

# Improving CONTACT for varying timesteps.

by

Niels van der Wekken

August 28, 2017

to obtain the degree of Master of Science  
at the Delft University of Technology,  
to be defended publicly on Thursday August 31, 2017 at 16:00 AM.

Student number: 1509594  
Project duration: April 1, 2016 – December 31, 2016  
Supervisors: Dr. ir. E. A. H. Vollebregt, VORtech  
Prof. dr. ir. C. Vuik, TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Theory</b>	<b>3</b>
2.1	Elasticity . . . . .	3
2.2	Tractions and slip . . . . .	4
2.3	Boundary Element Methods . . . . .	5
2.4	Halfspace approximation . . . . .	5
2.5	Quasi-identical behaviour . . . . .	6
2.6	Problem dependent simplifications . . . . .	6
2.6.1	2D vs 3D . . . . .	6
2.6.2	Time-dependency . . . . .	6
2.7	Discretization . . . . .	6
2.8	Implementation of CONTACT . . . . .	8
2.9	Piecewise linear approximation . . . . .	8
2.10	Iterative solvers . . . . .	8
<b>3</b>	<b>Discretisation in detail</b>	<b>9</b>
3.1	2D steady state and quasi-identical . . . . .	9
3.1.1	Active set algorithm . . . . .	9
3.1.2	Adhesion . . . . .	10
3.1.3	Slip . . . . .	10
3.2	2D transient and quasi-identical . . . . .	10
3.2.1	Adhesion . . . . .	11
3.2.2	Slip . . . . .	11
3.3	Implementation in Matlab . . . . .	11
<b>4</b>	<b>Smoothing</b>	<b>13</b>
<b>5</b>	<b>Problem statement and course of action</b>	<b>15</b>
5.1	Work done in this research . . . . .	15
5.1.1	2D steady-state rolling . . . . .	15
5.1.2	2D transient rolling . . . . .	16
5.1.3	Rate of convergence . . . . .	16
5.2	Research Questions . . . . .	16
<b>6</b>	<b>Using higher order basis functions</b>	<b>19</b>
6.1	Explicit expressions for the piecewise-constant case . . . . .	20
6.2	Influence coefficients using linear basisfunctions . . . . .	20
6.2.1	Integration using Maxima . . . . .	23
6.3	Fourier analysis . . . . .	26
6.4	Piecewise Quadratic Basisfunctions . . . . .	29
6.4.1	Rate of Convergence . . . . .	29
<b>7</b>	<b>Time-stepping</b>	<b>31</b>
7.1	Under-relaxation of $A'$ . . . . .	31
7.2	Numerical diffusion . . . . .	31
<b>8</b>	<b>Fractional matrix powers</b>	<b>35</b>
8.1	Fractional power computation . . . . .	35

<b>9</b>	<b>World Fixed implementation</b>	<b>39</b>
9.1	CONTACT libraries for Matlab . . . . .	39
9.2	Rolling cylinder . . . . .	39
9.3	Evolution in time . . . . .	40
	9.3.1 Velocity Verlet . . . . .	40
	9.3.2 Leapfrog . . . . .	40
9.4	Energy . . . . .	41
<b>10</b>	<b>Leading edge adjustments</b>	<b>43</b>
10.1	Analytic solution in 2D steady state . . . . .	43
10.2	Solution scheme using the leading edge displacement . . . . .	44
10.3	Effects on convergence . . . . .	44
10.4	Adjusting the Leading Edge basis function . . . . .	45
<b>11</b>	<b>Conclusion and recommendations</b>	<b>47</b>
11.1	Recommendations for further research . . . . .	47
	<b>Appendices</b>	<b>49</b>
<b>A</b>	<b>Explicit expression using constant basis functions</b>	<b>51</b>
<b>B</b>	<b>Influence coefficients based on piecewise quadratic basisfunctions</b>	<b>53</b>
<b>C</b>	<b>Leading Edge influence coefficient</b>	<b>59</b>
	<b>Bibliography</b>	<b>61</b>

# Preface

This master thesis will be done at VORtech under supervision of E.A.H. Vollebregt. Vollebregt, once a student of Prof. J.J. Kalker, is an expert in the field of (rolling) contact mechanics and has done much work on the software package CONTACT. Recently it has been discovered that for certain small values in the discretized timestep non-physical 'wiggles' show up in the results. My master thesis will focus on finding out what causes these wiggles to arise and how we can prevent this.

VORtech is a scientific software engineering company that produces, maintains, and optimizes scientific software. They have long-term contracts with big corporations like Shell, Rijkswaterstaat, Deltares, and TNO as well as smaller projects with SMEs. The majority of its 27 employees has a mathematical or physical background. In the first place VORtech works by letting clients hire VORtech's mathematicians to be an addition to their team, backing them up with specific expertise. Secondly VORtech develops and maintains software packages for clients that need certain software but do not develop software themselves. Lastly VORtech provides clients with mathematical consultancy.

The software package CONTACT is maintained by Vollebregt, co-founder of VORtech. CONTACT is mainly used in railway simulation packages to calculate the forces and other physical parameters that occur in the wheel-rail contact. My master thesis will revolve around the forces calculated by this software package.

*Delft, June 2016*



# Introduction

Contact mechanics describes the way two (elastic) objects interact with each other when they touch, make contact. As the two bodies are pressed against each other, at the point of contact the bodies want to take up the same spot in space, resulting in each of them applying a repelling force on the other. This force will deform the bodies so that a contact area appears where the resulting contact forces balance the forces pressing the bodies together. In a wheel-rail system the two bodies (the wheel and the rail) are not just pressed together, but the wheel will be rolling over the rail. This rolling adds a dynamical component to the system, in the form of tangential friction.

J.J. Kalker [19] first developed the software package DUVOROL in 1978 [12, 13], this package assumed an elliptic contact area, as described in the Hertz theory [9], and was limited to steady-state rolling contact. In 1982 he finished its successor CONTACT. This package could determine the actual contact area, and could calculate for an instant frictional shift and transient rolling contact. Both programs were written in FORTRAN IV. The underlying theory and the framework of the algorithms are described in Kalker's most cited work [14]. The book describes the problem of rolling contact mechanics and gives both analytic solutions for certain geometries and the numerical scheme for more complex geometries. Although the software version of 1990 is outdated, the core of the methodology has not changed.

In 1992-4 E.A.H. Vollebregt rewrote CONTACT in FORTRAN 77 [27], modernizing it in a way that is easier adaptable. After the publishing of Kalker's book in 1990 there has been much development on CONTACT. Adding more functionality, like extensions on the friction model [26, 29] and an extension towards conformal contact areas [30]. But also implementing faster numerical solvers, utilizing Conjugate Gradient methods and Fast Fourier Transformations [22, 25, 34] to improve the performance.

This thesis will deal with problems arising when varying the timestep. The focus will be on the occurrence of artificial numerical wiggles that arise when the timestep becomes small, see Figure 1.1 for an example. More specifically, the wiggles arise when the traversed distance per timestep  $\delta t \cdot V$  is small compared to the gridsize  $\delta x$ . For brevity this traversed distance per timestep will be simply called the "timestep"  $\delta q$  although  $\delta q$  is actually a distance. Its size compared to the gridsize is controlled by a parameter  $c$  given by  $\frac{\delta t \cdot V}{\delta x} = \frac{\delta q}{\delta x}$ . In the numerical scheme that arises the most influential factor is a matrix  $B$ . Without giving further details yet, there are a few things that happen to  $B$  when  $c \ll 1$ . When  $c = 0$  the matrix is singular and no solution exists. As  $c \downarrow 0$  the imaginary parts of the eigenvalues of  $B$  become bigger, relative to the real parts. Also the condition number grows rapidly as  $c$  decreases, meaning that the solution will be very sensitive to small deviations in the input, so little errors blow up in the results.

In [23] it is shown that the ratio  $c = \frac{\delta q}{\delta x}$  is an important factor concerning the accuracy of the model. Whereas in other applications it is typically found that a lower value of  $c$  results in better accuracy, here it is found that for  $c < 0.55$  the accuracy decreases again.

Another artifact that has been found is a form of smoothing of sharp peaks or kinks in the solution. This numerical diffusion-like phenomenon will be explained in the second part of the thesis and solutions for

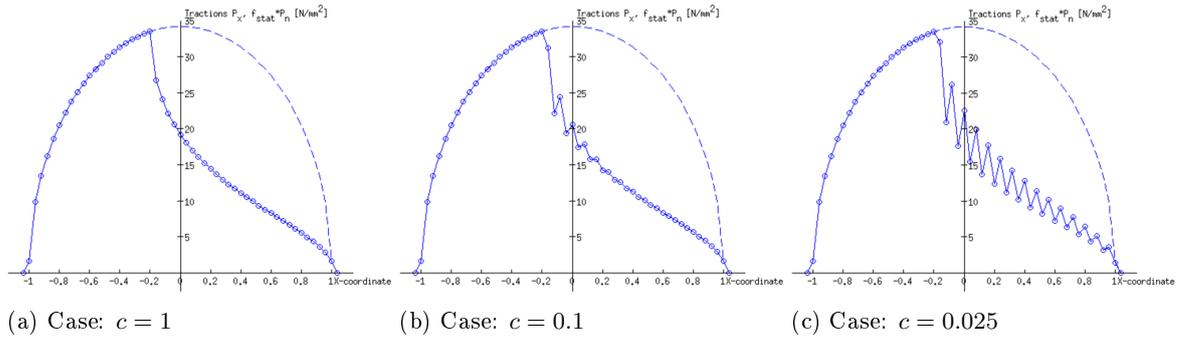


Figure 1.1: Computations of tractions in 2D in CONTACT choosing different values of  $\delta q$ . This is the 2D Carter/Fromm problem, see [27, section 5.2].

this problem are sought.

At last the convergence of the algorithm is investigated. Currently the algorithm only has first order convergence whereas second order convergence is expected. The last part of the thesis will show that the bottleneck for the convergence lies in the leading edge of the contact zone.

In Chapter 2 the necessary theoretical framework will be given. This includes the physical background that lies at the basis of contact mechanics (Sections 2.1 and 2.2), techniques used to simplify the problem (Sections 2.3 - 2.6), the discretization (Section 2.7), and implementation (Section 2.8). Also remarks on the use of piece-wise functions (Section 2.9) and the use of iterative solvers (Section 2.10) is given.

Chapter 3 will give a more detailed insight in how exactly the discretized slip-equations are solved, followed by a description of the problem of smoothing in Chapter 4.

Chapter 5 describes the problem that needs to be solved and the different cases that can be considered. In Section 5.1 a short description of the research done is given. Followed in Section 5.2 by the research questions for my master thesis.

In Chapter 6 the wiggle phenomenon will be investigated. Smoothing of the solution will be further discussed in Chapters 7.2-9. Chapter 10 will deal with the rate of convergence of the algorithm.

Finally a conclusion along with recommendations is given in Chapter 11.

# 2

## Theory

In the following sections more details will be given concerning the equations that need to be solved, and the way they are currently solved. The basic theory of elasticity can be found in Love [17]. This book includes theory on elasticity and the concepts of stress and strain. A good basis for the theory on contact mechanics can be found in Johnson [10].

### Elasticity

#### Deformation

In order to work with elastic contact problems we need to understand what elastic deformation is. Deformation of a physical body is the transformation of the positions of particles in the reference state to a their positions in the new state. When a particle had position  $\mathbf{x}$  and after deformation has position  $\mathbf{x} + \mathbf{u}$ , then the deformation is  $\mathbf{u}$ .

#### Strain

Now introduce the strain  $\epsilon$ . The strain consists of longitudinal strains  $\epsilon_{ii} = \frac{\partial u_i}{\partial x_i}$  corresponding to stretching of the body, and shear strains  $\epsilon_{ij} = \frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}$  for  $i \neq j$  corresponding to bending of the body. The strains describe the relative displacement of the particles in the body.

#### Stress

Strain in the body can be caused both by outside stress, forces like gravity or pressure acting on the body, as well as by internal elastic stress, resulting from the material resisting change. The strains are linked to (internal) stress  $\sigma$ , according to the generalized Hooke's Law through the elastic tensor  $C$  by  $\sigma_{ij} = C_{ijkl}\epsilon_{kl}$ . We will use Einstein summation convention throughout this documents unless specified (as we just did with the elastic tensor), there will be no summation over the  $\epsilon_{ii}$  though in the previous paragraph.

When the elastic behaviour of a material is independent on the direction of the stress- and strain-components the material is said to be isotropic. For example in the case of metals. An example where this is no longer valid is wood, because of the uniformly directed fibrous structure the wood reacts different under stress perpendicular or parallel to the direction of the fibres. The elastic tensor  $C_{ijkl}$  for an isotropic material reduces to two material properties  $\lambda$  and  $\mu$ . Here  $\mu$  is the rigidity and  $\lambda + \frac{2}{3}\mu$  the modulus of compression. Furthermore, when the material is homogeneous the  $\lambda$  and  $\mu$  are constant through the material. For an isotropic material we can give the stresses as functions of the strains through:  $\sigma_{ii} = \lambda(\epsilon_{11} + \epsilon_{22} + \epsilon_{33}) + 2\mu\epsilon_{ii}$  and  $\sigma_{ij} = \mu\epsilon_{ij}$ ,  $i \neq j$ . Again do not use summation over  $\sigma_{ii}$  and  $\epsilon_{ii}$ .

## Tractions and slip

The following scalar and vector quantities are important in the contact model. Before the bodies are brought into contact we refer to them by their coordinates  $\mathbf{x}$  in a right-handed coordinate system where the  $x$ -axis points in the rolling direction and the  $z$ -axis points upwards into the upper body. When the bodies are brought into contact stresses  $\sigma$ , strains  $\epsilon$  and displacements  $\mathbf{u}$  arise. We are particularly interested in the surface quantities where we call the surface stresses of body 1  $\mathbf{p}^{(1)}(\mathbf{x})$  and of body 2  $\mathbf{p}^{(2)}(\mathbf{x})$ . Because these surface stresses work against each other they have the same amplitude but opposite sign:  $\mathbf{p}^{(1)}(\mathbf{x}) = -\mathbf{p}^{(2)}(\mathbf{x})$ , so in the model we will only consider the surface stress on body 1 and call this  $\mathbf{p}(\mathbf{x}) = \mathbf{p}^{(1)}(\mathbf{x})$ . After the deformation the displacement of body 1 at a point  $\mathbf{x}$  is given by  $\mathbf{u}^{(1)}(\mathbf{x})$ . Now let the displacement difference be given by the difference in the displacements of bodies 1 and 2 at a given position:  $\mathbf{u}(\mathbf{x}) = \mathbf{u}^{(1)}(\mathbf{x}) - \mathbf{u}^{(2)}(\mathbf{x})$ . Furthermore we split the stress vector into a scalar value  $p_n$  for the normal stress, and the 2-vector  $\mathbf{p}_\tau$  for the tangential stresses, called tractions. The normal distance between the undeformed surfaces is given by  $h$ . Now the normal distance between the bodies in the deformed state is given by:

$$e := h + u_n \quad (2.1)$$

Finally we have a relative slip  $\mathbf{s}$  (also a 2-vector as there is no slip in the normal direction) which describes how fast the upper and lower body slide over each other compared to the rolling velocity given by [14, equation 1.25]:

$$\mathbf{s} = \mathbf{w} + \frac{\dot{\mathbf{u}}}{V} \quad (2.2)$$

$V$  is the rolling velocity magnitude, because we choose the velocity to always be in the  $x$ -direction we have that  $\mathbf{v} = [V, 0, 0]^T$ . Here  $\mathbf{w} = [\xi - \phi y, \eta + \phi x]^T$  is the relative rigid slip given as a function of the longitudinal and lateral creepage  $\xi$  and  $\eta$  and the spin creepage  $\phi$ . The relative slip and relative rigid slip are relative compared to the magnitude of velocity. In rolling the build up of tractions is mostly governed by the velocity of creeping relative to the overall rolling velocity. Therefore we define the relative slip velocity  $\mathbf{s}_{relative} = \mathbf{s}_{absolute}/V$  as a dimensionless slip velocity and abbreviate  $\mathbf{s} := \mathbf{s}_{relative}$ . The quantity  $\dot{\mathbf{u}}$  is the material derivative of the displacement given by:  $\dot{\mathbf{u}} = \frac{\partial \mathbf{u}}{\partial t} - V \frac{\partial \mathbf{u}}{\partial x}$ . The minus in front of the spatial derivative is because as the upper body moves in positive  $x$ -direction, this means that the surface particles actually move in the negative  $x$ -direction through the contact area.

Now we can define the contact conditions:

In the normal problem:

$$\text{in exterior area E : } e > 0, p_n = 0 \quad (2.3)$$

$$\text{in contact area C : } e = 0, p_n \geq 0 \quad (2.4)$$

In the tangential problem:

$$\text{in exterior area E : } \mathbf{s} \text{ free, } \mathbf{p}_\tau = \mathbf{0} \quad (2.5)$$

$$\text{in adhesion area H : } \|\mathbf{s}\| = 0, \|\mathbf{p}_\tau\| \leq g \quad (2.6)$$

$$\text{in slip area S : } \|\mathbf{s}\| > 0, \mathbf{p}_\tau = -g \frac{\mathbf{s}}{\|\mathbf{s}\|} \quad (2.7)$$

This means that in the exterior area E the bodies do not touch each other so there is no stress and the bodies move freely from each other so the slip can be anything. In the adhesion area H the bodies touch each other without slipping, the stress between them is bound by an upper bound  $g$ , in the slip area S the bodies slide over each other and the stress reaches its upper bound and is directed in the opposite direction from the slip. This upper bound is called the traction bound and is given by Coulomb's friction law [5] as

$$g = \mu p_n \quad (2.8)$$

where  $\mu$  is the coefficient of friction.

The displacements at  $\mathbf{x}$  can now be determined by integrating the product of the stress at the contact area  $C$  with an influence function using:

$$u_i(\mathbf{x}) = \int_C A_{ij}(\mathbf{x}, \mathbf{y}) p_j(\mathbf{y}) dS \quad (2.9)$$

This equation describes how  $\mathbf{u}$  is a function of the tractions  $\mathbf{p}$  at the contact surface  $C$ . The indices  $i, j$  in the function  $A_{ij}$  run over 1, 2, 3 and  $A_{ij}(\mathbf{x}, \mathbf{y})$  tells us how the traction  $p_j(\mathbf{y})$  influences the displacement  $u_i(\mathbf{x})$ . Note that we use Einstein summation convention over the index  $j$ .

## Boundary Element Methods

The Boundary Element Method (BEM) is a numerical approximation method for solving partial differential equations (PDEs). The BEM is derived by rewriting the PDE into an integral equation defined on a boundary. Discretizing this integral equation gives the BEM. An advantage of BEMs over finite element methods, finite volume methods, and finite difference methods is that the BEM only solves the equations on the boundary of the domain of interest. Especially when Greens functions are known and the influence can be calculated analytically it is interesting to solve the system on the boundary using a BEM. Only calculating for the boundary means that we have one dimension less to work with, so with the same grid coarseness as in other methods, less grid points need to be considered, beware though that the system of equations coming from a BEM is dense while a FEM generally gives a sparse system of equations. Another advantage is gained when the body of interest is unbounded but has a bounded boundary, thus resulting in a finite domain for the BEM.

The major drawback when using a BEM is of course the fact that you do not calculate anything that happens below the boundary. When you are interested in internal parameters that can not be calculated in a straightforward way from properties on the boundary (when there are no Greens functions or comparable relations) you introduce extra errors, usually in the form of extrapolation errors.

## Halfspace approximation

Although the full stress-strain equations are complex it is possible to make some assumptions to simplify the model. In our model we make use of the half-space approach. This means that we assume that the two contacting bodies are infinite half spaces. This approach can be made if the bodies look like the halfspaces in a zone where the elastic field is significant and only begin to differ significantly from the half spaces where the elastic field is very small. This means that the contact area must be small compared to the typical dimension of the bodies such that the radius of curvature is large near the contact area. See Figure 2.1. Results in Kalker [14, Figure 5.20] “*support the statement that the half-space approximation is justified, when the diameter of contact is less than 1/3 of the diameter of the contacting bodies*”.

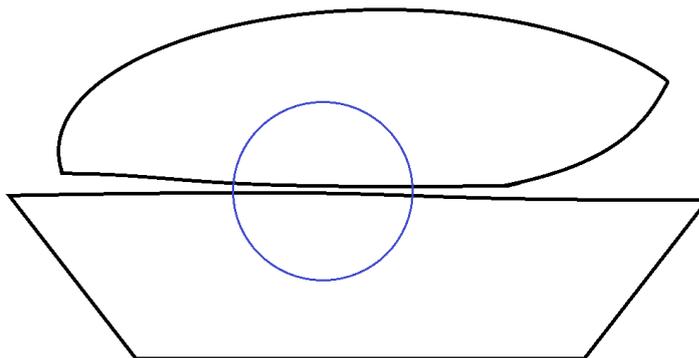


Figure 2.1: When the upper and lower body make contact the stresses and strains will only be ‘felt’ inside the blue circle because the influence of stresses and strains decreases as the distance increases away from the contact zone. So the geometry of the bodies outside the blue circle is unimportant and for contact calculations we may assume that the bodies are two half spaces.

The strenght of the halfspace approximation lies in the fact that the formula’s for the influence functions  $A_{ij}$  used in equation (2.9) are derived by Boussinesq [3] and Cerruti [4] for the halfspace. So explicit Greens functions can be used in the BEM.

## Quasi-identical behaviour

Another simplification occurs when the two bodies of contact are quasi-identical. This is the case when both the geometry and the elastic behaviour of the bodies are similar. The first is automatically the case when we already have applied the half-space approach, both bodies are halfspaces, and thus identical. The second condition is fulfilled when the Young's Modulus  $E$  and Poisson's Ratio  $\nu$  are equal. Stresses in one direction influence deformations in all three directions, when the two contacting bodies are not quasi-identical the normal and tangential problem influence each other. When the bodies are quasi-identical however the tangential displacements are the same for both bodies so there is no relative tangential displacement and we can separate the normal and tangential problem.

Applying a BEM to solve the contact between two quasi-identical half spaces greatly simplifies the model. Because now the normal and tangential problems are separated we first solve the normal problem on the contact surface and use the solution for the normal problem when solving the tangential problem on the contact surface.

## Problem dependent simplifications

Based on which assumptions we make we can adjust the equations to make them easier to solve. We will make distinctions between solving the system in 2D and 3D and between transient and steady-state rolling.

### 2D vs 3D

When we make the half-space approximation and the objects of contact are quasi-identical the normal and tangential problems are already separated. So in both the 2D and 3D case we will first solve the normal problem. In the 3D case the tangential contact area is 2D and equation (2.7) is a quadratic equation. In the 2D case however the tangential contact area becomes a 1D line and equation (2.7) is linear, thus easy to solve. This means that the 2D case is much easier to solve. As the wiggles and other discretization effects occur both in the 2D and 3D problem a solution is first sought for the 2D case since this case is easier to study and a solution in 2D might be applicable in the 3D case as well.

### Time-dependency

Equation (2.2) contains a time derivative. When we have a transient system we need an initial state. Once this initial state is known the full solution has to be determined by evolving the solution in time.

A steady-state solution can be found in two ways. The first is simply applying the algorithm to solve the transient rolling case and stop once the solution of the current time instance is the same as the solution of the previous time instance. This is also what was originally implemented in Kalker's DUVOROL software.

A more sophisticated approach is going back to the equation for the stress (2.2) and set the time-derivative to zero. Now  $\mathbf{p}_\tau$  and thereby  $\mathbf{u}_\tau$  can be calculated directly. This method will be called the direct approach.

## Discretization

Now that we have simplified the physical model into a mathematical description we can discretize the problem. As stated in [14, 23] in CONTACT the potential contact area is discretized using identical rectangles  $I$  with size  $\Delta x \times \Delta y$ . The centre of rectangle  $I$  is denoted by  $(x_I, y_I)$ , note that there is no  $z$ -component because the contact area lies in the x-y plane so  $z = 0$  is a constant on the whole contact area. Equation (2.3) through (2.7) are discretized by using piecewise constant functions  $\phi_I$  that are 1 on rectangle  $I$  and 0 everywhere else. The true solution  $\mathbf{p}$  is then approximated by the piecewise constant function  $\sum_i \phi_i \cdot \mathbf{p}_i$ .

The slip will be discretized by rewriting the slip equation. When looking at steady-state the time derivative in the material derivative in equation (2.2) drops out and we are left with:

$$\mathbf{s} = \mathbf{w} - \frac{\partial \mathbf{u}}{\partial x} \quad (2.10)$$

Before we can discretize this equation we need to approximate the spatial derivative. The slip equation is an advection equation, these equations are generally solved using an upwind-scheme. What happened upstream is known, so using data from upstream will generally give good results. Because the velocity is in the negative direction this upwind-scheme gives us a forward differentiation for the spatial derivative. A first order scheme is used because of its robustness. We may now approximate with a small enough  $\Delta x$  the derivative to  $x$  by using:  $\frac{\partial \mathbf{u}}{\partial x} \approx \frac{\mathbf{u}(\mathbf{x} + [\Delta x, 0]^T) - \mathbf{u}(\mathbf{x})}{\Delta x}$ . This will give us the expression:

$$\mathbf{s} = \mathbf{w} - \frac{\mathbf{u}(\mathbf{x} + [\Delta x, 0]^T) - \mathbf{u}(\mathbf{x})}{\Delta x} \quad (2.11)$$

If we now use  $\Delta x = V \cdot dt = dq$  and write  $\mathbf{x} + [dq, 0]^T = \mathbf{x}'$  then a particle that has position  $\mathbf{x}$  at time  $t$  had position  $\mathbf{x}'$  at time  $t' = t - dt$ . This turns the slip equation into:

$$\mathbf{s} = \mathbf{w} + \frac{\mathbf{u}(\mathbf{x}) - \mathbf{u}(\mathbf{x}')}{dq} \quad (2.12)$$

Now only looking at the values of  $\mathbf{s}$  and  $\mathbf{w}$  in the centres of the rectangles  $I$  we get the discretized form:

$$\mathbf{s}_I = \mathbf{w}_I + \frac{\mathbf{u}(\mathbf{x}_I) - \mathbf{u}(\mathbf{x}'_I)}{dq} \quad (2.13)$$

The solution for the displacement do not need to be discretized,  $\mathbf{u}(\mathbf{x})$  can be determined using equation (2.9) for any  $\mathbf{x}$ . Because  $\mathbf{p}$  has been replaced by a piecewise constant function the integral in equation (2.9), it can be solved analytically using the results by Boussinesq [3] and Cerruti [4]. The influence functions are discretized by calculating the displacements felt in element  $I$  as a result of tractions in element  $J$ . This means that influence coefficient  $A_{IiJj}$  indicates how the displacement in rectangle  $I$  in the  $i$  direction is influenced by tractions in rectangle  $J$  in the  $j$  direction. So we can write:

$$A_{IiJj} = \iint_S A_{ij}(\mathbf{z}) dS \quad (2.14)$$

Where  $S$  is the surface of a rectangle that has centre  $\mathbf{x}_J - \mathbf{x}_I$  with width  $\Delta x$  and height  $\Delta y$ . The integral can thus be written as:

$$A_{IiJj} = \int_{x_J - x_I - \frac{\Delta x}{2}}^{x_J - x_I + \frac{\Delta x}{2}} \int_{y_J - y_I - \frac{\Delta y}{2}}^{y_J - y_I + \frac{\Delta y}{2}} A_{ij}(\mathbf{z}) dz_2 dz_1 \quad (2.15)$$

Explicit expressions for the influence functions  $A_{ij}$  and solutions to the integrals for  $A_{IiJj}$  are given in [14, section 4.3.2].

When the tractions  $\mathbf{p}_J$  and influence functions  $A_{IiJj}$  are know we can now determine the values of  $\mathbf{u}(\mathbf{x}_I)$  by using:

$$u_i(\mathbf{x}_I) = \sum_{J_j} A_{IiJj} p_{Jj} \quad (2.16)$$

Using this in equation (2.13) we find for the discretized slip equation in case of steady-state:

$$s_{Ii} = w_{Ii} + \frac{(A_{IiJj} - A'_{IiJj}) p_{Jj}}{dq} \quad (2.17)$$

where  $A'_{IiJj}$  is calculated similar to  $A_{IiJj}$  but then with the centres of the rectangles  $I$  shifted a distance  $dq$  to the right.

## Implementation of CONTACT

In CONTACT the equations arising from the contact mechanical theory are modified into a minimization problem. This is done through Kalker's variational theory [11], using the principals of minimizing the virtual work and maximizing the virtual complementary energy [14, §4.1-4.2]. This minimization problem turns out to be a strictly convex quadratic problem, opening it up to a rich theory around Quadratic Programming on a convex objective function constrained to a convex feasible region [2, 6, 15, 18, 21].

Kalker used an algorithm that solves the convex minimization problem to solve the contact problem and calculate the tractions. This algorithm, KOMBI, computes both the normal and tangential tractions. In the case of quasi identical elastic bodies the KOMBI algorithm can be split into first solving the normal stresses  $p_{In}$  using NORM and then using the normal solution to solve the tangential tractions  $p_{I\tau}$  using TANG [14, §4.3].

These algorithms are active-set algorithms. Solving the equations for the tractions, either in adhesion or slip, and restoring discretized elements of the contact area to the slip or the adhesion region when the constraints (2.6) or (2.7) are exceeded.

Because of the convex quadratic programming approach, existence, uniqueness, and finite determinability of the active set algorithm are given.

## Piecewise linear approximation

The use of a piecewise constant approximation for the tractions  $\mathbf{p}$  might be a cause of the appearance of wiggles. To prevent or reduce these wiggles we will look at using piecewise linear approximations. One direct advantage these solutions will give is the fact that while the derivative at any location of a piecewise constant function is either zero or does not exist (the derivative could be seen as a sum of Dirac delta functions), the derivative of a piecewise linear function is a piecewise constant function. In 1D the shape of a piecewise linear function is trivial, however, in 2D the function can increase in the  $x$ -direction,  $y$ -direction or in both the  $x$ - and  $y$ -direction at the same time. The last case is called bi-linear.

Multiple groups have published on Boussinesq-Cerruti solutions for (piecewise) linear functions instead of using the (piecewise) constant function as applied in CONTACT. Svec and Gladwell [20] give solutions of normal deformation due to polynomial normal pressure distributions on a triangular surface area. Li and Berger [16] extend this by giving the full solutions for constant and (bi)linear pressure loads over a triangular surface area.

The advantage of dividing the contact domain in triangles is its high adaptation level to curved edges of the contact domain compared to rectangles. However, using rectangles simplifies the computational model. Also, when the edges of the contact area are unknown in advance a rectangular grid has the advantage over a triangular grid. CONTACT also uses rectangles to describe the contact domain, so for this thesis we will focus on rectangular domains. Dydo and Busby [7] give solutions to constant and (bi)linear pressure loads over a rectangular surface area.

## Iterative solvers

In the two dimensional case constraints (2.6) and (2.7) are linear, in the three dimensional case however these are not linear anymore. This means that the system of equations to solve the discretized slip is not linear either. In the early versions of CONTACT this system was solved using a Newton-Raphson method [33].

In 1993 an improvement was made by implementing a variation on the Gauss-Seidel method, later enhanced and stabilized by application of Successive Over Relaxation [22, 31].

Although these iterative solvers are used in the software, they are outside the area of interest of this thesis.

## Discretisation in detail

In this chapter the discretisation will be performed in more detail. In Section 3.3 details about the implementation will be given.

From equation (2.2) we have:

$$\mathbf{s} = \mathbf{w} + \frac{1}{V} \frac{\partial \mathbf{u}}{\partial t} - \frac{\partial \mathbf{u}}{\partial x} \quad (3.1)$$

where  $\mathbf{w} = [\xi - \phi y, \eta + \phi x]^T$  is given.

### 2D steady state and quasi-identical

In 2D steady state, equation (3.1) becomes:

$$s = w - \frac{\partial u}{\partial x} \quad (3.2)$$

Using first order upwind with a position step  $dq$  independent of the gridsize this becomes:

$$s = w - \frac{u(x + dq) - u(x)}{dq} \quad (3.3)$$

Discretization gives us:

$$s_I = w_I + \frac{u(x_I) - u(x_I + dq)}{dq} \quad (3.4)$$

Now use  $u(x_I) = \sum_{Jj} A_{I1Jj} p_{Jj}$  and  $u(x_I + dq) = \sum_{Jj} A'_{I1Jj} p_{Jj}$ , where  $A'_{I1Jj}$  is calculated as in equation (2.15) with in the limits  $x_I$  replaced by  $x_I + dq$ . Using Einstein summation we write  $\sum_{Jj} A_{I1Jj} p_{Jj} = A_{I1Jj} p_{Jj}$ . Normally  $j$  would take values 1, 2, 3 representing the  $x, y, z$  directions. However, in the 2D system there is no  $y$ -direction. Also when considering a quasi-identical system the influence function  $A_{1,3}$  is zero. We are therefore left with just  $u(x_I) = A_{I1J1} p_{J1}$ . This turns the equation into the system:

$$s_I = w_I + \frac{(A_{I1J1} - A'_{I1J1}) p_{J1}}{dq} \quad (3.5)$$

### Active set algorithm

Equation (3.5) seems to be problematic because it contains both the unknowns  $s_I$  and  $p_{J1}$ . To solve this problem an active set algorithm is used. This process is described by Kalker [14, section 4.3.1] and does the following:

- Fix the value of the slip.

- For elements where the slip is zero, calculate the tractions  $p_{J1}$ .  
For elements where the slip is nonzero, set the traction at the traction bound.
- Check if, with this solution, no constraints are violated. If there are, adjust the slip and adhesion regions accordingly and calculate new slip values  $s_I$ .
- Repeat from the first step until a feasible solution is found.

So after choosing an initial condition (zero slip everywhere is a good candidate) we solve equation (3.5) assuming  $s_I = 0$  to find the tractions in adhesion H, using these new found tractions we solve equation (3.8) where the  $p_{J1}$  are known to find the slip in S.

While doing this we have ignored the constraints in the slip-conditions (2.6) and (2.7). These constraints are possibly violated.

First check if in H the traction has crossed the traction bound. If we find elements  $I$  where  $|p_{I1}| > g_I$  these elements are taken out of H and moved to S. After this we solve (3.5) again, using the new division in H and S.

When all calculated tractions are within the traction bound, we check the value of the slip. For any element  $I$  where we find  $s_I \cdot p_I > 0$  we remove this element from S and move it to H. After this we solve (3.5) again, using the new division in H and S.

This process continues until a solution is found that solves the discretized equations and does not violate the constraints.

Now we have to divide our contact domain into two sections, one where the upper and lower body stick to each other, the adhesion area H, and one where the upper body slips over the lower body, the slip area S. Let  $N$  be the total number of elements in the contact area,  $N_H$  be the number of contact elements in the adhesion area H, and  $N_S$  be the number of contact elements in the slip area S.

## Adhesion

In the adhesion area we use equation (2.6), so with zero slip we get:

$$0 = w_I + \frac{(A_{I1J1} - A'_{I1J1})p_{J1}}{dq} \quad (3.6)$$

Note that we are only solving the equation for the tractions  $p_{J1}$  where element  $J$  lies in H, so this are  $N_H$  equations. As  $w_I$  is known we can, in adhesion, directly calculate the  $p_{J1}$  by:

$$p_{J1} = -(A_{I1J1} - A'_{I1J1})^{-1}w_I dq \quad (3.7)$$

## Slip

While in the slip area we use (2.7) and solve the slip equation as:

$$s_I = w_I + \frac{(A_{I1J1} - A'_{I1J1})p_{J1}}{dq} \quad (3.8)$$

These  $N_S$  equations immediately give an expression for the slip  $s_I$  in known terms.

## 2D transient and quasi-identical

When instead of looking at steady-state we look at the transient equations we can no longer ignore the time derivative in equation (3.1), now using a Lagrangian approach gives us:

$$s = w + \frac{1}{V} \frac{Du}{Dt}. \quad (3.9)$$

Because we are assuming a transient situation we need a solution at the starting time  $t = 0$ . Now we can discretize the equation in time. We indicate a time-discretised solution at time  $dt \cdot i$  by  $u^{(i)}$ . This transforms the time-derivative, using Euler backwards, into:  $\frac{Du}{Dt} \approx \frac{u^{(i)}(x(t)) - u^{(i-1)}(x(t-dt))}{dt}$ .

$$s^{(i)} = w + \frac{1}{V} \frac{u^{(i)}(x(t)) - u^{(i-1)}(x(t-dt))}{dt}. \quad (3.10)$$

Discretization in space and using that  $V \cdot dt = dq$  gives us:

$$s_I^{(i)} = w_I + \frac{u^{(i)}(x_I(t)) - u^{(i-1)}(x_I(t-dt))}{dq}. \quad (3.11)$$

We write  $u'$  for  $u$  at the previous timestep and drop the notation for the discretized time again. Also note that if a particle is at position  $x_I$  at time  $t$ , then it was at position  $x_I + V \cdot dt = x_I + dq$  at time  $t - dt$ . So write  $x_I$  for  $x$  at the current time and  $x'_I$  for  $x$  at the previous timestep.

$$s_I = w_I + \frac{u(x_I) - u'(x'_I)}{dq}. \quad (3.12)$$

Now use  $u_I = A_{I1Jj} p_{Jj}$  again. As we are still in the 2D quasi-identical system we are left with just  $u(x_I) = A_{I1J1} p_{J1}$ . This turns the equation into the system:

$$s_I = w_I + \frac{A_{I1J1} p_{J1} - A'_{I1J1} p'_{J1}}{dq}. \quad (3.13)$$

Here  $A_{I1J1}$  is defined in the same way as (3.5). Not only  $w_I$  is known but also  $p'_{J1}$  is known. This is the solution of the tractions calculated at the previous timestep (or the initial conditions during the first step). Following the same element division into H and S we can further specify the equations we use to solve the system in adhesion or slip.

## Adhesion

In the adhesion area we use equation (2.6), with zero slip we get:

$$0 = w_I + \frac{A_{I1J1} p_{J1} - A'_{I1J1} p'_{J1}}{dq} \quad (3.14)$$

So we can in adhesion directly calculate the  $p_{J1}$  by:

$$p_{J1} = A_{I1J1}^{-1} (-dq w_I + A'_{I1J1} p'_{J1}) \quad (3.15)$$

## Slip

In the slip area we use (2.7) and solve the slip equation as:

$$s_I = w_I + \frac{A_{I1J1} p_{J1} - A'_{I1J1} p'_{J1}}{dq} \quad (3.16)$$

This immediately gives an expression for the slip  $s_I$  in known terms.

## Implementation in Matlab

As already mentioned in Section 3.1 we do not calculate the tractions and slip in the full domain. In steady state equation (3.8) is valid in the full domain, however equations (3.6) and (3.2) are only valid in the adhesion and slip region respectively. This is why we implement these equations only in their respective areas. Let's take a closer look at the equations.

In *steady state* we work with a matrix  $M = (A_{I_1J_1} - A'_{I_1J_1})/dq$ . Because we do not have the same conditions in the adhesion and slip area we will make a division of the vectors  $\mathbf{p}$ ,  $\mathbf{s}$ ,  $\mathbf{w}$  and matrix  $M$  such that each subvector either contains elements corresponding to the adhesion or the slip area.

$$\mathbf{p} = \begin{bmatrix} \mathbf{p}_H \\ \mathbf{p}_S \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} \mathbf{s}_H \\ \mathbf{s}_S \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} \mathbf{w}_H \\ \mathbf{w}_S \end{bmatrix}, \quad M = \begin{bmatrix} M_{HH} & M_{HS} \\ M_{SH} & M_{SS} \end{bmatrix}. \quad (3.17)$$

Note that  $\mathbf{s}_H = \mathbf{0}$  because there is no slip in adhesion and  $\mathbf{p}_S$  is known because the traction has reached the traction bound in the slip area. This notation allows us to combine equations (3.6) and (3.8) to get:

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{s}_S \end{bmatrix} = \begin{bmatrix} \mathbf{w}_H \\ \mathbf{w}_S \end{bmatrix} + \begin{bmatrix} M_{HH} & M_{HS} \\ M_{SH} & M_{SS} \end{bmatrix} \begin{bmatrix} \mathbf{p}_H \\ \mathbf{p}_S \end{bmatrix}. \quad (3.18)$$

From this we can again find separate equations for  $\mathbf{p}_H$  and  $\mathbf{s}_S$ :

$$\mathbf{0} = \mathbf{w}_H + M_{HH}\mathbf{p}_H + M_{HS}\mathbf{p}_S, \quad (3.19)$$

which gives the result:

$$\mathbf{p}_H = -M_{HH}^{-1}(\mathbf{w}_H + M_{HS}\mathbf{p}_S). \quad (3.20)$$

The second equation gives:

$$\mathbf{s}_S = \mathbf{w}_S + M_{SH}\mathbf{p}_H + M_{SS}\mathbf{p}_S, \quad (3.21)$$

where we can use the solution of  $\mathbf{p}_H$  that we just found in the previous step and use this as a known value giving us the solution immediately.

In the *transient case* we keep the matrices  $A_{I_1J_1}/dq$  and  $A'_{I_1J_1}/dq$  as separate entities. Again we make the same subdivision into adhesion and slip areas. This notation allows us to combine equations (3.14) and (3.16) to get:

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{s}_S \end{bmatrix} = \begin{bmatrix} \mathbf{w}_H \\ \mathbf{w}_S \end{bmatrix} + \begin{bmatrix} A_{HH} & A_{HS} \\ A_{SH} & A_{SS} \end{bmatrix} \begin{bmatrix} \mathbf{p}_H \\ \mathbf{p}_S \end{bmatrix} - \begin{bmatrix} A'_{HH} & A'_{HS} \\ A'_{SH} & A'_{SS} \end{bmatrix} \begin{bmatrix} \mathbf{p}'_H \\ \mathbf{p}'_S \end{bmatrix}. \quad (3.22)$$

From this we can again find separate equations for  $\mathbf{p}_H$  and  $\mathbf{s}_S$ :

$$\mathbf{0} = \mathbf{w}_H + A_{HH}\mathbf{p}_H + A_{HS}\mathbf{p}_S - A'_{HH}\mathbf{p}'_H - A'_{HS}\mathbf{p}'_S, \quad (3.23)$$

which gives the result:

$$\mathbf{p}_H = -A_{HH}^{-1}(\mathbf{w}_H + A_{HS}\mathbf{p}_S - A'_{HH}\mathbf{p}'_H - A'_{HS}\mathbf{p}'_S). \quad (3.24)$$

The second equation gives:

$$\mathbf{s}_S = \mathbf{w}_S + A_{SH}\mathbf{p}_H + A_{SS}\mathbf{p}_S - A'_{SH}\mathbf{p}'_H - A'_{SS}\mathbf{p}'_S, \quad (3.25)$$

where we can use the solution of  $\mathbf{p}_H$  that we just found in the previous step and use this as a known value giving us the solution immediately.

## Smoothing

A different problem, apart from the wiggles, is the appearance of smoothing whenever  $c$  is not an integer. At first this is unexpected as the analytic influence coefficients should in theory result in a correct solution. To investigate this subject we will go back to equation (3.24). This shows that the newfound tractions  $\mathbf{p}$  in adhesion depend on four matrix-vector products. When we eliminate the brackets the equation becomes:

$$\mathbf{p}_H = -A_{HH}^{-1}\mathbf{w}_H - A_{HH}^{-1}A_{HS}\mathbf{p}_S + A_{HH}^{-1}A'_{HH}\mathbf{p}'_H + A_{HH}^{-1}A'_{HS}\mathbf{p}'_S. \quad (4.1)$$

For simplicity suppose that between two iterations the division between adhesion and slip elements does not change. Then  $\mathbf{p}_S = \mathbf{p}'_S$  because in the slip area the traction is at the traction bound. We can rewrite (4.1) to get:

$$\mathbf{p}_H = -A_{HH}^{-1}(\mathbf{w}_H + (A_{HS} - A'_{HS})\mathbf{p}'_S) + A_{HH}^{-1}A'_{HH}\mathbf{p}'_H. \quad (4.2)$$

Here the term  $A_{HH}^{-1}A'_{HH}\mathbf{p}'_H$  says how the old solution is deformed to a new position, the term  $-A_{HH}^{-1}(\mathbf{w}_H + (A_{HS} - A'_{HS})\mathbf{p}'_S)$  adds an extra contribution to the tractions that is caused by slipping of the system, note that this contribution is independent of the tractions in the adhesion area and is nothing more than a vertical shift. So let us take a look at what  $A_{HH}^{-1}A'_{HH}$  exactly does to  $\mathbf{p}'_H$ . This is easiest done by looking at what happens to a traction that has the value one at one point and is zero everywhere else. If we know what happens to such a traction we know what happens to a realistic traction as that is just a linear combination of such single point tractions.

When looking at the matrix  $A_{HH}^{-1}A'_{HH}$ , multiplying this with a vector with a one at the  $i^{th}$  position and zeros everywhere else results in the  $i^{th}$  column of the matrix. When using  $c = 1$  this matrix has, except for the first column, a one on the  $(i - 1)^{th}$  position and zeros everywhere else. For  $c = 0$  the matrix is just the identity matrix because when  $c = 0$  we have that  $A' = A$ . For  $0 < c < 1$  however the columns of  $A_{HH}^{-1}A'_{HH}$  have non-zero values in multiple rows, this means that the traction with a nonzero value at a single point in space becomes smeared out after the first transformation. Repetitive iterations only increase this effect.

Figure 4.1 shows how  $A_{HH}^{-1}A'_{HH}$  transforms a  $p'_H$  that is one at  $x_I = 0$  and zero everywhere else after one timestep. We see that in the case  $c = 1$  this transformation is nothing but a horizontal translation over one gridpoint. We would like this behaviour for all  $c$ , however for  $c < 1$  it is impossible to have this result as there are no gridpoints between 0 and  $-\Delta x$ . So we see a transformation that puts the weight of the new  $p_H$  in between 0 and  $-\Delta x$ . The columns of  $A_{HH}^{-1}A'_{HH}$  sum up to one for all  $c$  so the total traction does stay constant in all cases.

Figure 4.2 shows how  $p'_H$  is transformed after a traversed distance of  $50\Delta x$ . From this we can see that over time the information that should have moved from 0 to  $-50\Delta x$  has instead spread over a wide area. And when  $c$  becomes small (the green line, for  $c = 0.05$ ) it might even add to the appearance or amplification of wiggles.

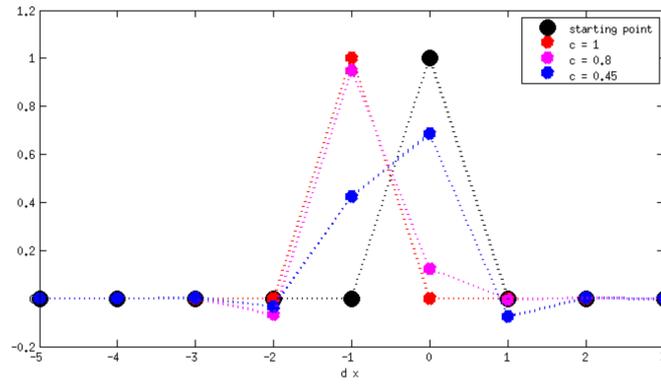


Figure 4.1: Solution of the traction  $p_x$  using different values for  $c$  after one iteration.

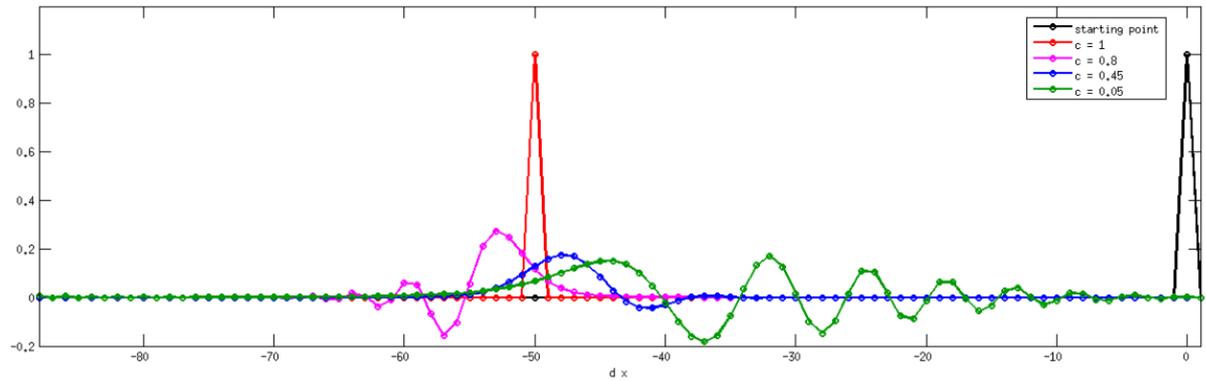


Figure 4.2: Solution of the traction  $p_x$  using different values for  $c$  after a traversed distance of  $50\Delta x$ .

Results of this smoothing can be seen in Figure 4.3, where the 2D Cattaneo to Carter [28, example 5.8] testcase has been used in order to see the effect of smoothing in a realistic situation. The behaviour of the smoothing looks very similar to the graphs in Figure 4.2. In case  $1 > c \geq \frac{1}{2}$  smoothing will slightly overestimate the forward motion of extrema (to the left, downstream) while when  $\frac{1}{2} > c > 0$  smoothing slightly underestimates the forward motion (to the right, upstream).

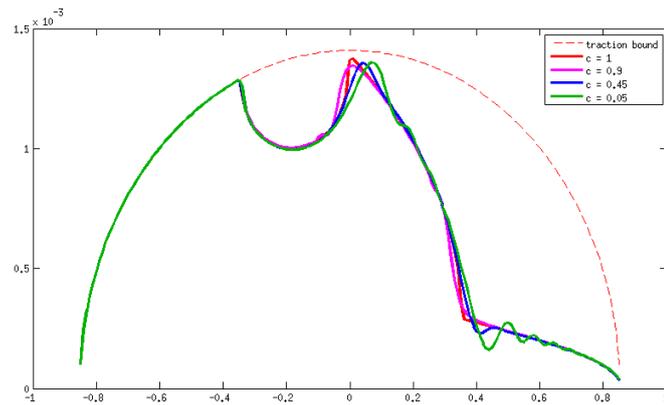


Figure 4.3: Solution of the traction  $p_x$  using different values for  $c$  after a traversed distance of  $50\Delta x$ .

## Problem statement and course of action

Problems arise when the factor  $c = \frac{dq}{dx}$  becomes small. Physically this means that the traversed distance per timestep is small compared to the gridsize. Mathematically this means that the discretized influence functions  $A(\mathbf{x}) - A'(\mathbf{x} + \mathbf{V} \cdot dt)$  have increasingly smaller eigenvalues. In [22] Vollebregt remarks that the coefficients in steady-state depend on the time-step, and below a certain point there will be either slower convergence or no convergence at all, *“the restriction on the time step is sometimes quite severe, so that physically attractive values cannot be used”*.

A simple solution to this is never choosing a too small factor  $c$ . This is a nice solution when only running the program CONTACT by itself, however CONTACT has been integrated in larger train simulation packages, in such cases the input parameters by CONTACT are no longer handpicked but fed to the software by the overall package. Ideally it would be best if CONTACT can be adjusted so it converges properly for any value of  $c$ . Finding out why the wiggles occur can also help solve this problem, because in that case CONTACT can check it’s input parameters and determine beforehand whether the parameters will result in a converging solution and give feedback (and adjust the critical parameters) when this is not the case.

Next to this problem a second phenomenon was observed. The solution of the tractions appears to suffer from a form of numerical diffusion that smooths out any sharp extrema. There seemed to be two different causes for this smoothing (the numerical diffusion and an effect of under-relaxation). It is shown that both are actually equivalent. This smoothing needs to be fixed to get detailed traction profiles in a transient system.

### Work done in this research

A possible cause for the appearance of wiggles is that in the numerical scheme a piecewise constant approximation is used. When taking the difference between two of such approximations, for example when approximating a derivative or determining the influence coefficients  $A - A'$ , the middle part completely drops out and only the edges are left. When instead a piecewise linear approximation is used there will be a net result over the full interval, see Figure 5.1. Moving to linear approximations not only gives a ‘smoother’ solution but might also increase accuracy or rate of convergence.

Although this might be a possible solution first it is important to determine what exactly causes the wiggles to appear. From what values of the factor  $c$  do the wiggles appear and why?

### 2D steady-state rolling

Because the problems arise in both the 2D and the 3D case the work in this thesis will focus on 2D rolling. The tangential slip-equation is much easier to solve in this case so it will be easier to focus on

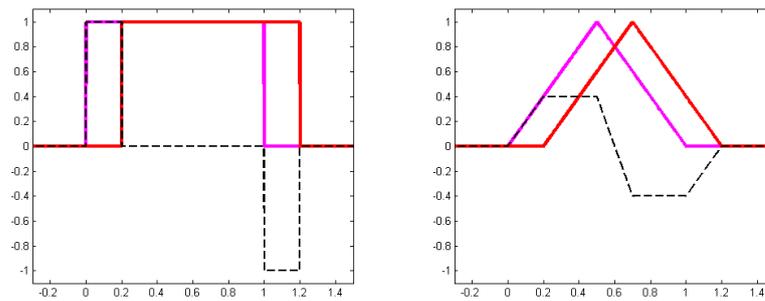


Figure 5.1: The magenta graph is the constant (left) or linear (right) basis function on  $(0, 1)$ . The red graph is the same basis function shifted slightly to the right, the black graph is the difference between them.

the real problem. In steady-state we do not have the sharp extrema that are subject to smoothing so only the wiggles will be investigated. Therefore the first course of action will be:

- solving the wiggle-problem on 2D steady-state rolling.

## 2D transient rolling

When using the direct approach to determine the steady-state solution the matrix-operator used is given by  $A_{IiJj} - A'_{IiJj}$ . We can calculate this matrix and use this to try and see what might cause the wiggles. The matrix-operator used in the transient situation seems to be just the matrix  $A_{IiJj}$ . But also in this situation wiggles are detected. These wiggles do not appear in the initial states of the solution but progress over time. This has to do with the fact that the solution at a later time is the result of repeated application of these matrix-operators. What makes this problem harder is that the problem is not caused by a single matrix operation but by an accumulation of operations. It would be convenient if a solution to the steady-state problem also solves the transient problem, this however has to be checked and possibly an extra solution has to be found. Just like for steady-state, also in the transient case we want a solution to hold when the traction bound is added to the equation. Additionally, in transient rolling we have to find a solution for the smoothing problem so this testcase is suitable for multiple problems:

- solving the wiggle-problem on 2D transient rolling.
- solving the smoothing-problem on 2D transient rolling.

## Rate of convergence

Additionally it was observed that an unexpectedly slow rate of convergence is achieved by the system. This will be investigated, with hopes that solving (one of) the other problems will also increase the rate of convergence of the system. Also refinements to the way the leading edge is treated are made to improve the rate of convergence. So the last part of the thesis is devoted to:

- improving the rate of convergence.

## Research Questions

As a final closure I will here state the research questions for my master thesis.

1. How can we solve the problem of wiggles in the solution?
  - What causes the wiggles that arise when the factor  $c = \frac{dq}{dx}$  becomes small?
  - Does replacing the piecewise constant basis functions by piecewise (bi)linear basis functions solve this problem?

- 
- Are there other ways to prevent the appearance of unphysical wiggles?
  2. How can we solve the problem of smoothing of the tractions?
    - What causes the smoothing effect?
    - How can we prevent this numerical smoothing to influence the tractions?
  3. How do any of the adjustments made influence the rate of convergence of the algorithm?
    - Are there other ways to improve the rate of convergence?



## Using higher order basis functions

In the original situation we determine the coefficient matrix while using piecewise-constant basis functions. We use an upwind discretization in equation (2.10). For appropriate  $c$  this is a robust method. Although central difference seemed like a good alternative this appears not to be the case. This is caused by the way the influence matrices are constructed.

When we take  $c = 1$  the matrix  $M_{IJ}^* = A_{IJ} - A'_{IJ}$  is a Toeplitz matrix. When assuming a 2D quasi-identical setting we are only concerned with the  $A_{IJ}$ , we will for brevity write  $M_{IJ} = M_{IJ}^*$ . With the diagonal elements  $M_{II}$  having the highest value, upper diagonal elements have positive value but their magnitude rapidly drops as they are further away from the diagonal. The elements  $M_{I+1I}$  have the lowest values, lower diagonal elements all have negative value but the magnitude again rapidly drops when we move away from the diagonal. See Figure 6.1. On the left we see the matrix  $M = A - A'$  for  $c = 1$  while on the right the inverse is plotted. Note that the main-diagonal holds the highest values.

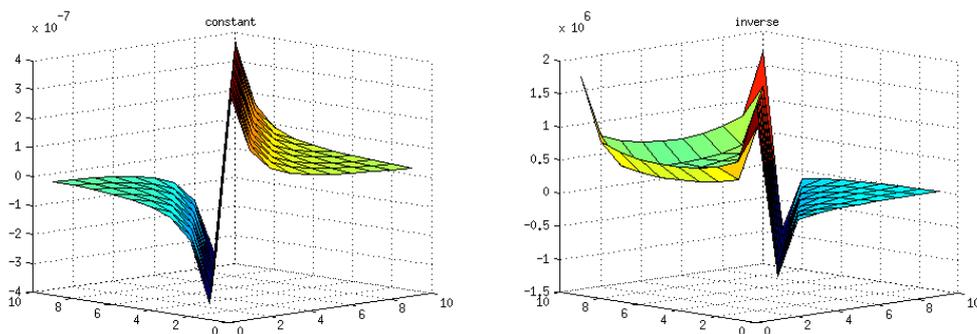


Figure 6.1: Matrix  $A - A'$  and its inverse for  $c = 1$ . The  $x$  and  $y$  axis represent the rows and columns of the matrix.

When we apply central differences the matrix  $M$  becomes  $M_{IJ} = A'_{J1I1} - A'_{I1J1}$ . This again is a Toeplitz matrix. However, now the highest value is attained by the elements  $M_{II+1}$  with the lowest values still at elements  $M_{I+1I}$ . Again the magnitude of elements away from the diagonal rapidly drops. For the diagonal elements however we have  $M_{II} = 0$ . Initially this does not seem to be that bad, nonetheless when we look at the inverse of this  $M$  we see that the off-diagonals of  $M^{-1}$  oscillate between positive and negative values.

As we go back to upwind again but let  $c$  slowly decrease we notice that the value of the elements  $M_{II+1}$  starts to rise while the value of the  $M_{II}$  goes to zero. For very small values of  $c$  the matrix attained from the upwind method is indistinguishable from the matrix attained from the central difference. This can be seen in Figure 6.2. Note that the main-diagonal has now become almost zero and the highest values are moved to the diagonal one above the main diagonal.

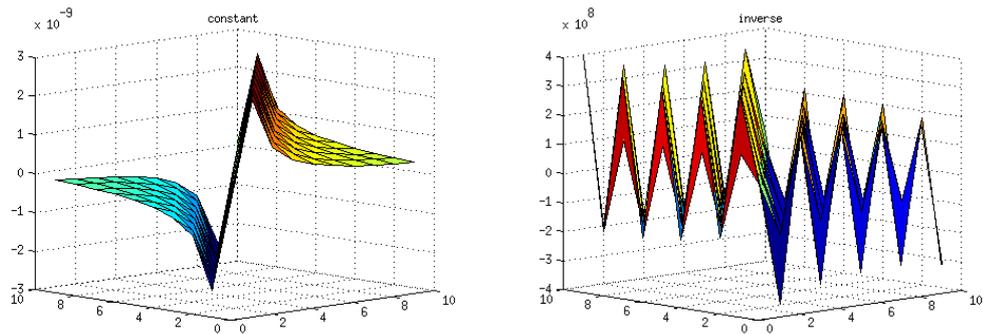


Figure 6.2: Matrix  $A - A'$  and its inverse for  $c = 0.01$ .

## Explicit expressions for the piecewise-constant case

To have a better understanding of what happens to the influence coefficients we will look at the analytical expression when using constant basisfunctions. Our main interest lies in what happens to the elements  $M_{II}$ ,  $M_{I+1I}$ , and  $M_{II+1}$ . Because  $M$  is a Toeplitz matrix it does not matter which  $i$  we take, so we will look at the expressions for  $M_{11}$ ,  $M_{21}$ , and  $M_{12}$ . Let  $\mathbf{x}_1 = (0, 0)^T$ ,  $\mathbf{x}_2 = (\Delta x, 0)^T$ , and  $0 < dq < dx$ . The analytic deduction can be found in Appendix A, this will primarily be bookkeeping applied to the solutions given in Kalker [14].

In Figure 6.3 we can see how the values of the main- and off-diagonals are functions of  $c$ . From these plots it becomes clear that although all values of the matrix go to zero as  $c$  goes to zero there is a clear difference in how fast they do this. At  $c = 1$  we have  $M_{12} = -M_{11}$  while  $M_{21}$  is more than half as small. However around  $c = 0.3939$  we find that  $M_{11} = M_{21}$ . For lower values of  $c$   $M_{21}$  gradually takes on the same value as  $-M_{12}$  while the significance of  $M_{11}$  falls off completely. Resulting in a case similar as using a central difference scheme.

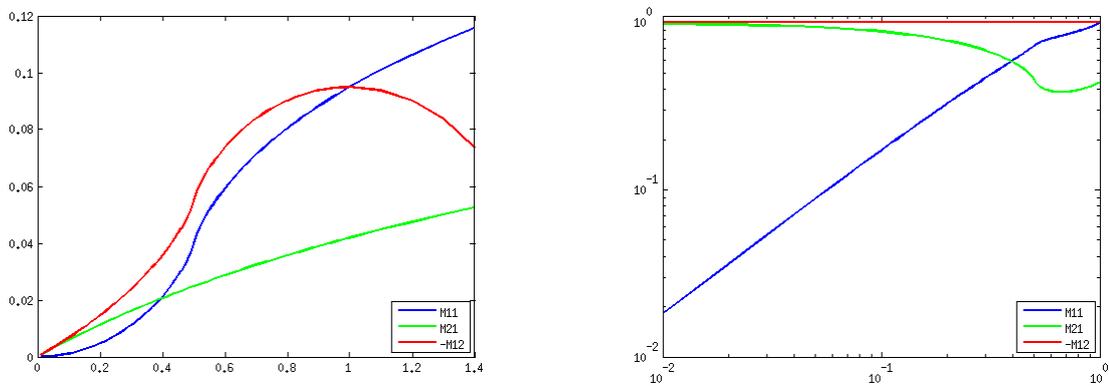


Figure 6.3: The left plot shows the values of the main- and off-diagonals, the right plot shows the same values but scaled so it becomes clear which values matter and which become unimportant. The horizontal axis is the value of  $c$ . Note that  $M_{12}$  is negative.

## Determining the influence coefficients using linear basisfunctions

When determining the coefficient matrix based on piecewise-linear basis functions a restrained approach will be followed. We will not immediately calculate all the different coefficient matrices  $A_{I_i J_j}$  using a bi-linear form. Instead the simplest case is used and results from that case will be examined to see if there was any gain.

For the 2D quasi-identical problem we are only interested in the relationship between  $u_1$  and  $p_1$ , so the

influence functions  $A_{I1J1}$ . For now we assume we are only interested in the displacements at the surface, so we set  $z = 0$ .

Following Vollebregt's approach we get [24, equation 36a]:

$$u_x(x, y, 0) = \frac{P_x}{\pi G} \left( \frac{1 - \nu}{\rho} + \nu \frac{(x - \xi)^2}{\rho^3} \right). \quad (6.1)$$

This is the displacement we get at  $(x, y, 0)$  from a point stress  $P_x$  at  $(\xi, \eta)$ . If we now want to know the displacement we get at  $(x, y, 0)$  as a result of the whole contact area  $C$  we need to sum over all point pressures  $p_x(\xi, \eta)$  times the surface  $\delta\xi \cdot \delta\eta$  of each point. Summing over these infinitesimal points results in the integral:

$$u_x(x, y, 0) = \frac{1}{\pi G} \iint_C \left( \frac{1 - \nu}{\rho} + \nu \frac{(x - \xi)^2}{\rho^3} \right) p_x(\xi, \eta) d\xi d\eta. \quad (6.2)$$

This lets us define the influence function:

$$A_{ux}(x, y, 0, \xi, \eta) = \frac{1}{\pi G} \left( \frac{1 - \nu}{\rho} + \nu \frac{(x - \xi)^2}{\rho^3} \right). \quad (6.3)$$

Now introduce the basisfunction on the origin:

$$\varphi(x', y') = \begin{cases} \frac{x}{\Delta x} + 1 & \text{when } -\Delta x < x' \leq 0 \wedge |y| < \frac{\Delta y}{2} \\ \frac{-x}{\Delta x} + 1 & \text{when } 0 < x' < \Delta x \wedge |y| < \frac{\Delta y}{2} \\ 0 & \text{when } |x'| \geq \Delta x \vee |y| \geq \frac{\Delta y}{2} \end{cases}. \quad (6.4)$$

We can use this to define a basisfunction for each element as:

$$\varphi_I(x, y) = \varphi(x - x_I, y - y_I). \quad (6.5)$$

This basisfunction  $\varphi_I$  is now only nonzero when  $|x - x_I| < \Delta x$ . We can approximate the tractionfield with a continuous, piece-wise linear,  $p_x$  given by:

$$p_x(x, y) = \sum_I p_{Ix} \varphi_I(x, y). \quad (6.6)$$

The  $p_{Ix}$  are the coefficients to be determined. We can substitute the discretized traction (B.7) into (6.2) and find:

$$u_x(x, y, 0) = \iint_C A_{ux}(x, y, 0, \xi, \eta) \sum_J \{p_{Jx} \varphi_J(\xi, \eta)\} d\xi d\eta. \quad (6.7)$$

The order of integration and summation may be reversed, and the constants  $p_{Jx}$  can be taken out of the integral:

$$u_x(x, y, 0) = \sum_J \left\{ p_{Jx} \iint_C A_{ux}(x, y, 0, \xi, \eta) \varphi_J(\xi, \eta) d\xi d\eta \right\}. \quad (6.8)$$

We now define the influence coefficients  $A_{IuJx}$  as the integrals in the last equation. So we have:

$$u_x(x_I, y_I, 0) = \sum_J A_{IuJx} p_{Jx}, \quad (6.9)$$

with

$$A_{IuJx} = \iint_C A_{ux}(x_I, y_I, 0, \xi, \eta) \varphi_J(\xi, \eta) d\xi d\eta. \quad (6.10)$$

Use relation (B.6) and only integrate over the nonzero parts to get:

$$A_{IuJx} = \iint_{\mathcal{I}(\varphi_J)} A_{ux}(x_I, y_I, 0, \xi, \eta) \varphi_J(\xi, \eta) d\xi d\eta. \quad (6.11)$$

Here  $\mathcal{I}(\varphi_J)$  is the area where  $\varphi_J$  is nonzero, so the area  $(x_J - \Delta x, x_J + \Delta x) \times (y_J - \frac{\Delta y}{2}, y_J + \frac{\Delta y}{2})$ . Apply a transformation of variables, going from  $x_I, y_I, \xi, \eta$  to:

$$x'_I = x_I - x_J, \quad y'_I = y_I - y_J, \quad \xi' = \xi - x_J, \quad \eta' = \eta - y_J. \quad (6.12)$$

This translates the area of integration so that the origin lies at the centre. Using equation (B.3) we know that  $A_{ux}$  is only a function of  $(x_I - \xi)$  and  $(y_I - \eta)$ , which is equal to  $(x'_I - \xi')$  and  $(y'_I - \eta')$ , so  $A_{ux}(x_I, y_J, 0, \xi, \eta) = A_{ux}(x'_I, y'_I, 0, \xi', \eta')$ . From (B.6) we have that  $\varphi_J(\xi, \eta) = \varphi(\xi', \eta')$ . So we can write the equation as:

$$A_{IuJx} = \iint_{\mathcal{I}(\varphi)} A_{ux}(x'_I, y'_I, 0, \xi', \eta') \varphi(\xi', \eta') \delta\xi' \delta\eta'. \quad (6.13)$$

From the definition given in (6.4) we have different expressions for  $\varphi(\xi, \eta)$  when  $\xi \leq 0$  and  $\xi > 0$ . So we split the integral into two parts and for convenience drop the primes on  $\xi$  and  $\eta$ .

$$\begin{aligned} A_{IuJx} &= \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_{-\Delta x}^0 A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) \left( \frac{\xi}{\Delta x} + 1 \right) d\xi d\eta \\ &\quad + \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_0^{\Delta x} A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) \left( \frac{-\xi}{\Delta x} + 1 \right) d\xi d\eta. \end{aligned} \quad (6.14)$$

The next challenge is finding explicit expressions for these integrals. Integrating the  $A_{ux} \cdot 1$  term has been done already in [14] as this is required for the piece-wise constant approximation. The difficulty when finding an antiderivative for  $A_{ux} \cdot \xi$  is that  $A_{ux}$  contains terms  $(x_I - x_J - \xi)$ , which are then multiplied by plain  $\xi$ 's. Therefore note that:

$$\xi = (x_I - x_J) - (x_I - x_J) + \xi = (x_I - x_J) - (x_I - x_J - \xi). \quad (6.15)$$

In the integral the term  $(x_I - x_J)$  is just a constant so we can rewrite  $\frac{\xi}{\Delta x} + 1$  and  $\frac{-\xi}{\Delta x} + 1$  into terms with  $(x_I - x_J - \xi)$  and a constant term as:

$$\frac{\xi}{\Delta x} + 1 = \frac{(x_I - x_J) - (x_I - x_J - \xi)}{\Delta x} + 1 = -\frac{(x_I - x_J - \xi)}{\Delta x} + \left( \frac{x_I - x_J}{\Delta x} + 1 \right), \quad (6.16)$$

$$\frac{-\xi}{\Delta x} + 1 = \frac{(x_I - x_J - \xi) - (x_I - x_J)}{\Delta x} + 1 = \frac{(x_I - x_J - \xi)}{\Delta x} - \left( \frac{x_I - x_J}{\Delta x} - 1 \right). \quad (6.17)$$

Recall from (B.3) that we have to find explicit expressions for:

$$\iint \frac{x_I - x_J - \xi}{\rho} d\xi d\eta, \quad (6.18)$$

$$\iint \frac{(x_I - x_J - \xi)^3}{\rho^3} d\xi d\eta, \quad (6.19)$$

where  $\rho = \sqrt{(x_I - x_J - \xi)^2 + (y_I - y_J - \eta)^2}$ .

### Integration using Maxima

The integration is performed using the software package `Maxima`, a continuation of the package `Macsyma`. This gives the following antiderivatives when setting the integration constant to zero:

$$\begin{aligned} F_0(x, y, \xi, \eta) &:= \iint \frac{1}{\sqrt{(x-\xi)^2 + (y-\eta)^2}} d\xi d\eta & (6.20) \\ &= (x-\xi) \sinh^{-1} \left( \frac{(y-\eta)}{|(x-\xi)|} \right) + (y-\eta) \sinh^{-1} \left( \frac{(x-\xi)}{|(y-\eta)|} \right), \end{aligned}$$

$$\begin{aligned} F_1(x, y, \xi, \eta) &:= \iint \frac{(x-\xi)}{\sqrt{(x-\xi)^2 + (y-\eta)^2}} d\xi d\eta & (6.21) \\ &= \frac{1}{2} \left[ (x-\xi)^2 \sinh^{-1} \left( \frac{(y-\eta)}{|(x-\xi)|} \right) + (y-\eta) \sqrt{(x-\xi)^2 + (y-\eta)^2} \right], \end{aligned}$$

$$\begin{aligned} F_2(x, y, \xi, \eta) &:= \iint \frac{(x-\xi)^2}{\sqrt{(x-\xi)^2 + (y-\eta)^2}^3} d\xi d\eta & (6.22) \\ &= (y-\eta) \sinh^{-1} \left( \frac{(x-\xi)}{|(y-\eta)|} \right), \end{aligned}$$

$$\begin{aligned} F_3(x, y, \xi, \eta) &:= \iint \frac{(x-\xi)^3}{\sqrt{(x-\xi)^2 + (y-\eta)^2}^3} d\xi d\eta & (6.23) \\ &= (y-\eta) \sqrt{(x-\xi)^2 + (y-\eta)^2}. \end{aligned}$$

Combining all results again into (6.14) gives:

$$\begin{aligned} A_{IuJx} &= \left( \frac{x_I - x_J}{\Delta x} + 1 \right) \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_{-\Delta x}^0 A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) d\xi d\eta & (6.24) \\ &\quad - \frac{1}{\Delta x} \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_{-\Delta x}^0 (x_I - x_J - \xi) A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) d\xi d\eta \\ &\quad - \left( \frac{x_I - x_J}{\Delta x} - 1 \right) \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_0^{\Delta x} A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) d\xi d\eta \\ &\quad + \frac{1}{\Delta x} \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_0^{\Delta x} (x_I - x_J - \xi) A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) d\xi d\eta, \end{aligned}$$

with:

$$A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) = \frac{1}{\pi G} \left( \frac{1-\nu}{\rho} + \nu \frac{(x_I - x_J - \xi)^2}{\rho^3} \right). \quad (6.25)$$

In order to shorten the notation a bit we write (for  $n = 0..3$ ):

$$\begin{aligned} \mathcal{F}^-(F_n, x, y) &= F_n \left( x, y, 0, \frac{\Delta y}{2} \right) - F_n \left( x, y, -\Delta x, \frac{\Delta y}{2} \right) & (6.26) \\ &\quad - F_n \left( x, y, 0, -\frac{\Delta y}{2} \right) + F_n \left( x, y, -\Delta x, -\frac{\Delta y}{2} \right), \end{aligned}$$

$$\begin{aligned} \mathcal{F}^+(F_n, x, y) &= F_n \left( x, y, \Delta x, \frac{\Delta y}{2} \right) - F_n \left( x, y, 0, \frac{\Delta y}{2} \right) & (6.27) \\ &\quad - F_n \left( x, y, \Delta x, -\frac{\Delta y}{2} \right) + F_n \left( x, y, 0, -\frac{\Delta y}{2} \right). \end{aligned}$$

Now the explicit expression for  $A_{IuJx}$  becomes:

$$\begin{aligned} \pi G A_{IuJx} = & \left( \frac{x_I - x_J}{\Delta x} + 1 \right) \left( (1 - \nu) \mathcal{F}^-(F_0, x_I - x_J, y_I - y_J) + \nu \mathcal{F}^-(F_2, x_I - x_J, y_I - y_J) \right) \quad (6.28) \\ & - \frac{1}{\Delta x} \left( (1 - \nu) \mathcal{F}^-(F_1, x_I - x_J, y_I - y_J) + \nu \mathcal{F}^-(F_3, x_I - x_J, y_I - y_J) \right) \\ & - \left( \frac{x_I - x_J}{\Delta x} - 1 \right) \left( (1 - \nu) \mathcal{F}^+(F_0, x_I - x_J, y_I - y_J) + \nu \mathcal{F}^+(F_2, x_I - x_J, y_I - y_J) \right) \\ & + \frac{1}{\Delta x} \left( (1 - \nu) \mathcal{F}^+(F_1, x_I - x_J, y_I - y_J) + \nu \mathcal{F}^+(F_3, x_I - x_J, y_I - y_J) \right), \end{aligned}$$

Rearranging terms gives us the final result (dropping the terms  $x_I - x_J, y_I - y_J$  in the notation to keep some brevity):

$$\begin{aligned} \pi G A_{IuJx} = & \left( \frac{x_I - x_J}{\Delta x} \right) \left( (1 - \nu) (\mathcal{F}^-(F_0) - \mathcal{F}^+(F_0)) + \nu (\mathcal{F}^-(F_2) - \mathcal{F}^+(F_2)) \right) \quad (6.29) \\ & + \left( (1 - \nu) (\mathcal{F}^-(F_0) + \mathcal{F}^+(F_0)) + \nu (\mathcal{F}^-(F_2) + \mathcal{F}^+(F_2)) \right) \\ & - \frac{1}{\Delta x} \left( (1 - \nu) (\mathcal{F}^-(F_1) - \mathcal{F}^+(F_1)) + \nu (\mathcal{F}^-(F_3) - \mathcal{F}^+(F_3)) \right), \end{aligned}$$

Note that:

$$\begin{aligned} \mathcal{F}^-(F_n) + \mathcal{F}^+(F_n) = & F_n \left( \Delta x, \frac{\Delta y}{2} \right) - F_n \left( -\Delta x, \frac{\Delta y}{2} \right) \quad (6.30) \\ & - F_n \left( \Delta x, -\frac{\Delta y}{2} \right) + F_n \left( -\Delta x, -\frac{\Delta y}{2} \right), \end{aligned}$$

$$\begin{aligned} \mathcal{F}^-(F_n) - \mathcal{F}^+(F_n) = & -F_n \left( -\Delta x, \frac{\Delta y}{2} \right) + 2F_n \left( 0, \frac{\Delta y}{2} \right) - F_n \left( \Delta x, \frac{\Delta y}{2} \right) \quad (6.31) \\ & + F_n \left( -\Delta x, -\frac{\Delta y}{2} \right) - 2F_n \left( 0, -\frac{\Delta y}{2} \right) + F_n \left( \Delta x, -\frac{\Delta y}{2} \right). \end{aligned}$$

Using these results we can compare the coefficient matrix found when using constant or linear functions. Figure 6.4 shows on the left the matrix  $M = A - A'$  for  $c = 1$  while on the right the inverse is plotted. This result is very similar to the case using constant basis functions as found in Figure 6.1.

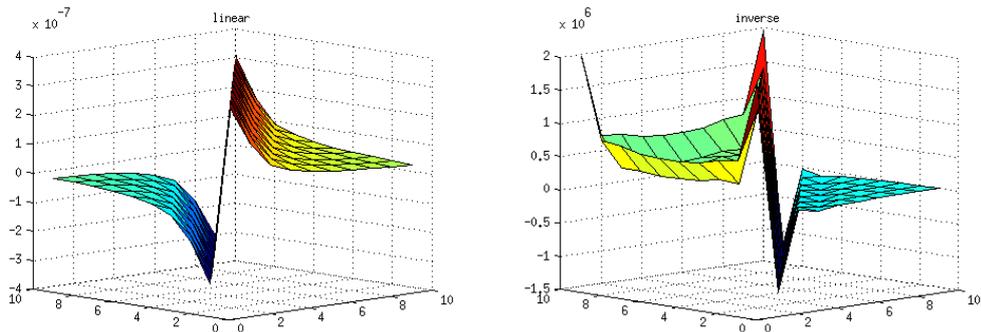


Figure 6.4: Matrix  $A - A'$  and its inverse for  $c = 1$  using piecewise-linear basis functions.

However, when taking small  $c$  we see similar effects as before. In Figure 6.5 the case for  $c = 0.01$  is shown. Comparing this with the result in Figure 6.2 shows that the wiggles are almost the same.

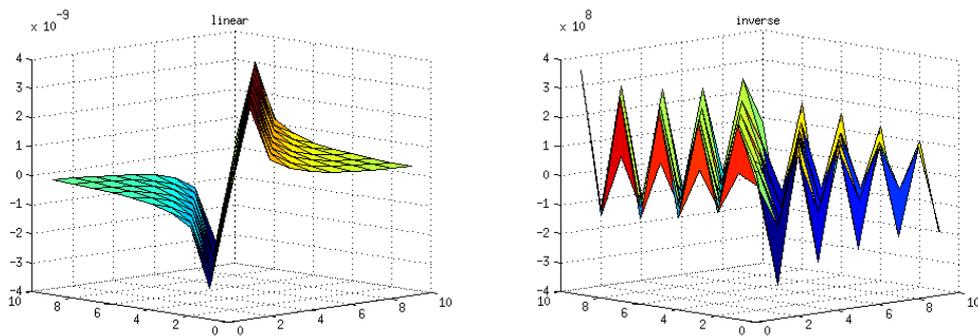


Figure 6.5: Matrix  $A - A'$  and its inverse for  $c = 0.01$  using piecewise-linear basis functions.

Now we can perform the same analysis as in Figure 6.3, which is shown in Figure 6.6. From these results we can conclude that in the case using linear basis functions the insignificance of the main diagonal for small  $c$  is still present. The problem has been slightly reduced but this is not enough by far.

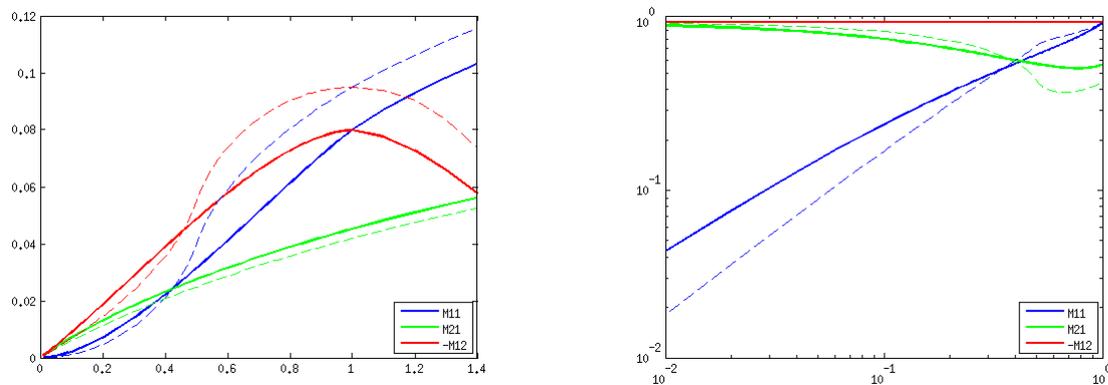


Figure 6.6: Left: values of  $M_{IJ}$  on a linear scale. Right: values of  $M_{IJ}$  scaled so the highest of the three values is exactly 1, on a logarithmic scale. The drawn lines show the matrix entry values when using basis functions linear in  $x$ , the dashed lines show the old values when using constant basis functions.

## Fourier analysis

A way to analyse the results is by use of Fourier Transformations. Using results from Vollebregts paper [25] we have a way to approximate the inverse of a Toeplitz matrix using a Fast Fourier Transformation (FFT). Here we will not use this with the intent of finding the inverse matrices but to gain insight in the behaviour of the inverse matrix.

Use the following approach:

1. Take the column of the original sufficiently large (Toeplitz) matrix.
2. Remove all elements above the diagonal and place them at the end of the resulting vector, which would be the same as a cyclic extension and placing the diagonal at the origin.
3. Apply a FFT on the transformed column.
4. Take the pointwise multiplicative inverse of all the Fourier coefficients.
5. Apply an inverse FFT on the resultant Fourier coefficients.
6. Reverse the transformation done in the second step to get a column of a (Toeplitz) matrix again.

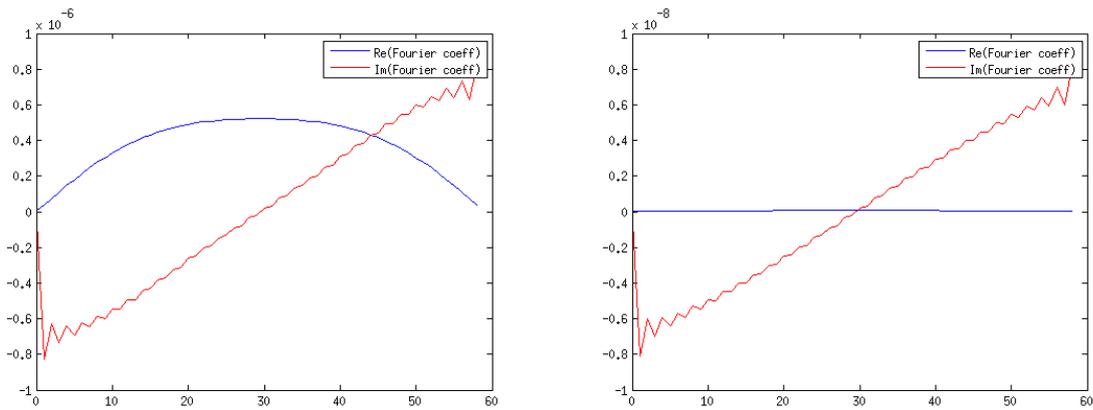
The matrix column we get is what a column of the inverse would look like if we ignore boundary effects.

7. Use the single column to extend this to a Toeplitz matrix.

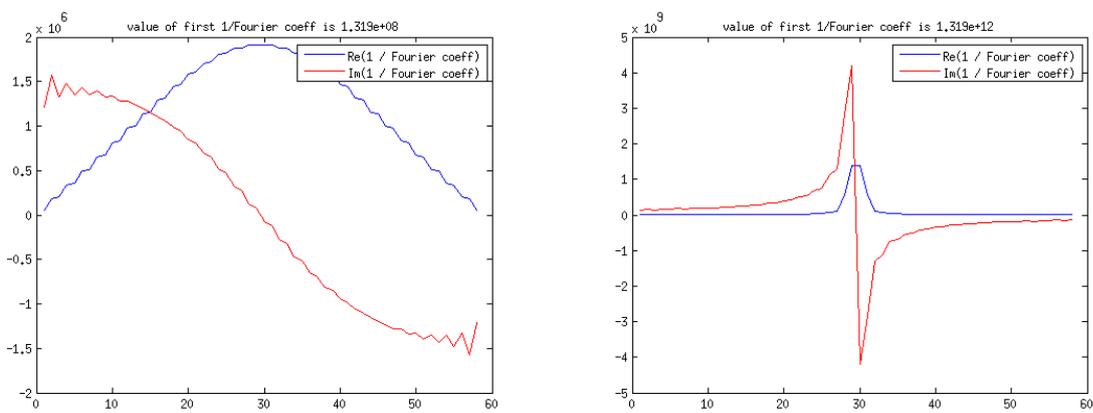
This results in an approximate inverse of a Toeplitz matrix that has some minor errors arising at the boundary. See Figure 6.7 for two applications of this method.

The absolute value, or amplitude, of the Fourier coefficients shows how important certain frequencies are in the columns. We see that in case  $c \approx 1$  all amplitudes are important and the inverse looks smooth. In case  $c \downarrow 0$  the high frequencies (Fourier coefficients around the 30th position) become very unimportant. This means that when we take the multiplicative inverse of the Fourier coefficients to find the Fourier coefficients for the approximate inverse these high frequencies become very important. This behaviour is also seen in the resulting inverse matrix, where the high frequencies dominate the behaviour of the system. So the absence of high frequencies in the Fourier domain of the influence coefficients causes the high frequency dominance in the inverse matrices.

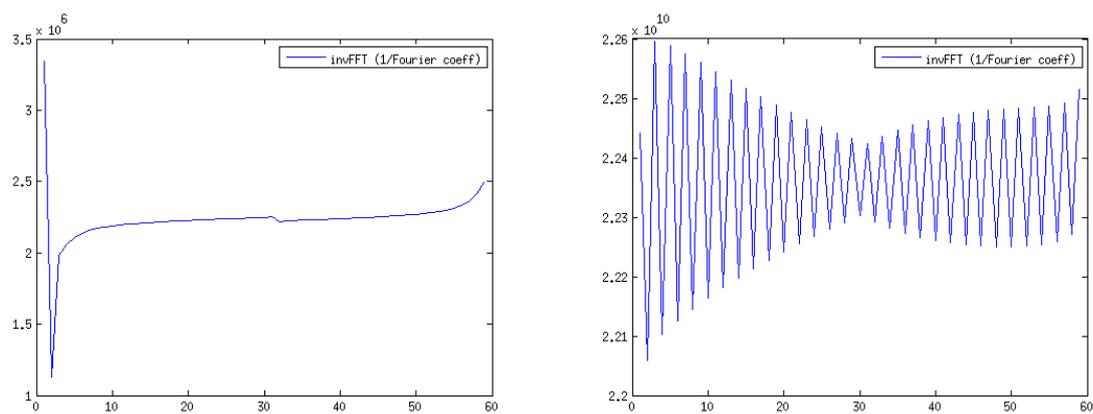
When abandoning the piecewise constant approximation for a piecewise linear approximation this problem is not solved because the Fourier Coefficients look very similar and the high frequency dominance for the inverse is still present, see Figure 6.8.



(a) Fourier coefficients.

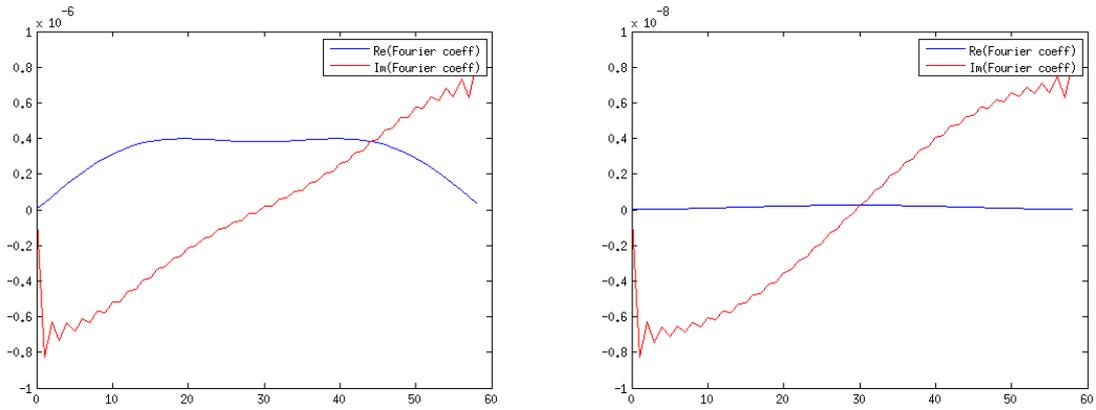


(b) One divided by Fourier coefficients.

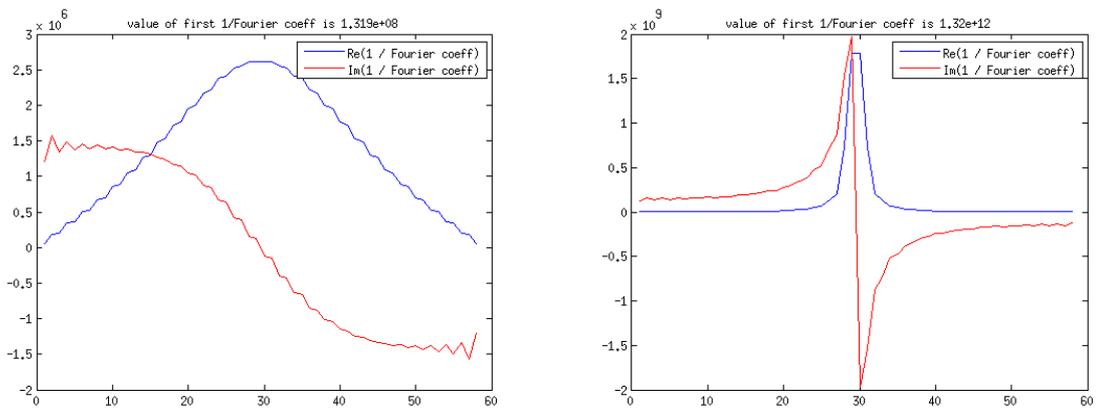


(c) Inverse Fourier transformation.

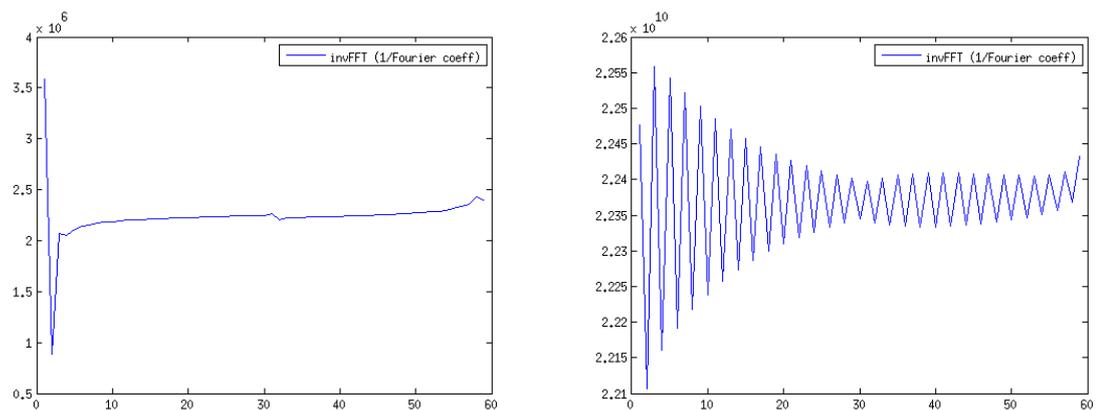
Figure 6.7: Results from steps 3, 4 and 5 in the Fourier analysis process, left using  $c = 1$ , right using  $c = 0.01$ . Using piecewise-constant basis functions.



(a) Fourier coefficients.



(b) One divided by Fourier coefficients.



(c) Inverse Fourier transformation.

Figure 6.8: Results from steps 3, 4 and 5 in the Fourier analysis process, left using  $c = 1$ , right using  $c = 0.01$ . Using piecewise-linear basis functions.

## Piecewise Quadratic Basisfunctions

When determining the coefficient matrix based on piecewise-quadratic basis functions a restrained approach will be followed. We will not immediately calculate all the different coefficient matrices  $A_{I_i J_j}$  using a bi-quadratic form. Instead the simplest case is used and results from that case will be examined to see if there was any gain.

For the 2D quasi-identical problem we are only interested in the relationship between  $u_1$  and  $p_1$ , so the influence functions  $A_{I_1 J_1}$ . For now we assume we are only interested in the displacements at the surface, so we set  $z = 0$ .

The math, going from the continuous influence functions to discrete influence coefficients based on quadratic basis functions will be very similar to the steps taken in Section 6.2. This can be found in Appendix B. We will now move on to results.

### Rate of Convergence

Now that we have multiple basis functions it is interesting to see if using one of them results in a faster rate of convergence than the others.

Before we talk about the rate of convergence, we first have to determine what quantity we are looking at to converge. One quantity to look at would be the total traction over the contact surface. Another would be the traction at a certain position in the contact surface. The advantage of the first option is that it uses all the relevant information. The downside however is that when calculating the total traction we have to be careful about how we do this. As the total traction is just the traction integrated over the contact surface we could end up determining the rate of convergence of our Riemann integral. The advantage of the second option is that you are looking at the pointwise convergence and are not influenced by other convergence mechanisms.

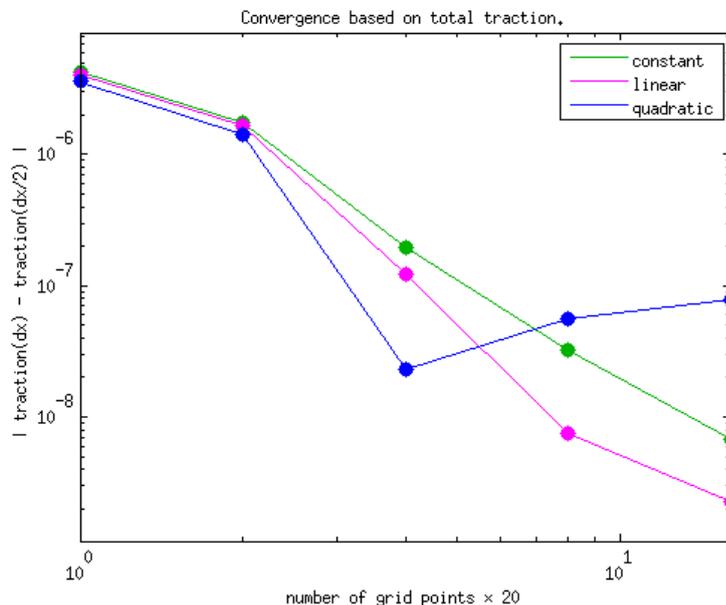


Figure 6.9: Convergence of the total traction for increasingly fine grids.

Figure 6.9 suggests that constant and linear basis functions perform equally well (linear basis functions having a slightly lower error but no faster convergence). The quadratic basis functions however perform not too well. At first convergence looks fine, but after some refinement of the grid no more improvement is noticed. These results reject the use of quadratic basis functions as they actually make the convergence worse.



## Time-stepping

We have concluded that even when using higher order approximations for the traction it is unfavourable to directly calculate the influence coefficients when using a small value of  $c$ . This brings us to the next approach; under-relaxation. Assume that the influence coefficients behave as a linear function of  $c$  around the origin and write:

$$A'(c) \approx \left(1 - \frac{c}{c^*}\right) A'(0) + \frac{c}{c^*} A'(c^*). \quad (7.1)$$

Here  $A'(\xi)$  is the influence coefficient matrix  $A$  calculated as in (2.15) with the rectangle of which we integrate is shifted a distance  $\xi$  to the right. For clarity, note that  $A'(0)$  is what we previously just called  $A$ . Instead of directly calculating the influence coefficients  $A'(c)$  for the needed  $c$  we can now calculate the influence coefficients  $A'(c^*)$  for a sufficient large  $c^*$  and approximate  $A'(c)$  using this alternative  $A'(c^*)$ .

### Under-relaxation of $A'$

Now call the ratio of  $c$  over  $c^*$  the relaxation parameter  $\theta = \frac{c}{c^*}$ . We will write for the coefficient matrix calculated with the larger timestep using  $c^*$  the matrix  $A^*$ . The new matrix  $A'$ , calculated using the relaxation parameter  $\theta$  and the matrix  $A$  and  $A^*$  now becomes  $A' = (1 - \theta)A + \theta A^*$ .

The problems arose when we started working with the inverse of  $A - A'$  for a small value of  $c$ . Now however  $A - A'$  becomes  $A - ((1 - \theta)A + \theta A^*) = \theta(A - A^*)$ . Where the quantity  $A - A^*$  behaves nicely, also inverted, because we chose  $c^*$  large enough.

Figure 7.1 shows the effects of under-relaxation. From this plot we can see that the wiggles that were caused by the alternating semi-diagonals of the inverse of  $A - A'$  using the original time stepping for small values of  $c$  are absent when using a scaled version  $A - A^*$  instead. These results were generated in `Matlab` for the 2D Cattaneo to Carter testcase [28, example 5.8].

### Numerical diffusion

One big flaw of this method is the appearance of numerical diffusion. This happens whenever  $c \neq 1$  or  $c^* \neq 1$ . When  $c^* < 1$  this is caused by the smoothing behaviour explained in Chapter 4. Here we will discuss the effects we see when using  $c^* = 1$ . The reason to see this as a separate problem is because this kind of smoothing is much less dependent on the size of  $c$ , as the previous occurrence shows very distinct behaviour between for example  $c = 0.8$  and  $c = 0.2$ . Also, this smoothing comes in the absence of wiggles for small values of  $c$ .

When using  $c^* = 1$  we use a matrix  $A^*$  where  $\delta q^* = \delta x$ . In a 2D case we determine the tractions by solving equation (3.14). When we take equation (3.14) and stick the timestep to the matrices, we find

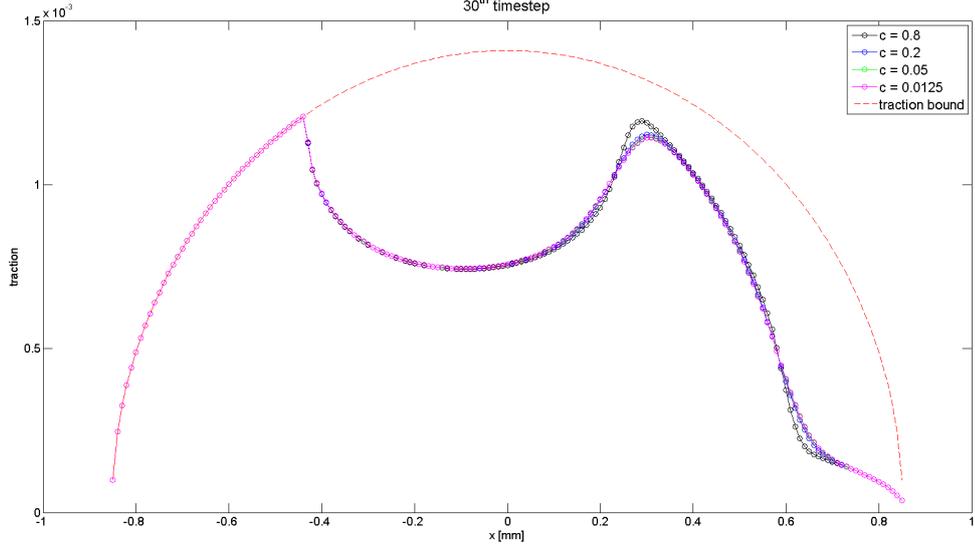


Figure 7.1: Solutions using increasingly smaller values of  $c$  after a traversed distance of  $24\Delta x$ .

an alternative expression for (3.15) in the form of:

$$p_{J1} = \delta q A_{I1J1}^{-1} \left( -w_I + \frac{A'_{I1J1}}{\delta q} p'_{J1} \right), \quad (7.2)$$

where the matrices, instead of the quantity  $w_I$ , are scaled by the timestep. Now we can plug in equation (7.2) our approximation for  $A'$  as given by (7.1) using the simpler notation:  $A' = (1-\theta)A + \theta A^*$ . This results in the following equation for the traction, where  $\delta q = c\delta x$  and  $\delta q^* = c^*\delta x$ :

$$p = \delta q A^{-1} \left( -w + \frac{A'}{\delta q} p' \right) \quad (7.3)$$

$$= \delta q A^{-1} \left( -w + \left( (1-\theta) \frac{A}{\delta q} + \theta \frac{A^*}{\delta q} \right) p' \right) \quad (7.4)$$

$$= \delta q A^{-1} \left( -w + (1-\theta) \frac{A}{\delta q} p' + \theta \frac{A^*}{\delta q} p' \right) \quad (7.5)$$

$$= -\delta q A^{-1} w + (1-\theta) \delta q A^{-1} \frac{A}{\delta q} p' + \theta \delta q A^{-1} \frac{A^*}{\delta q} p' \quad (7.6)$$

$$= -\delta q A^{-1} w + (1-\theta) p' + \theta \delta q A^{-1} \frac{A^*}{\delta q} p' \quad (7.7)$$

$$= (1-\theta) p' + \delta q A^{-1} \left( -w + \theta \frac{A^*}{\delta q} p' \right) \quad (7.8)$$

$$= (1-\theta) p' + \theta \delta q^* A^{-1} \left( -w + \frac{A^*}{\delta q^*} p' \right) \quad (7.9)$$

$$= (1-\theta) p' + \theta p^* \quad (7.10)$$

In the last but one step we used that, as  $\delta q = c\delta x$  and  $\theta = \frac{c}{c^*}$ , we can write  $\delta q = \frac{c}{c^*} c^* \delta x = \theta \delta q^*$  and  $\frac{\theta}{\delta q} = \frac{c}{c^* c \delta x} = \frac{1}{\delta q^*}$ . The  $p^*$  we introduced in (7.10) is what the new solution of  $p$  would be if instead of using  $c$  and a timestep  $\delta q$  we just used  $c^*$  and timestep  $\delta q^*$ . Therefore solving  $p$  by approximating the matrix  $A'$  by taking a linear combination of the matrix  $A$  and  $A^*$  gives the exact same solution as taking a linear combination of the old tractions  $p'$  and the traction  $p^*$ .

In practice however these two methods will not always give the exact same solution. When using relaxation to adjust the matrix  $A'$  as in (7.4) we are still working with the small timestep  $\delta q$ , however when using relaxation on the solution  $p$  as in (7.10) we work with a larger timestep  $\delta q$ . This means

that during the TANG [14, §4.3] algorithm process there can be a different partition of adhesion and slip elements. So (7.4) and (7.10) only give the same solution when using timesteps  $\delta q$  and  $\delta q^*$  dictate the same element partition as a different partition means that a different equation, either (3.15) or (3.16), is used on the elements that are in adhesion in one case and in slip in the other.

When we now apply this new method in the algorithm the first observation is that indeed there appear no artificial wiggles. The second observation however is that solutions obtained with a  $c < c^* = 1$  stay near the solution found using  $c = c^* = 1$  in a big part of the domain, but deviate from the solution around sharp edges. See Figure 7.2 for an example of this deviation.

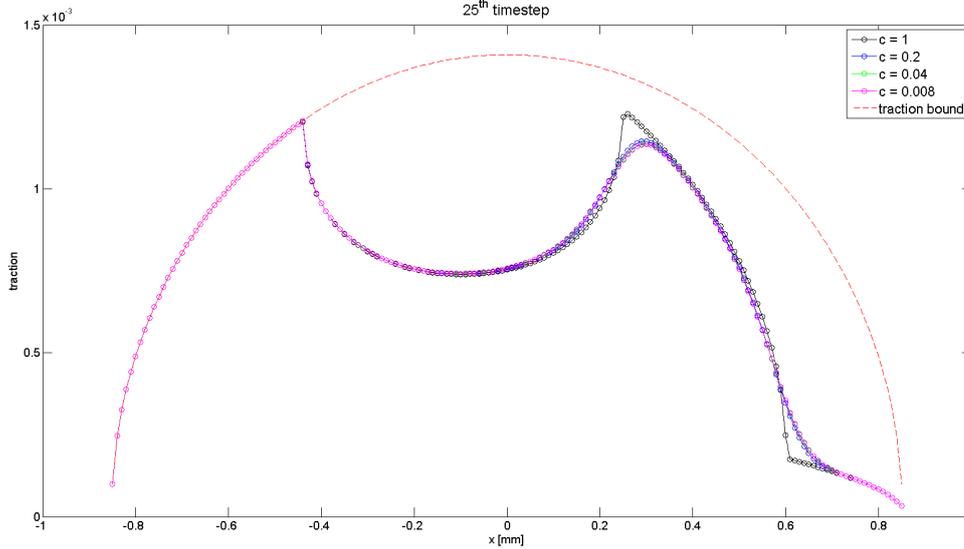


Figure 7.2: Solution of the traction  $p_x$  using different values for  $c$  after a traversed distance of  $25\Delta x$ .

The appearance of this deviation from the original solution ( $c = 1$ ), which looks like suppression of extrema, has the characteristics of numerical diffusion. This can be explained best by looking at the relaxation process in the way of equation (7.10). This says that the solution  $p$  is constructed as a weighted average of the solution  $p'$  at the previous timestep and the solution  $p^*$  of a point ahead in time (ahead by a factor  $\theta^{-1}$ ). Because of the advective nature of the slip-equation (2.2) the tractions gradually move from right to left in the contact area (because of the left to right motion of the rolling object). The solution  $p^*$  is close to the solution  $p$  translated a distance  $\delta q^*$  to the left. This means that at the sharp points in the solution, like the peak around 0.26 in Figure 7.2, this peak is replaced by the average of two sharp solutions. This averaging will slightly flatten out the sharp point. The transient character of the problem will make sure that as we progress in time the flattening accumulates, smoothing out the sharp point completely. In Figure 7.2 we also see one sharp point in the solution that stays intact. This is because here the solution moves into the traction bound, so although averaging does still occur the averaging happens beyond the traction bound, and is thus cut off in the TANG algorithm.



## Fractional matrix powers

In Chapter 4 we see that unphysical smoothing of sharp corners is caused by the fact that the matrix product  $A_{HH}^{-1}A'_{HH}$  is supposed to move the solution of the traction  $p_H$  a distance  $dq$  to the left. However this is only achieved without error when that distance is a multiple of the gridspacing  $\Delta x$ . Let  $A_{HH}^*$  be the matrix we find when we have  $c = 1$  and  $A'_{HH}$  the matrix when we use  $c < 1$ . Now for example when  $c = \frac{2}{3}$  we have that  $3dq = 2\Delta x$ . So after three timesteps the traction should have moved two gridspacings to the left. This means that we want  $(A_{HH}^{-1}A'_{HH})^3 \mathbf{p}'_H = (A_{HH}^{-1}A_{HH}^*)^2 \mathbf{p}'_H$ . This would be achieved when we take  $A'_{HH}$  to meet the following:

$$(A_{HH}^{-1}A'_{HH})^3 = (A_{HH}^{-1}A_{HH}^*)^2 \quad (8.1)$$

$$A_{HH}^{-1}A'_{HH} = (A_{HH}^{-1}A_{HH}^*)^{\frac{2}{3}} \quad (8.2)$$

$$A'_{HH} = A_{HH}(A_{HH}^{-1}A_{HH}^*)^{\frac{2}{3}}. \quad (8.3)$$

This example gives us a new way to approximate  $A'$ , now ensuring that we get an  $A'$  that correctly translates the tractions after a time  $n \cdot dq$  when  $n \cdot dq = m \cdot \Delta x$  for some  $n, m \in \mathbb{N}_{>0}$ . The used approximation for  $A'$  is:

$$A' = A(A^{-1}A^*)^c, \quad (8.4)$$

where  $A^*$  is the influence matrix calculated using  $c = 1$ .

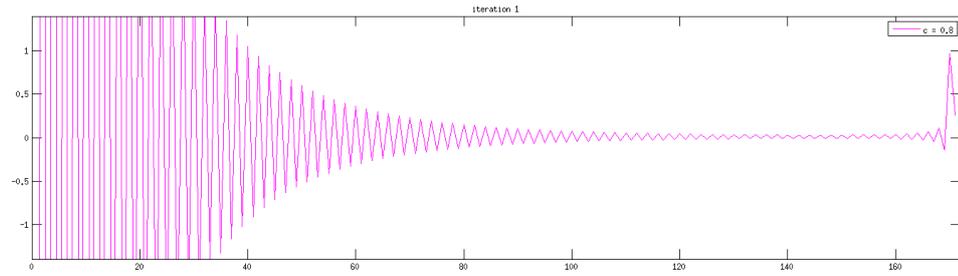
To see what this new  $A'$  does to a traction  $\mathbf{p}$  we will look at a number of different cases. First look at Figure 8.1a to see how  $A^{-1}A'$  works on a  $\mathbf{p}$  consisting of a single one at the far right end. Now look at Figures 8.1b and 8.1c to see what happens after 33 and 133 iterations. In this example  $c = 0.8$  is used.

From Figure 8.1 we see that ahead of the position of where the spike should be there are huge wiggles. There are a few small wiggles trailing behind, but at least these are damped. Next we will look at  $\mathbf{p}$  that have different shapes and see how the wiggles behave around these shapes. Results are shown in Figure 8.2, again using  $c = 0.8$ .

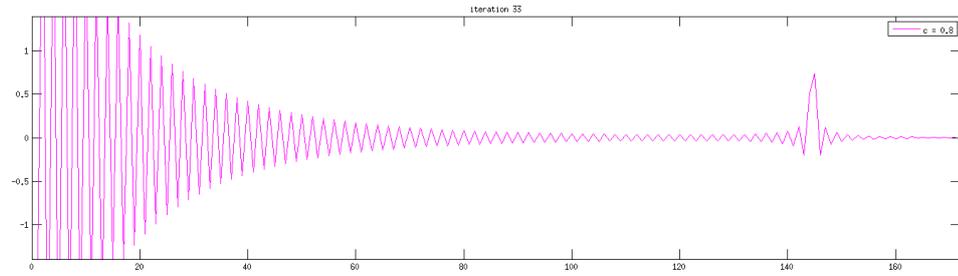
Judging from the results shown in Figure 8.2 it looks like using a relatively smooth  $\mathbf{p}$  prevents wiggles to arise in the proximity of the data. Now we will look at how well this approximation  $A'$  behaves in the same testcase as used in Figure 7.2. A similar plot is shown in Figure 8.3, this makes clear that when looking after a traversed distance of  $25\Delta x$  the solutions obtained using the approximate  $A'$  with different  $c$  are all on spot with the solution obtained using  $c = 1$ . Figure 8.4 shows the solutions obtained when using the directly calculated  $A'$  for  $c = 0.8$  and the approximated  $A'$  for  $c = 0.8$ . From this plot we see that although the new approximation does keep the sharp edges that are smoothed out in the old solution, a new type of wiggles arise when we have not traversed an integer multiple of  $\Delta x$ .

### Fractional power computation

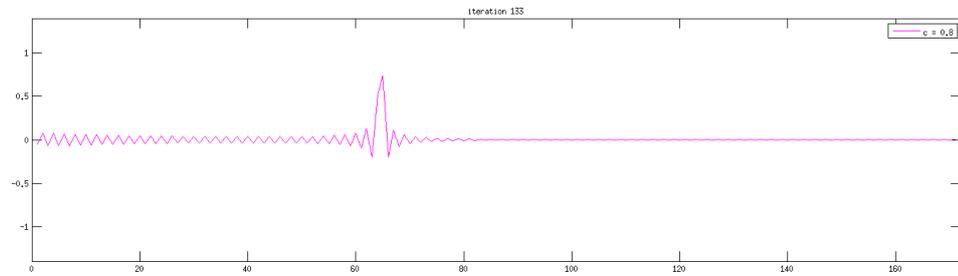
Although this solution looks nice (especially after using a filter to get rid of the wiggles in the intermediate results) the computational cost of determining non-integer powers of a matrix are unacceptably



(a)  $A^{-1}A'\mathbf{p}$  for a traction with a single one.



(b)  $(A^{-1}A')^{33}\mathbf{p}$  for a traction with a single one.



(c)  $(A^{-1}A')^{133}\mathbf{p}$  for a traction with a single one.

Figure 8.1: Results after 1, 33, 133 iterations.

high. Therefore this solution is unacceptable due to the computational costs it brings.

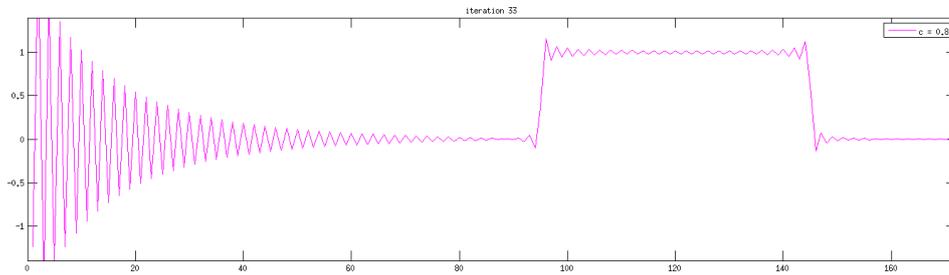
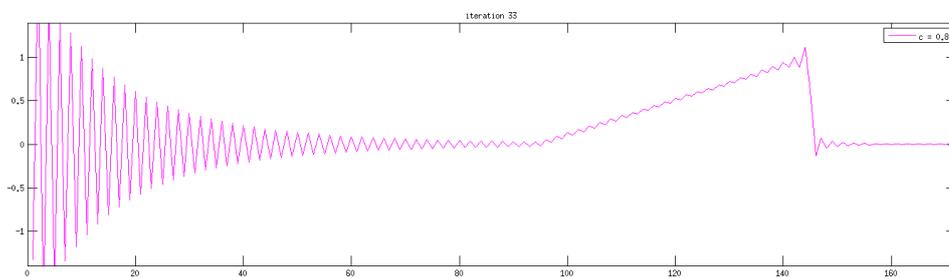
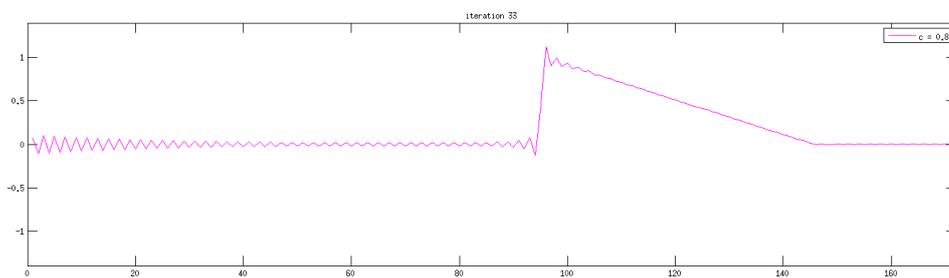
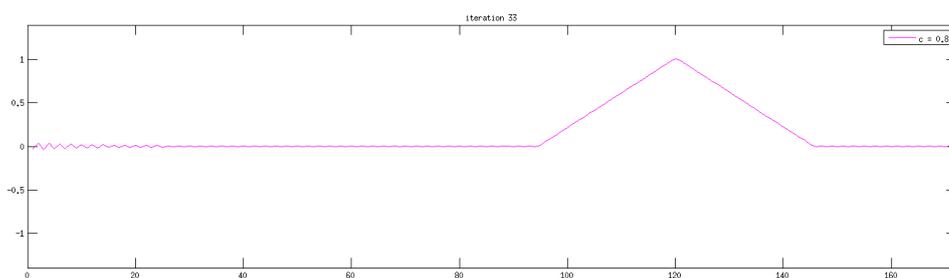
(a) Using a plateau as initial  $\mathbf{p}$ .(b) Using a ramp starting low and ending high as initial  $\mathbf{p}$ .(c) Using a ramp starting high and ending low as initial  $\mathbf{p}$ .(d) Using a pyramid as initial  $\mathbf{p}$ .

Figure 8.2: Results after 33, iterations.

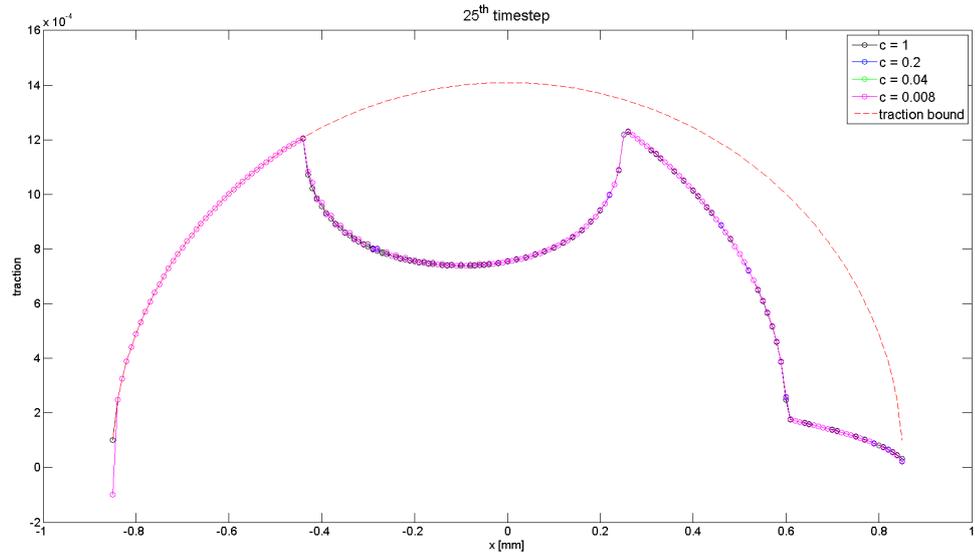


Figure 8.3: Solution of the traction  $p_x$  using different values for  $c$  after a traversed distance of  $25\Delta x$ .

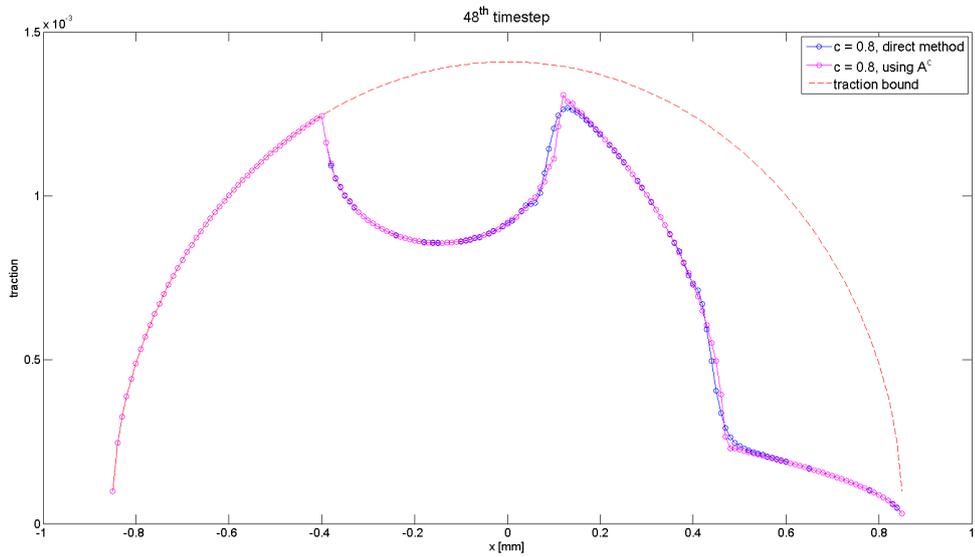


Figure 8.4: Solution of the traction  $p_x$  using different values for  $c$  after a traversed distance of  $38.4\Delta x$ .

## World Fixed implementation

Previously we looked at the contact problem in a contact fixed framework. The motivation for this approach is that the grid only has to be as large as the contact area and moves along with the contact area. For short simulations this is not a real issue, but when doing simulations where the total displacement covers a (large) multiple of the contact area, the grid we need to use in a framework that is not contact fixed needs to be that same multiple times as big as the grid used in a contact fixed framework.

Another approach is using a world fixed framework. Here the grid is fixed to the world and the contact area moves over this grid. The motivation for using this approach is the appearance of the smoothing effect for non-integer values of  $c$  as described in Chapter 4. The key difference is that in the contact-fixed approach we have traction  $\tau_i$  at position  $x_i$  and want to move that traction to somewhere in between position  $x_{i-1}$  and  $x_i$ , while in a world-fixed approach we have traction  $\tau_i$  at position  $x_i$  and want to find the traction in the next timestep, still at position  $x_i$ .

### CONTACT libraries for Matlab

During this process the actual CONTACT software is used. This is done by use of shared libraries that can be called from Matlab (or another programming language like C++). From Matlab the input settings are given, in the .dll file the contact problem is solved, and afterwards the output can be retrieved in Matlab to use the results. Now to use a contact fixed approach we have to set the CONTACT control parameter [28, section 3.5] T to 2 while for a world fixed approach we have to set T to 1. One thing to be careful about is the fact that in the contact fixed situation we can define the traction bound at the start and keep it constant, while in the world fixed case the traction bound is different in each step because the contact area moves over the chosen grid.

### Rolling cylinder

The situation we will investigate involves a solid cylinder on a hill. To make things easy the hill and cylinder will be made of the same material so we can assume quasi-identical behaviour. The hill and cylinder are in vacuum, so we do not concern ourselves with drag. As the cylinder is pulled down by gravity the friction between the cylinder and the hill will cause slipping and a traction on the cylinder. This traction will make the cylinder start to roll. As the rolling increases the traction will be reduced, resulting in less slipping and less rolling-resistance. This on its turn slows down the process that makes the cylinder roll faster while the cylinder does keep catching up velocity because of the constant gravitational pull. Thus again returning to a situation where it will slip and starts to roll faster. This process will cause an oscillating traction profile on the contact area until the system finally finds an equilibrium state where the increase in velocity and increase in rolling match each other and a stable traction profile is found. This system is interesting because of the constantly changing behaviour of the tractions. The 'peaks' in the traction profile as seen in Figure 8.3 keep forming in the contact area for

a long time so any smoothing effects, see Chapter 4, are persistent and could significantly influence the results.

## Evolution in time

Because the velocity and thus position of the system are not as straight-forward as in the much simpler Cattaneo-to-Carter testcase [28, section 5.8] considered before, we must take care of the proper evolution of the system in time. In order to do this both Velocity Verlet and the Leapfrog method [1] are implemented and the results of these two methods compared. These methods assume that we have a constant time-step, the change in position is the velocity, the change in velocity the acceleration, and the acceleration a function of the position only. This means that the methods can not be used in their original form, because we do not have a constant time-step and the acceleration is a function of both position and velocity.

First let us define the physical quantities that need to be calculated. Let  $t$  be the time,  $X$  the middle of the contact area,  $v$  the velocity, and  $a$  the acceleration.

Groot and Warren [8, equation 9] used an adapted version of Velocity Verlet that introduces a factor  $\lambda$  in the intermediate velocity calculation. Using  $\lambda = \frac{1}{2}$  produces the normal Velocity Verlet algorithm in case the acceleration is only a function of position. In their experiments they primary choose  $\lambda = \frac{1}{2}$ . In our case this coice of  $\lambda$  is also the right choice because of the backwards nature of the solver used in CONTACT. From now on, when we talk about Velocity Verlet we refer to their adapted version that allows for the acceleration to be a function of both position and velocity.

## Velocity Verlet

Velocity Verlet actually is a variable time implementation of the Verlet method already. This means that when using Velocity Verlet we do not have to worry about varying timesteps. What sets Velocity Verlet apart from Leapfrog is that the position, velocity, and acceleration are all determined at the same point in time. In pseudocode the method comes down to the following (note that Groot and Warren's facotor  $\lambda$  has been replaced by 0.5 already):

```
initialize: x(0), v(0), a(0)

for i = 0:N-1 {
  dt(i) = dt(v(i))
  v'(i+1) = v(i) + dt(i)*a(i)*0.5
  x(i+1) = x(i) + dt(i)*v'(i+1)
  a(i+1) = Fa( x(i+1),v'(i+1) )
  v(i+1) = v(i) + dt(i) * (a(i)+a(i+1))/2
}
```

This process can be visualised by the following scheme:

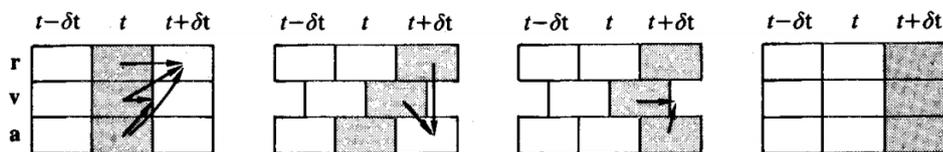


Figure 9.1: Visualisation of the Velocity Verlet scheme, the scheme is an edited version of the one found in [1].

## Leapfrog

The Leapfrog method does not accept a varying timestep. Therefore we need to calculate a new timestep within the loop. This is because the second half of the scheme will use the updated value of the timestep.

What sets Leapfrog apart from Velocity Verlet is that the position and acceleration are determined at one point in time, but the velocity at half a timestep later. The pseudocode is given by:

```
initialize: x(0), v(0), a(0)

dt(0) = dt(v(0))
v(0) = v(0) + dt(0)*a(0)/2

for i = 0:N-1 {
  x(i+1) = x(i) + dt(i)*v(i)
  a(i+1) = Fa( x(i+1),v(i) )
  v'(i+1) = v(i) + dt(i)*a(i+1)/2
  dt(i+1) = dt(v'(i+1))
  v(i+1) = v'(i+1) + dt(i+1)*a(i+1)/2
}
```

This process can be visualised by the following scheme:

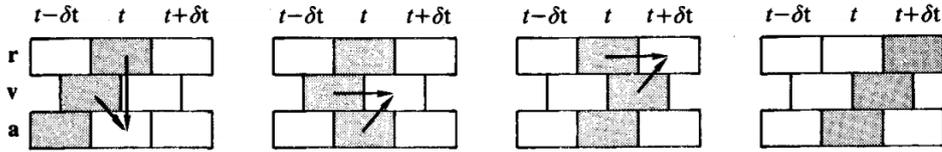


Figure 9.2: Visualisation of the Leapfrog scheme, the scheme is an edited version of the one found in [1]. The first step in the pseudocode is actually the third step in the visualisation of the scheme.

From this visualisation it becomes clear where Leapfrog gets its name from. The moments in time on which the most recent position and most recent velocity are know leap over each other.

Next to velocity and acceleration we also have to determine the angular velocity and angular acceleration. This is done parallel to the normal velocity and acceleration calculations in a similar fashion.

## Energy

For this situation no analytical or experimental results are known. This means that another method of validation is needed. One of the physical values we can determine is the total energy in the system. Because of the law of conservation of energy we know that the total energy of the system is a constant. So we need to have that at any time the total difference in energy with the begin situation should be zero. Conservation of energy holds in a closed system, in our case this system is not only the rolling cylinder but also the hill, because energy that is lost by friction will be converted into heat and ends up in both the cylinder and the hill. The total energy can be found by taking the sum over all forms of energy in the system. This is given in the following formula:

$$E_{total} = E_{potential} + E_{velocity} + E_{rotational} + E_{elastic} + E_{work} \quad (9.1)$$

Here the kinetic energy is split into kinetic energy due to velocity and angular velocity. The individual forms of energy can be expressed by:

$$E_{potential} = (x_0 - x) \cdot \sin(\alpha) \cdot m \cdot g \quad (9.2)$$

$$E_{velocity} = \frac{1}{2} \cdot m \cdot v^2 \quad (9.3)$$

$$E_{rotational} = \frac{1}{2} \cdot I \cdot \omega^2 \quad (9.4)$$

$$E_{elastic} = \frac{1}{2} \cdot k \cdot u^2 \quad (9.5)$$

$$E_{work} = \int_0^t F_{fric} \cdot (x_0 - x) \, d\tau \quad (9.6)$$

where  $\alpha$  is the slope of the hill,  $m$  the mass of the cylinder,  $g$  the gravitational acceleration,  $I$  the moment of inertia, and  $k$  the ‘spring constant’ of the cylinder.

In the system we will not work with the term  $\sin(\alpha) \cdot m \cdot g$  but simply use a given  $F_{cm}$ , the force due to gravity. The moment of inertia  $I$  is also given. Using the output given by CONTACT the energy equations can be rewritten as:

$$E_{potential} = -X \cdot F_{cm} \quad (9.7)$$

$$E_{velocity} = \frac{1}{2} \cdot m \cdot v^2 \quad (9.8)$$

$$E_{rotational} = \frac{1}{2} \cdot I_y \cdot \omega^2 \quad (9.9)$$

$$E_{elastic} = \frac{1}{2} \cdot \iint_S u_x \cdot p_x \, dx dy \quad (9.10)$$

$$E_{work} = - \int_0^t \iint_S (v \cdot s_x) \cdot p_x \, dx dy \, d\tau \quad (9.11)$$

where the surface integrals are taken over the contact area. The quantities given by CONTACT are: the displacements  $u_x$ , the tractions  $p_x$ , and the relative slip  $s_x$ . We have to take care with the slip. When we use a contact fixed approach we get as output the relative slip  $s_x$  which we can immediately use this. When we use a world fixed approach we get the absolute slip  $S_x = dq \cdot s_x$ .

Now that we can calculate the energy in the system at any moment we can check how much the calculated energy change differs from the theoretical value of 0. This is not only useful for checking whether a solution is physically possible, but also to analyse the convergence of a method. Results indicate that doubling the number of gridpoints approximately divides the error in the energy by 1.7. This suggests a slow convergence. There is no difference between the use of Velocity Verlet or Leapfrog, so either method can be used.

## Leading edge adjustments

In order to improve the convergence of the overall scheme we will focus on the leading edge. The leading edge is the first grid point that enters the contact area during rolling. So in the examples used in this theses, where motion goes from left to right the leading edge is at the right side of the contact area. In theory the numerical solver should have second order convergence. However, in practice we do not see faster than first order convergence when refining the grid. We suspect that the reason for this is caused by a first order error in the leading edge. Because of the upwind nature of the system, this first order error is transported through the whole system resulting in an overall first order error.

### Analytic solution in 2D steady state

An analytic solution for the Carter-Fromm problem [28, example 5.2] is given by Kalker [14, equation 2.12]. In our test case we have a contact area that is centered around the origin with a width of 1mm where 60% ( $a = 0.6$ ) of the contact area is in adhesion. The normalized analytic solution for the tractions  $p(x)$  is given as:

$$p(x) = \begin{cases} -\sqrt{1-x^2} & \text{when } -1 \leq x < 1-2a \\ -\sqrt{1-x^2} + \sqrt{a^2 - (x-1+a)^2} & \text{when } 1-2a \leq x \leq 1 \\ 0 & \text{else} \end{cases} . \quad (10.1)$$

Using this solution and the influence function  $A$  we can calculate the analytic solution for the displacements  $u(x)$ . This gives the following expression (write  $u(x) = u(x, 0, 0)$ ):

$$\begin{aligned} u(x) &= \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_{-1}^1 A(x, 0, x', y') p(x') dx' dy' \\ &= \frac{1}{\pi G} \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \left( \int_{-1}^1 \frac{(\nu-1)\sqrt{1-x'^2}}{\sqrt{(x-x')^2 + y'^2}} - \nu \frac{(x-x')^2 \sqrt{1-x'^2}}{\sqrt{(x-x')^2 + y'^2}^3} dx' \right. \\ &\quad \left. + \int_{1-2a}^1 \frac{(1-\nu)\sqrt{a^2 - (x'-1+a)^2}}{\sqrt{(x-x')^2 + y'^2}} + \nu \frac{(x-x')^2 \sqrt{a^2 - (x'-1+a)^2}}{\sqrt{(x-x')^2 + y'^2}^3} dx' \right) dy. \end{aligned} \quad (10.2)$$

The integration over  $x'$  can not be done analytically, the integration over  $y'$  can be done analytically. So we will first integrate over  $y'$  analytically and then do the numerical integration over  $x'$ . The integrals over  $y'$  are all of the forms:

$$\int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \frac{r}{\sqrt{s^2 + y'^2}} dy' = 2r \operatorname{arcsinh} \left( \frac{\Delta y}{2|s|} \right) \quad (10.3)$$

and

$$\int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \frac{s^2 r}{\sqrt{s^2 + y'^2}^3} dy' = 2 \frac{r \Delta y}{\sqrt{4s^2 + (\Delta y)^2}}. \quad (10.4)$$

This turns equation (10.2) into:

$$\begin{aligned} \frac{\pi G}{2} u(x) &= (\nu - 1) \int_{-1}^1 \sqrt{1 - x'^2} \operatorname{arcsinh} \left( \frac{\Delta y}{2|x - x'|} \right) dx' \\ &- \nu \int_{-1}^1 \frac{\sqrt{1 - x'^2} \Delta y}{\sqrt{4(x - x')^2 + (\Delta y)^2}} dx' \\ &+ (1 - \nu) \int_{1-2a}^1 \sqrt{a^2 - (x' - 1 + a)^2} \operatorname{arcsinh} \left( \frac{\Delta y}{2|x - x'|} \right) dx' \\ &+ \nu \int_{1-2a}^1 \frac{\sqrt{a^2 - (x' - 1 + a)^2} \Delta y}{\sqrt{4(x - x')^2 + (\Delta y)^2}} dx' \end{aligned} \quad (10.5)$$

Using numerical integration this gives us a (semi-) analytic value for  $u(x)$ . In order to test how well the CONTACT algorithm converges when  $u$  is known at the boundary we will use this to determine the value at the leading edge. Define  $u_{LE} = u(1)$ , where the exact value of  $u$  is used.

## Solution scheme using the leading edge displacement

When solving equation (2.10) we are interested in the interior elements of the contact area. Here we replace  $\mathbf{u}$  by the matrix-vector product  $A\mathbf{p}$  and  $\mathbf{u}'$  by the matrix-vector product  $A'\mathbf{p}$ . The last entry in the vector  $\mathbf{u}'$  however is the displacement one gridpoint upstream of the last gridpoint in the interior, in other words, *at* the boundary. Tests are done in MATLAB and implementation will be similar to Section 3.3.

We use that  $\mathbf{u}' = A'\mathbf{p}$ . Here all but the last entry of the vector are unknowns, the last entry will be  $u_{LE}$ . This means that we can replace the matrix  $A'$  by a submatrix that is equal to  $A'$  in all but it's last row (notate this part by  $A'^{m-1 \times m}$ ), and has a row with zeros as it's last row. Now we can write:

$$\mathbf{u}' = \begin{bmatrix} A'^{m-1 \times m} \\ 0 \cdots 0 \end{bmatrix} \mathbf{p} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ u_{LE} \end{bmatrix} \quad (10.6)$$

This means that we can use the same framework as in equation (3.18), where the vector with  $u_{LE}$  will be merged with the vector containing the rigid slip  $\mathbf{w}$  and in the matrix  $M$  will use the adjusted  $A'$  with one row of zeros instead of the full matrix  $A'$ .

## Effects on convergence

The first result we can look at is comparing how knowing the exact value of the displacement and using this influences the solution of the steady-state 2D problem. In Figure 10.1 we can see that the solution is much closer to the analytic Carter-Fromm solution when using the exact displacement at the leading edge. This is also made quantitative when we look at the error in the 1-Norm.

The more interesting effect though is the effect on the convergence. When using the original scheme going from 15 to 30, 60, 120, and 240 grid points reduces the error with a factor 1.83 on average, and a factor 2.09 in optimal situations. But when the displacement at the leading edge is known, so the traction at the leading edge can be calculated more accurately we see that the same grid refinements result in a reduction in the error with a factor of 2.89 on average, and a factor 4.96 in the optimal situation.

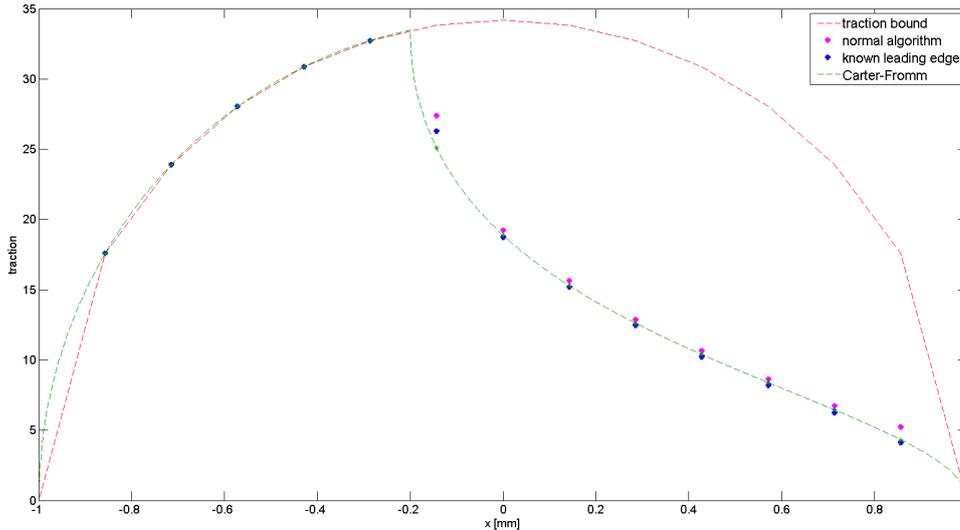


Figure 10.1: Difference between using the exact solution for the leading edge displacement or not in the 2D Carter-Fromm problem.

## Adjusting the Leading Edge basis function

The results from the previous section look great, but they are not usable in practice. We could only do this because we started with the analytic solution in the first place. It does however indicate strongly that when the fact that we only observe first order convergence in the normal case is primarily caused by the error made at the leading edge. If the traction in this one gridpoint can be calculated more accurately the total traction will be more accurate as well. How well the solution at the leading edge can be calculated depends on how the gridpoints are located with respect to the edge. When using a linear basis function this will give rise to a triangle function with the peak at the last gridpoint  $x_N$ , and nonzero values up to the last gridpoint plus a distance  $\Delta x$ . Let us assume that the contact area runs from  $-1$  to  $1$ . Define a variable  $\alpha = \frac{1-x_N}{\Delta x}$ . This means that  $0 \leq \alpha < 1$  because if  $\alpha \geq 1$  there would be another gridpoint in the contact area after  $x_N$ . For values of  $\alpha$  close to one there will only be a small contribution of the last basis function beyond the contact area. But when  $\alpha$  becomes small, the greater part of the last basis function will lie outside the contact area. There the traction is zero, so any contribution will be nonphysical. So we will adjust the leading edge basis function so that the basis function will be zero outside the contact area, see Figure 10.2 for this adjustment.



Figure 10.2: On the left, the general basis function, on the right the adjusted version used at the leading edge.

This adjustment has been tested in the 2D steady-state Carter-Fromm testcase. The discrete traction  $p_N$  at the leading edge is overestimated using this method, however this new value, combined with its adjusted basis function do lead to improved  $p_n$  for all other tractions in the adhesion area. This gives an overall improvement on the traction in 1-norm. Sadly though, except for a select range of values for  $\alpha$  the rate of convergence is not significantly improved. An average factor of a 1.7146 smaller 1-norm error is gained however. See Figure 10.3 for an example using  $\alpha = 0.4$ .

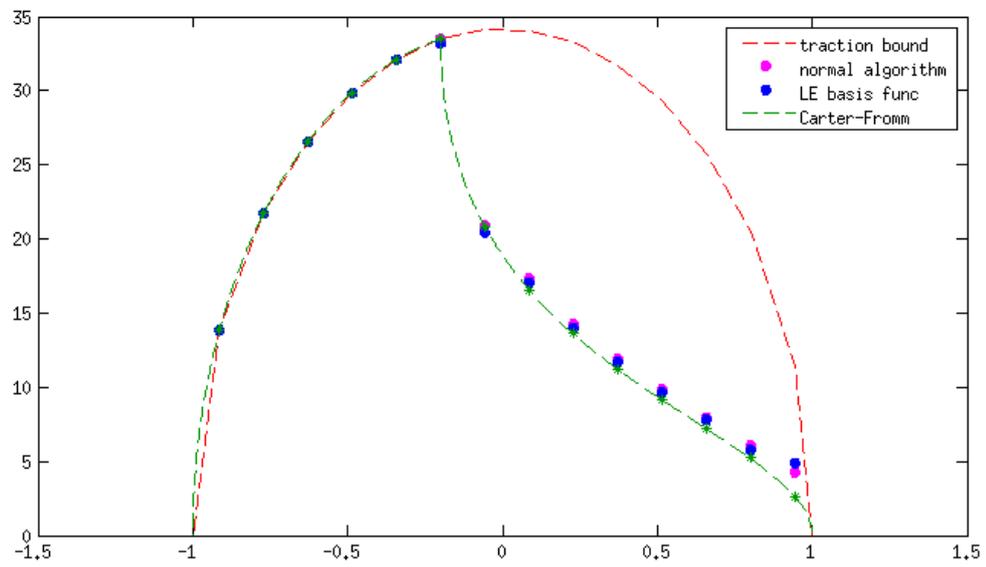


Figure 10.3: The Carter-Fromm testcase, using normal linear basisfunctions and a basis function adjusted for the leading edge with  $\alpha = 0.4$ .

## Conclusion and recommendations

In this thesis three problems have been investigated and several attempts have been made to solve them, with varying degrees of success.

First of all, the appearance of wiggles arising for small time steps has been investigated. This was caused by the amplification of high frequencies by the operator dealing with the spatial derivative in the slip-equation. An attempt to solve this problem was the use of higher order basis functions. This however did not help against the wiggles at all. An adjustment to the time-stepping scheme, similar to under-relaxation, did remove the wiggles, even at very small time steps.

Secondly, both the original scheme as the one improved by the new time-stepping suffered from numerical smoothing of extrema in the solution. A solution using fractional matrix powers was found, but due to the high computational cost this was dropped in favour of the use of a world-fixed implementation of the solution. This comes down to looking at the problem in an Eulerian frame of reference instead of the previously more used Lagrangian frame of reference. Both of these subjects have been presented in a conference in Madeira [32].

Finally the order of convergence of the solver has been investigated. In theory we would expect second order convergence, this however is not observed. The introduction of higher order basis functions was also expected to increase the rate of convergence, but this seems not to be the case. Using analytic solutions it was found that the primary cause of the low rate of convergence is the first order error made when determining the leading edge traction. This error propagates through the whole contact area and limits the overall rate of convergence. Therefore an adjustment for basis function used at the leading edge was made. This did reduce the error made, compared to the analytic solution. Sadly there was no significant increase in the rate of convergence save for a small range of input values.

So this means that the problem with wiggles is solved. It looks like the smoothing can be stopped as well, however further tests show odd behaviour when using the world-fixed implementation in certain cases. The cause for the low order of convergence has been found, a solution is still not presented however.

### Recommendations for further research

The use of higher order basisfunctions might improve the accuracy of the solution in the subsurface of the material. So although they were not useful in this thesis, they could be useful for the stress calculation.

The problems arising in the world-fixed frame need to be investigated. It is still unknown why they appear.

A way to improve the solution at the leading edge is valuable as it will likely improve the overall order of convergence, thereby drastically decreasing computational time to reach a satisfying accuracy.



# Appendices



# A

## Explicit expression for the influence coefficients using constant basis functions

$$A_{11111} = \int_{-\frac{1}{2}\Delta x}^{\frac{1}{2}\Delta x} \int_{-\frac{1}{2}\Delta y}^{\frac{1}{2}\Delta y} A_{11}(\mathbf{z}) dz_2 dz_1 \quad A'_{11111} = \int_{-\frac{1}{2}\Delta x-dq}^{\frac{1}{2}\Delta x-dq} \int_{-\frac{1}{2}\Delta y}^{\frac{1}{2}\Delta y} A_{11}(\mathbf{z}) dz_2 dz_1 \quad (\text{A.1})$$

$$A_{21111} = \int_{-\frac{3}{2}\Delta x}^{-\frac{1}{2}\Delta x} \int_{-\frac{1}{2}\Delta y}^{\frac{1}{2}\Delta y} A_{11}(\mathbf{z}) dz_2 dz_1 \quad A'_{21111} = \int_{-\frac{3}{2}\Delta x-dq}^{-\frac{1}{2}\Delta x-dq} \int_{-\frac{1}{2}\Delta y}^{\frac{1}{2}\Delta y} A_{11}(\mathbf{z}) dz_2 dz_1 \quad (\text{A.2})$$

$$A_{11121} = \int_{\frac{1}{2}\Delta x}^{\frac{3}{2}\Delta x} \int_{-\frac{1}{2}\Delta y}^{\frac{1}{2}\Delta y} A_{11}(\mathbf{z}) dz_2 dz_1 \quad A'_{11121} = \int_{\frac{1}{2}\Delta x-dq}^{\frac{3}{2}\Delta x-dq} \int_{-\frac{1}{2}\Delta y}^{\frac{1}{2}\Delta y} A_{11}(\mathbf{z}) dz_2 dz_1 \quad (\text{A.3})$$

Subtracting  $A'_{I_1 J_1}$  from  $A_{I_1 J_1}$  to get  $M_{IJ}$  gives us:

$$M_{11} = - \int_{-\frac{1}{2}\Delta x-dq}^{-\frac{1}{2}\Delta x} \int_{-\frac{1}{2}\Delta y}^{\frac{1}{2}\Delta y} A_{11}(\mathbf{z}) dz_2 dz_1 + \int_{\frac{1}{2}\Delta x-dq}^{\frac{1}{2}\Delta x} \int_{-\frac{1}{2}\Delta y}^{\frac{1}{2}\Delta y} A_{11}(\mathbf{z}) dz_2 dz_1 \quad (\text{A.4})$$

$$M_{21} = - \int_{-\frac{3}{2}\Delta x-dq}^{-\frac{3}{2}\Delta x} \int_{-\frac{1}{2}\Delta y}^{\frac{1}{2}\Delta y} A_{11}(\mathbf{z}) dz_2 dz_1 + \int_{-\frac{1}{2}\Delta x-dq}^{-\frac{1}{2}\Delta x} \int_{-\frac{1}{2}\Delta y}^{\frac{1}{2}\Delta y} A_{11}(\mathbf{z}) dz_2 dz_1 \quad (\text{A.5})$$

$$M_{12} = - \int_{\frac{1}{2}\Delta x-dq}^{\frac{1}{2}\Delta x} \int_{-\frac{1}{2}\Delta y}^{\frac{1}{2}\Delta y} A_{11}(\mathbf{z}) dz_2 dz_1 + \int_{\frac{3}{2}\Delta x-dq}^{\frac{3}{2}\Delta x} \int_{-\frac{1}{2}\Delta y}^{\frac{1}{2}\Delta y} A_{11}(\mathbf{z}) dz_2 dz_1 \quad (\text{A.6})$$

Where, using  $R = \sqrt{z_1^2 + z_2^2}$ , we have [14, equation (4.39d)]:

$$A_{11} = \frac{R^{-1} - \nu z_2^2 R^{-3}}{\pi G} \quad (\text{A.7})$$

This can be reformulated as:

$$\begin{aligned} \pi G A_{11} &= \frac{1}{R} - \nu \frac{z_2^2}{R^3} \\ &= \frac{z_1^2 + z_2^2}{R^3} - \nu \frac{z_2^2}{R^3} \\ &= \frac{z_1^2}{R^3} + (1 - \nu) \frac{z_2^2}{R^3} \end{aligned} \quad (\text{A.8})$$

We can write [14, equation (4.41a,b)]:

$$\iint \frac{z_1^2}{R^3} dS = z_2 \log(z_1 + R) \quad (\text{A.9})$$

$$\iint \frac{z_2^2}{R^3} dS = z_1 \log(z_2 + R) \quad (\text{A.10})$$

Combining equations (A.8), (A.9), and (A.10) gives us:

$$\pi G \iint A_{11} dS = z_2 \log(z_1 + R) + (1 - \nu) z_1 \log(z_2 + R) \quad (\text{A.11})$$

Now we can use this to write down expressions for  $M_{11}$ . In order to make the expression more readable we will substitute:

$$\begin{aligned} x'_l &= -\frac{1}{2}\Delta x - dq & x'_u &= \frac{1}{2}\Delta x - dq \\ x_l &= -\frac{1}{2}\Delta x & x_u &= \frac{1}{2}\Delta x \\ y_l &= -\frac{1}{2}\Delta y & y_u &= \frac{1}{2}\Delta y \end{aligned}$$

Note that:

$$\int_{x_l}^{x_u} \int_{y_l}^{y_u} \frac{\partial^2 f(\mathbf{z})}{\partial z_1 \partial z_2} dz_2 dz_1 = f(x_u, y_u) - f(x_u, y_l) - f(x_l, y_u) + f(x_l, y_l) \quad (\text{A.12})$$

Using the substitution and equations (A.11) and (A.12) gives us the following expression for  $M_{11}$ :

$$\begin{aligned} M_{11} &= - \left( y_u \log \left( x_l + \sqrt{x_l^2 + y_u^2} \right) + (1 - \nu) x_l \log \left( y_u + \sqrt{x_l^2 + y_u^2} \right) \right) \\ &\quad + \left( y_l \log \left( x_l + \sqrt{x_l^2 + y_l^2} \right) + (1 - \nu) x_l \log \left( y_l + \sqrt{x_l^2 + y_l^2} \right) \right) \\ &\quad + \left( y_u \log \left( x'_l + \sqrt{x_l'^2 + y_u^2} \right) + (1 - \nu) x'_l \log \left( y_u + \sqrt{x_l'^2 + y_u^2} \right) \right) \\ &\quad - \left( y_l \log \left( x'_l + \sqrt{x_l'^2 + y_l^2} \right) + (1 - \nu) x'_l \log \left( y_l + \sqrt{x_l'^2 + y_l^2} \right) \right) \\ &\quad + \left( y_u \log \left( x_u + \sqrt{x_u^2 + y_u^2} \right) + (1 - \nu) x_u \log \left( y_u + \sqrt{x_u^2 + y_u^2} \right) \right) \\ &\quad - \left( y_l \log \left( x_u + \sqrt{x_u^2 + y_l^2} \right) + (1 - \nu) x_u \log \left( y_l + \sqrt{x_u^2 + y_l^2} \right) \right) \\ &\quad - \left( y_u \log \left( x'_u + \sqrt{x_u'^2 + y_u^2} \right) + (1 - \nu) x'_u \log \left( y_u + \sqrt{x_u'^2 + y_u^2} \right) \right) \\ &\quad + \left( y_l \log \left( x'_u + \sqrt{x_u'^2 + y_l^2} \right) + (1 - \nu) x'_u \log \left( y_l + \sqrt{x_u'^2 + y_l^2} \right) \right) \quad (\text{A.13}) \end{aligned}$$

For  $M_{21}$  and  $M_{12}$  we have the same expression as formula (A.13) when using a different substitution.

# B

## Influence coefficients based on piecewise quadratic basisfunctions

Following Vollebregt's approach we get [24, equation 36a]:

$$u_x(x, y, 0) = \frac{P_x}{\pi G} \left( \frac{1 - \nu}{\rho} + \nu \frac{(x - \xi)^2}{\rho^3} \right). \quad (\text{B.1})$$

This is the displacement we get at  $(x, y, 0)$  from a point stress  $P_x$  at  $(\xi, \eta)$ . If we now want to know the displacement we get at  $(x, y, 0)$  as a result of the whole contact area  $C$  we need to sum over all point pressures  $p_x(\xi, \eta)$  times the surface  $\delta\xi \cdot \delta\eta$  of each point. Summing over these infinitesimal points results in the integral:

$$u_x(x, y, 0) = \frac{1}{\pi G} \iint_C \left( \frac{1 - \nu}{\rho} + \nu \frac{(x - \xi)^2}{\rho^3} \right) p_x(\xi, \eta) d\xi d\eta. \quad (\text{B.2})$$

This lets us define the influence function:

$$A_{ux}(x, y, 0, \xi, \eta) = \frac{1}{\pi G} \left( \frac{1 - \nu}{\rho} + \nu \frac{(x - \xi)^2}{\rho^3} \right). \quad (\text{B.3})$$

Now introduce two sets of basisfunctions. One is a parabola contained within an element, the other is a Lagrange polynomial spanning two elements and having the centre of the basisfunction at the edge of two elements. The basisfunctions defined on the origin are:

$$\varphi^o(x', y') = \begin{cases} 1 - \left( \frac{x'}{\Delta x} \right)^2 & \text{when } |x'| \leq \Delta x \wedge |y| < \frac{\Delta y}{2} \\ 0 & \text{else} \end{cases}, \quad (\text{B.4})$$

$$\varphi^e(x', y') = \begin{cases} \frac{1}{2} \left( \frac{x'}{\Delta x} + \frac{3}{2} \right)^2 - \frac{1}{8} & \text{when } -2\Delta x \leq x' < 0 \wedge |y| < \frac{\Delta y}{2} \\ \frac{1}{2} \left( \frac{x'}{\Delta x} - \frac{3}{2} \right)^2 - \frac{1}{8} & \text{when } 0 \leq x' < 2\Delta x \wedge |y| < \frac{\Delta y}{2} \\ 0 & \text{else} \end{cases}. \quad (\text{B.5})$$

Note that we not only look at the centres  $(x_j)$  of the elements but also at the edges of the elements  $(x_j + \frac{\Delta x}{2})$ . From here on let the  $x_j = (j - 1) \frac{\Delta x}{2}$  denote both the centres and the edges of the elements. We can use this to define the basisfunctions for each element as:

$$\varphi_J(x, y) = \begin{cases} \varphi^e(x - x_J, y - y_J) & \text{when } J \text{ is odd} \\ \varphi^o(x - x_J, y - y_J) & \text{when } J \text{ is even} \end{cases}. \quad (\text{B.6})$$

The basisfunction  $\varphi_J$  is now only nonzero when  $|x - x_K| < \frac{\Delta x}{2}$  for  $J$  even, while  $\varphi_J$  is only nonzero when  $|x - x_K| < \Delta x$  for  $J$  odd. We can approximate the tractionfield with a continuous, piece-wise quadratic,  $p_x$  given by:

$$p_x(x, y) = \sum_J p_J \varphi_J(x, y). \quad (\text{B.7})$$

The  $p_J$  are the coefficients to be determined. We can substitute the discretized traction (B.7) into (B.2) and find:

$$u_x(x, y, 0) = \iint_C A_{ux}(x, y, 0, \xi, \eta) \sum_J p_J \varphi_J(x, y) d\xi d\eta. \quad (\text{B.8})$$

The order of integration and summation may be reversed, and the constants  $p_J$  can be taken out of the integral:

$$u_x(x, y, 0) = \sum_J p_J \iint_C A_{ux}(x, y, 0, \xi, \eta) \varphi_J(\xi, \eta) d\xi d\eta. \quad (\text{B.9})$$

We now define the influence coefficients  $A_{IuJx}$  as the integrals in the last equation. So we have:

$$u_x(x_I, y_I, 0) = \sum_J A_{IuJx} p_J, \quad (\text{B.10})$$

with

$$A_{IuJx} = \iint_C A_{ux}(x_I, y_I, 0, \xi, \eta) \varphi_J(\xi, \eta) d\xi d\eta. \quad (\text{B.11})$$

Use relation (B.6) and only integrate over the nonzero parts to get:

$$A_{IuJx} = \iint_{\mathcal{I}(\varphi_J)} A_{ux}(x_I, y_I, 0, \xi, \eta) \varphi_J(\xi, \eta) d\xi d\eta. \quad (\text{B.12})$$

Here  $\mathcal{I}(\varphi_J)$  is the area where  $\varphi_J$  is nonzero, so:  $\mathcal{I}(\varphi_J) = (x_J - \Delta x, x_J + \Delta x) \times (y_J - \frac{\Delta y}{2}, y_J + \frac{\Delta y}{2})$  for  $J$  even and  $\mathcal{I}(\varphi_J) = (x_J - 2\Delta x, x_J + 2\Delta x) \times (y_J - \frac{\Delta y}{2}, y_J + \frac{\Delta y}{2})$  for  $J$  odd. Apply a transformation of variables, going from  $x_I, y_I, \xi, \eta$  to:

$$x'_I = x_I - x_J, \quad y'_I = y_I - y_J, \quad \xi' = \xi - x_J, \quad \eta' = \eta - y_J. \quad (\text{B.13})$$

This translates the area of integration so that the origin lies at the centre. Using equation (B.3) we know that  $A_{ux}$  is only a function of  $(x_I - \xi)$  and  $(y_I - \eta)$ , which is equal to  $(x'_I - \xi')$  and  $(y'_I - \eta')$ , so  $A_{ux}(x_I, y_I, 0, \xi, \eta) = A_{ux}(x'_I, y'_I, 0, \xi', \eta')$ . From (B.6) we have that  $\varphi_J(\xi, \eta) = \varphi(\xi', \eta')$ . So we can write the equation as:

$$A_{IuJx} = \iint_{\mathcal{I}(\varphi)} A_{ux}(x'_I, y'_I, 0, \xi', \eta') \varphi(\xi', \eta') \delta\xi' \delta\eta'. \quad (\text{B.14})$$

From the definition given in (B.4) and (B.5) we have the expression for  $\varphi^o(\xi, \eta)$  and  $\varphi^e(\xi, \eta)$  and for convenience drop the primes on  $\xi$  and  $\eta$ . This gives the expressions for the influence coefficients for even and odd  $J$  respectively:

$$A_{IuJx} = \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_{-\Delta x}^{\Delta x} A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) \left(1 - \left(\frac{\xi}{\Delta x}\right)^2\right) d\xi d\eta, \quad (\text{B.15})$$

$$\begin{aligned} A_{IuJx} &= \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_{-2\Delta x}^0 A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) \left(\frac{1}{2} \left(\frac{\xi}{\Delta x} + \frac{3}{2}\right)^2 - \frac{1}{8}\right) d\xi d\eta \\ &+ \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_0^{2\Delta x} A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) \left(\frac{1}{2} \left(\frac{\xi}{\Delta x} - \frac{3}{2}\right)^2 - \frac{1}{8}\right) d\xi d\eta. \end{aligned} \quad (\text{B.16})$$

The next challenge is finding explicit expressions for these integrals. Integrating the  $A_{ux} \cdot 1$  term has been done already in [14] as this is required for the piece-wise constant approximation. The difficulty when finding an antiderivative for  $A_{ux} \cdot \xi$  and  $A_{ux} \cdot \xi^2$  is that  $A_{ux}$  contains terms  $(x_I - x_J - \xi)$ , which are then multiplied by plain  $\xi$ 's. Therefore note that:

$$\xi^2 = (x_I - x_J - \xi)^2 - 2(x_I - x_J)(x_I - x_J - \xi) + (x_I - x_J)^2. \quad (\text{B.17})$$

In the integral the term  $(x_I - x_J)$  is just a constant so we can rewrite  $1 - \left(\frac{\xi}{\Delta x}\right)^2$  into terms with  $(x_I - x_J - \xi)$  and a constant term as:

$$\begin{aligned} 1 - \left(\frac{\xi}{\Delta x}\right)^2 &= 1 - \frac{(x_I - x_J - \xi)^2}{(\Delta x)^2} + \frac{2(x_I - x_J)}{(\Delta x)^2} (x_I - x_J - \xi) - \frac{(x_I - x_J)^2}{(\Delta x)^2} \\ &= \frac{1}{(\Delta x)^2} \left( (\Delta x)^2 - (x_I - x_J)^2 + 2(x_I - x_J)(x_I - x_J - \xi) - (x_I - x_J - \xi)^2 \right). \end{aligned} \quad (\text{B.18})$$

In a similar fashion we deal with the terms  $\frac{1}{2} \left( \frac{\xi}{\Delta x} + \frac{3}{2} \right)^2 - \frac{1}{8}$  and  $\frac{1}{2} \left( \frac{\xi}{\Delta x} - \frac{3}{2} \right)^2 - \frac{1}{8}$  in order to rewrite them as:

$$\begin{aligned} \frac{1}{2} \left( \frac{\xi}{\Delta x} + \frac{3}{2} \right)^2 - \frac{1}{8} &= \frac{\xi^2}{2(\Delta x)^2} + \frac{3\xi}{2\Delta x} + 1 & (\text{B.19}) \\ &= \frac{(x_I - x_J - \xi)^2 - 2(x_I - x_J)(x_I - x_J - \xi) + (x_I - x_J)^2}{2(\Delta x)^2} + \frac{3(x_I - x_J) - 3(x_I - x_J - \xi)}{2\Delta x} + 1 \\ &= \frac{(2(\Delta x)^2 + 3\Delta x(x_I - x_J) + (x_I - x_J)^2 - (3\Delta x + 2(x_I - x_J))(x_I - x_J - \xi) + (x_I - x_J - \xi)^2)}{2(\Delta x)^2} \end{aligned}$$

$$\begin{aligned} \frac{1}{2} \left( \frac{\xi}{\Delta x} - \frac{3}{2} \right)^2 - \frac{1}{8} &= \frac{\xi^2}{2(\Delta x)^2} - \frac{3\xi}{2\Delta x} + 1 & (\text{B.20}) \\ &= \frac{(x_I - x_J - \xi)^2 - 2(x_I - x_J)(x_I - x_J - \xi) + (x_I - x_J)^2}{2(\Delta x)^2} - \frac{3(x_I - x_J) - 3(x_I - x_J - \xi)}{2\Delta x} + 1 \\ &= \frac{(2(\Delta x)^2 - 3\Delta x(x_I - x_J) + (x_I - x_J)^2 + (3\Delta x - 2(x_I - x_J))(x_I - x_J - \xi) + (x_I - x_J - \xi)^2)}{2(\Delta x)^2} \end{aligned}$$

Recall from (B.3) that we have to find explicit expressions for:

$$\iint \frac{(x_I - x_J - \xi)^n}{\rho} d\xi d\eta, \quad (\text{B.21})$$

$$\iint \frac{(x_I - x_J - \xi)^{n+2}}{\rho^3} d\xi d\eta, \quad (\text{B.22})$$

for  $n = 0, 1, 2$  where  $\rho = \sqrt{(x_I - x_J - \xi)^2 + (y_I - y_J - \eta)^2}$ .

The integration is performed using the software package **Maxima**, a continuation of the package **Macsyma**.

This gives the following antiderivatives when setting the integration constant to zero:

$$\begin{aligned} F_{01}(x, y, \xi, \eta) &:= \iint \frac{1}{\sqrt{(x-\xi)^2 + (y-\eta)^2}} d\xi d\eta & (B.23) \\ &= (x-\xi) \sinh^{-1} \left( \frac{(y-\eta)}{|(x-\xi)|} \right) + (y-\eta) \sinh^{-1} \left( \frac{(x-\xi)}{|(y-\eta)|} \right), \end{aligned}$$

$$\begin{aligned} F_{11}(x, y, \xi, \eta) &:= \iint \frac{(x-\xi)}{\sqrt{(x-\xi)^2 + (y-\eta)^2}} d\xi d\eta & (B.24) \\ &= \frac{1}{2} \left[ (x-\xi)^2 \sinh^{-1} \left( \frac{(y-\eta)}{|(x-\xi)|} \right) + (y-\eta) \sqrt{(x-\xi)^2 + (y-\eta)^2} \right], \end{aligned}$$

$$\begin{aligned} F_{21}(x, y, \xi, \eta) &:= \iint \frac{(x-\xi)^2}{\sqrt{(x-\xi)^2 + (y-\eta)^2}} d\xi d\eta & (B.25) \\ &= \frac{1}{6} \left[ (\eta^3 - 3y\eta^2 + 3y^2\eta - y^3) \sinh^{-1} \left( \frac{x-\xi}{|y-\eta|} \right) \right. \\ &\quad \left. + 2(x-\xi)^3 \sinh^{-1} \left( \frac{y-\eta}{|x-\xi|} \right) + (x-\xi)(y-\eta) \sqrt{(x-\xi)^2 + (y-\eta)^2} \right], \end{aligned}$$

$$\begin{aligned} F_{23}(x, y, \xi, \eta) &:= \iint \frac{(x-\xi)^2}{\sqrt{(x-\xi)^2 + (y-\eta)^2}^3} d\xi d\eta & (B.26) \\ &= (y-\eta) \sinh^{-1} \left( \frac{(x-\xi)}{|(y-\eta)|} \right), \end{aligned}$$

$$\begin{aligned} F_{33}(x, y, \xi, \eta) &:= \iint \frac{(x-\xi)^3}{\sqrt{(x-\xi)^2 + (y-\eta)^2}^3} d\xi d\eta & (B.27) \\ &= (y-\eta) \sqrt{(x-\xi)^2 + (y-\eta)^2}, \end{aligned}$$

$$\begin{aligned} F_{43}(x, y, \xi, \eta) &:= \iint \frac{(x-\xi)^4}{\sqrt{(x-\xi)^2 + (y-\eta)^2}^3} d\xi d\eta & (B.28) \\ &= \frac{1}{2} \left[ (\eta^3 - 3y\eta^2 + 3y^2\eta - y^3) \sinh^{-1} \left( \frac{x-\xi}{|y-\eta|} \right) \right. \\ &\quad \left. + (x-\xi)(y-\eta) \sqrt{(x-\xi)^2 + (y-\eta)^2} \right]. \end{aligned}$$

Combining all results again into (B.15) gives for the even  $J$ :

$$\begin{aligned} (\Delta x)^2 A_{IuJx}^0 &= ((\Delta x)^2 - (x_I - x_J)^2) \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_{-\Delta x}^{\Delta x} A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) d\xi d\eta & (B.29) \\ &\quad + 2(x_I - x_J) \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_{-\Delta x}^{\Delta x} (x_I - x_J - \xi) A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) d\xi d\eta \\ &\quad - \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_{-\Delta x}^{\Delta x} (x_I - x_J - \xi)^2 A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) d\xi d\eta, \end{aligned}$$

and into (B.16) gives for the odd  $J$ :

$$\begin{aligned}
2(\Delta x)^2 A_{IuJx}^+ &= (2(\Delta x)^2 + 3\Delta x(x_I - x_J) + (x_I - x_J)^2) \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_{-\Delta x}^0 A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) d\xi d\eta \\
&\quad - (3\Delta x + 2(x_I - x_J)) \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_{-\Delta x}^0 (x_I - x_J - \xi) A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) d\xi d\eta \\
&\quad + \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_{-\Delta x}^0 (x_I - x_J - \xi)^2 A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) d\xi d\eta \\
&\quad + (2(\Delta x)^2 - 3\Delta x(x_I - x_J) + (x_I - x_J)^2) \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_0^{\Delta x} A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) d\xi d\eta \\
&\quad + (3\Delta x - 2(x_I - x_J)) \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_0^{\Delta x} (x_I - x_J - \xi) A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) d\xi d\eta \\
&\quad + \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_0^{\Delta x} (x_I - x_J - \xi)^2 A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) d\xi d\eta,
\end{aligned} \tag{B.30}$$

with:

$$A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) = \frac{1}{\pi G} \left( \frac{1 - \nu}{\rho} + \nu \frac{(x_I - x_J - \xi)^2}{\rho^3} \right). \tag{B.31}$$

In order to shorten the notation a bit we write:

$$\mathcal{F}(F_{nm}, x, y) = F_{nm} \left( x, y, \Delta x, \frac{\Delta y}{2} \right) - F_{nm} \left( x, y, -\Delta x, \frac{\Delta y}{2} \right) \tag{B.32}$$

$$- F_{nm} \left( x, y, \Delta x, -\frac{\Delta y}{2} \right) + F_{nm} \left( x, y, -\Delta x, -\frac{\Delta y}{2} \right),$$

$$\mathcal{F}^-(F_{nm}, x, y) = F_{nm} \left( x, y, 0, \frac{\Delta y}{2} \right) - F_{nm} \left( x, y, -2\Delta x, \frac{\Delta y}{2} \right) \tag{B.33}$$

$$- F_{nm} \left( x, y, 0, -\frac{\Delta y}{2} \right) + F_{nm} \left( x, y, -2\Delta x, -\frac{\Delta y}{2} \right),$$

$$\mathcal{F}^+(F_{nm}, x, y) = F_{nm} \left( x, y, 2\Delta x, \frac{\Delta y}{2} \right) - F_{nm} \left( x, y, 0, \frac{\Delta y}{2} \right) \tag{B.34}$$

$$- F_{nm} \left( x, y, 2\Delta x, -\frac{\Delta y}{2} \right) + F_{nm} \left( x, y, 0, -\frac{\Delta y}{2} \right).$$

Now the explicit expressions for the discretised influence functions become:

$$(\Delta x)^2 \pi G A_{IuJx} = ((\Delta x)^2 - (x_I - x_J)^2). \quad (\text{B.35})$$

$$\begin{aligned} & ((1 - \nu)\mathcal{F}(F_{01}, x_I - x_J, y_I - y_J) + \nu\mathcal{F}(F_{23}, x_I - x_J, y_I - y_J)) \\ & + 2(x_I - x_J) ((1 - \nu)\mathcal{F}(F_{11}, x_I - x_J, y_I - y_J) + \nu\mathcal{F}(F_{33}, x_I - x_J, y_I - y_J)) \\ & - ((1 - \nu)\mathcal{F}(F_{21}, x_I - x_J, y_I - y_J) + \nu\mathcal{F}(F_{43}, x_I - x_J, y_I - y_J)), \end{aligned}$$

$$2(\Delta x)^2 \pi G A_{IuJx}^+ = (2(\Delta x)^2 + 3\Delta x(x_I - x_J) + (x_I - x_J)^2). \quad (\text{B.36})$$

$$\begin{aligned} & ((1 - \nu)\mathcal{F}^-(F_{01}, x_I - x_J, y_I - y_J) + \nu\mathcal{F}^-(F_{23}, x_I - x_J, y_I - y_J)) \\ & - (3\Delta x + 2(x_I - x_J)) \cdot \\ & ((1 - \nu)\mathcal{F}^-(F_{11}, x_I - x_J, y_I - y_J) + \nu\mathcal{F}^-(F_{33}, x_I - x_J, y_I - y_J)) \\ & + ((1 - \nu)\mathcal{F}^-(F_{21}, x_I - x_J, y_I - y_J) + \nu\mathcal{F}^-(F_{43}, x_I - x_J, y_I - y_J)) \\ & + (2(\Delta x)^2 - 3\Delta x(x_I - x_J) + (x_I - x_J)^2) \cdot \\ & ((1 - \nu)\mathcal{F}^+(F_{01}, x_I - x_J, y_I - y_J) + \nu\mathcal{F}^+(F_{23}, x_I - x_J, y_I - y_J)) \\ & + (3\Delta x - 2(x_I - x_J)) \cdot \\ & ((1 - \nu)\mathcal{F}^+(F_{11}, x_I - x_J, y_I - y_J) + \nu\mathcal{F}^+(F_{33}, x_I - x_J, y_I - y_J)) \\ & + ((1 - \nu)\mathcal{F}^+(F_{21}, x_I - x_J, y_I - y_J) + \nu\mathcal{F}^+(F_{43}, x_I - x_J, y_I - y_J)). \end{aligned}$$

## Leading Edge influence coefficient

We have  $N$  constant basisfunctions for each element around the  $x_I$  in the contact area, and the basisfunction:

$$\varphi_{N+1}(x) = \begin{cases} 2\frac{x_N-x}{\Delta x} & : x_N - \frac{\Delta x}{2} < x < x_N + \frac{\Delta x}{2} \\ 0 & : \text{else} \end{cases} \quad (\text{C.1})$$

We now approximate the tractionfield with  $p_x$  given by:

$$p_x(x, y) = \sum_{I=1}^{N+1} p_{Ix} \varphi_I(x, y). \quad (\text{C.2})$$

The  $p_{Ix}$  are the coefficients to be determined. We can substitute the discretized traction (B.7) into (6.2) and find:

$$u_x(x, y, 0) = \iint_C A_{ux}(x, y, 0, \xi, \eta) \sum_J \{p_{Jx} \varphi_J(\xi, \eta)\} d\xi d\eta. \quad (\text{C.3})$$

Following the same steps as in the previous appendix, this leads to the influence coefficient equations:

$$A_{IuJx} = \iint_C A_{ux}(x_I, y_I, 0, \xi, \eta) \varphi_J(\xi, \eta) d\xi d\eta. \quad (\text{C.4})$$

These integrals have been calculated already for the constant basisfunctions. So only the case  $J = N+1$  will be evaluated here. Knowing where  $\varphi_{N+1}$  is nonzero we can write:

$$A_{Iu(N+1)x} = \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_{x_N - \frac{\Delta x}{2}}^{x_N + \frac{\Delta x}{2}} A_{ux}(x_I, y_I, 0, \xi, \eta) \left(2\frac{x_N - \xi}{\Delta x}\right) d\xi d\eta \quad (\text{C.5})$$

The next challenge is finding explicit expressions for these integrals. Integrating the  $A_{ux} \cdot 1$  term has been done already in [14] as this is required for the piece-wise constant approximation. The difficulty when finding an antiderivative for  $A_{ux} \cdot \xi$  is that  $A_{ux}$  contains terms  $(x_I - \xi)$ , which are then multiplied by plain  $\xi$ 's. Therefore note that:

$$\xi = x_I - x_I + \xi = x_I - (x_I - \xi). \quad (\text{C.6})$$

In the integral the term  $x_I$  is just a constant so we can rewrite  $2\frac{x_N - \xi}{\Delta x}$  into terms with  $(x_I - \xi)$  and a constant term as:

$$2\frac{x_N - \xi}{\Delta x} = 2\frac{x_N - (x_I - (x_I - \xi))}{\Delta x} = 2\frac{x_N - x_I}{\Delta x} + 2\frac{x_I - \xi}{\Delta x}. \quad (\text{C.7})$$

Recall from (B.3) that we have to find explicit expressions for:

$$\iint \frac{x_I - \xi}{\rho} d\xi d\eta, \quad (\text{C.8})$$

$$\iint \frac{(x_I - \xi)^3}{\rho^3} d\xi d\eta, \quad (\text{C.9})$$

where  $\rho = \sqrt{(x_I - \xi)^2 + (y_I - \eta)^2}$ .

The integration has already been performed in Section 6.2 and the same equations (6.20)-(6.23) are used.

Combining all results again into (C.5) gives:

$$\begin{aligned} A_{Iu(N+1)x} &= 2 \frac{x_N - x_I}{\Delta x} \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_{x_N - \frac{\Delta x}{2}}^{x_N + \frac{\Delta x}{2}} A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) d\xi d\eta \\ &+ \frac{2}{\Delta x} \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_{x_N - \frac{\Delta x}{2}}^{x_N + \frac{\Delta x}{2}} (x_I - \xi) A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) d\xi d\eta, \end{aligned} \quad (\text{C.10})$$

with:

$$A_{ux}(x_I - x_J, y_I - y_J, 0, \xi, \eta) = \frac{1}{\pi G} \left( \frac{1 - \nu}{\rho} + \nu \frac{(x_I - x_J - \xi)^2}{\rho^3} \right). \quad (\text{C.11})$$

In order to shorten the notation a bit we write (for  $n = 0..3$ ):

$$\begin{aligned} \mathcal{F}(F_n, x, y) &= F_n \left( x, y, x_N + \frac{\Delta x}{2}, \frac{\Delta y}{2} \right) - F_n \left( x, y, x_N - \frac{\Delta x}{2}, \frac{\Delta y}{2} \right) \\ &- F_n \left( x, y, x_N + \frac{\Delta x}{2}, -\frac{\Delta y}{2} \right) + F_n \left( x, y, x_N - \frac{\Delta x}{2}, -\frac{\Delta y}{2} \right). \end{aligned} \quad (\text{C.12})$$

Now the explicit expression for  $A_{Iu(N+1)x}$  becomes:

$$\begin{aligned} \pi G A_{Iu(N+1)x} &= 2 \frac{x_N - x_I}{\Delta x} \left( (1 - \nu) \mathcal{F}^-(F_0, x_I, y_I) + \nu \mathcal{F}^-(F_2, x_I, y_I) \right) \\ &+ \frac{2}{\Delta x} \left( (1 - \nu) \mathcal{F}^-(F_1, x_I, y_I) + \nu \mathcal{F}^-(F_3, x_I, y_I) \right), \end{aligned} \quad (\text{C.13})$$

# Bibliography

- [1] M.P. Allen and D.J. Tildesley. *Computer Simulation of Liquids*. Clarendon press, Oxford, 1987.
- [2] E.M.L. Beale. On quadratic programming. *Naval Research and Logistics Quarterly*, 6:227–243, 1959.
- [3] J. Boussinesq. *Application des Potentiels à l'Étude de l'Équilibre et du Mouvement des Solides Élastiques*. Gauthier-Villars, Paris, 1885.
- [4] V. Cerruti. Ricerche intorno all'equilibrio dei corpi elastici isotropi. *Reale Accademia dei Lincei*, 13, 1882.
- [5] C.A. Coulomb. *Théorie des machines simples*. Mémoire de Mathématique et de Physique de l'Académie Royale, 1821.
- [6] D. den Hertog, C. Roos, and T. Terlaky. A polynomial method of weighted centers for convex quadratic programming. *J. of Information and Optimization Sciences*, 12(2):187–205, 1991.
- [7] J.R. Dydo and H.R. Busby. Elasticity solutions for constant and linearly varying loads applied to a rectangular surface patch on the elastic half-space. *Journal of Elasticity*, 38:153–163, 1995.
- [8] R.D. Groot and P.B. Warren. Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. *The Journal of Chemical Physics*, 107(11):4423–4435, 1997.
- [9] H. Hertz. Über die Berührung fester elastischer Körper. *Journal für reine und angewandte Mathematik*, 92:156–171, 1882.
- [10] K.L. Johnson. *Contact Mechanics*. Cambridge University Press, Cambridge (UK), 1985.
- [11] J.J. Kalker. Variational principles of contact elastostatics. *Journal of the Institute of Mathematics and its Applications*, 20:199–219, 1977.
- [12] J.J. Kalker. The computation of three-dimensional rolling contact with dry friction. *Int. Journ. for Numerical Methods in Engineering*, 14:1293–1307, 1979.
- [13] J.J. Kalker. Introduction of the Fortran IV programs Duvorol and Contact for the solution of 3 D elastostatic half-space contact problems with and without friction. Technical Report 82-29, Delft University of Technology, Department of Mathematics and Informatics, 1982.
- [14] J.J. Kalker. *Three-Dimensional Elastic Bodies in Rolling Contact*. Solid Mechanics and its Applications. Kluwer Academic Publishers, Dordrecht, Netherlands, 1990.
- [15] M.K. Kozlov, S.P. Tarasov, and L.G. Khachiyan. The polynomial solvability of convex quadratic programming. *USSR Computational Mathematics and Mathematical Physics*, 20(5):223 – 228, 1980.
- [16] J. Li and E.J. Berger. A Boussinesq-Cerruti solution set for constant and linear distribution of normal and tangential load over a triangular area. *Journal of Elasticity*, 63:137–151, 2001.
- [17] A.E.H. Love. *A Treatise on the Mathematical Theory of Elasticity, 4th ed.* University Press, Cambridge, 1952.
- [18] K.G. Murty. *Linear complementarity, linear and nonlinear programming*. Heldermann Verlag, Berlin, 1988.

- [19] A.L. Schwab and C.M. Kalker-Kalkman. Joost J. Kalker (1933-2006): A life in rolling contact. *Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, September 4-7 2007.
- [20] O.J. Svec and G.M.L. Gladwell. An explicit Boussinesq solution for a polynomial distribution of pressure over a triangular region. *Journal of Elasticity*, 1:167–170, 1971.
- [21] P. Tseng. A simple polynomial-time algorithm for convex quadratic programming. Technical report, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Oktober 1988.
- [22] E.A.H. Vollebregt. A Gauss-Seidel type solver for special convex programs, with application to frictional contact mechanics. *Journal of Optimization Theory and Applications*, 87(1):47–67, 1995.
- [23] E.A.H. Vollebregt. Refinement of Kalker’s rolling contact model. In A. Bracciali, editor, *Proceedings of the 8th International Conference on Contact Mechanics and Wear of Rail/Wheel Systems*, pages 149–156, Firenze, Italy, 2009.
- [24] E.A.H. Vollebregt. m09002: Integration of the Boussinesq-Cerruti kernels for bilinear elements. VORtech, 2011.
- [25] E.A.H. Vollebregt. A new solver for the elastic normal contact problem using conjugate gradients, deflation, and an FFT-based preconditioner. *J. of Computational Physics*, 257, Part A:333–351, 2014.
- [26] E.A.H. Vollebregt. Numerical modeling of measured railway creep versus creep-force curves with CONTACT. *Wear*, 314:87–95, 2014.
- [27] E.A.H. Vollebregt. User guide for CONTACT, Rolling and sliding contact with friction. Technical Report TR09-03, version 16.1, VORtech, 2016. See [www.kalkersoftware.org](http://www.kalkersoftware.org).
- [28] E.A.H. Vollebregt. User guide for CONTACT, Rolling and sliding contact with friction. Technical Report TR09-03, version 16.1, VORtech, 2016. See [www.kalkersoftware.org](http://www.kalkersoftware.org).
- [29] E.A.H. Vollebregt and H.M. Schuttelaars. Quasi-static analysis of 2-dimensional rolling contact with slip-velocity dependent friction. *J. of Sound and Vibration*, 331(9):2141–2155, 2012. doi:10.1016/j.jsv.2012.01.011.
- [30] E.A.H. Vollebregt and A. Segal. Solving conformal wheel-rail rolling contact problems. *Vehicle System Dynamics*, 51(suppl. 1):455–468, 2014. URL <http://dx.doi.org/10.1080/00423114.2014.906634>.
- [31] E.A.H. Vollebregt, J.J. Kalker, and H.X. Lin. A fast solver for normal and tangential contact mechanics in the half-space. In M.H. Aliabadi and C.A. Brebbia, editors, *Contact Mechanics, Computational Techniques*, Southampton, 1993. Computational Mechanics Publications.
- [32] C.D. van der Wekken, E.A.H. Vollebregt, and C. Vuik. Occurrence and removal of wiggles in transient rolling contact simulation. In J. Ambrósio, W. Schielen, and J. Pombo, editors, *Proceedings of EuroMech colloquium 578, Rolling Contact Mechanics for Multibody System Dynamics*, pages 1–11, Lisbon, Portugal, 2017. IDMEC.
- [33] T.J. Ypma. Historical development of the Newton-Raphson method. *Society for Industrial and Applied Mathematics Review*, 37(4):531–551, 1995.
- [34] J. Zhao, E.A.H. Vollebregt, and C.W. Oosterlee. A fast nonlinear conjugate gradient based method for 3D concentrated frictional contact problems. *Journal of Computational Physics*, 288:86–100, 2015.