of augmented Lagrangians

The application of the method of augmented Lagrangians proceeds in an identical fashion as the application of the penalty method. The method is discussed also by Jones and Papadopoulos (2000). What changes is how the predictor is computed. For the augmented Lagrangian case this is:

$$\mathbf{t}_{\mathrm{T}}^{p} = \boldsymbol{\lambda}_{\mathrm{T}} + p\mathbf{d}_{\mathrm{T}}. \tag{3.61}$$

The actual value of the contact tractions can then be obtained from (3.60).

As with the method of augmented Lagrangians for contact in the normal direction, $\boldsymbol{\lambda}_{\mathrm{T}}$ is initialised to $\mathbf{0}$. When the increment has converged to a certain value of $\boldsymbol{\lambda}_{\mathrm{T}}$, its estimation can be improved by setting it to the final estimated value for the tangential tractions from the increment:

$$\boldsymbol{\lambda}_{\mathrm{T}} = \mathbf{t}_{\mathrm{T}}(\mathbf{d}_{\mathrm{T}}). \tag{3.62}$$

Using these improved value, the increment can then be recomputed with a smaller violation of the slip condition.

When using the method of augmented Lagrangians both for the contact tractions and the frictional tractions, we have the option to make the contact method completely associated again. The advantages of this are symmetry of the matrix, so that less storage is required. Additionally symmetrical matrices can be solved more efficiently due to the additional structure. The symmetric matrix is obtained, if we set the value of $\mathrm{t}_{\mathrm{N}}$ in (3.57a) and (3.60) to $\lambda_{\mathrm{N}}$, which is the estimation to the contact normal traction. When employed in this manner, the friction is always one step in accuracy behind the contact normal traction, since it is using the normal traction from a previous augmentation. However, due to the additional efficiency of using symmetric matrices, a solution with the same accuracy may still be computed in less time.

## 3.6 Conclusions

In this chapter, we discussed several methods that can be employed to solve the contact problem in finite element methods. First we discussed the mixed method, which is computationally expensive, and the constraint method, that is very efficient but can present problems for example in plate forming simulations were the detection of contact is very important. Moreover, there may be too many constraints in the plate forming simulation to use an effective active set method.

After that, we discussed the general form of regularisation problems, and noted that all of them stem from a basic mold. Everything is really decided by selecting the form of a specific penalisation function $\Xi$. This penalisation function is introduced in the optimisation setting For the more general variational setting, one just fills in the results for the first order conditions of the optimisation problem. The different methods discussed were:

- The penalty method, for which:

$$\begin{aligned} \Xi(\mathrm{d}_\mathrm{N}, \{p\}) &= \frac{p}{2}\langle -\mathrm{d}_\mathrm{N}\rangle^2, &\text{(3.63a)}\\ \mathrm{t}_\mathrm{N}(\mathrm{d}_\mathrm{N}, \{p\}) &= -p\langle -\mathrm{d}_\mathrm{N}\rangle. &\text{(3.63b)}\end{aligned}$$

- The method of augmented Lagrangians, for which:

$$\begin{aligned} \Xi(\mathrm{d}_\mathrm{N}, \{p, \lambda_\mathrm{N}\}) &= \frac{1}{2p}\left\langle -(\lambda_\mathrm{N} + p\mathrm{d}_\mathrm{N})\right\rangle^2, &\text{(3.64a)}\\ \mathrm{t}_\mathrm{N}(\mathrm{d}_\mathrm{N}, \{p, \lambda_\mathrm{N}\}) &= -\langle -(\lambda_\mathrm{N} + p\mathrm{d}_\mathrm{N})\rangle. &\text{(3.64b)}\end{aligned}$$

- The modified barrier method for which:

$$\begin{aligned} \Xi(\mathrm{d}_\mathrm{N}, \{p, \lambda_\mathrm{N}\}) &= \frac{\lambda_\mathrm{N}}{p}\log\left(1 + p\mathrm{d}_\mathrm{N}\right), &\text{(3.65a)}\\ \mathrm{t}_\mathrm{N}(\mathrm{d}_\mathrm{N}, \{p, \lambda_\mathrm{N}\}) &= \lambda_\mathrm{N}\frac{1}{1 + p\mathrm{d}_\mathrm{N}}. &\text{(3.65b)}\end{aligned}$$

In the same fashion, two methods were discussed that enforce friction, they rely on a predictor-corrector scheme. Only the predictor is determined differently for the two schemes that were considered. These two schemes are:

- The penalty method for friction, in which case the predictor is set to:

$$\mathbf{t}_\mathrm{T}^p = p\mathbf{d}_\mathrm{T}. \tag{3.66}$$

- The method of augmented Lagrangians, in which case the predictor is set to:

$$\mathbf{t}_\mathrm{T}^p = \boldsymbol{\lambda}_\mathrm{T} + p\mathbf{d}_\mathrm{T}. \tag{3.67}$$

From the predictor, the correct frictional traction is computed, by using either the fixed estimate to the normal contact traction $\lambda_\mathrm{N}$ or the current estimate $\mathrm{t}_\mathrm{N}$. The frictional traction now follows from:

$$\mathbf{t}_\mathrm{T} = \begin{cases} \mathbf{t}_\mathrm{T}^p & \text{if } \|\mathbf{t}_\mathrm{T}^p\| \leq |\mathrm{t}_\mathrm{N}|, \\ \mu|\mathrm{t}_\mathrm{N}|\frac{\mathbf{t}_\mathrm{T}^p}{\|\mathbf{t}_\mathrm{T}^p\|} & \text{otherwise.} \end{cases} \tag{3.68}$$

Chapter 4

# DISTANCE FUNCTIONS

## 4.1 Introduction

In Chapter 2 the equations were formed that characterise the contact problem. In Chapter 3 several methods were discussed to convert the contact integral to a displacement based form only. In this chapter, the final piece of machinery is developed and discussed to simulate contact in forming: The actual evaluation of the contact integral.

In this chapter, it is assumed that a regularisation method is chosen to approximate the tractions. In that case there is only one fundamental unknown left, which is the computation of the distance function.

There are three main issues which are paramount to the success of a contact algorithm, these are stability, accuracy and efficiency. Each of these three issues is also important in the selection of the regularisation method, and this is the topic of most of the articles that appear on contact. However, even when such a regularisation method is already chosen, which does not cause instabilities, can satisfy the contact constraints accurately, and is relatively efficient, still stability problems can arise if the integral is not properly considered. Moreover the overall accuracy and efficiency depends on other factors than the regularisation method alone.

Upon the introduction of a discretisation for both the master and slave boundary, mesh incompatibilities can be introduced between the master and slave body. By mesh incompatibilities we mean that the element sizes of master and slave may not fit together when meshed independently. An alternative would be to mesh the master and slave together, to form an (initially) compatible contact topology.

Let us consider the first key factor: stability. When there is large sliding contact between the incompatible meshes, the gradient of the distance function can change abruptly. This abrupt change can cause stability issues. A careful examination of when these changes occur, can catch the instabilities before they distort the convergence of the finite element method. An alternative solution is to avoid the mesh incompatibilities by smoothing the master boundary. The situation is then reverted to that of the previous chapter. As an example consider El-Abbasi and Meguid (2001); Chamoret et al. (2001) or Wriggers et al. (2001). Good books on the geometry of surfaces are by Farin (1988) and Piegl and Tiller (1997). Additionally, one can attempt to stabilise the Newton method itself, as in Christensen et al. (1998); Leung et al. (1998); Kane et al. (1999). Stability problems in general are discussed by Esche et al. (1997); Pang (1990).

The second point in our discussion is that an accurate result is desired. The final accuracy of the simulation is depends on two things:

1. The integration scheme that is employed to numerically approximate the contact integral.

2. The discretisation that is used in the simulation of the problem.

The two aspects are not completely independent. The selection of the most appropriate integration method depends strongly on how the discretisation is performed. The aforementioned mesh incompatibilities can cause inaccurate results. Meshing the master and slave boundaries simultaneously and employing contact elements does not solve this problem: Due to sliding during the simulation, the contact elements can become severely distorted, and their results are then again inaccurate.

The third point in our discussion is efficiency: The computation of the distance functions should not take too much time. Having rejected contact elements due to possible accuracy problems, the projection location of the integration points is no longer fixed. Consequently, we need an algorithm to find the projection of the integration points on the master surface. Assuming that the master is meshed using linear elements, this means that the nearest segment on the master surface needs to be found. Various algorithms have been proposed. A hierarchical based search method is proposed by Zhong and Nilsson (1988) and Zhong and Nilsson (1990). The pinball algorithm was proposed by Belytschko and Neal (1991). An algorithm incorporated in the dynamic finite element package DYNA3D is by Oldenburg and Nilsson (1994). The method originally employed in the finite element package DIEKA is the block search method, discussed in Atzema (1994) and Carleer (1997). Further alternatives are discussed in Wang and Nakamachi (1997); Wang and Makinouchi (2000) and Wang et al. (2001). To illustrate the efficiency that can be gained by exploiting structure, the reader is referred to Munjiza and Andrews (1998). In this thesis we present a new search method, based on computational geometric data structures discussed in Berg et al. (1997).

The discussion in this chapter follows roughly the reasoning as presented above. In Section 4.2 the discretisation of the boundary is introduced, and the results for the distance function. From it a slightly adjusted formulation for the contact integrals is obtained. In Section 4.3 various integration methods are discussed to approximate the discretised integrals. In Section 4.4, the alternative of smoothing the master boundary is considered. In Section 4.5 we present a new method to quickly locate the projection locations for each of the integration points on a piecewise linear boundary. Finally in Section 4.6 the conclusions are presented.

## 4.2 Discretisation of the contact integral

In this section the discretisation of the contacting boundaries is discussed. The discretisation influences the manner in which the contact integral is to be computed. The boundaries can be either meshed independently of one another, or simultaneously. The latter method gives rise to the use of contact elements. The advantages and disadvantages of both methods are discussed. The result is that the contact elementless method is chosen in the remainder of this chapter as the more generally applicable one.

### 4.2.1 The discrete integral formulation

For an arbitrary variation $\delta\boldsymbol{\varphi}$, let us again state the integral to be computed for the regularised virtual displacement case, see (2.71):

$$G_c(\boldsymbol{\varphi}, \delta\boldsymbol{\varphi}) = \int_\Gamma \mathrm{t_N}(\mathrm{d_N}(\boldsymbol{\varphi}(\mathbf{X})))D_{\delta\boldsymbol{\varphi}}[\mathrm{d_N}(\boldsymbol{\varphi}(\mathbf{X}))]\,\mathrm{d}\Gamma. \tag{4.1}$$

Upon discretising the above, with $\boldsymbol{\varphi}^h \in \mathcal{S}^h$ and $\delta\boldsymbol{\varphi}^h \in \mathcal{V}^h$, we arrive at:

$$G_c(\boldsymbol{\varphi}^h, \delta\boldsymbol{\varphi}^h) = \int_{\Gamma^h} \mathrm{t_N}(\mathrm{d_N}(\boldsymbol{\varphi}^h(\mathbf{X})))D_{\delta\boldsymbol{\varphi}^h}[\mathrm{d_N}(\boldsymbol{\varphi}^h(\mathbf{X}))]\,\mathrm{d}\Gamma. \tag{4.2}$$

And since $\boldsymbol{\varphi}^h(\mathbf{X}) = N(\mathbf{X}) \cdot x$ and $\delta\boldsymbol{\varphi}^h(\mathbf{X}) = N(\mathbf{X}) \cdot \delta x$, we can convert the directional derivative in the integral in the following manner:

$$\begin{aligned} D_{\delta\boldsymbol{\varphi}^h}[\mathrm{d_N}(\boldsymbol{\varphi}^h(\mathbf{X}))] &= D_{N(\mathbf{X})\cdot\delta x}[\mathrm{d_N}(N(\mathbf{X}) \cdot x)] \\ &= \left.\frac{\mathrm{d}}{\mathrm{d}\epsilon}\right|_{\epsilon=0} [\mathrm{d_N}(N(\mathbf{X}) \cdot x + \epsilon N(\mathbf{X}) \cdot \delta x)] \\ &= (\nabla_x \mathrm{d_N}(N(\mathbf{X}) \cdot x)) \cdot \delta x. \end{aligned} \tag{4.3}$$

Substituting this result back into (4.2) results in:

$$\begin{aligned} G_c^h(x, \delta x) &= \int_{\Gamma^h} \mathrm{t_N}(\mathrm{d_N}(x; \mathbf{X}))\,(\nabla_x \mathrm{d_N}(x; \mathbf{X})) \cdot \delta x\,\,\mathrm{d}\Gamma \\ &= \left(\int_{\Gamma^h} \mathrm{t_N} \nabla_x \mathrm{d_N}\,\mathrm{d}\Gamma\right) \cdot \delta x. \end{aligned} \tag{4.4}$$

In the latter equation the specific dependencies of $\mathrm{d_N}$ on the field variables were omitted for notational clarity. What remains to be done is to compute the integral between the parenthesis.

### 4.2.2 The appearance of geometric incompatibilities

Whilst going from (4.1) to (4.2), an implicit assumption was made. The assumption is that (4.2) is actually a good approximation to (4.1). This is in general not as good as would at first sight be expected. The problem is, that in the derivation of (4.1) we used the fact that where $\mathrm{d_N} = 0$ in the final solution, we have $\Gamma = \Gamma^{(1)} \cap \Gamma^{(2)}$. The remainder of the potential contact boundary, it was argued, does not matter in the computation. Upon discretisation, however, there may not be a proper contacting boundary left. In most of the situations, there will no longer be a contacting boundary left, only some contacting points.

Let us clarify the statement with some qualitative examples. The first example presents the contact between a discretised plate and a non-discretised roll. The roll is modelled using a circle. The second example is primarily the same as the first, only now the roll is discretised.

The examples are illustrated in Figure 4.1. The left picture in this figure illustrates the contact without penetration in the case the plate is discretised and the roll is smoothly modelled with a circle. As can be seen, to have a non-penetrating solution, there is only contact between the roll and the plate at two distinct locations. In fact, each of the surface segments can have at most one point of contact with the circle in an allowable solution. This is an essential geometric property of contact between a line segment and a circle. As a consequence, refinement of the mesh may increase the number of contacting points, but the number of these points will remain finite. This implies that there is no proper boundary to integrate over, merely a set of contacting points.

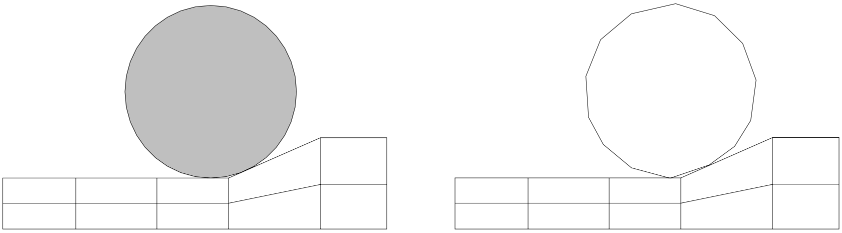


Figure 4.1: Contact between roll and plate, both smooth and discrete.

It is clear that such point wise constraints do not generally occur within the actual problem that is being modelled. In fact, in the actual problem, there will be a region of contact, not a set of points. The discretisation operation has rendered the two boundaries geometrically incompatible.

The situation does not improve upon discretisation of both the boundaries as is shown in Figure 4.1 in the picture to the right. However, the situation does not really seem to deteriorate either. There are still two contacting points.

To overcome the boundary incompatibilities, one may choose to employ contact elements, whereby both bodies are meshed simultaneously. However, when there is sliding occurring in the simulations, the contact elements are no longer guaranteed to have properly facing boundary segments. The situation then reverses to that illustrated in Figure 4.1 to the right, with possible penetration occurring. This problem can be avoided by employing an Arbitrary Lagrangian Eulerian scheme (see Huétink, 1986; van der Lugt, 1988). This type of procedure can keep the contact elements facing each other properly during the simulation. The result is geometrically compatible, but this comes at the price of having to perform a Eulerian step at each iteration. Also within some simulations it is nearly impossible to keep the contacting nodes at their proper locations and still have an acceptable mesh to continue the computation. Other attempts to overcome the inaccuracy arising due to geometric incompatibilities come from domain decomposition theory, (see Belgacem et al., 1998; Hild et al., 1998; Hild, 2000) or just trying to connect the meshes Dohrmann et al. (2000).

### 4.2.3 Dealing with geometric incompatibilities

The introduction of the geometric incompatibilities has a profound effect on the values of the distance function, and consequently on the outcome of the contact integral. The best that can be hoped for, is that the point-wise satisfaction of the contact constraints due to the discretisation approximates the values of the continuum problem in these points. If this is the case, then these discrete points can be used in the numerical approximation of the integral, which is discussed in Section 4.3.

The geometric incompatibilities cause the distance functions to be non-differentiable with respect to the boundary parametrisation, even though the master boundary may be smooth. The situation is illustrated in Figure 4.2

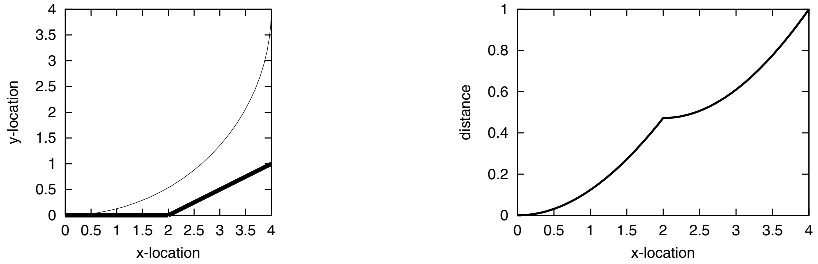


Figure 4.2: Illustration of non-differentiability with respect to boundary parametrisation.

On the left side of this figure, part of a roll is modelled by a circle. Below it, two line segments are given, which could be assumed to belong to a discretised body lying underneath. In the right side of the figure, the distance with respect to the roll is given as a function of the x-coordinate. From this figure, it can immediately be seen that the distance function is not smooth with respect to this coordinate.

This type of non-smoothness may cause the gradient to be discontinuous. To make sure that the evaluation will be accurate, the integral has to be split into its piece-wise smooth parts. It is assumed here that the master body is smooth, thus as long as the slave body is smooth, then the distance function will be smooth. The jumps only occur at element boundaries, thus the integral is to be computed by:

$$\int_{\Gamma^h} \mathrm{t_N} \nabla_x \mathrm{d_N}\, \mathrm{d}\Gamma = \sum_e \int_{\Gamma^h_e} \mathrm{t_N} \nabla_x \mathrm{d_N}\, \mathrm{d}\Gamma. \tag{4.5}$$

This all seems still quite trivial. However, assume that $\mathbf{X}$ is a material point lying on the boundary of both $\Gamma^h_1$ and $\Gamma^h_2$. If the master boundary is continuous, then we will still have that $\mathrm{d}_{\mathrm{N},1}(x;\mathbf{X}) = \mathrm{d}_{\mathrm{N},2}(x;\mathbf{X})$. The gradients at these points may now differ. This has no influence on the results of the integral if an analytical scheme was used. However, it can influence the numerical integration scheme.

We wish to make one remark on the analysis: Although the distance function is not smooth with respect to the boundary discretisation it can still be smooth with respect to the unknowns (displacements) in the vector $x$. It is the non-smoothness with respect to $x$ which influences the convergence characteristics of the Newton-Raphson process. The non smoothness with respect to the boundary parametrisation influences the final accuracy of the problem if not taken into account during the integration process.

### 4.2.4 Conclusions

We can conclude that in general problems that do not employ an ALE method will have to live with the geometric incompatibilities. In the case that both bodies are to be meshed, this means that a method which does not employ contact elements is to be preferred. The contact regions will in that case generally be limited to a number of points. The integral is furthermore to be split into smooth parts for an accurate result.

## 4.3 Evaluating the discretised contact integral

In this section several integration schemes are presented that can be used for evaluating the contact integral. Each of the methods has a theoretical accuracy associated with it, and initially one would assume that a higher order rule will yield more accurate results than a lower order one. Unfortunately, this is not so, and geometric arguments are given as to why this is the case.

The outline of this section is as follows: First in Section 4.3.1 several different numerical integration schemes are introduced. From these integration schemes the trapezoidal rule is selected as the most appropriate method for approximating the contact integral. The analysis is made with appeal to a simple master surface geometry. Next in Section 4.3.2 a short discussion is given on how to compute the distance with respect to more complex boundaries. From this, the piecewise linear boundary stands out as the more difficult to consider. Subsequently in Section 4.3.3, 4.3.4 and 4.3.5 the problems of non-smoothness of distance function with respect to the global variables, the non-smoothness of the distance function with respect to the boundary parametrisation and their solutions are discussed for the piecewise linear boundary. Finally, in Section 4.3.6 the conclusions are presented.

### 4.3.1 Numerical integration schemes

In Section 4.2.3, the total contact integral was split into individual smooth parts. The requirement for this was that the master boundary is smooth. For the case of independently meshed boundaries we do not have this property. Before we consider such cases, we first consider the problem of computing the integral of the deformable body in contact with the simplest possible smooth boundary: a straight line. In this specific case geometric compatibility is guaranteed and hence, it is possible to approximate the integral within the accuracy of the integration algorithm to be employed.

### Numerical integration schemes

Numerical integration schemes are typically characterised through the general integral problem:

$$\int_{-1}^{1} f(\eta)\,\mathrm{d}\eta = \sum_{i=1}^{N_p} w_i f(\eta_i). \tag{4.6}$$

Where the integral is approximated by a set of function evaluations at a number of specific locations $\eta_i$, known as integration points, multiplied by a series of weights $w_i$. The more general case of an integral over a domain $[a, b]$ is handled by a simple reparametrisation $x = a + \frac{b-a}{2}(\eta + 1)$. In this case, we get:

$$\int_{a}^{b} f(x)\,\mathrm{d}x = \int_{-1}^{1} f(\eta)\frac{\mathrm{d}x}{\mathrm{d}\eta}\,\mathrm{d}\eta = \frac{b-a}{2}\int_{-1}^{1} f(\eta)\,\mathrm{d}\eta. \tag{4.7}$$

Several different methods are shown in Table 4.1. The parameter $\bar{\eta}$ is a point in the interval $[-1, 1]$.

| Rule Name | $N_p$ | $w_1$ | $w_2$ | $\eta_1$ | $\eta_2$ | Accuracy |
|---|---|---|---|---|---|---|
| Gauss-1 | 1 | 2 | | 0 | | $\frac{1}{3}\frac{\mathrm{d}^2 f}{\mathrm{d}\eta^2}(\bar{\eta})$ |
| Gauss-2 | 2 | 1 | 1 | $\frac{-1}{\sqrt{3}}$ | $\frac{1}{\sqrt{3}}$ | $\frac{1}{15750}\frac{\mathrm{d}^4 f}{\mathrm{d}\eta^4}(\bar{\eta})$ |
| trapezoidal | 2 | 1 | 1 | -1 | 1 | $-\frac{2}{3}\frac{\mathrm{d}^2 f}{\mathrm{d}\eta^2}(\bar{\eta})$ |

Table 4.1: Several numerical integration methods

One look at this table, suggests that one would use a one-point Gaussian integration scheme for data which needs to be second order accurate and a two point Gauss rule if more accuracy is required. Clearly, the nodal (i.e. trapezoidal) integration scheme is the least efficient, since it requires two points evaluated for only second order accuracy. Still it is the most widely applied rule in contact mechanics.

### Integration rule-geometry coupling

When the interpretation of the integral is brought back into the analysis, a conclusion as reached in the previous section is not valid. First assume that a 1-D Gaussian integration scheme is used to approximate the normal contact integral. In that case, a geometrically allowable solution involving this integral may have the configuration depicted in Figure 4.3.

In this figure, the bottom part is the proposed smooth boundary contour being a line. Above that, a slave boundary is depicted. Alternatively, one could think of the elements as beam elements. The configuration in the figure satisfies the contact conditions under the integration rule. Upon application of an integration rule, one enforces the impenetrability condition in a finite number of points, which are exactly the integration points. It is clear from the picture that the impenetrability conditions are satisfied in these integration points.

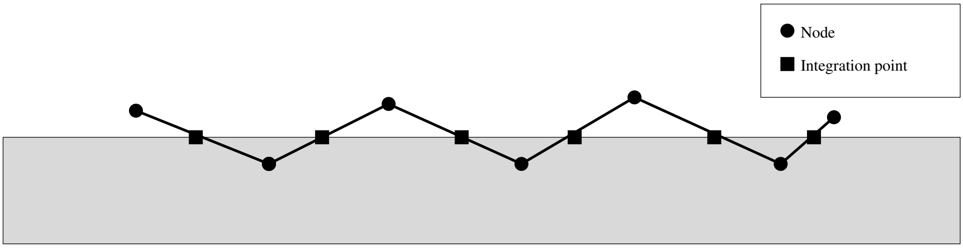


Figure 4.3: Example of possible problem with 1-points Gauss rule

What went wrong, is that the integration points do not sufficiently suppress the geometrical freedom of the slave boundary. The situation is comparable to that in incompressibility when zero energy modes arise. So although the contact integral may be computed within a second order accurate result using the 1D Gauss rule, it does so with respect to the configuration. And as can be seen from the example, the configuration is not always acceptable.

The location of the integration points determines the locations where contact is enforced. Employing a 2-points Gauss rule, improves the situation, as depicted in Figure 4.3, but does not completely solve it: It is still possible to have penetration without the algorithm noticing it. The only algorithm that enforces the contact constraints correctly for this example is nodal integration.

**Constraint counting**

There is an additional possible problem with the 2 points Gaussian rule that we wish to mention, and that is the problem of normal traction oscillations. Each integration point introduces a constraint as was already mentioned. In turn, each boundary node introduces a degree of freedom. When there are as many degrees of freedom as there are constraints, then the situation is stable, and sufficiently accurate. For $n$ boundary segments, there are $n+1$ degrees of freedom (in the $x$-direction). For a 1-point Gauss rule there are $n$ constraints (one integration point for each boundary segment). Consequently, there is one degree of freedom left, which can cause the zero energy mode.

The 2-points Gauss rule introduces $2n$ constraints. Which are $n-1$ constraints too many. This results in the property that the contact constraints can divide the normal pressures in an arbitrary way to come up with a valid solution for the aggregated normal tractions in the node. This can lead to normal pressure oscillations. The trapezoidal rule has exactly as many degrees of freedom as there are constraints.

The example in the previous section plus the constraint counting argument leads us to select the nodal integration rule.

### 4.3.2 Integrating complex master boundaries

In Section 4.3.1, the nodal integration scheme was selected as the most appropriate to compute the contact integral with respect to a simple smooth geometry: a line. In this section, we discuss what additional measures have to be taken if more complex geometries are involved.

### Smooth geometries

When dealing with smooth geometries, we nearly automatically no longer have geometric compatibility. An additional problem is that by selecting an integration rule a-priori, it becomes nearly impossible to avoid penetration from the smooth master body into the slave. As an example consider again the contact of a roll with a plate, and no matter which integration rule is selected, there is always possible penetration of the roll into the element.

This does not necessarily poses a real problem though. Even if penetration is possible, this will always be within the interior of a boundary segment with the nodal integration case. Since the linear boundary segment is only an approximation of the actual smooth boundary curve, the overlap can be either smaller or larger than observed in the simulation. The important point here is that the most accurate locations in the simulation, the nodes, are at least correctly constrained.

### Non-smooth geometries

The same problem of geometric incompatibility and the a-priori selection of constraint location holds for non-smooth geometries. The interpenetrations can also be larger.

Sometimes a two-pass procedure is proposed to overcome the problem (See Chabrand et al., 2001). In a two-pass procedure, first boundary A is assumed to be the master and boundary B the slave, and next boundary A is assumed to be the slave and boundary B the master. The average of the two integrals is then set to be the value of the contact integral.

Although this can be geometrically advantageous in some situations, in others, it can actually be worse. A definite drawback, however, is again the possible oscillation of the contact normal pressure over the contact boundary. This oscillation arises from the arbitrary distribution of pressures over the two integrals, since they are in large parts clearly interdependent. An example of a geometric advantageous situation for a two-pass procedure versus a one-pass procedure is illustrated in Figure 4.4.

On the left hand side of the figure the two-pass result is illustrated. On the right hand side the one-pass result or direct evaluation of the contact integral is given. Due to the more expensive evaluation time of a two-pass procedure, and the fact that it may improve the geometric configuration, but may deteriorate the quality of the normal pressure, a decision is made to stick to the one-pass nodal integration rule.

An important remark to be made on the example that is shown in Figure 4.4 is that the discretisation is really poor. If one would want to approximate the sliding of a block, the discretisation should be much finer near the corner point. In that case the computational differences between a one-pass and two-pass procedure are really only limited to one element.

Other methods claiming better results by employing a separate discretisation of the interface are the mortar finite element methods, such as discussed in Belgacem et al. (1998), Hild et al. (1998) and McDevitt and Laursen (2000).

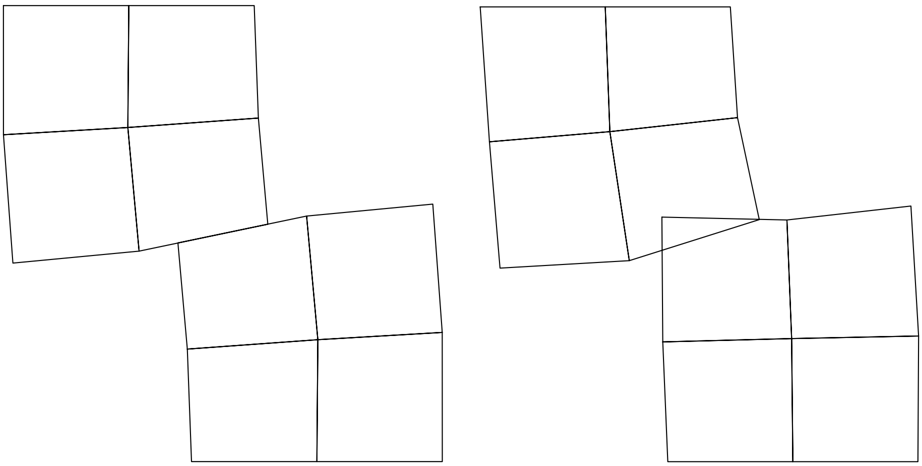


Figure 4.4: Example of a geometric configuration where a 2-pass procedure is better than a 1-pass procedure.

### 4.3.3 Piecewise linear boundaries: problems in sliding

In the previous sections, we motivated the selection of a trapezoidal integration rule for one-pass master slave 2D contact problems, even if the geometric incompatibilities give rise to penetrations. In this section we come back to the smoothness problems when the master mesh is discretised. In this case we need to consider both the non-smoothness with respect to the boundary parametrisation, as well as the non-smoothness with respect to the fundamental unknowns in the vector $x$.

We start the discussion of the latter. In Figure 4.5 on the left, part of a discretised master boundary is shown, along with the path of a node in a possible simulation. On the right, the distance of the function is sketched, with respect to its trajectory parametrisation.

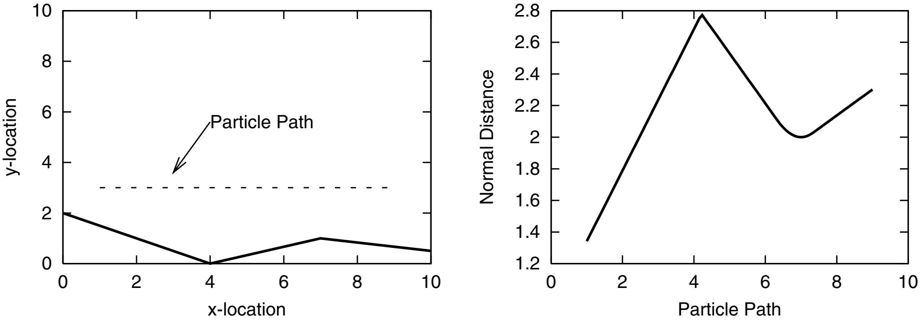


Figure 4.5: Non-differentiabilities in the discrete setting

In the figure two distinct transition points can be noted. The first transition point is when the projection of the slave node changes from the first to the second segment. This change of projection is abrupt, and has a discontinuous gradient associated with it. The second transition point is when the projection of the slave node changes from the second to the third segment, and this transition appears smooth. For the three-dimensional case likewise transitions appear along edges.

The abrupt changes in the gradient do not necessarily pose a problem in the computation of the gradient as was already discussed in Section 4.2.3. It can, however, distort the convergence of the Newton method used to iteratively solve the simulation. Due to the sudden change of projection the approximations can either oscillate around the actual solution, or convergence can be painfully slow. Using line-search can overcome the first problem, but the global convergence can remain very slow.

When dealing with smooth boundaries, the situation improves when approaching the contact boundary. Due to finite curvature, there is a distance after which the distance function is smooth with respect to a trajectory parametrisation, as long as the trajectory stays within that distance. However, in a discretisation, we do not have a finite curvature at the node locations of the master. No matter how close the node approaches the master curve, the gradient will remain discontinuous. This property holds for all the convex sections in the master boundary

Interestingly, the convex parts seem to be smooth. This is only so in appearance, however. The closer the slave node moves to the master node, the shorter the transition length will become. Upon touching the master node, it will even be 0. At that point the gradient is discontinuous, and remains so after penetration. (As was to be expected, since the convex and concave parts are complementary).

To stabilise the global simulation procedure, a projection will only be made once per step: at the beginning of the step. This will solve the non-smoothness with respect to variables in $x$.

### 4.3.4 Piecewise linear boundaries: problems in computing

As was already stated, the non-smoothness of the distance function with respect to the boundary parametrisation only influences the accuracy of the final (geometric) solution, but does not directly influence the rate of convergence.

In Section 4.3.1, the integration scheme for a single integral was presented. In Section 4.2.3, the boundary integral was split into assumed locally smooth parts. We again state the integral here:

$$\int_{\Gamma^h} \mathrm{t}_\mathrm{N} \nabla_x \mathrm{d}_\mathrm{N} \, \mathrm{d}\Gamma = \sum_e \int_{\Gamma^h_e} \mathrm{t}_\mathrm{N} \nabla_x \mathrm{d}_\mathrm{N} \, \mathrm{d}\Gamma. \tag{4.8}$$

The integral can then be approximated by:

$$\sum_e \frac{h_e}{2} \left( \mathrm{t}_{\mathrm{N},e_1} \nabla_x \mathrm{d}_{\mathrm{N},e_1} + \mathrm{t}_{\mathrm{N},e_2} \nabla_x \mathrm{d}_{\mathrm{N},e_2} \right), \tag{4.9}$$

where use was made of the trapezoidal rule. In this, $e_1$ is assumed to be one of the two end-points of the segment, and $e_2$ is assumed to be the other end point.

In the smooth case, the trapezoidal rule can be chained for obtaining a more efficient evaluation, since the end-points of the different integrals coincide. However, due to the non-smoothness with respect to the boundary parametrisation this is no longer necessarily so. For an illustration consider Figure 4.6.

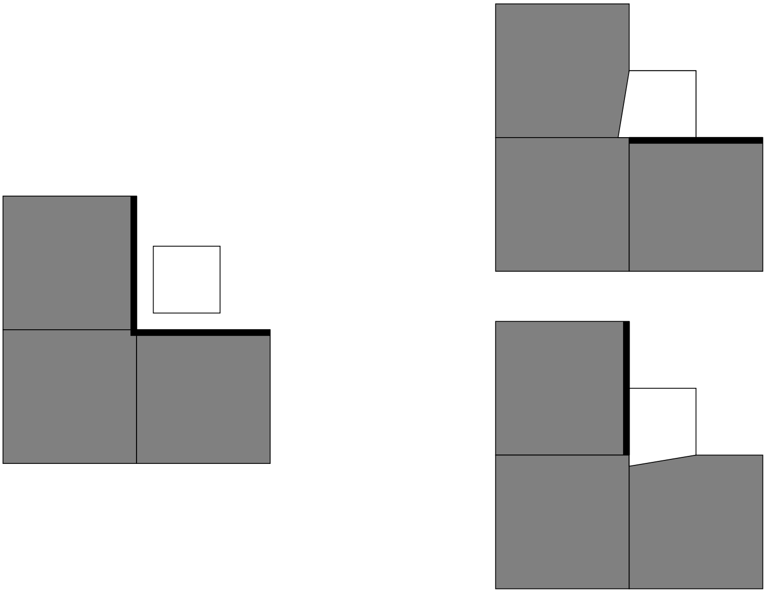


Figure 4.6: The corner problem with nodal integration and only one projection

If a chained trapezoidal integration rule would be employed, then only one projection for the lower left corner of the white box in the figure can be chosen. In the left of the figure the possible projection segments are given by a fat line. In the picture to the top right, the final solution is presented if the lower segment is chosen as the projection location. And in the picture to the bottom right the final solution is presented if the left segment is chosen as the projection location. Both selections do not deliver the correct final solution.

The reason for this is, that the lower left corner of the white box needs to project both on the lower as well as on the left segment. The necessity is driven by the non-smoothness of the distance function with respect to the boundary parametrisation.

### 4.3.5 Piecewise linear boundaries: projecting

To make sure the problems illustrated do not occur, we present a projection scheme for non-smooth linear boundaries that assures us that the integral is approximated accurately.

The way to solve this is by considering each slave integral by itself. Thus, by applying an un-chained trapezoidal rule on a possibly discontinuous function. Each slave node can then be projected multiple times on the master boundary: once for each slave segment it belongs to. In this case, a node is associated with the normal of the segment for which it is to function as an integration point. A node projects on a master segment only if the normal associated with the integration point, is in the opposite direction from the normal pointing away from the master segment. An illustration of the process for a 2-dimensional case is given in Figure 4.7. In this figure a part of a slave boundary and part of a master boundary are shown. We show how the node $q_3$ of the slave boundary is to be projected on the master boundary. Note that the node $q_3$ is part of both the edge $(q_2, q_3)$ as well as the edge $(q_3, q_4)$. Both cases are to be considered separately. First the nearest projection location on the master boundary is to be found which has an opposing normal. From the picture it can quickly be seen that this is the node $v_3$ for both cases. Now what is the projection of $q_3$ if it is the integration point for the edge $(q_3, q_4)$? For this the normals need to be checked. The normal for $(q_3, q_4)$ is drawn in the figure marked as $n$. The maximum opposing normal for $n$ of the adjacent edges of $v_3$ is that of $(v_2, v_3)$. Hence, the projection segment of $q_3$ for $(q_3, q_4)$ is $(v_2, v_3)$. A similar argumentation leads to the conclusion that $q_3$ for $(q_2, q_3)$ projects on $(v_3, v_4)$.

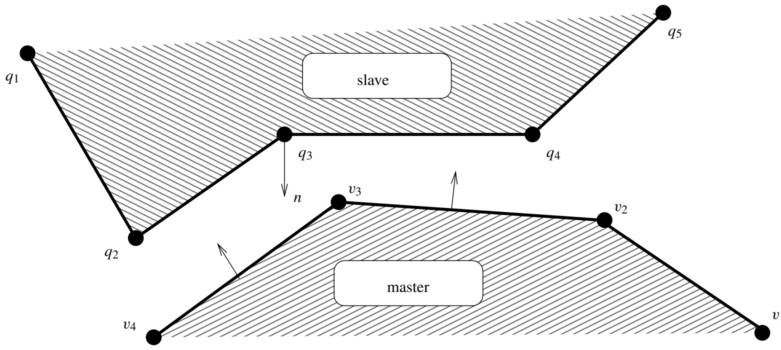


Figure 4.7: Illustration of projection of discontinuities

This method can also be used for self-contact. The method automatically rejects all faces that have normals pointing in the same direction.

Most integration points will turn out to project on the same master segment, irrespective of the segments they are attached to. In this case, we can join the integration points to save computation time. The method does require additional time in making the global projections. A complete discussion on how to find the projections efficiently is presented in Section 4.5.

### 4.3.6 Conclusions

From the above discussion, it can be seen that the trapezoidal integration rule is best for use in simulations where both bodies are approximated with piece-wise linear surfaces. The piece-wise linear boundaries introduces problems when sliding occurs and to stabilise this, projections should only be made once per step and remain fixed. Additionally, to be able to perform an accurate computation in the case of non-smoothness with respect to the boundary parametrisation, more work has to be performed to find all the projection locations.

To summarise, the reason for the problems in Section 4.2 and this section, is the non-smoothness of the distance function. By smoothing the master boundary, all these problems might be overcome. Smoothing the master boundary is the topic of the next section.

## 4.4 Smoothing the master boundary

For reasons of numerical stability, one might choose to smooth the master boundary. This will get rid of the necessity of performing a discontinuous integration scheme. Another advantage is that the accuracy of the boundary description may slightly improve with respect to the original piece-wise linear model. A disadvantage is that smoothing may decrease the efficiency of the overall method.

The most general description of a piece-wise linear master surface in 2D is by means of a polygon. For 3D the most general description is a triangulation. Smoothing this type of linear surface is considerably more complex than smoothing the 2D equivalent. If additional structure could be detected, for example if the surface is meshed with quadrilaterals in a regular fashion, then more efficient methods can be employed. We will only concern ourselves with the more general triangulations.

The topic of curves and surfaces are discussed in standard computational geometrical works, such as Farin (1988) and Piegl and Tiller (1997). The application of these smooth surfaces to overcome the discontinuity sliding problem have been investigated by El-Abbasi and Meguid (2001) for bi-cubic B-splines. Two-dimensional problems using Hermite and Bernstein polynomials have been investigated by Wriggers et al. (2001).

The layout of this section is as follows: The smoothing of polygons is the topic of Section 4.4.2. Smoothing the 3D triangulation is presented in Section 4.4.3. Finally, a short overview of the results is given in 4.4.4.

### 4.4.1 Goals to attain when smoothing

Before we can fully discuss the different methods that can be used for smoothing, something is to be said on what goals we are after when smoothing a master boundary. Smoothing a boundary can be a complex procedure, and we have to decide how smooth the result should be before we deem the surface description to have improved.

The ultimate goal of smoothing surfaces is to perform the simulation involving contact more efficiently. This sets the bounds on the selection of the available procedures. The non-smoothness of the distance function with respect to the fundamental unknowns $x$, which is equivalent to non-smoothness in sliding, is the property that is to be smoothed. This can be achieved by smoothing the boundary. So first the topic of smoothness is to be discussed.

Consider two curves $\mathbf{a}(u)$, and $\mathbf{b}(v)$ in $\mathbb{R}^2$, where both $u$ and $v$ are parameters from a parameter set. For the moment, let us assume that both $u$ and $v$ are selected from [0, 1]. The composite curve $\mathbf{c}(w)$ for $w \in [0, 2]$ is defined as:

$$\mathbf{c}(w) = \begin{cases} \mathbf{a}(w) & \text{if } 0 \leq w < 1, \\ \mathbf{b}(w-1) & \text{if } 1 \leq w \leq 2. \end{cases} \tag{4.10}$$

The composite curve is said to be continuous if $\mathbf{a}(1) = \mathbf{b}(0)$. It is said to be continuously differentiable of order one, if:

$$\lim_{w\uparrow 1} \frac{\mathrm{d}\mathbf{c}}{\mathrm{d}w} = \lim_{w\downarrow 1} \frac{\mathrm{d}\mathbf{c}}{\mathrm{d}w} \iff \frac{\mathrm{d}\mathbf{a}}{\mathrm{d}u}(1) = \frac{\mathrm{d}\mathbf{b}}{\mathrm{d}v}(0). \tag{4.11}$$

In an identical fashion higher order derivatives are defined.

However, it must be noted, that the above definition of continuity is with respect to a specific parametrisation. A parametrisation may change without affecting the shape of the curve. Consider the planar composite curve $\mathbf{c}^{(1)}$ defined as:

$$\mathbf{c}^{(1)}(w) = \begin{cases} (w, w) & \text{if } 0 \leq w < 1, \\ (2w-1, 2w-1) & \text{if } 1 \leq w \leq 2. \end{cases} \tag{4.12}$$

which is a straight line and is smooth. However, when checking the derivatives at $w = 1$, one can observe that the curve is not continuously differentiable with respect to the parameter $w$. This can easily be overcome by changing the parametrisation.

Since, the distance is a geometric invariant, the surface parametrisation should not matter. To this end a different kind of smoothness is required, one that does not depend on the parametrisation. This is achieved through the concept of geometric continuity. A curve is geometrically continuous of degree, or $G^0$ if it is connected. A curve is geometrically continuous of degree 1, or $G^1$ if the curve tangents have the same direction, but not necessarily the same magnitude. Another property of a $G^1$ curve is that it can be made $C^1$ by selecting an appropriate parametrisation. In an equivalent manner, higher orders of smoothness can be defined.

Hence, what we require is a surface that is at least $G^1$. A surface that is $G^2$ would be even better, since that would mean that the tangent matrix would even be continuous. However, constructing $G^2$ surfaces is considerably more expensive than constructing $G^1$ surfaces, and gaining time was our primary objective. From this we can set up the following demands:

- The smoothed master surface needs to be at least $G^1$.
- The construction time of the smoothing surface should be low, unless we are dealing with a fixed tool, in which case the smoothing surface needs only to be constructed once.
- The evaluation of the distance function, as well as its derivatives needs to be not too complex. Otherwise the time required in computing it, may become too great an overhead.
- Projecting a slave node on the smoothed master surface should not be too complex for the same reasons as mentioned for the evaluation of the distance function.

In the following sections these demands will guide the decisions in selecting the way in which the piecewise linear surfaces are to be smoothed.

### 4.4.2 Smoothing curves

In this section the smoothing of polygonal curves is presented. There are several ways of achieving this. The most common method, is using a spline curve. A spline curve is a piecewise polynomial curve, with a specified smoothness between the pieces. We describe the polynomial curve pieces by Bézier polynomials. The advantage of the Bézier description over the other polynomial basis functions, are their geometric properties, which will be discussed more fully.

#### Splines

The boundary is assumed to be a polygon, with vertices $v_1, v_2, \dots, v_N$. Each of the vertices has an a coordinate, which is labeled similarly as $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$. The $i$-th segment of this boundary is then a line-segment which runs from $\mathbf{X}_i$ to $\mathbf{X}_{i+1}$. Each segment of the boundary is going to be replaced by a polynomial curve.

All the polynomial curves pasted together form a piecewise polynomial curve known as a spline. The global smoothness of the spline is the minimum smoothness over each of the connections of adjacent polynomial pieces. The maximum degree of polynomial pieces is called the degree of the spline. A spline with degree $p$ and smoothness $q$ is an element of the spline space $\mathcal{S}_q^p$. It is trivial to note that $\mathcal{S}_q^p \subset \mathcal{S}_{q-1}^p$, and $\mathcal{S}_q^{p-1} \subset \mathcal{S}_q^p$. Also note that the original polygonal boundary is already a spline, namely one from $\mathcal{S}_0^1$.

#### Bézier curves

A Bézier curve is defined in terms of the Bernstein polynomials. The $i$-th Bernstein polynomial of degree $n$ is defined through:

$$B_i^n(\zeta) = \binom{n}{i}\zeta^i(1-\zeta)^{n-i}. \tag{4.13}$$

It furthermore requires a set of control points $\mathbf{P}_i \in \mathbb{R}^n$ to complete the description. The curve can now be represented by:

$$\mathbf{C}(\zeta) = \sum_{i=0}^{n} \mathbf{P}_i B_i^n(\zeta). \tag{4.14}$$

Bézier curves posses some nice properties that make them a natural choice for modelling surfaces (see Piegl and Tiller (1997)):

- Convex hull property: The curves are contained in the convex hulls of their defining control points: (the $\mathbf{P}_i$). This property is of importance in Section 4.5, that is why it is mentioned here.
- End point interpolation: $\mathbf{C}(0) = \mathbf{P}_0$, $\mathbf{C}(1) = \mathbf{P}_n$.
- The $k$-th derivative at $\zeta = 0$ ($\zeta = 1$) depends on the first (last) $k+1$ control points. In particular $\mathbf{C}'(0)$ and $\mathbf{C}'(1)$ are parallel to $\mathbf{P}_1 - \mathbf{P}_0$ and $\mathbf{P}_n - \mathbf{P}_{n-1}$ respectively.

Using the last two properties, we can thus produce a globally order one smooth cubic interpolating Bézier spline, which has only local support. The spline is thus from the space $\mathcal{S}_1^3$.

The local polynomial curve for the $k$-th segment takes the form:

$$\mathbf{C}_k(\zeta) = \sum_{i=0}^{3} \mathbf{P}_i^{(k)} B_i^3(\zeta), \tag{4.15}$$

where $\zeta$ is a local coordinate which runs from 0 to 1, and the global curve $\mathbf{C}(\xi)$ is the union of all these curves. To form the $k$-th local curve segment, we require the points $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$, as well as the average outward unit normals in these points $\mathbf{n}_k$ and $\mathbf{n}_{k+1}$. The normals are computed by averaging the normals to the adjacent line segments of the point. For the begin and end point no averaging is required.

### Construction of the curve

We now think of the line segment as if it were aligned with the $x$-axis, and assume that a parameter $\zeta$ runs along it from 0 to 1. At 0 it is at $\mathbf{x}_k$, and by the end point interpolation property, $\mathbf{P}_0^{(k)}$ must thus be $\mathbf{x}_k$. In an identical manner we find that $\mathbf{P}_3^{(k)} = \mathbf{x}_{k+1}$. We now assume that above $\zeta = \frac{1}{3}$ there lies $\mathbf{P}_1^{(k)}$ and above $\zeta = \frac{2}{3}$ there lies $\mathbf{P}_2^{(k)}$. The exact location of $\mathbf{P}_1^{(k)}$ is computed through the intersection of the line orthogonal $\mathbf{n}_k$ and the line orthogonal to the $k$-th line segment and emanating from $\frac{2}{3}\mathbf{x}_k + \frac{1}{3}\mathbf{x}_{k+1}$. In an identical fashion we find the point $\mathbf{P}_2^{(k)}$. The idea is illustrated in Figure 4.8.

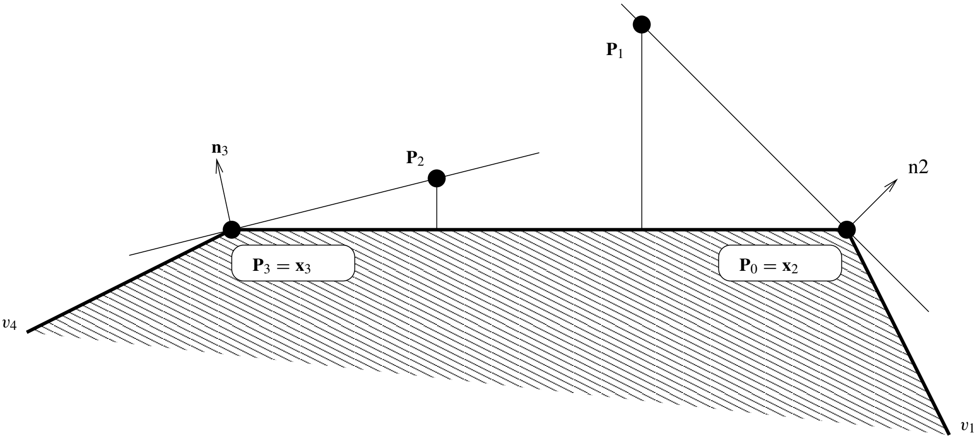


Figure 4.8: Illustration of Bézier approximation of line segment

From the construction it can be seen that the curve is continuous, since each curve piece interpolates the nodes at its end points. Furthermore, we can see that the derivatives of adjacent curve pieces lie along the same direction by the derivative property. Specifically, the derivative of the local curve segments take the following form:

$$\frac{\mathrm{d}\mathbf{C}_k}{\mathrm{d}\zeta}(\zeta) = \sum_{i=0}^{2} 3\left(\mathbf{P}_{i+1}^{(k)} - \mathbf{P}_i^{(k)}\right) B_i^2(\zeta). \tag{4.16}$$

As a result, the curve is geometrically smooth of order 1 as was required.

**Conclusions**

As can be seen, the construction of a spline, using Bézier polynomials is relatively easy. By using cubic basis functions the description is even completely local, which makes for efficient evaluation. This type of efficiency is absolutely necessary if both curves are the boundary of a body that is modelled as possibly deformable. In the next section we see how smoothing can be performed on three-dimensional triangulated boundaries.

### 4.4.3 Smoothing 3-dimensional surfaces

In this section the smoothing of unstructured triangular meshes is presented. Such an unstructured mesh may form the boundary of a solid body, or a plate whose deformation is studied. If the body is itself deformable, then performing a smoothing algorithm should not be too costly with respect to the time required to perform an iteration. Since the smoothing needs to be applied at each such iteration, the cost will then rise quite quickly. As a result, constructing the curve should be done locally, whereas the spline should be at least order 1 smooth, otherwise the work is for naught. Constructing a globally smooth surface, using local interpolants, can be achieved by using a quintic Bézier basis function on each of the individual triangles, and applying geometric constraints to their control points. How this is done, is the topic of this section.

**Splines in 3D**

The boundary in the three-dimensional case is assumed to consist of triangles. A triangulation is constructed from vertices $v_1, v_2, \ldots, v_N$. Each of the vertices has a coordinate, which is labeled as $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ as in the two-dimensional case. A triangle $T$ in the triangulation consists of three different vertices $v_{1_T}, v_{2_T}, v_{3_T}$. Each of the triangles in the triangulation is to be replaced by a smooth polynomial surface patch.

Again, pasting all the patches together, gives rise to a piecewise polynomial surface, which is also called a spline. The same smoothness requirements hold as in the two-dimensional case. In this manner, we can see that a proper triangulation, is a spline from $\mathcal{S}_0^1$, and our aim is to replace this spline by one from $\mathcal{S}_1^p$.

**Barycentric coordinates**

A topic that was not discussed in Section 4.4.2, but used nonetheless is that of barycentric coordinates. Barycentric coordinates are interior coordinates on a simplex. In the three-dimensional case, a simplex is a triangle. Points on the triangle can be either given by their

Cartesian coordinates, or by a local coordinate system relative to the coordinates of the vertices.

The local coordinates are given by the vector $\boldsymbol{\lambda}$, whose components are defined as follows:

$$\mathbf{x}\lambda_1\mathbf{x}_1 + \lambda_2\mathbf{x}_2 + \lambda_3\mathbf{x}_3, \tag{4.17a}$$

$$0 \le \lambda_i \le 1, \tag{4.17b}$$

$$\sum_i \lambda_i = 1. \tag{4.17c}$$

From the above, it can be concluded, that although there are three distinct numbers given for the local coordinates, there are only two really independent ones. This will cause all points described with barycentric coordinates to lie in the plane of the triangle.

Comparing this with what we did for the two-dimensional coordinate $\zeta$ running from 0 to 1 is the second barycentric coordinate for the two-dimensional simplex which is a line segment.

### Bézier patches

A Bézier patch is given with respect to the triangular Bézier basis function. The triangular Bézier basis functions are given with respect to the barycentric coordinates over the triangle. The $(i, j, k)$-th triangular basis function of degree $n$ is given by:

$$B^n_{ijk}(\boldsymbol{\lambda}) = \frac{n!}{i!j!k!}\lambda_1^i\lambda_2^j\lambda_3^k, \tag{4.18}$$

where $i + j + k = n$, and $i$, $j$ and $k$ are all 0 or larger.

The local patch is now given by:

$$\mathbf{S}(\boldsymbol{\lambda}) = \sum_{i+j+k=n} \mathbf{P}_{ijk}B^n_{ijk}(\boldsymbol{\lambda}). \tag{4.19}$$

The coefficients $\mathbf{P}_{ijk}$ are called the control points, and are elements of $\mathbb{R}^3$. The triangulation of the control points in 3D forms the control net. The control net is an approximation to the patch given in (4.19).

As with their 2D counterparts, Bézier patches posses some nice properties which makes their use convenient:

- Convex hull property: The surface is contained in the convex hull of the defining control points. This property is used in Section 4.5.
- Vertex interpolation:

$$\mathbf{S}(1,0,0) = \mathbf{P}_{n00}, \tag{4.20a}$$

$$\mathbf{S}(0,1,0) = \mathbf{P}_{0n0}, \tag{4.20b}$$

$$\mathbf{S}(0,0,1) = \mathbf{P}_{00n}. \tag{4.20c}$$

Furthermore, the boundary curve over an edge only depends on the control points defined over that edge.

- The tangent planes over the boundaries are coplanar with the triangles of the Bézier net that are edge incident with the boundaries.

These properties are more easily explained, when illustrated with a picture. In Figure 4.9 the control point locations for a 5th degree Bézier patch are shown when projected onto the parent triangle. Above each of these locations a control point is specified. The boundary curve along the left edge of the illustrated triangle only depends on the control points specified by locations 500, 401, 302, 203, 104 and 005. In an identical fashion, the boundary curve over the lower edge of the triangle only depends on the control points for locations 050, 140, 230, 320, 410 and 500. Thus if two patches are to connect continuously, then the control points over their shared edge should coincide.

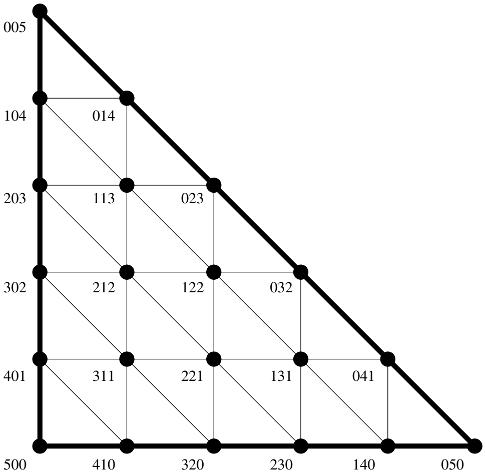


Figure 4.9: The control point locations for the 5th degree Bézier basis functions.

The continuity conditions are illustrated for two adjacent degree 4 Bézier patches in Figure 4.10. In this figure, we see two triangles drawn. The triangle with as vertices $v_1$, $v_2$, $v_3$ has a Bézier net with coefficients $\mathbf{P}_{ijk}$ and is of degree 4. The triangle with vertices $v_4$, $v_2$, $v_3$ also has a smoothing polynomial of degree 4 defined on it, with coefficients $\mathbf{Q}_{ijk}$.

The smoothness conditions from Lai (1997) now tell us that for the patches to join $G^1$ continuously, the control net triangles that are edge incident with the joining edge of the parent triangle need to be coplanar. The exact conditions for the triangle presented in Figure 4.10 are then for $G^0$ continuity:

$$\begin{aligned}\mathbf{P}_{040} &= \mathbf{Q}_{004},\\ \mathbf{P}_{031} &= \mathbf{Q}_{013},\\ \mathbf{P}_{022} &= \mathbf{Q}_{022},\\ \mathbf{P}_{013} &= \mathbf{Q}_{031},\\ \mathbf{P}_{004} &= \mathbf{Q}_{040}.\end{aligned}$$

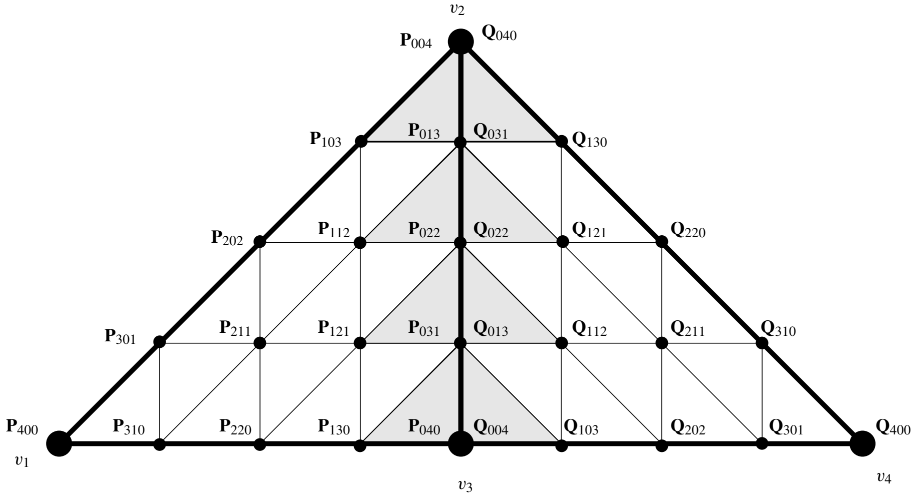


Figure 4.10: Smoothness condition

The $G^1$ continuity conditions are:

- $(\mathbf{P}_{130}, \mathbf{P}_{040}, \mathbf{P}_{031})$ is coplanar with $(\mathbf{Q}_{004}, \mathbf{Q}_{103}, \mathbf{Q}_{013})$.
- $(\mathbf{P}_{121}, \mathbf{P}_{031}, \mathbf{P}_{022})$ is coplanar with $(\mathbf{Q}_{013}, \mathbf{Q}_{112}, \mathbf{Q}_{022})$.
- $(\mathbf{P}_{112}, \mathbf{P}_{022}, \mathbf{P}_{013})$ is coplanar with $(\mathbf{Q}_{022}, \mathbf{Q}_{121}, \mathbf{Q}_{031})$.
- $(\mathbf{P}_{103}, \mathbf{P}_{013}, \mathbf{P}_{004})$ is coplanar with $(\mathbf{Q}_{031}, \mathbf{Q}_{130}, \mathbf{Q}_{040})$.

### Construction of the surface

We are going to use the above conditions to locally construct Bézier patches that are globally smooth. The degree of the patch that is used, is 5. This seems quite high, compared to the degree 3 curves as were employed in Section 4.4.2. However, it is the minimum degree for which a local solution is possible. This can be seen by repeating the exercise below for a lower degree triangle.

The control points will be referred to only by their numbers, and we assume a configuration as in Figure 4.9. The control point 500 is associated with $v_1$, 050 with $v_2$ and 005 with $v_3$. First we compute the normals for the triangle at each of the vertices $v_i$ by averaging the normals over each of the triangles that are incident with the corner node. This gives normals $\mathbf{n}_1$, $\mathbf{n_2}$ and $\mathbf{n}_3$ in each of the corner nodes. The patch should interpolate the vertices (for continuity) and the normals (for smoothness). So due to the property that the derivatives over the boundaries are coplanar with the Bézier net triangles, we end up with fixing: $\mathbf{P}_{500} = \mathbf{x}_1$. Also, because the normal is to be interpolated, $\mathbf{P}_{410}$ and $\mathbf{P}_{401}$ are chosen thus, that the plane defined by the three points is orthogonal to the average normal. In an identical fashion the control points 005, 104 and 014 are fixed as well as 050, 140 and 041.

We now note that due to the smoothness conditions, the control points at 410, 320 and 311 need to be coplanar with the triangle attached to the displayed one along the edge $(v_1, v_2)$. Also, 401, 302 and 311 need to be coplanar with the triangle attached along the edge $(v_3, v_1)$. Both the constraints share the control point located at 311. This means that the system is no longer local, unless some scheme is fixed for choosing the control point at 311. The scheme we propose is to fix the control points at 320, 311 and 302 such that the 410, 320 and 311 is also coplanar with 500, 410 and 401. Symmetrically, 401, 311 and 302 are also selected coplanar with 500, 410 and 401. Applying the same scheme later for the triangle below and to the left, will render the transition smooth of the required degree. Applying the same scheme in the other corners leaves us with only three control points to compute: 221, 122 and 212.

The remaining 3 control points are selected by demanding smoothness over the edge. The approximation for the normal to the edge tangent plane is computed by averaging the normals of the two connected triangles along that edge. Subsequently, the projection of the normal onto the boundary edge is subtracted to form a new normal which is orthogonal to the edge. The control point at 212 is now chosen such that the Bézier net triangle 302, 212, 203 is orthogonal to the corrected normal. This scheme yields a method that, when applied to each individual triangle, produces a spline surface that is $G^1$.

### Example

An example of the scheme applied to a data set describing a human scalp, is shown in Figure 4.11. In the left figure, the original triangulation is shown for the head. In the right figure a refinement of the triangulation is given, where the refined points are located on the smoothing surface. Sharp corners, where adjacent normals have an inner product of less than 0.7 are omitted from the averaging process. This leaves the sharp corners at the bottom of the head and on the nose. As can be seen, the nose remains angular, and from this we can conclude that the original mesh was too rough.

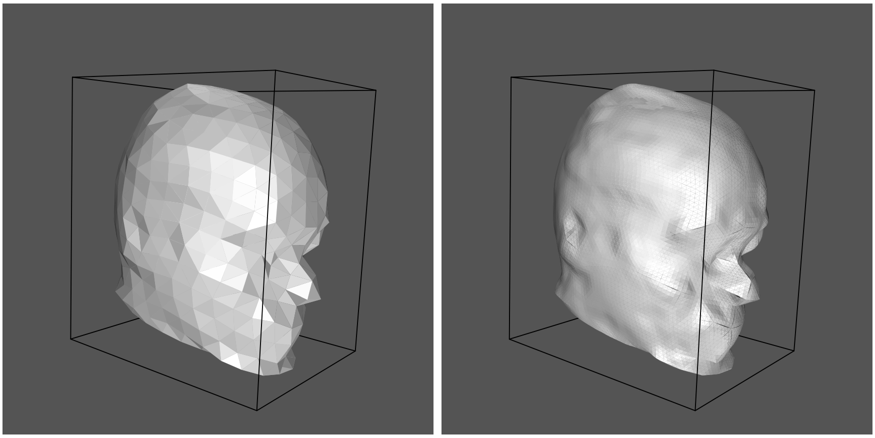


Figure 4.11: The smoothing of a head data set

### 4.4.4 Conclusions

The two dimensional smoothing problem turns out to be (relatively) efficient in terms of finding the projection point and constructing the curve.

It can be seen that in comparison with the two-dimensional case, the three-dimensional smoothing procedure is costly. This is so, despite the fact that the procedure is completely local. The reason is that a quite high degree polynomial is required to be able to keep the construction local. If a lower degree polynomial is used, a global system of equations needs to be solved, which makes things even more complex. This is not a problem when the body being smoothed is rigid. However, if the body is deformable, then smoothing needs to be performed at each iteration of the simulation, which is then prohibitively costly.

Apart from the construction cost, the localisation of the projection point is also significantly more expensive for the three-dimensional case than it is for the two-dimensional one. The reason for this is that a full fledged constrained optimisation problem has to be solved for each integration point. This makes things even more costly. After some simple tests, it turns out that the additional stability of the method does not outweigh the additional iterations required for the non-smoothed case. What was observed, however, is that the time required in projecting the integration points is a significant with respect to the total simulation time.

## 4.5 Efficient projection finding

In this section a new method is presented for performing a global search query. After application of this method, a local search still has to be performed to find the actual projection face. We use the term face here, so that the two and three dimensional cases can be discussed within one framework. A face is basically a $n_d - 1$ dimensional simplex, which in our discussion boils down to and edge for the two-dimensional problem and a triangle for the three-dimensional one.

In a finite element simulation, a significant amount of time can be spent on finding the projection face for each of the constrained points[1]. For the single point projection problem the solution procedure is relatively easy: First compute the *absolute* distance with respect to each of the master faces. Secondly select the master segment which has the smallest absolute distance as the projection face. The difference between projected distances and absolute distances is illustrated in Figure 4.12.

The projected distance has a sign, which depends on the chosen normal direction for the boundary under consideration. It is then the distance with respect to the line through the line segment and, depending whether it is above or below that line with respect to the normal, is signed. The real distance is the actual distance with respect to the line segment. It is not signed, since it is impossible to determine on which side the point is lying. It expresses a true measure of distance with respect to the segment, but is not usable as a constraint measure, since there is no sign to tell us whether there is penetration or not. However, it is useful and necessary in determining which line segment should be chosen to compute the projected distance for.

[1] Note that we do not use the term nodes here, since the constrained point may depend on the integration algorithm

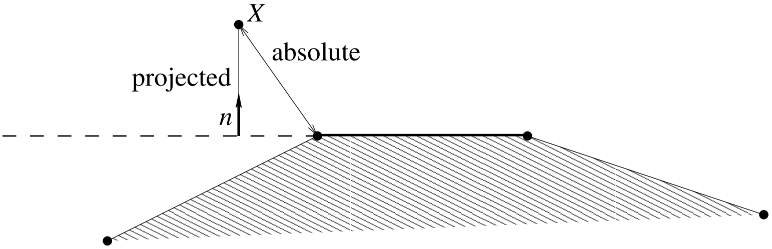


Figure 4.12: Difference between projected distance and absolute distance

Returning our attention to the single point projection problem: Assume that there are $N$ master faces. In that case we require an amount of operations to find the projection face which lies in the order of magnitude of $N$. Increasing $N$ by a factor of two, would approximately increase the amount of computation time by a factor of 2. For this we denote that the single point projection algorithm costs order $N$ operations, or more concisely $O(N)$ operations.

Applying the one point projection algorithm to a set of $M$ points leads to an algorithm that requires $O(MN)$ operations. If $M$ lies in the order of magnitude of $N$, then we could also write $O(N^2)$. For typical large scale problems, the matrix solution time for a direct solver lies in the order of $O(Nk^2)$ where $k$ is the bandwidth of the problem. Since, $N$ clearly dominates $k$ and even $k^2$, this would mean that the total simulation time is dominated by the time which is required to find the projections. This is unacceptable. To improve things, we can set up a search data structure on the faces. If the cost of creating this data structure is less than $O(N^2)$, and if it would bring down the search time for a single query to less than $O(N)$, then the time necessary to find all the projections using this data structure is less than $O(N^2)$.

Due to the possible time requirements of the projection problem, a small body of literature for solving it has been established. Each article arrives at a solution procedure from first principles. Instead of attempting to solve the problem directly, one could also look for problems that are similar in structure and have an efficient algorithm to solve them. Modifying the projection problem slightly so that it resembles an already solved problem then becomes our goal. Similar looking problems can be found in the field of computational geometry.

To apply the available geometric algorithms requires the split of the problem into two stages: first a global search is made using modified computational geometrical algorithms. The result of this global search is a set of faces which lie in the vicinity of the point to project. Subsequently a local search is made within this set of candidate faces to determine which is the actual projection face.

Depending on the structure of the problem, the result of the global search may in fact contain (a significant portion of) all the segments. This can occur in problems in which there is a large difference in magnitude between the smallest and largest element, or when the step-size is very large with respect to the average element size. But usually, the method is very efficient.

### 4.5.1 The capture box problem

In this section the original projection problem is moulded into a different form. The intent of this is to be able to apply an algorithm which is based on answering queries of the form: Given a point $q$, which intervals contain that point $q$?

We commence by forming the bounding box for each of the faces. A bounding box of a face, is the minimum sized box, aligned with the coordinate axes which contains the face.

Let us assume for a moment that the master faces are fixed. Furthermore, we make an estimation of the maximum displacements on any of the slave nodes for each of the coordinate directions. This gives rise to $\Delta\mathbf{x}_{\text{max}}$. These maximum displacements are used to enlarge the bounding box in each direction to form a capture box. The process is illustrated in Figures 4.13 and 4.14.

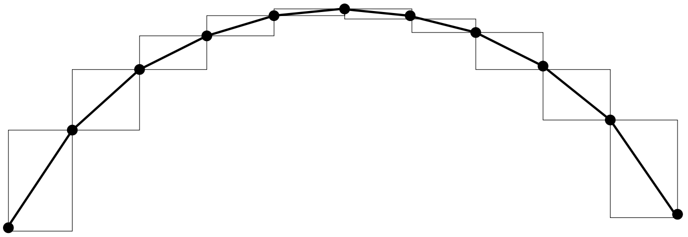


Figure 4.13: Illustration of bounding box.

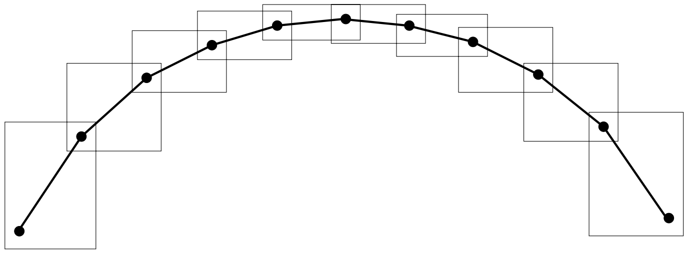


Figure 4.14: Extension to capture box.

The rationale behind the choice of this size of box is as follows. If the point is inside the capture box, it *may* come into contact with the face that corresponds to the capture box. If the point is outside the capture box, it can never enter the bounding box during the step. Thus, it can never come into contact with the face that is contained within the bounding box.

When given a point, we wish to select those capture boxes that contain it. These capture boxes contain the candidate projection faces. Which one of these candidate faces is the actual projection face is to be determined by a local search strategy. A good candidate for the local search is the brute force approach, since the total number of candidates is small.

In case the master surface is non-deformable, but is free to move, then this movement can be incorporated into the size of the capture box. Additionally, if the master surface is even free to deform, then we add twice the maximum displacement to the capture box. For accurate forming simulations, the maximum displacements do not usually exceed the size of the elements. Thus, the boxes are typically no more than three times the size of the original elements.

### 4.5.2 Building the data structure

We are going to solve the capture box problem presented in the previous section by a divide and conquer approach. First a coordinate direction is chosen. In this direction, the boxes in the total set of boxes $\mathcal{I}$ are sorted on their right end point. Next the median of these right end-points is selected. The median divides the set of boxes in three parts:

- There are boxes which lie fully to the left of the cutting plane specified through the median. These form the set $\mathcal{I}_{\text{left}}$.

- There are boxes which lie fully to the right of the cutting plane, these fall into the set $\mathcal{I}_{\text{right}}$.

- There are some boxes which intersect the cutting plane, these are placed in $\mathcal{I}_{\text{mid}}$.

The three sets are obviously disjoint, and together they again form the set $\mathcal{I}$.

The selection situation is shown in Figure 4.15. The boxes in bold belong to the set $\mathcal{I}_{\text{mid}}$.

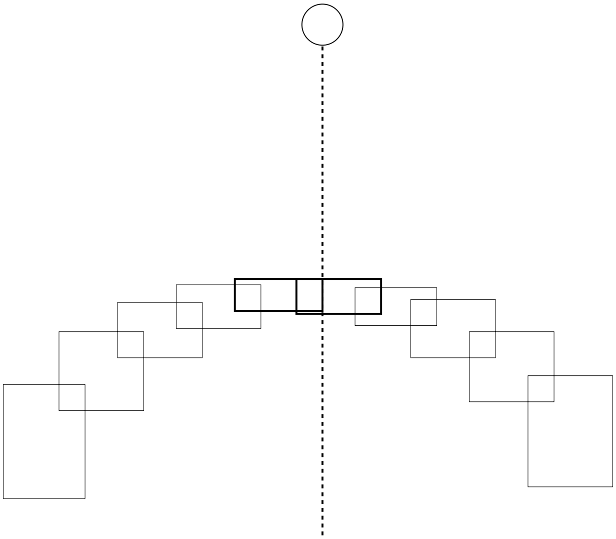


Figure 4.15: First subdivision example.

We are building a tree on the complete set of boxes $\mathcal{I}$. The root partitions the set into the three mentioned sets. The left child is to further contain a data structure of the boxes in $\mathcal{I}_{\text{left}}$. Equivalently the right child is to contain a further data structure on the boxes in $\mathcal{I}_{\text{right}}$. At the node itself, we store the set $\mathcal{I}_{\text{mid}}$ in a so-called associated data structure.

The left set and right set of boxes do not overlap each other in space, and are unstructured sets, just like the set $\mathcal{I}$ was originally. Hence, we can again repeat the process on the left and right child to obtain sets $\mathcal{I}_{\text{left}_{\text{left}}}$, $\mathcal{I}_{\text{left}_{\text{mid}}}$ and $\mathcal{I}_{\text{left}_{\text{right}}}$. In an identical manner the set for the right child is subdivided. As an illustration consider Figures 4.16 and 4.17 the tree is shown after the first and second subdivision step on the first coordinate direction.

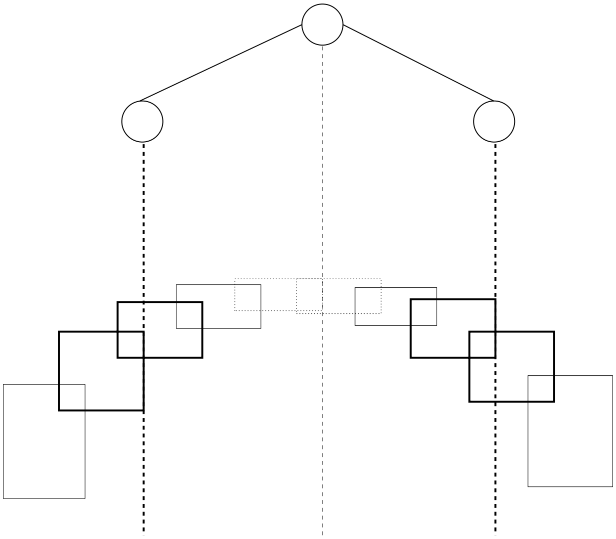


Figure 4.16: Second subdivision example.

As can be seen the data structure is a recursive one. Each node contains a set of boxes that are split on the first dimension. What remains is to structure all the middle sets that were created. The boxes in the middle set are already partitioned on the first coordinate dimension, and no additional separation is to be expected on the first coordinate direction. Instead, we have a second coordinate direction which is not yet taken into account.

As boxes in the original set $\mathcal{I}$ are ordered on the first coordinate direction, all the boxes in $\mathcal{I}_{\text{mid}}$ are ordered on the second coordinate direction. The approach for splitting $\mathcal{I}$ in the first coordinate direction, can be repeated for $\mathcal{I}_{\text{mid}}$ for the second coordinate direction. In the case of three-dimensional problems, such as sheet-forming, a further subdivision can be made on the third coordinate direction.

After we have changed dimension 3 times, each middle-set can no longer be subdivided in the proposed manner with any further separation of boxes to be expected. The maximum number of boxes left in any middle set after the third subdivision is called $K$. This number is the maximum number of boxes overlapping a certain point in space. For a typical simulation in two dimensions, $K$ is 2, since the capture boxes spanned by the faces attached

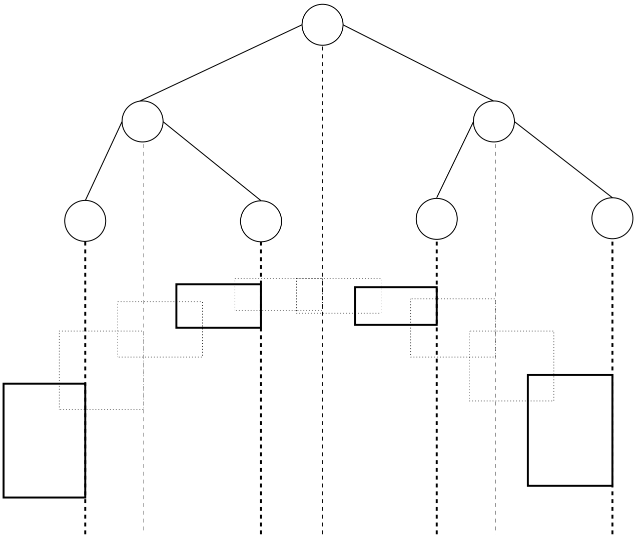


Figure 4.17: Third subdivision example.

to a single node will overlap. For a simulation in three dimensions using triangles the average number of elements attached to a node is 6, and thus $K$ is in a regular simulation approximately 6.

### 4.5.3 Querying the data structure

Querying the tree now proceeds in the following way: We start at the root node of the tree. Associated with this root node is a cutting line through some **x** on the first dimension.

The middle subtree of the root contains all the boxes that intersect the cutting line. Since these boxes lie both in the left space of the cutting line, as well as the right space of the cutting line, we need to search the complete middle set. This will report all boxes in the middle set that contain the query point.

After the middle set is checked, we compare the coordinate of the query point with that of the cutting line. If it is smaller, we continue the search in the left child. If it is larger, we continue the search in the right child.

The worst case scenario for a search gives an upper-bound on the time complexity of the problem. The tree on $x$ has a depth of $O(\log N)$. Each of the subtrees on $y$ has a depth of at most $O(\log N)$. Each of the subtrees on $z$ has also a depth of at most $O(\log N)$. Finally each of the middle sets in the $z$ subtrees contains at most $K$ boxes.

The complexity of the search can then be deduced as follows, for a search we need to go down at most $O(\log N)$ nodes in the $x$-tree. So the total work is $O(\log N)$ times the work required at each associated data-structure in these nodes. An equivalent reasoning holds for the subtree on $y$ and $z$, which gives us a total work of at most $O(K \log^3 N)$, thus a search complexity of $O(\log^3 N)$. For the two-dimensional case it is $O(\log^2 N)$.

We finish this section by making some remarks: The worst case search time for the algorithm presented is widely overestimated. In fact the pre-factor starts to be very important if $\log N$ is not too large. After a longer analysis, it turns out that the pre-factor can be decreased to at least $K/4$.

The worst case bound is quite strict, because it assumes a certain substructuring in which half of the boxes fall into the left set, half of the boxes fall into the middle set and the right set is empty.

## 4.6 Conclusions

In this chapter, the integration of the contact integral was discussed. We saw that quite a few problems remain, even when a good regularisation method was already selected. One of the key factors that arises is that of geometric incompatibility when the problem under consideration is discretised. The geometric incompatibility between the slave and master surface can cause the traction approximations to be inaccurate, and when the master surface is non-smooth can deteriorate convergence.

The geometric incompatibility also limits the choice of integration algorithms. By selecting a nodal integration algorithm, the accuracy problems can be alleviated. The influence of the arising stability problems can be diminished by employing a line search procedure. Alternatively, we can smooth the master boundary. The procedure for smoothing general 2D and 3D boundaries was shown to be effective for the 2D problem, but rather cumbersome for the 3D problem.

A third factor apart from accuracy and stability is efficiency. In the last part of this chapter, a method was presented to improve the efficiency of finding the projection locations for the integration points.

With this, the theoretical description of how to deal with contact in simulations is complete, and in the next chapter we take a look at several numerical experiments and applications to compare the different methodologies.

Chapter 5

# Numerical Experiments and Applications

## 5.1 Introduction

In this chapter the theory that was developed in this thesis is illustrated by several examples. In Section 5.2 we compare the different traction regularisations. In Section 5.3 the accuracy of the contact normal tractions are tested. In Section 5.4, the accuracy of the implementation of the friction algorithm is tested. In Section 5.5, a non-unique projection problem is illustrated. In Section 5.6 an example is given of a large sliding problem, with deformable tools. In Section 5.7 an industrial application is presented. We end the chapter in Section 5.8 with the conclusions.

## 5.2 Efficiency of traction regularisations

In this first example we investigate the different regularisation schemes (see Chapter 3). The example is an upsetting problem. The problem is symmetric, and only the right side is modelled. The initial configuration is illustrated in Figure 5.2. The left boundary of the billet is prescribed with a zero $x$-displacement condition.

The bottom slab is the foundation on which the block is resting. All the degrees of freedom of the foundation are suppressed. The constraint for the lower interface of the block is set to be a contact interface with the foundation. Due to internal stresses, during the upsetting process the block will rise up, therefore the interesting feature of loss of contact is observed during the simulation. Friction is taken into account during the simulation. To take care of the incompressibility constraint a selective reduced integration method is applied, where the pressure is only integrated in the center point of the quads.

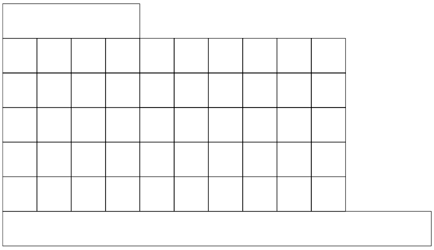


Figure 5.1: The configuration of the upsetting problem.

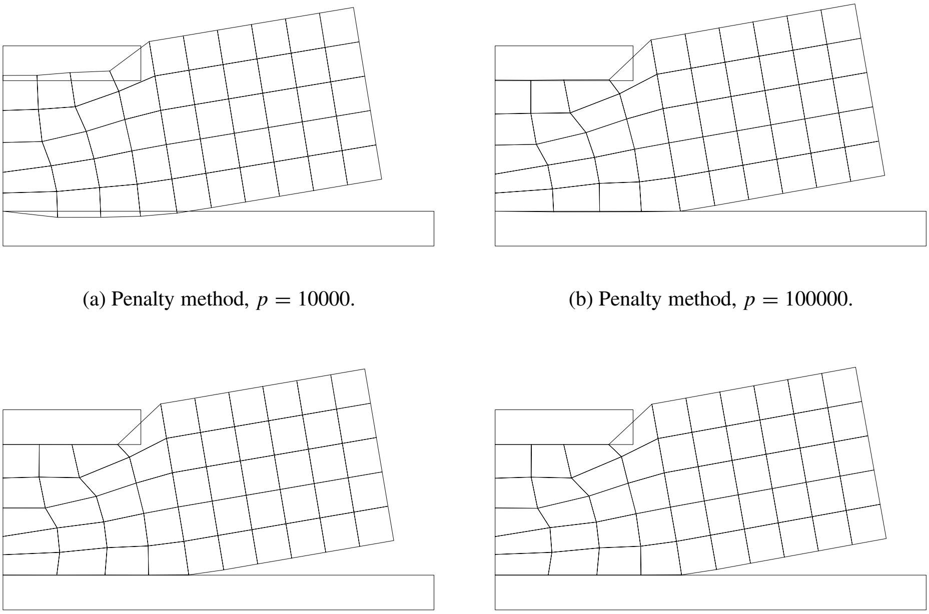
The top slab is the punch for which the displacements are prescribed. Again a contact constraint is set for the interface for the punch with the block. The problem is simulated using the penalty method, the method of augmented Lagrangians, and the barrier method.

For the penalty parameter for normal contact in all the penalty and augmented Lagrangian scheme we use 10000. The penalty for the modified barrier method was set to 100. A distinction between the two penalties is made, since the contribution to the stiffness matrix for the penalty based methods and the barrier methods are different. For the penalty method the penalty could be interpreted as force per unit area per unit penetration depth $[\mathrm{N}\cdot\mathrm{m}^{-3}]$, whereas in the barrier methods, the interpretation would be: per unit penetration depth $[\mathrm{m}^{-1}]$. Choosing them to be the same would cause completely different convergence characteristics. The penalty parameter for the frictional part is set to 10000, and the friction coefficient is chosen as 0.1. For the barrier method and method of augmented Lagrangians, we set a maximum allowable penetration of 0.01. The default projection method as presented in Chapter 4 is chosen.

The results are illustrated in Figure 5.2. As can be seen the penalty method causes large penetrations for this example. Increasing the penalty causes the method to have slightly poorer convergence, but solves the problem of the large penetrations. This result illustrates the problems with convergence of the penalty method that were discussed in Chapter 3.4.1: If the penalty is chosen too small there is unacceptable overlap, if it is chosen too large, one gets convergence problems. From the solutions of the augmented Lagrangian and modified barrier method it can be seen that these methods deliver a geometrically good solution with the specified penalties.

There is furthermore a large geometric overlap for each of the methods for the last element sliding away under the punch. This overlap arises from the application of the same basic projection scheme for each of the methods. What occurs is the same situation as the one that was depicted in Figure 4.4. In this figure an element is shown to slide of a corner. At this point a choice is to be made: Either the complete element is in contact, or the complete element is not in contact. Either selection is incorrect, since the actual situation is that the element is only partly in contact. One might be tempted by preferring the non-overlapping solution, since it is geometrically more pleasing.

A non-overlapping solution for the upsetting problem can be obtained by applying a two-pass procedure as was proposed in Chapter 4.3.2. Upon evaluation of the results in Figures 5.2 we could decide to use either the modified barrier method or the method of



(a) Penalty method, $p = 10000$.

(b) Penalty method, $p = 100000$.

(c) Augmented Lagrangians, $p = 10000$.

(d) Modified barrier, $p = 100$.

Figure 5.2: Results of the different regularisation schemes

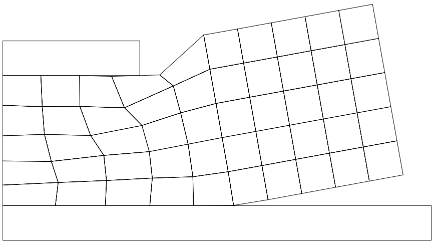


Figure 5.3: Solution of the 2-pass problem.

augmented Lagrangians as the method to solve the two pass problem. Interestingly, the method of augmented Lagrangians does not converge within a reasonable number of iterations anymore. The modified barrier method does converge. An explanation could be that the modified barrier method is more non-linear and can predict larger changes in contact tractions with smaller changes in displacements. This additional non-linearity, however, does slow down the rate of convergence.

The final result is illustrated in Figure 5.2. The solution now seems to be geometrically more correct. The actual solution, however, lies somewhere between the results of the one-pass and two-pass solution. This can easily be understood by realising that the line segment which is lying partly under the block should be split into two parts: One part which lies under the punch, and a part that does not. In the case of the one-pass procedure the line-segment loses contact early. In the case of the two-pass procedure, the line-segment loses contact too late. Thus the one and two-pass procedures form a lower and upper bound on the actual solution. These bounds can be improved by mesh-refinement, which brings us back to our initial observation that the large overlap is really due to a discretisation error, and is not an artifact of the application of the contact method.

As a conclusion of the above simulation we can state that the accuracy of the penalty method is strongly dependent on the magnitude of the penalty, and thus after a simulation a check should be run to see if there are no unacceptable violations. The modified barrier and augmented Lagrangian method do not have this problem. Moreover, the modified barrier method is somewhat more non-linear than the method of augmented Lagrangians, which can help in some situations as with the solution of the two-pass procedure in which the local material non-linearities are also stronger. If this is not the case, then probably the method of augmented Lagrangians is the best choice: it gives the highest accuracy with the least local increase of non-linearities.

## 5.3 Accuracy of contact

The accuracy of a numerical method is always an important topic to consider. To test the accuracy of a method one requires a non-trivial test for which an analytical solution is available. In the case of contact this non-trivial test is given by the Hertzian contact formulas Hertz (1881). The Hertzian contact formulas describe the contact pressure distribution between two cylinders (line-contact) or between two spheres (point-contact).

The Hertzian contact formulas have been used to test the previous incarnation of the contact algorithm in DIEKA, the results of which are presented in ter Haar (1996). In that work, it was concluded that with the contact algorithm as it was present at that time, it was not possible to accurately predict the contact normal pressure.

The Hertzian contact formulas that we consider in this example are those for line contact, so that we can suffice with a two-dimensional plain strain simulation. The contact pressure distribution between two cylinders is given by:

$$p = p_{\max}\sqrt{1-\xi^2}, \tag{5.1}$$

where $\xi$ is a normalised coordinate, $\xi = x/a$. The coordinate $x$ is the usual Cartesian $x$-coordinate, and $a$ is the half contact width.

The maximum pressure is given by:

$$p = \sqrt{\frac{F_n \cdot E^*}{2\pi B R^*}}. \tag{5.2}$$

Here $F_n$ is the total exerted normal force, $E^*$ is the combined elasticity modulus, $B$ is the height of the cylinders and $R^*$ is the combined radius. Using the same quantities, the half contact width is given as:

$$a = \sqrt{\frac{8 F_n R^*}{\pi B E^*}}. \tag{5.3}$$

The combined elasticity modulus can be computed by using the elasticity moduli of the two cylinders $E_1$ and $E_2$ and the Poisson ratio of both cylinders $\nu_1$ and $\nu_2$. The combination is given by:

$$E^* = \frac{2E_1E_2}{E_2(1-\nu_1^2) + E_1(1-\nu_2^2)}. \tag{5.4}$$

The combined radius is computed by using the radii of the two cylinders $R_1$ and $R_2$ as follows:

$$R^* = \frac{R_1R_2}{R_1 + R_2}. \tag{5.5}$$

To validate the implementation using these formulas we set up a contacting interface between a cylinder and a plate. We assume the cylinder and plate to be infinitely high, so we can model the problem in a two-dimensional setting. The plate can be considered as a cylinder with an infinite radius. For the combined radius we then find:

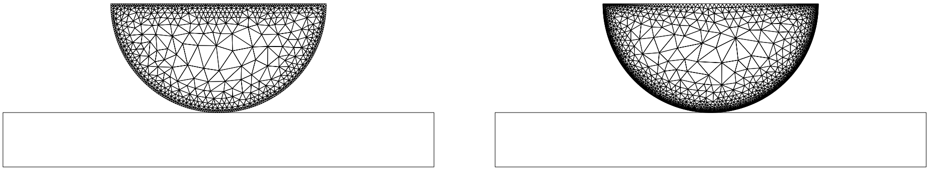
$$R^* = \lim_{R_2 \longrightarrow \infty} \frac{R_1R_2}{R_1 + R_2} = R_1. \tag{5.6}$$

Furthermore we assume that the plate is undeformable. To achieve this, we make the plate much stiffer than the cylinder. For the combined elasticity radius this means:

$$E^* = \lim_{E_2 \longrightarrow \infty} \frac{2E_1E_2}{E_2(1-\nu_1^2) + E_1(1-\nu_2^2)} = \frac{2E_1}{1-\nu_1^2}. \tag{5.7}$$

In the simulation the radius of the cylinder was set to 50 mm. The elasticity modulus for the cylinder was set to 210000 MPa and the Poisson ratio was selected as 0.31. Two different meshes composed of triangular elements were used to compute the resulting normal pressure. They are given in Figure 5.4. The second mesh contains twice as many elements along the circular part as the first. The top of the cylinder is suppressed in the $y$-direction and the sides of the plate are suppressed in the $x$-direction. A total force of 35000 N is applied to the plate in the $y$ direction. To achieve an accurate result, the method of augmented Lagrangians was employed to obtain a maximum penetration of $1.0 \cdot 10^{-5}$.

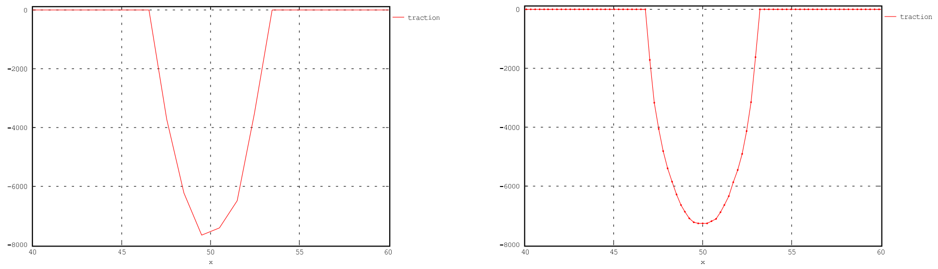
The results for the normal tractions are presented in Figure 5.5. The value for the half contact width is from the Hertzian formulas: $a = 3.10$ mm. The value for the maximum pressure is: $p_{\max} = 7195$ N. The results are presented in Table 5.1. As can be seen from the table, the correct half contact width lies nicely within the upper and lower bounds of the left and right contact widths. And the pressure is approximated more accurately if the



(a) Mesh 1

(b) Mesh 2

Figure 5.4: The two meshes employed in the simulation of Hertzian contact.



(a) Contact pressure for mesh 1

(b) Contact pressure for mesh2

Figure 5.5: The two meshes employed in the simulation of Hertzian contact.

| Property | Mesh 1 | Mesh 2 | Exact |
|---|---|---|---|
| Left contact width lower | 2.47 | 2.97 | 3.10 |
| Left contact width upper | 3.47 | 3.22 | 3.10 |
| Right contact width lower | 2.48 | 2.97 | 3.10 |
| Right contact width upper | 3.47 | 3.22 | 3.10 |
| Maximum pressure | 7618 | 7464 | 7195 |

Table 5.1: Results of Hertzian contact simulation.

number of elements is increased. What is interesting, is the fact that the maximum contact normal pressure is overestimated by the algorithm, whereas the original results by ter Haar (1996) showed an underestimation, and a wider contact area. The latter is probably due to using a penalty method with a too low penalty value. The result of that simulation will show more overlap, consequently it has a larger contact with and hence on average lower contact normal tractions.

Another thing that is observed from the simulations is that the maximum pressure is overestimated. For the first mesh, the overestimation is approximately 6%. The second mesh that has twice as fine a boundary discretisation has an overestimation of the maximum pressure of approximately 3.5%. Since local inaccuracies can play an important role in global instabilities, we can conclude that a high number of elements is required to be able to accurately simulate the contact behaviour.

### 5.3.1 Stribeck friction

Another friction algorithm that can be used instead of Coulomb friction, is Stribeck friction. The difference between the two is that Coulomb friction is assuming a constant coefficient of friction, whereas in the case of Stribeck friction, the coefficient is dependent on the pressure, viscosity of a lubricant between the contacting surfaces, the sum velocity and the roughness of the two surfaces. The dependency on the sum velocity really comes from the use of rotating balls in which lubricant is transported into the contact area if the balls are rotating in the same direction. In forming simulations one can assume that the tools are static, thus moving with a very low velocity. In that case the sum velocity can actually be approximated with the differential velocity of the surfaces. See also ter Haar (1996) and Westeneng (2001).

The usage of the regularisation method as discussed in Section 3.5 is identical for the Stribeck model as it was for the Coulomb model. The following data is required for the Stribeck model to function:

$\eta$ The dynamic viscosity of the lubricant in [$^{\mathrm{Ns}}/_{\mathrm{m}^2}$].

$R_a$ The combined surface roughness in [$\mu$m].

$\mu_{\mathbf{hl}}$ The coefficient of friction for hydro dynamic lubrication.

$\mu_{\mathbf{bl}}$ The coefficient of friction for boundary lubrication.

$L_{\mathbf{hl}}$ The $L$-parameter for the transition of mixed lubrication to hydro dynamic lubrication.

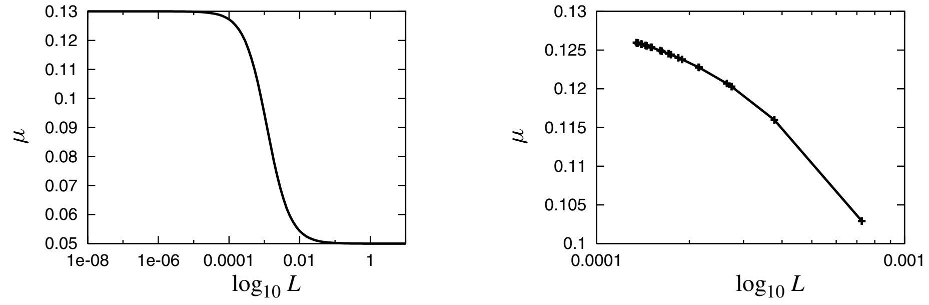
$L_{\mathbf{bl}}$ The $L$-parameter for the transition of mixed lubrication to boundary lubrication.

$v_+$ The sum-velocity in [$^{\mathrm{m}}/_{\mathrm{s}}$].

$p$ The pressure in [$^{\mathrm{N}}/_{\mathrm{m}^2}$].

The $L$-parameter is defined as:

$$L = \frac{\eta v_+}{p R_a}. \tag{5.8}$$



(a) Example of Stribeck curve.

(b) $L$ values versus $\mu$ from simulation.

Figure 5.6: Stribeck curves example, and data from simulation.

Using the above definitions the coefficient of friction is defined as:

$$\mu = \mu_{\text{hl}} + (\mu_{\text{bl}} - \mu_{\text{hl}}) \left( \frac{1}{2} - \frac{1}{2} \tanh \left( c \cdot \log \left( \frac{L}{L_0} \right) \right) \right). \tag{5.9}$$

The parameters $c$ and $L_0$ are calculated as:

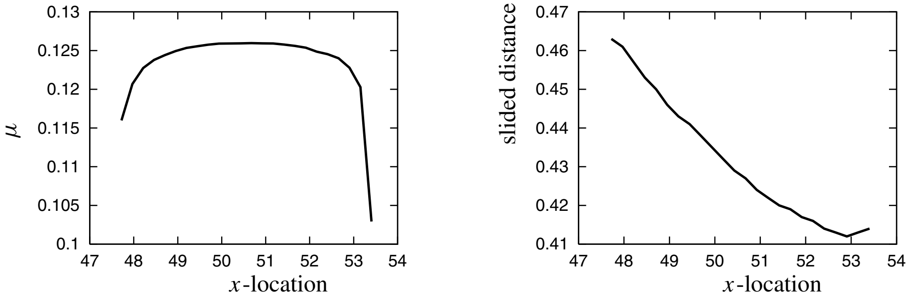
$$c = -\frac{2}{\log \frac{L_{\text{bl}}}{L_{\text{hl}}}}, \tag{5.10}$$

$$L_0 = \sqrt{L_{\text{bl}} \cdot L_{\text{hl}}}. \tag{5.11}$$

We have performed a sliding simulation using a Stribeck friction algorithm. The lubricant viscosity was set to $1\,\mathrm{Ns/m^2}$, the surface roughness was assumed to be $1\,\mu$m. The sum velocity and pressure are obtained from the simulation. The friction parameters that were used are as follows: $L_{\text{hl}} = 4.3 \cdot 10^{-3}$, $L_{\text{bl}} = 2.7 \cdot 10^{-4}$, $\mu_{\text{hl}} = 0.05$ and $\mu_{\text{bl}} = 0.13$. A graph of the corresponding Stribeck curve is given in Figure 5.6(a) The simulation performs 5 steps with a displacement of 0.1 mm in the positive $x$-direction each 0.1 s. The other parameters required are identical to that in Section 5.3.

At the fifth step, all the nodes are sliding, and the $L$-parameters and coefficients of friction can be extracted from the simulation. The result is given in Figure 5.6(b). From this figure we can see that most of the nodes are in the mixed lubrication regime.

The distribution of the coefficients of friction over the contact area are given in Figure 5.7(a). From it can be seen that the higher pressure on the nodes in the centre of the contacting area is clearly causing higher coefficients of friction. In Figure 5.7(b) a graph is presented of the total amount of sliding at the end of the fifth step. From this it can be seen that the largest amount of sliding is done by the trailing nodes.



(a) Distribution of friction parameters over the contact area.

(b) Slided distance of the contact area.

Figure 5.7: Graphs of the distribution of sliding and friction coefficient over the contact area.

## 5.4 Accuracy of friction

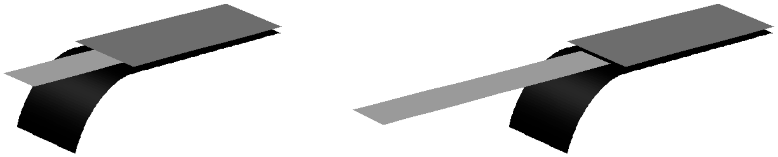
To test the friction behaviour of the contact algorithm, two model problems are studied for which the analytical solutions are known. Both problems model the pull-out of a strip from between a die-blankholder pair.

The first problem that is considered is the sliding of a strip along a flat surface in Section 5.4.1. The second problem is the sliding of a strip along both a flat plane and a cylinder, which is discussed in Section 5.4.2.

### 5.4.1 Sliding along the plane

The first problem to test the accuracy of the implementation of the friction algorithm is the sliding of a plate along a surface. A plate is clamped between a flat die and a flat blankholder. The blankholder is subsequently loaded with a force of 500 N in the $z$–direction. The blankholder movements are suppressed in the $x$- and $y$-direction. The die movements are fully suppressed. The strip is half the width of the die and blankholder and placed between them. The strip is meshed using membrane elements, and set to a thickness of 1 mm. Next the displacement of the strip in the $x$-direction is prescribed along the minimum $x$-boundary of the strip. The total prescribed displacement of the strip is 30 mm in the negative $x$-direction. The initial length of the strip is 50 mm and its width is set to 10 mm. The initial and final configuration of the simulation is illustrated in Figure 5.8.

The coefficient of friction in the simulation was set to 0.1. In each converged increment, the total normal force the top surface is 500 N, which is the same for the bottom surface of the plate. Hence, the friction force exerted on the top and bottom surface is 50 N each, amounting to a total pull-out force of 100 N. The pull-out force can also be obtained from the simulation and is plotted in Figure 5.9. As can be observed from the figure, the theoretical value and numerical value are in good agreement.



(a) Initial configuration

(b) Final configuration

Figure 5.8: The initial and final configuration of the pulling problem.

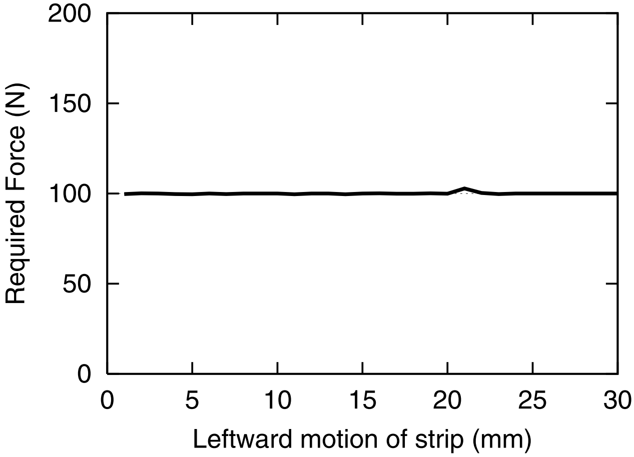
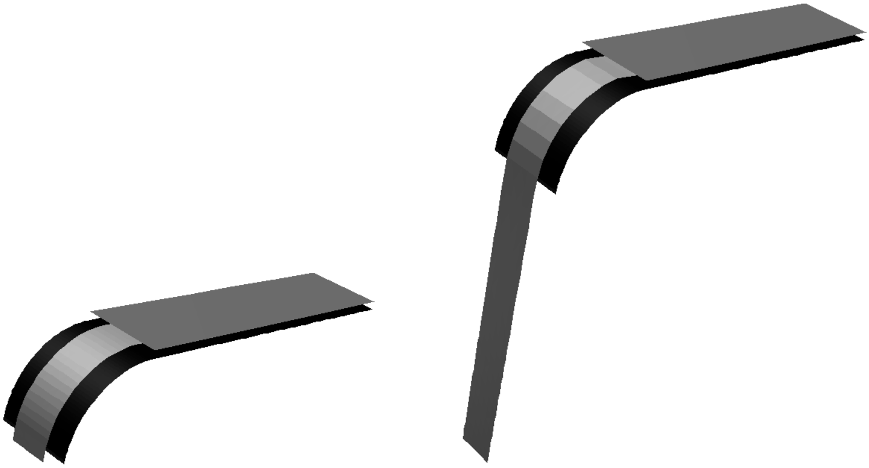


Figure 5.9: The computed pull-out force on the leftmost edge of the strip.



(a) Initial configuration

(b) Final configuration

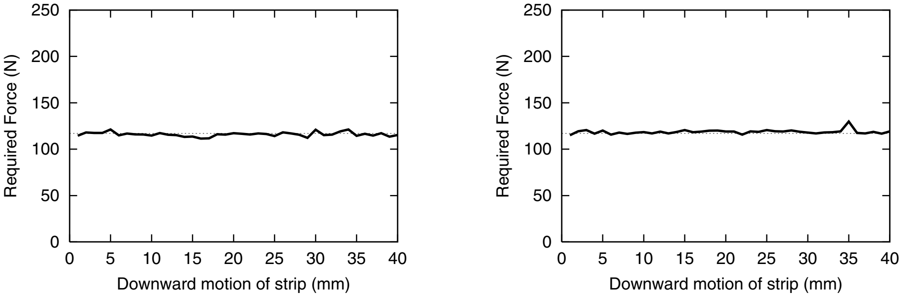
Figure 5.10: The initial and final configuration of the pulling problem.

### 5.4.2 Sliding along the plane and cylinder

A more complicated problem is obtained, when the strip is not pulled out along the direction of the die and blankholder, but orthogonal to it, along a cylinder. The initial and final configurations for this more complex simulation are illustrated in Figure 5.10. The width is again 10 mm, the circular part of the die has a radius of 10 mm. The strip has a length of 40 mm clamped between the die and blankholder, the remaining length of the strip is such that it lies in its entirety along the cylindrical part of the die.

Once again the strip is modelled using membrane elements, so bending does not influence the forces required to pull the strip down. The plate is 1mm thick, and the coefficient of friction is set to 0.1. The cylindrical extension of the die is modelled using flat triangular elements. This is to see if the contact model as proposed shows parasitic stiffness, as was discussed by Vreede (1992). To see if the approximation error of the triangular elements to the cylinder is of influence in the computation of the force, two distinct discretisations are chosen for the die. A total displacement of 40 mm is prescribed on the edge at the minimum $x$-coordinate in the $z$-direction. The results are illustrated in Figure 5.11.

The total force that is required can be obtained from the solution of a simple differential equation. Let us consider a small arc of the cylindrical extension with an angle $\Delta\theta$, for an illustration take a look at Figure 5.4.2 The arc is considered infinitesimal, thus it can be adequately approximated by a line segment. On the line segment a force $T_0$ is operating which is “pulling” downwards to the right along the tangent of the cylinder, and a force $T_1$ which is “pulling” it downwards to the left along the tangent of the cylinder. Without loss of generality, we assume the plate is slipping towards the left in the picture. The plate is



(a) Rough die mesh results

(b) Finer die mesh results

Figure 5.11: The computed pull-out force on the leftmost edge of the strip.

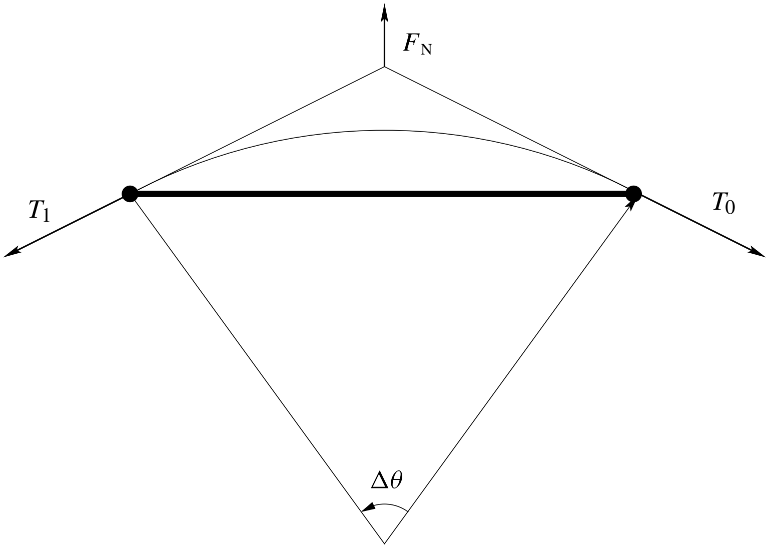


Figure 5.12: A small arc of the cylinder.

not accelerated, thus there is equilibrium of forces. The force normal to the line segment exerted by $T_0$ is $T_0\frac{1}{2}\Delta\theta$. Equivalently so, the force exerted by $T_1$ exerted normal to the line segment is $T_1\frac{1}{2}\Delta\theta$. The total force exerted by the plate on the cylinder is thus:

$$\frac{1}{2}\Delta\theta\,(T_0 + T_1)\,. \tag{5.12}$$

This force is in equilibrium with the force exerted by the cylinder on the plate, which we name $F_{\mathrm{N}}$. Taking into account Coulomb's law, to let the plate slip to the left, a force has to be exerted in the left direction, which is $\mu F_{\mathrm{N}}$ larger than the force exerted to the right. Putting this all in formula form leads to:

$$\begin{aligned} T_1 - T_0 &= \mu F_{\mathrm{N}} \Longleftrightarrow \\ \Delta T &= \mu\frac{1}{2}\Delta\theta\,(T_0 + T_1)\,. \end{aligned} \tag{5.13}$$

Taking the limit $\Delta\theta \to 0$, leads to the differential equation:

$$\frac{\mathrm{d}T}{\mathrm{d}\theta} = \mu T. \tag{5.14}$$

The solution of this differential equation is:

$$T(\theta) = T(0)\exp\mu\theta. \tag{5.15}$$

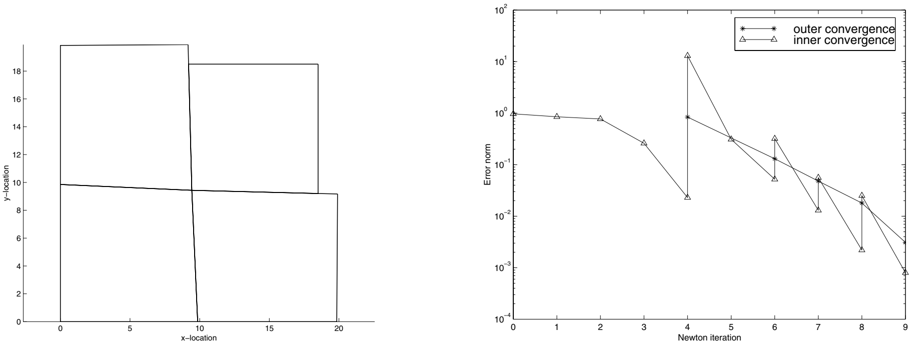
In our case $T(0) = 100\,\mathrm{N}$ according to the results in Section 5.4.1. The total angle traversed is a quarter circle, which means that the resulting force should be: $T(\frac{\pi}{2}) = 100\exp\left(0.1\frac{\pi}{2}\right) = 117\,\mathrm{N}$.

Comparing this value with the results that were obtained by the simulations we see that both the rough and the fine approximation of the die agree well with the analytic value of 117 N. We do not encounter any parasitic stiffnesses discussed by Vreede (1992). From the flat pull, and the pull along a cylinder, we can conclude that the friction algorithm is implemented correctly.

## 5.5 Correct projections

In this section we consider the problem of a block of material being pushed into a corner. The mesh is very course, see also Schreppers et al. (1992). The problem is considered difficult, since it is hard for an algorithm to decide what the correct contactor segments are. The simulation essentially checks the implementation of the projection algorithm as discussed in Section 4.3 (see also Figure 4.6).

The implementation of this problem has been tested in MATLAB. The problem has been simulated using the modified barrier method. The setup of the problem is shown in Figure 5.13. The foundation is assumed to deform elastically and is supported on the bottom and on the left. The block in the upper right corner is then in one increment pressed into the corner, by prescribing downward displacements on the top and leftward displacements on the right nodes. We refer back to Section 3.4.3 on how the barrier method is applied.



(a) Final result

(b) Convergence characteristics

Figure 5.13: Results of the corner problem.

A distinction can be made between inner iterations, in which for a fixed penalty parameter and fixed estimations to the normal tractions an equilibrium solution is computed, and outer iterations in which the normal contact tractions are updated, and the penalty parameter can be adjusted.

The convergence results are plotted in Figure 5.13 on the right side, where the inner convergence shows the progress within each subproblem loop, i.e. for set values of the penalty parameter $p$ and the estimations of the contact normal traction $\lambda$. The outer convergence shows the progress of the total convergence. Just as with augmented Lagrangian schemes, the total scheme converges linearly. Each inner problem converges quadratically, when sufficiently close to the optimum for that subproblem. The zigzag pattern is due to fact that when we update the Lagrange multipliers and the value of the penalty parameter $p$, the error in our initial estimate increases.

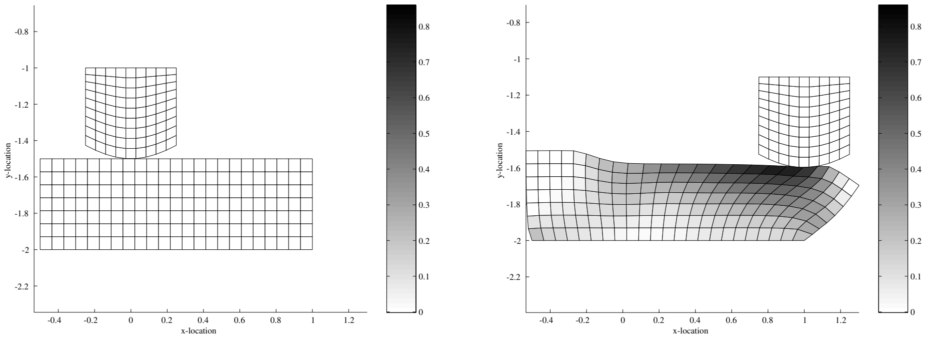
From the simulation we can see that the projection algorithm as discussed in Chapter 4.3 handles the non-unique projection problem well. It furthermore shows that the modified barrier method converges linearly to the solution in one iteration per augmentation, which makes it a competitor for the method of choice in contact simulations.

## 5.6 A 2-D large sliding problem

In this section the augmented Lagrangian scheme and modified barrier scheme are compared. The computation is performed in Matlab. Our interest lies in comparing the convergence characteristics of the two methods.

The example considered is an elastic deformable indenter working on an elasto-plastic material. Thus both bodies involved in the computation are deformable. The example is a model problem, so no distinct material is considered.

The setup of the model problem is shown in Figure 5.14. A Von Mises flow rule is



(a) Equivalent plastic strain in the initial state.

(b) Equivalent plastic strain in the final state.

Figure 5.14: The initial and final state of the sliding problem.

| Property | Value for indenter | Value for block |
|---|---|---|
| $E$ | 1000 | 100 |
| $\nu$ | 0.31 | 0.31 |
| $C$ | | 6 |
| $\varepsilon_0$ | | $1.033 \cdot 10^{-5}$ |
| $n$ | | 0.13 |

Table 5.2: Material properties for model problem.

used in conjunction with a Ludwik–Nadai hardening law is used, this takes the form $\sigma_Y = C(\varepsilon_0 + \varepsilon_p)^n$. In this formula $\sigma_Y$ is the yield stress and $\varepsilon_p$ is the equivalent plastic strain.

The parameter setting that was used is given in Table 5.2. The indenter remains elastic throughout the simulation and thus no plastic property values for the indenter are listed.

The results of the computation are shown in Figure 5.14. The equivalent plastic strain is shown in the deformed mesh. The top row of nodes on the indenter is prescribed. The bottom nodes on the block are all suppressed in both directions. The indenter has prescribed movement on the top nodes. First a 0.1 downward movement is prescribed on these nodes. The $x$-displacement on the indenter top nodes are then suppressed. After that a 1.0 displacement to the right is prescribed on the top nodes, as well as suppressing the $y$-displacement. The downward part of the motion is performed in 8 steps, whereas the right displacement is performed in 80 steps. This corresponds to steps of approximately 25% of the element size used. The time discretisation used is an Euler-backward scheme.

In the computation of both the modified barrier scheme and augmented Lagrangians algorithm, we can distinguish between inner and outer iterations. The inner iterations are the normal Newton steps, where we keep the estimations for the tractions fixed. Whereas in the outer iterations we augment the estimations for the tractions. The complexity of the problem can be expressed in the cumulative amount of inner iterations required to achieve convergence of the contact traction estimations. The measure of convergence we use is the following:

$$e = \frac{\sum_i |\lambda_i||d_i|}{1 + \|\boldsymbol{\lambda}\|}. \tag{5.16}$$

For the convergence of the inner iterations we find that it does not have to exceed the convergence of the outer iteration, since that will be wasted effort. This does not hold for the step before the first augmentation, since we do not know the accuracy of the traction estimates then.

In this $e$ represents the error. The vector $\boldsymbol{\lambda}$ represents the estimation for the contact tractions in the boundary nodes, and $d_i$ is the distance at node $i$. The division is done by the given term to have no conflicts when $\boldsymbol{\lambda}$ is equal to $\mathbf{0}$. Hence, the error approximates the average penetration. We have set our convergence goal to $e = 1.0 \cdot 10^{-5}$. A plot of this parameter for the Augmented Lagrangian algorithm and the modified barrier method is shown in Figure 5.15.

From this picture it can be seen that the modified barrier method requires less augmentations to converge to the required accuracy. The question is, however, how many more inner iterations will this take? Since, each inner iteration requires the assembly of a tangent stiffness matrix, and the solution of a linear system of equations.

The convergence results can be seen in Tables 5.3 and 5.4. Here we see that the modified barrier method requires more inner iteration per outer iteration than the method of augmented Lagrangians. However, the modified barrier method requires two less augmentations to converge. This results in an equivalent number of inner iterations required to converge.

## 5.7 An industrial application

As a final example of the implementation of the contact algorithm, we consider a stretch forming process. The stretch forming process is used by Fokker aerostructures to manufacture parts for aircraft skins. To create a good part at minimal cost, certain process parameters need to be optimised. Using a finite element simulation to analyse the stretch forming process increases the insight into the process and can aid in the reduction of cost. In this section we study the influence of friction on the major and minor strains for a complex saddle shaped part.

The tools that are being used are a stretch block and two clamps The clamps are modelled by cylinders. The initial configuration is illustrated in Figure 5.16, where the cylinders are omitted.

The part is made of aluminium sheet. The sheet is assumed to be initially 3.5 mm thick. A Young's modulus of 70 GPa was taken and a Poisson ratio of 0.3. The yield function that was used is modelled with a Von Mises yield surface, hence no anisotropy of the sheet was taken into account. The work hardening is simulated using a Ludwik–Nadai curve. The Ludwik–Nadai stress-strain relation is given as:

$$\sigma_y(\varepsilon_p) = C(\varepsilon_0 + \varepsilon_p)^n. \tag{5.17}$$

| cum iter | outer iter | sub iter | inner conv | outer conv |
|---|---|---|---|---|
| | 0 | 0 | $9.47 \cdot 10^{-1}$ | |
| 1 | 0 | 1 | $8.00 \cdot 10^{-3}$ | |
| 2 | 0 | 2 | $2.40 \cdot 10^{-3}$ | |
| 3 | 0 | 3 | $1.15 \cdot 10^{-4}$ | |
| 4 | 0 | 4 | $4.13 \cdot 10^{-6}$ | |
| 5 | 0 | 5 | $7.48 \cdot 10^{-11}$ | $6.68 \cdot 10^{-4}$ |
| | 1 | 0 | $4.66 \cdot 10^{-3}$ | |
| 6 | 1 | 1 | $4.67 \cdot 10^{-5}$ | $2.43 \cdot 10^{-4}$ |
| | 2 | 0 | $2.82 \cdot 10^{-3}$ | |
| 7 | 2 | 1 | $1.44 \cdot 10^{-5}$ | $1.04 \cdot 10^{-4}$ |
| | 3 | 0 | $1.97 \cdot 10^{-3}$ | |
| 8 | 3 | 1 | $4.96 \cdot 10^{-6}$ | $4.31 \cdot 10^{-5}$ |
| | 4 | 0 | $1.50 \cdot 10^{-3}$ | |
| 9 | 4 | 1 | $8.10 \cdot 10^{-7}$ | $1.66 \cdot 10^{-5}$ |
| | 5 | 0 | $1.25 \cdot 10^{-3}$ | |
| 10 | 5 | 1 | $4.35 \cdot 10^{-7}$ | $4.53 \cdot 10^{-6}$ |

Table 5.3: Convergence for the augmented Lagrangian method.

| cum iter | outer iter | sub iter | inner conv | outer conv |
|---|---|---|---|---|
| | 0 | 0 | $9.47 \cdot 10^{-1}$ | |
| 1 | 0 | 1 | $2.66 \cdot 10^{-2}$ | |
| 2 | 0 | 2 | $6.48 \cdot 10^{-3}$ | |
| 3 | 0 | 3 | $1.30 \cdot 10^{-4}$ | |
| 4 | 0 | 4 | $3.29 \cdot 10^{-7}$ | $1.17 \cdot 10^{-4}$ |
| | 1 | 0 | $3.84 \cdot 10^{-3}$ | |
| 5 | 1 | 1 | $2.25 \cdot 10^{-4}$ | |
| 6 | 1 | 2 | $9.04 \cdot 10^{-7}$ | $3.09 \cdot 10^{-5}$ |
| | 2 | 0 | $8.75 \cdot 10^{-4}$ | |
| 7 | 2 | 1 | $1.36 \cdot 10^{-5}$ | |
| 8 | 2 | 2 | $4.88 \cdot 10^{-9}$ | $1.55 \cdot 10^{-5}$ |
| | 3 | 0 | $4.01 \cdot 10^{-4}$ | |
| 9 | 3 | 1 | $1.83 \cdot 10^{-6}$ | |
| 10 | 3 | 2 | $1.13 \cdot 10^{-10}$ | $9.67 \cdot 10^{-6}$ |

Table 5.4: Convergence for the modified barrier method.

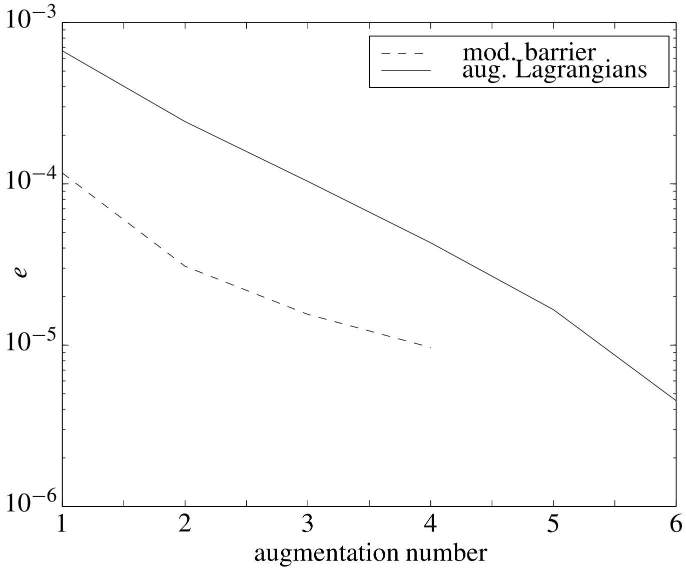


Figure 5.15: Convergence of the augmentations.

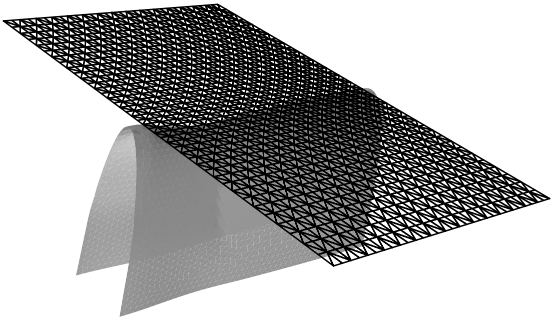


Figure 5.16: Initial configuration of the stretching process.

In this equation $\sigma_y$ is the yield stress for a certain equivalent plastic strain $\varepsilon_p$. The point $\varepsilon_0$ is the initial yield strain, and this is set at $1.2 \cdot 10^{-3}$. The hardening coefficient $C$ is set to 500 MPa, and the parameter $n$ is chosen as 0.2. Combining this all leads to an initial yield stress of 150 MPa. The last parameter important in the simulation is the coefficient of friction, which is chosen successively as 0.1 and 0.2.

The sheet is modelled with discrete Kirchhoff triangular elements. For the clamping mechanism ideal clamping is assumed, so that a displacement on the boundary nodes can be prescribed. The sheet is meshed with 1200 elements and has a refinement along the top. Since the simulation is fully symmetric, we only model half of the plate. The contact model that was employed to obtain the solution is the method of augmented Lagrangians.

The results are shown in Figures 5.17 and 5.18. In Figure 5.17 the thickness of the final plate is shown after using a coefficient of friction of 0.1. The result in Figure 5.18 shows the thickness after using a coefficient of friction of 0.2.

From the pictures a clear dependence on the friction coefficient can be seen. The value of the frictional coefficient can be influenced by the roughness of the die and the type of lubricant that is employed. It is this type of analysis which provides insight into the process of stretch forming.

## 5.8 Conclusions

From the examples we can see that the way contact is modelled has a large influence on the rate of convergence. The results from the upsetting problem in Section 5.2 show us that the augmented Lagrangian method and the modified barrier method give good, geometrically acceptable solutions. In the same section it was shown that having a fully conforming geometrical solution does not necessarily mean that the simulation was performed very accurately. This in fact depends strongly on the discretisation.

The result of the Hertzian contact problem in Section 5.3 shows us that the contact normal tractions can be computed accurately within the error of the discretisation.

The result of the pull-out problem in Section 5.4 shows us that the friction algorithm was implemented correctly and can be simulated accurately using the method of augmented Lagrangians for both the contact normal tractions as well as the tangential tractions. It furthermore shows that there are no parasitic tangential stiffnesses.

The corner contact problem in Section 5.5 illustrates that the correct projection segments are obtained by the projection algorithm.

The large sliding problem in Section 5.6 further illustrates the capabilities of both the augmented Lagrangian as well as the modified barrier method as traction estimation methods.

Finally from the stretch forming problem discussed in Section 5.7 it follows that the contact model is an essential component in simulating a forming process. In this section one can see the influence of the coefficient of friction has on the final thicknesses on the plate. This type of insight can then be used to adjust the forming process itself to obtain a better product.

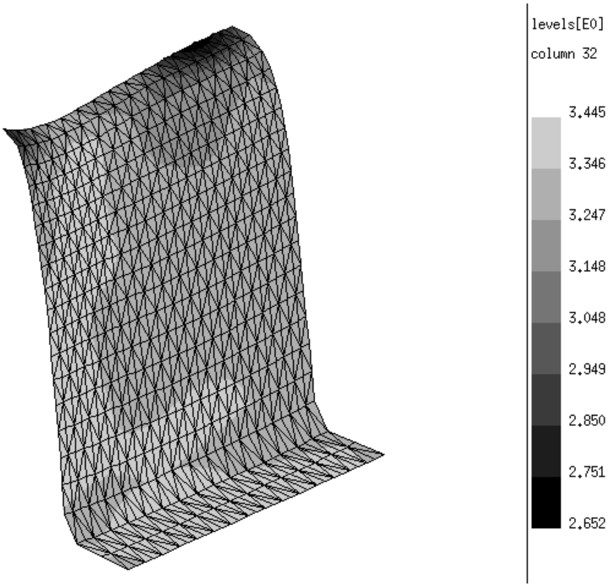


Figure 5.17: Thickness of plate with coefficient of friction at 0.1

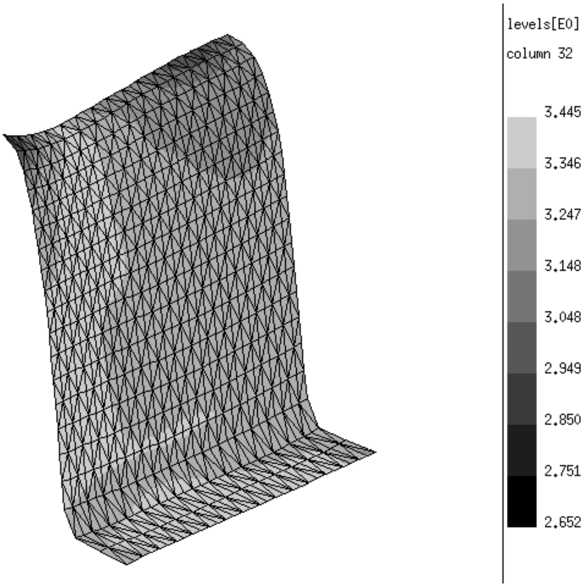


Figure 5.18: Thickness of plate with coefficient of friction at 0.2

# Chapter 6

# CONCLUSIONS AND RECOMMENDATIONS

As it was stated in the the introduction of this thesis, our aim is to study the contact problem applied to metal forming simulations. To see where it originates, formulate the equations and constraints, discuss the various ways in which to enforce those constraints, and finally how to incorporate it all into a finite element framework. By scrutinising the complete path of development of the equations to the implementation, we intend to identify all the possible pitfalls occurring when contact methods are applied.

The first part of our aim was to formulate the equations and constraints. This part of the work is demonstrated in Chapter 2. The derivation is presented for the weak form of equilibrium including contact. The equivalence of the necessary conditions of the contact problem in elasticity and the final one for the virtual displacement case is shown. From this it can be seen that there is a strong coupling between variational problems and the minimisation of functionals.

An advantage of the insight gained by the fact that there exists an association between the variation and minimisation problems, is that we can use methods that solve the contact problem for the minimisation problem and use it for the variation problems.

The following equation in general form represents the weak form with contact:

$$G(\boldsymbol{\varphi}, \mathbf{w}) + G_c(\boldsymbol{\varphi}, \mathbf{w}) = 0 \tag{6.1}$$

The first functional in this equation can be computed in the usual sense by a finite element method. The second functional contains the contribution of contact. This latter functional can be expanded by interpreting **w** either as virtual displacements or as virtual velocities. For virtual displacements, the result is:

$$G_c\left(\boldsymbol{\varphi}, \delta\boldsymbol{\varphi}\right) = \int_{\gamma^{(1)}} \mathrm{t}_N \delta \mathrm{d}_N \,\mathrm{d}\gamma + \int_{\gamma^{(1)}} \mathrm{t}_{T_\alpha} \delta \bar{\xi}^{(2)\alpha} \,\mathrm{d}\gamma \tag{6.2}$$

We can conclude that what needs to be done is to compute the contact tractions, which enforce the impenetrability constraints. Several ways in which this can be done are discussed in Chapter 3.

First we discussed the mixed method, which is computationally expensive, and the constraint method, which is very efficient but can present problems for example in plate forming simulations where the detection of contact is very important. Moreover, there may be too many constraints in the plate forming simulation to use an effective active set method.

The methods that were dealt with more in depthly are methods that regularise the contact traction by estimating them through allowing small violations of the impenetrability condition. By taking this viewpoint, all the different methods can be seen to stem from a basic mold. The method is specified by assuming the form of a specific penalisation function $\Xi$. This penalisation function is introduced in the optimisation setting For the more general variational setting, one just fills in the results for the first order conditions of the optimisation problem. The different methods discussed are:

- The penalty method, for which:

$$\begin{aligned} \Xi(\mathrm{d}_\mathrm{N}, \{p\}) &= \frac{p}{2}\langle -\mathrm{d}_\mathrm{N}\rangle^2, && (6.3a)\\ \mathrm{t}_\mathrm{N}(\mathrm{d}_\mathrm{N}, \{p\}) &= -p\langle -\mathrm{d}_\mathrm{N}\rangle. && (6.3b)\end{aligned}$$

- The method of augmented Lagrangians, for which:

$$\begin{aligned} \Xi(\mathrm{d}_\mathrm{N}, \{p, \lambda_\mathrm{N}\}) &= \frac{1}{2p}\langle -(\lambda_\mathrm{N} + p\mathrm{d}_\mathrm{N})\rangle^2, && (6.4a)\\ \mathrm{t}_\mathrm{N}(\mathrm{d}_\mathrm{N}, \{p, \lambda_\mathrm{N}\}) &= -\langle -(\lambda_\mathrm{N} + p\mathrm{d}_\mathrm{N})\rangle. && (6.4b)\end{aligned}$$

- The modified barrier method for which:

$$\begin{aligned} \Xi(\mathrm{d}_\mathrm{N}, \{p, \lambda_\mathrm{N}\}) &= \frac{\lambda_\mathrm{N}}{p}\log\left(1 + p\mathrm{d}_\mathrm{N}\right), && (6.5a)\\ \mathrm{t}_\mathrm{N}(\mathrm{d}_\mathrm{N}, \{p, \lambda_\mathrm{N}\}) &= \lambda_\mathrm{N}\frac{1}{1 + p\mathrm{d}_\mathrm{N}}. && (6.5b)\end{aligned}$$

Using the same reasoning but immediately in the variational setting, the contact frictional tractions can be regularised. These methods rely on a predictor-corrector scheme. Only the predictor is determined differently for the two schemes that were considered. These two schemes are:

- The penalty method for friction, in which case the predictor is set to:

$$\mathbf{t}^p_\mathrm{T} = p\mathbf{d}_\mathrm{T}. \tag{6.6}$$

- The method of augmented Lagrangians, in which case the predictor is set to:

$$\mathbf{t}^p_\mathrm{T} = \boldsymbol{\lambda}_\mathrm{T} + p\mathbf{d}_\mathrm{T}. \tag{6.7}$$

From the predictor, the correct frictional traction is computed, by using either the fixed estimate to the normal contact traction $\lambda_{\mathrm{N}}$ or the current estimate $\mathrm{t}_{\mathrm{N}}$. The frictional traction now follows from:

$$\mathbf{t}_{\mathrm{T}} = \begin{cases} \mathbf{t}_{\mathrm{T}}^{p} & \text{if } \|\mathbf{t}_{\mathrm{T}}^{p}\| \leq |\mathrm{t}_{\mathrm{N}}|, \\ \mu|\mathrm{t}_{\mathrm{N}}|\frac{\mathbf{t}_{\mathrm{T}}^{p}}{\|\mathbf{t}_{\mathrm{T}}^{p}\|} & \text{otherwise.} \end{cases} \tag{6.8}$$

In this work we only considered regularisation methods and shortly discussed the constraint method. It might be interesting to see these two methods compared more in depth. Additionally one could solve the intermediate problems that arise during an SQP-step (see Section 3.3.2) with either the augmented Lagrangian or modified barrier method, instead of performing direct eliminations.

Next in Chapter 4, the integration of the contact integral is discussed. As it turns out, a lot of the convergence problems observed with contact arise from geometric incompatibilities. The geometric incompatibility between the slave and master surface can cause the traction approximations to be inaccurate, and when the master surface is non-smooth it can deteriorate convergence.

The geometric incompatibility also limits the choice of integration algorithms. By selecting a nodal integration algorithm, the accuracy problems can be alleviated. The influence of the arising stability problems can be diminished by employing a line search procedure. Alternatively, we can smooth the master boundary. The procedure for smoothing general 2D and 3D boundaries is shown to be effective for the 2D problem, but rather cumbersome for the 3D problem. It is therefore recommended that additional research is performed on finding more efficient local smoothing methods for 3D problems.

A third factor apart from accuracy and stability is efficiency. In the last part of Chapter 4, a method is presented to improve the efficiency of finding the projection locations for the integration points. The total time that is required for the simulation can increase significantly if the projections are performed in a straightforward manner, it is therefore recommended to always use a more sophisticated search algorithm.

Finally in Chapter 5, several numerical problems are studied to illustrate the operation of the different parts of the contact algorithm. These are an upsetting problem to illustrate the difference between the several regularisation methods. A pull-out problem shows that the methods implemented do not contain parasitic stiffnesses that may be encountered when using contact elements, and it furthermore shows that the friction algorithm is implemented correctly. The corner contact problem illustrates that the correct projection segments are obtained by the projection algorithm. The large sliding problem further illustrates the capabilities of both the augmented Lagrangian as well as the modified barrier method as traction estimation methods. Finally a stretch forming problem is analysed, from which we can conclude that the contact model has an influence on the final product geometry.

A chain is only as strong as its weakest link, so far the contact problem has been held responsible for the failure of a lot of simulations. It is with this work that many of the stability, accuracy and efficiency issues that arise due to contact are addressed, so that research may again focus on other areas which require attention for the correct simulation of forming processes.

Appendix A

# MISTAKES IN THE CONVECTED DESCRIPTION

In this appendix, we discuss the small mistake that was made in the much referred paper Laursen and Simo (1993). The paper goes into great depth in describing the continuum mechanics framework for contact. One of the claims made, is that the contact slip velocity can be made frame-indifferent when one would just use a convected description expressed in the dual basis. In fact, it is argued in this article, that the description of the velocity in this basis is a constitutive decision. The first author even goes into greater detail in Laursen (1994) to clarify the choices made. The author is hoping to provide a general and simple theory to solve a very complex class of problems. Unfortunately, this is not so.

To see this, we need to introduce some theory concerning covariant and contravariant vectors. Using this theory, we can quickly locate the error in both articles.

## A.1 Vectors

By the term vector, two things can be meant. In linear algebra, a vector is usually considered to be just a bunch of numbers. These numbers can be stacked in a column vector, or put sequentially in a row vector. In the physical world, a vector is a quantity which is defined to have a direction and a length. The number of space dimensions that the vector "lives in", is usually 2 or 3. We only concern ourselves with 3 dimensional space. A vector, being a physical quantity, is independent of the representation which is used, it just is.

To work with arbitrary vectors, we usually represent them with respect to a basis. The most preferred basis is the orthonormal Cartesian basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. A vector $\mathbf{q}$ is represented with respect to this Cartesian basis as:

$$\mathbf{q} = q_i \mathbf{e}_i, \tag{A.1}$$

where we used the summation convention. If we are well aware of what basis is being used, the basis vectors are often omitted:

$$\mathbf{q} = q_i \tag{A.2}$$

An interesting thing to note is that using a Cartesian basis still depends on the location of an observer.

Now, we consider a set of three independent vectors $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$, which are not necessarily orthogonal, nor of unit length. In order to represent an arbitrary vector $\mathbf{a}$ in terms of these vectors, we need to find components $(a^1, a^2, a^3)$ such that

$$\mathbf{a} = a^1\mathbf{v}_1 + a^2\mathbf{v}_2 + a^3\mathbf{v}_3. \tag{A.3}$$

One way of doing this is by forming a reciprocal basis $(\mathbf{v}^1, \mathbf{v}^2, \mathbf{v}^3)$. The reciprocal basis vectors satisfy the following relationship:

$$\mathbf{v}_i \cdot \mathbf{v}^j = \delta_i^j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}. \tag{A.4}$$

Using the reciprocal basis, the components $a^i$ are easily found by:

$$\mathbf{a} \cdot \mathbf{v}^j = a^i\mathbf{v}_i \cdot \mathbf{v}^j = a^i\delta_i^j = a^j. \tag{A.5}$$

In an identical fashion, the components of $\mathbf{a}$ with respect to the reciprocal basis can be found. In that case, $\mathbf{a}$ takes the form:

$$\mathbf{a} = a_j\mathbf{v}^j, \tag{A.6}$$

and the components can be found by

$$a_i = \mathbf{a} \cdot \mathbf{v}_i. \tag{A.7}$$

The components with respect to the original basis vectors $\mathbf{v}_i$, are called the *contravariant* components of $\mathbf{a}$. The components with respect to the reciprocal basis vectors $\mathbf{v}^j$ are called the *covariant* components of $\mathbf{a}$. Note that the numbers $a_i$ and $a^i$ may differ, but that in combination with the basis vectors, they represent the exact same vector.

An additional simple relationship between the components exists, which is given now. First introduce the numbers $g_{ij}$ and $g^{ij}$ as

$$g_{ij} = \mathbf{v}_i \cdot \mathbf{v}_j. \tag{A.8}$$

In an identical fashion:

$$g^{ij} = \mathbf{v}^i \cdot \mathbf{v}^j. \tag{A.9}$$

From the properties of the inner product, we can immediately conclude that $g_{ij} = g_{ji}$ and $g^{ij} = g^{ji}$. The numbers $g_{ij}$ are called the metric components of the space and the numbers $g^{ij}$ are called the conjugate metric components of the space. Using this notation, we can quickly find that

$$a_i = g_{ij}a^j, \tag{A.10}$$

and

$$a^i = g^{ij}a_j. \tag{A.11}$$

## A.2 The mistake

In Laursen and Simo (1993) and in Laursen (1994), the transition is made from a vector described with contravariant components, using a convected basis, to a vector described with covariant components, using the reciprocal basis to the convected basis. Specifically in Laursen and Simo (1993), the conversion is given for the velocity field. In the article, the convected basis is given as $\boldsymbol{\tau}_\alpha$, $\alpha = 1, 2$, together with the unit normal $n$, which is normal to both vectors. The convected basis vectors are given as the partial derivatives of the map $\boldsymbol{\Psi}_t$ (See also Chapter 2), as:

$$\boldsymbol{\tau}_\alpha = \boldsymbol{\Psi}_{t,\alpha}. \tag{A.12}$$

Using the fact that

$$\boldsymbol{\Psi}_t = \boldsymbol{\varphi} \circ \boldsymbol{\Psi}_0, \tag{A.13}$$

we find after some elaboration that

$$\boldsymbol{\tau}_\alpha = \mathbf{F} \cdot \mathbf{T}_\alpha, \tag{A.14}$$

where

$$\begin{aligned} \mathbf{F} &= \frac{\partial \boldsymbol{\varphi}}{\partial \mathbf{X}} && \text{(A.15)} \\ \mathbf{T}_\alpha &= \boldsymbol{\Psi}_{0,\alpha} && \text{(A.16)} \end{aligned}$$

The tangential velocity $\mathbf{v}_T$ is expressed in the current coordinate system as:

$$\mathbf{v}_T = \mathrm{v}_T^\alpha \boldsymbol{\tau}_\alpha. \tag{A.17}$$

The pull-back of $\mathbf{v}_T$, is defined as:

$$\begin{aligned} \mathbf{V}_T &= F^{-1} \cdot \mathbf{v}_T && \text{(A.18)} \\ &= F^{-1} \left( \mathrm{v}_T^\alpha \boldsymbol{\tau}_\alpha \right) && \text{(A.19)} \\ &= F^{-1} \left( \mathrm{v}_T^\alpha \mathbf{F} \cdot \mathbf{T}_\alpha \right) && \text{(A.20)} \\ &= \mathrm{v}_T^\alpha \mathbf{T}_\alpha && \text{(A.21)} \end{aligned}$$

In the article, the components of $\mathrm{v}_T^\alpha$ are shown to be

$$\mathrm{v}_T^\alpha = \dot{\xi}^\alpha(\mathbf{X}, t), \tag{A.22}$$

but this is inconsequential for the discussion, and we stick to the usage of $\mathrm{v}_T^\alpha$.

The metric components (as discussed in the previous section) for the space in the original configuration on the contact surface can be expressed as

$$M_{\alpha\beta} = \mathbf{T}_\alpha \cdot \mathbf{T}_\beta. \tag{A.23}$$

Using the metric, the currently contravariantly represented vector $\mathbf{V}_T$ can also be written using the covariant basis:

$$\mathbf{V}_T = \mathrm{v}_T^\alpha \mathbf{T}_\alpha = M_{\alpha\beta} \mathrm{v}_T^\alpha \mathbf{T}^\beta. \tag{A.24}$$

The distinction in representation in the article is made by attaching a "flat" sign to the symbol, thus a distinction is made between

$$\mathbf{V}_T = \mathrm{v}_T^{\alpha} \mathbf{T}_{\alpha} \tag{A.25}$$

and

$$\mathbf{V}_T^{\flat} = M_{\alpha\beta} \mathrm{v}_T^{\alpha} \mathbf{T}^{\beta}. \tag{A.26}$$

But we stress here that the distinction is only in representation! The vectors $\mathbf{V}_T^{\flat}$ and $\mathbf{V}_T$ are in fact totally identical. It is only the selection of the basis, which changes the components.

In an identical manner as we constructed a reciprocal basis in the undeformed configuration, we can also construct one in the deformed configuration. This gives rise to a basis $\boldsymbol{\tau}^{\beta}$, with a metric

$$m_{\alpha\beta} = \boldsymbol{\tau}_{\alpha} \cdot \boldsymbol{\tau}_{\beta}. \tag{A.27}$$

Representation of the original vector $\mathbf{v}_T$ in this basis would give rise to:

$$\mathbf{v}_T = m_{\alpha\beta} \mathrm{v}_T^{\alpha} \boldsymbol{\tau}^{\beta}. \tag{A.28}$$

The "claim to fame" in the articles now arises from the statement that pushing forward the vector $\mathbf{V}_T^{\flat}$ gives rise to an invariant object to describe the friction with. The reasoning goes as follows: The covariantly represented vector $\mathbf{V}_T^{\flat}$ is given by:

$$\mathbf{V}_T^{\flat} = M_{\alpha\beta} \mathrm{v}_T^{\alpha} \mathbf{T}^{\beta}. \tag{A.29}$$

Pushing the vector forward, would yield according to the article (but incorrectly so):

$$\mathbf{v}_T^{\flat} = M_{\alpha\beta} \mathrm{v}_T^{\alpha} \boldsymbol{\tau}^{\beta}. \tag{A.30}$$

Unfortunately, the push forward of $\mathbf{T}^{\beta}$ is *not* $\boldsymbol{\tau}^{\beta}$. The correct computation of the push-forward, will require the reversal of the coordinate transformation to the reciprocal basis, obtaining the contravariant tensor. The contravariant tensor can then be pushed-forward. Transforming the push-forward again to covariant form yields (A.28).

The description (A.30) is no longer a description of the velocity. The mistake made lies in the incorrect transformation of a covariant tensorial quantity, which needs to be pushed forward in a different manner than contravariant tensorial quantities, which is attempted in both articles. It seems the author, must have had some feeling there was something wrong, by his elaborate derivation using contravariant tensorial quantities in Laursen (1994), upon which follows the sentence: "One may readily conclude from these facts that the same relationship holds for the dual basis,...", which is obviously not so.

## A.3 Conclusions

The conclusion we can draw here is that using the push-forward of the covariant description in the undeformed configuration of the velocity, does not yield the result as proposed in Laursen and Simo (1993) and Laursen (1994).

# Bibliography

Atzema, E. (1994, February). *Formability of Sheet Metal and Sandwhich Laminates*. Ph. D. thesis, Universiteit Twente, 7500 AE, Enschede.

Barlam, D. and E. Zahavi (1999). The reliability of solutions in contact problems. *Computers and Structures 70*, 35–45.

Bathe, K. J. and P. A. Bouzinov (1997). On the constraint function method for contact problems. *Computers and Structures 64*, 1069–1085.

Bazaraa, M. S. and C. M. Shetty (1979). *Nonlinear programming: theory and algorithms*. Wiley.

Belgacem, F. B., P. Hild, and P. Laborde (1998). The mortar finite element method for contact problems. *Mathematical and Computer Modelling 28*, 263–271.

Belytschko, T., W. K. Liu, and B. Moran (2000). *Nonlinear Finite Elements for Continua and Structures*. Wiley.

Belytschko, T. and M. O. Neal (1991). Contact-impact by the pinball algorithm with penalty and lagrangian methods. *International Journal for Numerical Methods in Engineering 31*, 547–572.

Berg, M., M. Kreveld, M. Overmars, and O. Schwarzkopf (1997). *Computational Geometry: Algorithms and Applications*. Springer Verlag.

Bertsekas, D. P. (1982). *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press.

BittenCourt, E. and G. J. Creus (1998). Finite element analysis of three-dimensional contact and impact in large deformation problems. *Computers and Structures 69*, 219–234.

Bonet, J. and R. D. Wood (1997). *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press.

Breitfeld, M. G. and D. F. Shanno (1994, April). A globally convergent penalty–barrier algorithm for nonlinear programming and its computational performance. Technical Report RRR 12-94, RUTCOR.

Breitfeld, M. G. and D. F. Shanno (1996). Computational experience with penalty-barrier methods for nonlinear programming. *Annals of Operations Research 62*, 439–464.

Carleer, B. (1997, March). *Finite Element Analysis of Deep Drawing*. Ph. D. thesis, Universiteit Twente, 7500 AE, Enschede.

Chabrand, P., O. Chertier, and F. Dubois (2001). Complementarity methods for multibody friction contact problems in finite deformations. *International Journal for Numerical Methods in Engineering 51*, 553–578.

Chabrand, P., F. Dubois, and M. Raous (1998). Various numerical methods for solving unilateral contact problems with friction. *Mathematical and Computer Modelling 28*, 97–108.

Chamoret, D., J. M. Bergheau, A. Rassineux, and P. Villon (2001). Modelling of contact surface by local hermite diffuse interpolation. In A. M. Habraken (Ed.), *The international 4th Esaform conference on Materials Forming*, pp. 179–182.

Chawla, V. and T. A. Laursen (1998). Energy consistent algorithms for frictional contact problems. *International Journal for Numerical Methods in Engineering 42*, 799–827.

Chenot, J. L. and L. Fourment (1998). Numerical formulations and algorithms for solving contact problems in metal forming simulation. In *Computational Mechanics, New Trends and Applications*.

Christensen, P. W., A. Klarbring, J. S. Pang, and N. Strömberg (1998). Formulation and comparison of algorithms for frictional contact problems. *International Journal for Numerical Methods in Engineering 42*, 145–173.

Coorevits, P., P. Hild, and J. Pelle (2000). A posteriori error for unilateral contact with matching and non-matching meshes. *Computer Methods in Applied Mechanics and Engineering 186*, 65–83.

Crisfield, M. A. (2000). Re-visiting the contact patch test. *International Journal for Numerical Methods in Engineering 48*, 435–449.

Czekanski, A., S. A. Meguid, N. El-Abbasi, and M. H. Refaat (2001). On the elastodynamic solution of frictional contact problems using variational inequalities. *International Journal for Numerical Methods in Engineering 50*, 611–627.

do Carmo, M. P. (1976). *Differential Geometry of Curves and Surfaces*. Prentice-Hall.

Dohrmann, C. R., S. W. Key, and M. W. Heinstein (2000). A method for connecting dissimilar finite element meshes in two dimensions. *International Journal for Numerical Methods in Engineering 48*, 655–678.

Dong, C. Y. (1999). A simple benchmark problem to test frictional contact. *Computer Methods in Applied Mechanics and Engineering 177*, 153–162.

El-Abbasi, N. and K. J. Bathe (2001). Stability and patch test performance of contact discretizations and a new solution algorithm. *Computers and Structures 79*, 1473–1486.

El-Abbasi, N. and S. A. Meguid (1999). On the treatment of frictional contact in shell structures using variational inequalities. *International Journal for Numerical Methods in Engineering 46*, 275–295.

El-Abbasi, N. and S. A. Meguid (2001). On the modelling of smooth contact surfaces using cubic splines. *International Journal for Numerical Methods in Engineering 50*, 953–967.

Esche, S. K., G. L. Kinzel, and T. Altan (1997). Issues in convergence improvements for non-linear finite element programs. *International Journal for Numerical Methods in Engineering 40*, 4577–4594.

Farahani, K., M. Mofid, and A. Vafai (2000). A solution method for general contact-impact problems. *Computer Methods in Applied Mechanics and Engineering 187*, 69–77.

Farahani, K., M. Mofid, and A. Vafai (2001). United elements method for general contact-impact problems. *Computer Methods in Applied Mechanics and Engineering 191*, 843–860.

Farin, G. E. (1988). *Curves and surfaces for computer aided geometric design: a practical guide*. Academic Press.

Fiacco, A. V. and G. P. McCormick (1968). *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. John Wiley & Sons.

Franz, J., M. Liepelt, and K. Schittkowski (1995). Penalty-barrier-methods for nonlinear optimization: Implementation and computational results. Technical report, Department of Mathematics, University of Bayreuth, Germany.

Frisch, K. R. (1955, May 13). The logarithmic potential method of convex programming. Technical report, University of Economics, Oslo, Norway.

Giannakopoulos, A. E. (1989). The return mapping method for the integration of friction constitutive relations. *Computers & Structures 32*, 157–167.

Givoli, D. and I. Doukhovni (1996). Finite element-quadratic programming approach for contact problems with geometrical nonlinearity. *Computers and Structures 61*, 31–41.

Habraken, A. M. and S. Cescotto (1998). Contact between deformable solids: The fully coupled approach. *Mathematical and Computer Modelling 28*, 153–169.

Heinstein, M. W. and T. A. Laursen (1999). An algorithm for the matrix-free solution of quasistatic frictional contact problems. *International Journal for Numerical Methods in Engineering 44*, 1205–1226.

Hertz, H. (1881). über die berührung fester elastischer körper. *J. Reine Angew. Mathm. 92*, 156–171.

Hild, P. (2000). Numerical implementation of two nonconforming finite element methods for unilateral contact. *Computer Methods in Applied Mechanics and Engineering 184*, 99–123.

Hild, P., F. B. Belgacem, and P. Laborde (1998). Unilateral contact for non-matching finite element meshes: the global contact condition. In S. Idelsohn, E. O. nate, and E. Dvorkin (Eds.), *Computation Mechanics, New Trends and Applications*.

Huétink, J. (1986, June). *On the Simulation of Thermo-Mechanical Forming Processes*. Ph. D. thesis, Universiteit Twente, 7500 AE, Enschede.

Hughes, T. J. R. (1987). *The Finite Element Method: Linear Static and Dynamic Analysis*. Prentice-Hall.

Jean, M. (1999). The non-smooth contact dynamics method. *Computer Methods in Applied Mechanics and Engineering 177*, 235–257.

Jones, R. E. and P. Papadopoulos (2000). A yield-limited lagrange multiplier formulation for frictional contact. *International Journal for Numerical Methods in Engineering 48*, 1127–1149.

Jones, R. E. and P. Papadopoulos (2001). A novel three-dimensional contact finite element based on smooth pressure interpolations. *International Journal for Numerical Methods in Engineering 51*, 791–811.

Kane, C., E. A. Repetto, M. Ortiz, and J. E. Marsden (1999). Finite element analysis of nonsmooth contact. *Computer Methods in Applied Mechanics and Engineering 180*, 1–26.

Kikuchi, N. and J. T. Oden (1988). *Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods*. SIAM.

Klarbring, A. and G. Björkman (1988). A mathematical programming approach to contact problems with friction and varying contact surface. *Computers and Structures 30*, 1185–1198.

Kloosterman, G., R. M. J. van Damme, A. H. van den Boogaard, and J. Huétink (2001). A geometrical based contact algorithm using a barrier method. *International Journal for Numerical methods in Engineering 51*, 865–882.

Lai, M.-J. (1997). Geometric interpretation of smoothness conditions of triangular polynomial patches. *Computer Aided Geometric Design 14*, 191–199.

Laursen, T. A. (1994). The convected description in large deformation frictional contact problems. *International Journal of Solids and Structures 31*, 669–681.

Laursen, T. A. and J. C. Simo (1992). Algorithmic symmetrization of coulomb frictional problems using augmented lagrangians. *Computer Methods in Applied Mechanics and Engineering 108*, 133–146.

Laursen, T. A. and J. C. Simo (1993). A continuum-based finite element formulation for the implicit solution of multibody large deformation frictional contact problems. *International Journal for Numerical Methods in Engineering 36*, 3451–3485.

Lei, X. (2001). Contact friction analysis with a simple interface element. *Computer Methods in Applied Mechanics and Engineering 190*, 1955–1965.

Leung, A. Y. T., C. Guoqing, and C. Wanji (1998). Smoothing newton method for solving two- and three-dimensional frictional contact problems. *International Journal for Numerical Methods in Engineering 41*, 1001–1027.

Lie-heng, W. (1998). On the duality methods for the contact problem in elasticity. *Computer Methods in Applied Mechanics and Engineering 167*, 275–282.

Luenberger, D. G. (1973). *Introduction to Linear and Nonlinear Programming*. Addison-Wesley.

Malvern, L. E. (1969). *Introduction to the Mechanics of a Continuous Medium*. Prentice-Hall.

McCormick, J. P. (1983). *Nonlinear Programming*. John Wiley & Sons.

McDevitt, T. W. and T. A. Laursen (2000). A mortar-finite element formulation for frictional contact problems. *International Journal for Numerical Methods in Engineering 48*, 1525–1547.

Munjiza, A. and K. R. F. Andrews (1998). Nbs contact detection algorithms for bodies of similar size. *International Journal for Numerical Methods in Engineering 43*, 131–149.

Oldenburg, M. and L. Nilsson (1994). The position code algorithm for contact searching. *International Journal for Numerical Methods in Engineering 37*, 359–386.

Pang, J. S. (1990). Newton's method for *b*–differentiable equations. *Mathematics of Operations Research 15*, 311–341.

Papadopoulos, P. and J. M. Solberg (1998). A lagrange multiplier method for the finite element solution of frictionless contact problems. *Mathematical and Computer Modelling 28*, 373–384.

Parisch, H. and C. Lübbing (1997). A formulation of arbitrarily shaped surface elements for three-dimensional large deformation contact with friction. *International Journal for Numerical Methods in Engineering 40*, 3359–3383.

Piegl, L. and W. Tiller (1997). *The NURBS book*. Springer.

Pietrzak, G. and A. Curnier (1999). Large deformation frictional contact mechanics: Continuum formulation and augmented lagrangian treatment. *Computer Methods in Applied Mechanics and Engineering 177*, 351–381.

Polyak, R. (1992). Modified barrier functions (theory and methods). *Mathematical Programming 54*, 177–222.

Refaat, M. H. and S. A. Meguid (1997). Updated lagrangian formulation of contact problems using variational inequalities. *International Journal for Numerical Methods in Engineering 40*, 2975–2993.

Refaat, M. H. and S. A. Meguid (1998). A new strategy for the solution of frictional contact problems. *International Journal for Numerical Methods in Engineering 43*, 1053–1068.

Rieger, A. and P. Wriggers (2001). Adaptive methods for frictionless contact problems. *Computers and Structures 79*, 2197–2208.

Rogovoy, A. and B. Ivanov (1997). Displacement formulation of the friction conditions on the contact surface. *Computers and Structures 62*, 133–139.

Schreppers, G. J. M. A., W. A. M. Brekelmans, and A. A. H. J. Sauren (1992). A finite element formulation of the large sliding contact. *International Journal for Numerical Methods in Engineering 35*, 133–143.

Shimizu, T. and T. Sano (1995). An application of a penalty method contact and friction algorithm to a 3-dimensional tool surface expressed by a b-spline patch. *Journal of Materials Processing Technology 28*, 207–213.

Simo, J. C. (1988a). A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition. part i: Continuum formulation. *Computer Methods in Applied Mechanics and Engineering 66*, 199–219.

Simo, J. C. (1988b). A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition. part ii: Computational aspects. *Computer Methods in Applied Mechanics and Engineering 68*, 1–31.

Simo, J. C. and T. J. R. Hughes (1998). *Computational Inelasticity*. Springer-Verlag New York.

Simo, J. C. and T. A. Laursen (1992). An augmented lagrangian treatment of contact problems involving friction. *Computers & Structures 42*, 97–116.

Stupkiewicz, S. (2001). Extension of the node-to-segment contact element for surface-expansion-dependent contact laws. *International Journal for Numerical Methods in Engineering 50*, 739–759.

ter Haar, R. (1996, May). *Friction in Sheet Metal Forming*. Ph. D. thesis, Universiteit Twente, 7500 AE, Enschede.

van der Lugt, J. (1988, October). *A Finite Element Method for the Simulation of Thermo-Mechanical Contact Problems in Forming Processes*. Ph. D. thesis, Universiteit Twente, 7500AE, Enschede.

Vreede, P. T. (1992, December). *A Finite Element Method for Simulations of 3-Dimensional Sheet Metal Forming*. Ph. D. thesis, Universiteit Twente, 7500AE, Enschede.

Wang, F., J. Cheng, and Z. Yao (2001). Ffs contact searching algorithm for dynamic finite element analysis. *International Journal for Numerical Methods in Engineering 52*, 655–672.

Wang, S. and A. Makinouchi (2000). Contact search strategies for fem simulation of the blow molding process. *International Journal for Numerical Methods in Engineering 48*, 501–521.

Wang, S. P. and E. Nakamachi (1997). The inside-outside contact search algorithm for finite element analysis. *International Journal for Numerical Methods in Engineering 40*, 3665–3685.

Westeneng, A. (2001, Mar). *Modelling of Contact and Friction in Deep Drawing Processes*. Ph. D. thesis, Universiteit Twente, 7500 AE, Enschede.

Wriggers, P., L. Krstulovic-Opara, and J. Korelc (2001). Smooth $c^1$-interpolations for two-dimensional frictional contact problems. *International Journal for Numerical Methods in Engineering 51*, 1469–1495.

Zavarise, G. and P. Wriggers (1998). A segment-to-segment contact strategy. *Mathematical and Computer Modelling 28*, 497–515.

Zavarise, G. and P. Wriggers (1999). A superlinear convergent augmented lagrangian procedure for contact problems. *Engineering Computations 16*, 88–119.

Zavarise, G., P. Wriggers, and B. A. Schrefler (1995). On augmented lagrangian algorithms for thermomechanical contact problems with friction. *International Journal for Numerical Methods in Engineering 38*, 2929–2949.

Zavarise, G., P. Wriggers, and B. A. Schrefler (1998). A method for solving contact problems. *International Journal for Numerical Methods in Engineering 42*, 473–498.

Zhong, Z. H. and L. Nilsson (1988). A contact searching algorithm for general contact problems. *Computers & Structures 33*, 197–209.

Zhong, Z. H. and L. Nilsson (1990). A contact searching algorithm for general 3d contact-impact problems. *Computers & Structures 34*, 327–335.

# List of Symbols

| | |
|---|---|
| $\mathcal{A}$ | The parameter space for the surface mapping |
| $B_i^n$ | The $i$-th Bézier basis function of degree $n$ |
| $\mathbf{C}$ | A curve |
| $D$ | The directional derivative operator |
| $\mathrm{d}_{\mathrm{N}}$ | The signed normal distance |
| $\mathbf{d}_{\mathrm{N}}$ | The normal distance vector |
| $\mathrm{d}_{\mathrm{T}}$ | The magnitude of tangential slip |
| $\mathbf{d}_{\mathrm{T}}$ | The tangential slip vector |
| $e_i$ | Vector of zeros, except for a 1 at location $i$ |
| $\mathbf{F}$ | The deformation gradient |
| $F$ | The nodal force vector |
| $G$ | Functional containing regular terms of the weak form |
| $g$ | Gravitational acceleration |
| $G_c$ | Functional containing contact terms of the weak form |
| $\mathbf{I}$ | The identity operator |
| $I$ | Set of constraint labels |
| $\mathcal{I}$ | A set of capture boxes |
| $I_{\text{free}}$ | Set of node indices that are not displacement prescribed |
| $K$ | The stiffness matrix |
| $\mathcal{L}$ | A Lagrangian functional |
| $\boldsymbol{\lambda}$ | A barycentric coordinate |

| | |
|---|---|
| $m$ | The number of nodes |
| $\mathbf{n}_z$ | Normal vector in the $z$-direction |
| $N$ | The composite matrix of weighing functions |
| $\mathbf{n}$ | A unit normal vector |
| $n_d$ | The number of space dimensions |
| $N_i$ | A shape function for the $i$-th node |
| $\mathbf{P}$ | A control point for the Bézier description of a surface |
| $P$ | A potential function or functional |
| $p$ | A penalty parameter |
| $P_i$ | A control point |
| $\mathbb{P}$ | The Projection operator |
| $\mathbf{q}$ | A set of constraint parameters |
| $\mathbb{R}$ | The Euclidean space |
| $\mathbf{S}$ | A Bézier patch |
| $\mathcal{S}$ | Sobolev space |
| $s$ | A scaling term |
| $\mathcal{S}^h$ | Finite dimensional Sobolev space |
| $\mathbf{T}$ | A vector tangent to the boundary in the reference configuration |
| $\mathbf{t}$ | The traction vector |
| $t$ | A point in time |
| $\mathbf{t}_\mathrm{N}$ | The contact normal traction vector |
| $\mathrm{t}_\mathrm{N}$ | The magnitude of the normal traction |
| $\mathbf{t}_\mathrm{T}$ | The contact tangential traction vector |
| $\mathbf{v}$ | The velocity of a material point |
| $\mathcal{V}$ | Sobolev space of weighing functions |
| $\mathcal{V}^h$ | Finite dimensional Sobolev space of weighing functions |
| $\mathbf{v}_\mathrm{N}$ | The normal separation velocity |
| $\mathbf{v}_\mathrm{T}$ | The tangential slip velocity |

| | |
|---|---|
| $\mathbf{w}$ | A weighing function |
| $\mathbf{X}$ | A material point in the reference configuration |
| $\mathbf{x}$ | A material point in the current configuration |
| $x$ | Stacked column vector of nodal points |

| | |
|---|---|
| $\beta$ | The relative extrapolation point |
| $\Gamma$ | A set of boundary points in the reference configuration |
| $\Gamma_{\varphi}$ | The part of the reference boundary where displacements are prescribed |
| $\Gamma_{\sigma}$ | The part of the reference boundary where tractions are prescribed |
| $\gamma$ | A set of boundary points in the current configuration |
| $\gamma_{\varphi}$ | The part of the current boundary where displacements are prescribed |
| $\gamma_{\sigma}$ | The part of the current boundary where tractions are prescribed |
| $\epsilon$ | The inner loop termination criterion |
| $\boldsymbol{\varphi}$ | The mapping operator of points from the reference configuration to the current configuration |
| $\lambda_{\mathrm{N}}$ | Lagrange multiplier for the impenetrability constraint |
| $\boldsymbol{\lambda}_{\mathrm{T}}$ | The tangential Lagrange multiplier vector |
| $\mu$ | Friction coefficient |
| $\partial\Omega$ | The boundary of the set of material points |
| $\Phi$ | The Coulomb yield function |
| $\delta\boldsymbol{\varphi}$ | A variation of the displacement mapping |
| $\boldsymbol{\Psi}$ | A mapping from a parameter space to the contact boundary |
| $\rho$ | Density |
| $\boldsymbol{\sigma}$ | The Cauchy stress tensor |
| $\tau$ | A vector tangent to the boundary in the current configuration |
| $\tau$ | The outer loop termination criterion |
| $\boldsymbol{\xi}$ | A parameter for the surface mapping |
| $\Xi$ | A penalisation functional |

| | |
|---|---|
| $\Omega$ | A set of material points in the reference configuration |
| $\omega$ | A set of material points in the current configuration |
| $\zeta$ | consistency parameter for slip |
| $H(\cdot)$ | The Heaviside function |
| $\langle\cdot\rangle$ | The Macaulay bracket operator |

# ACKNOWLEDGEMENTS

This project has been made possible by the financial support of the Netherlands Institute for Metals Research, and the scientific support of the group of Technical Mechanics of the University of Twente, their essential input is gratefully acknowledged. An additional acknowledgement goes to Fokker aerostructures for allowing me the use of a toolshape for the stretch forming simulation in this thesis.

The first person I want to thank is Ton van den Boogaard, my supervisor during the project. Our different backgrounds would cause quite a few discussions to result in agreeing that we had disagreed over something we both agreed on. Off course also thanks to Han Huétink for having me work in his group in the first place, and for being so enthusiastic in discussing technical difficulties. Your "ooh, I have a thesis that contains something somewhere in a chapter on this," has certainly increased my stack of booklets. Yu Yuhong is thanked for bearing with me, when my enthusiasm on understanding something would mean listening to yet another story on contact. Also she has made sure I can intrigue chinese restaurant attendants by ordering szeh chuan chicken in chinese.

A big thank you also goes to Ruud van Damme who has set me on my tracks to go and do this work, and who has been a big help ever since he supervised me through my Masters degree. How quickly you understand stuff is incredible, as well as your ability to find corrections as if the page was initially written in pidgin English.

Thanks to Herman van Corbach for the tech-talk, at least there is someone working here with a passion for gadgets and Linux, and together with Nico van Vliet for taking care of the computer thing in general. Annemarie Teunissen, Debbie Vrieze, Tanja Gerritsen and Jacqueline Emmerich are thanked for helping with the office practicalities, where it needs to be remarked that office practicalities comprise everything from handing out paperclips to offering moral support. Also thanks to all the other colleagues in the applied mechanics group for providing such a relaxed working environment.

Apart from colleagues and work related things, I also want to thank the people who have made my stay in Enschede so much more enjoyable. In one particular order there is Gert Brendel, with whom I suffered through a whole year of organising fun activities. Then there is Mark Arends, with whom I have spent quite a few nights watching movies and doing a Waldorf and Stadler. And finally there is Sander Korthouwer, who is just too incorrigible for his own good. The latter two have even agreed to being my paranimfs. And to make sure I do not forget anyone: a thank you to all my friends for providing the ideal background for doing a PhD.

A special note of appreciation goes to my parents for pushing me to go and study, and making sure I had every opportunity to do so. And the last thank you goes to Daniëlle for giving the best support anyone could want.

# About the author

Gertjan Kloosterman was born in Leeuwarden, the Netherlands on March 5, 1973. In 1985 he started his secondary education (VWO) at the Rijksscholengemeenschap Simon Vestdijk in Harlingen, graduating in 1991. Having been told that science is the way to go, his final subjects comprised Mathematics, Physics, Chemistry and Biology, as well as the compulsory subjects English and Dutch, although languages and history have kept an interest ever since.

In September of 1991, he started his study of Mathematical Sciences at the University of Twente in Enschede, the Netherlands. During his studies he participated for two years in the "cabaretvereniging" Contramime, where he and two others wrote and performed a genuine production. For his final work and practical training he spent eight months in Australia, working on initial surface triangulations. The work was supervised by Dr. Ruud van Damme, in the Numerical Analysis group of Professor Traas. He received his Masters degree in January 1997.

From March 1997 till October 1997, he worked at Philips Display Components in Eindhoven. The work carried the title of improving surface description of shadow masks in cathode ray tubes. The final product was a software kit implementing several strategies for fitting and manipulating B-splines.

From October 1997 until September 1998 he worked for Paragon Decision Technologies in Haarlem on Branch-and-Bound algorithms for non-linear programming. This gave him the opportunity to learn an exciting and widely applicable field of mathematics: optimisation.

From September 1998 until January 1999, he briefly worked at the faculty of Mathematical Sciences of the University of Twente, yet again on surface triangulations. This resulted in an internal report and a joint paper on best data-dependent triangulations.

From January 1999 he started his PhD research on contact methods in finite element methods, joined DIOK to play badminton and learned FORTRAN to be able to program in the local finite element code DIEKA, (and after 4 years still thinks it is less expressive than C++).

Thanks to the participation in the graduate school of Engineering Mechanics, he had the opportunity to attend courses only touching his own research such as: Micromechanics of Materials, Structural Optimisation and Reliability and Structural Acoustics. All the work was performed as an employee of the Netherlands Institute for Metals Research (NIMR) and was performed at the faculty of Engineering Technology in the group of Applied Mechanics and Composites of the University of Twente.