Simulation of steel-concrete bond-slip with sequentially linear analysis using interface elements

Sebastiaan Ensink

May 2010

Preface

The report at hand is for my Msc Thesis at the faculty of Civil Engineering and Geosciences, department of Computational Modeling of Structures at Delft University of Technology.

I would very much like to thank Professor Jan Rots for giving me the opportunity on working on such an interesting topic. Furthermore I would especially like to thank Max Hendriks for the endless support and guidance he has provided which has been of great help to me. I would also like to thank Joop den Uijl for the interesting talks about concrete sciences and giving me a constant reminder that things are usually more complicated then I could imagine. I also want to thank Anne van de Graaf for all the help and support he has given for writing the Fortan code and helping out with numerous other problems.

I hope you will enjoy reading this report.

Den Hoorn, May 2010

Examination committee:

Prof.dr.ir. J.G. Rots
Dr.ir. M.A.N. Hendriks
Ir. J.A. den Uijl
A.V. v.d. Graaf Msc

Contents

Preface **3**

1 Introduction **9**

1.1 General introduction to thesis 10

1.2 Steel-concrete bond-slip theory 11

1.2.1 The bond mechanism 11

1.2.2 Factors influencing the bond strength 12

1.2.3 Bond-slip experiments 13

1.2.4 Bond-slip relations 14

1.3 Computational modeling of bond-slip 15

1.4 Main objectives thesis 16

2 The bond-slip model by Den Uijl and Bigaj **17**

2.1 Introduction 18

2.2 General description 19

2.3 Parameters used 20

2.4 Calculation of stages I, II and III 21

2.5 Calculation of splitting failure 21

2.6 Calculation of pull-out failure 22

2.7 Results for splitting failure 24

2.8 Results for pull-out failure 25

3 Extensions to an SLA software implementation **27**

3.1 Introduction 28

3.2 Program overview 28

3.3 Axi-symmetric elements 29

3.4 Interface elements 30

3.5 Cable (truss) elements 32

3.6 Post processing 32

4 Sawtooth approximation and sawtooth generators **33**

4.1 Introduction 34

4.2 Sawtooth modeling of Hordijk tension softening 34

4.2.1 Nonlinear formulation 34

4.2.2 Automatic sawtooth generator 35

4.3 Sawtooth modeling of Den Uijl and Bigaj bond-slip model 38

4.4 Sawtooth modeling of steel plasticity 39

5 Nonlinear calculations of the tension-pull experiment by Gijsbers 41

5.1 Introduction 42
5.2 Concrete crack model 42
5.3 Interface bond-slip model 43
5.4 Modeling of imperfections 44
5.5 Axi-symmetric calculation 46
5.5.1 Model setup 46
5.5.2 Results 47
5.6 3D calculation 50
5.6.1 Model setup 50
5.6.2 Results 51
5.7 Modeling and implementation of alternative bond-slip unloading schemes 53
5.7.1 Unloading in the axi-symmetric calculation 53
5.7.2 Programming user interface 55
5.7.3 Experimental results from cyclic tests 56
5.7.4 Results of alternative unloading schemes 57
5.8 Summary and conclusions chapter 5 59

6 SLA calculations of the tension-pull experiment by Gijsbers 61

6.1 Introduction 62
6.2 Adopted sawtooth models 62
6.3 Results using imperfections 64
6.4 Results without imperfections 67
6.5 Reducing number of teeth 71
6.6 Changing the shear retention behaviour 75
6.7 Using mesh refinements 78
6.7.1 Used refinements 79
6.7.2 Results 79
6.8 Summary and conclusions chapter 6 83

7 SLA calculations of the tension-pull experiments by Mayer 87

7.1 Introduction 88
7.2 Discretization options 89
7.3 Overview of calculated experiments 89
7.4 Material properties 91
7.5 Bond-slip sawtooth diagrams 92
7.6 Results tension-pull experiments 93
7.6.1 Influence of reinforcement ratio 93
7.6.2 Influence of concrete type 95
7.6.3 Influence of rebar diameter 97
7.6.4 Stress-average strain plots 100
7.6.5 Overview of results, number of cracks and crack spacing . 101
7.7 Summary and conclusions chapter 7 102

8 SLA calculation of a beam in bending experiment by Walraven 105
8.1 Introduction 106
8.2 Results using full bond 108
8.3 Results with bond-slip 109
8.4 Load-displacement curve 110

9 Conclusions and recommendations 111
9.1 Conclusions 112
9.2 Guidelines for simulating tension-pull experiments 114
9.3 Recommendations 116

Summary 117

A Matlab bond model: code 123

B Matlab bond model: verification 139

C Bond model: calculation of stage III 147

D Bond model: pull-out failure, derivation of radial strain - slip relations 151

E Derivation of constitutive matrix for axi-symmetric elements in SLA 159

F Verification of SLA code for axi-symmetric elements 163

G Overview of routines for SLA software extensions 165

H Matlab sawtooth generator for Hordijk tension softening: code 167

I Nonlinear axi-symmetric calculation: *DIANA* program settings 175

J *DIANA* User interface models *BOND3N, BOND3S* and *BOND3E* code and verification 177

K Nonlinear 3D calculation: *DIANA* program settings 185

L Sawtooth diagrams used in section 6.5 187

M Sawtooth diagrams used in section 6.7 191

N Sawtooth diagrams used in chapter 7 195

O Mayer S4D12-A (B25) 199

P Mayer S8D12-A (B25) 201

Q Mayer S8D12-A (B45) 205

R Mayer S4D16-A (B25) 209

S Mayer S6D16-A (B25) 211

T Mayer S2D25-A (B25) 215

U Mayer S4D25-A (B25) 219

Chapter 1

Introduction

1.1 General introduction to thesis

In reinforced concrete structures cracking and crack width is controlled by the layout and the amount of the reinforcement. Being able to predict the crack width and spacing is important for a number of reasons for instance with regard to durability or structures that need to be impermeable to fluids. Of strong influence to the cracking behaviour of reinforced concrete structures is the bond that exist between the steel and the concrete. This bond is however a complex phenomenon and cannot be readily predicted. With experiments a bond-slip relation can be determined and with testing of long specimens it is also possible to determine the crack spacing.
This thesis sets out to explore the possibilities of simulating steel-concrete bond-slip behaviour numerically. For numerical simulations conventional non-linear analysis using the finite element method can be used to predict and calculate cracking and crack development in reinforced concrete structures. In these simulations the bond between the steel and the concrete can be taken into account in various ways. Often however the bond is not taken into account and instead full bond is assumed.
A recent development at Delft University is the use of sequentially linear analysis or SLA for short. This method is an alternative to non-linear analysis and has already showed good results for simulating cracking behaviour in brittle materials such as concrete, glass and masonry. This thesis will explore the possibilities of using SLA with the object of simulating bond-slip behaviour. Of special interest are the aforementioned experiments on long specimens since these tests reveal a fully developed crack pattern. Therefore the possibilities to simulate these experiments using conventional non-linear analysis as well as SLA will be explored.

First the general bond-slip theory will be described in the next section. In section 1.4 the main objectives for this thesis will be stated.

1.2 Steel-concrete bond-slip theory

1.2.1 The bond mechanism

There are many sources in literature that describe the bond mechanism in detail [1], [7], [9]. Here only a basic description will be given to describe the most important aspects of the bond mechanism between concrete and reinforcement. The description given here applies to reinforcing bars with ribs only.
The bond mechanism consists of three main components: (chemical) adhesion, friction and mechanical interlocking (bearing), see figure 1.1. The adhesion (a) is very small and lost almost immediately upon loading. The mechanical interlocking can also be viewed as friction. The inclined bearing forces (c) can be divided into two components: the longitudinal part, this is called the bond stress, and the radial part which is called the splitting stress. The inclined forces which radiate outwards into the surrounding concrete are balanced by ring tensile stresses see figure 1.2. If the ring tensile stresses become too large the concrete will crack resulting in longitudinal splitting cracks.

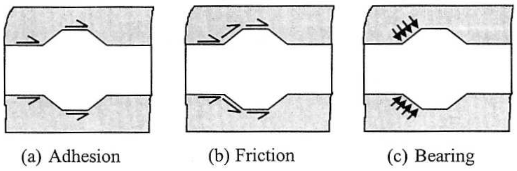


Figure 1.1: Force transfer mechanism

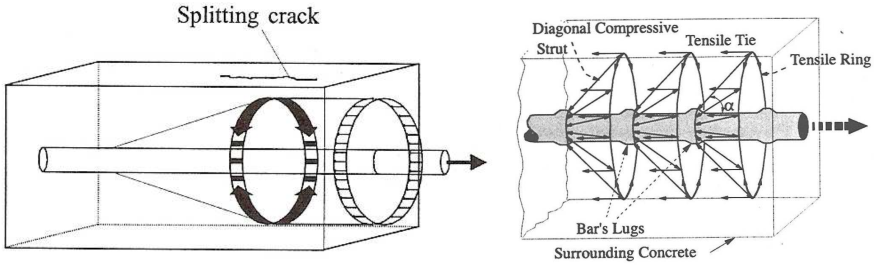


Figure 1.2: Ring tensile stresses

There are two distinct bond failure mechanisms:

- Splitting failure.
- Pull-out failure.

Splitting failure will occur when splitting cracks can reach an outer surface before the bar is pulled out, for instance when there is only a small concrete cover. In this case the normal stresses between the concrete and the rebar are reduced drastically causing much lower bond stress transfer.
The other failure mechanism, pull-out failure, will occur if the confinement (i.e. the concrete cover or transverse reinforcement) around the bar is sufficient to withstand (balance) the splitting stresses. In this case bond failure is ultimately caused by shearing off of the concrete between the rebar and the ribs. Figure 1.3 explains

the difference in load transfer between pull-out and splitting failure (taken from [8]).

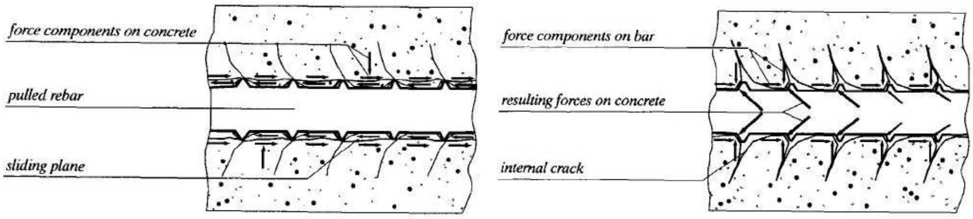


Figure 1.3: Bond pull-out failure (left) and bond splitting failure (right)

The difference in bond-slip behaviour in terms of a bond stress-slip relation is illustrated in figure 1.4 below.

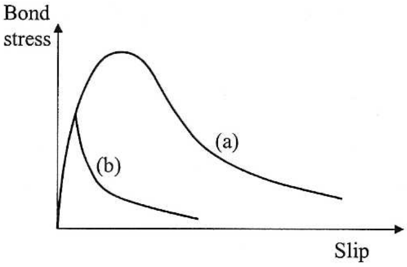


Figure 1.4: Difference in bond-slip behaviour, (a) pull-out failure, (b) splitting failure

1.2.2 Factors influencing the bond strength

Since bond performance is not only dependent on pure material properties but also on the structure surrounding the rebar there are many aspects influencing the bond performance.[1] Some factors are listed here (see also [7], [16]):

- concrete compressive strength, especially for pull-out failure;
- concrete tensile strength, especially for splitting failure;
- level of confinement (active or passive), i.e. concrete cover or transverse reinforcement;
- size effects;
- concrete type (HSC, NSC, FRC);
- rebar geometry (diameter, relative rib area);
- type of loading, load repetition (load history).

[1] this also implies bond performance to be non-unique meaning it can differ at different locations along the rebar

1.2.3 Bond-slip experiments

Experiments on reinforced concrete specimens involving bond-slip can be used among other things to derive bond stress-slip relations. This section very shortly describes several types of experimental setups that are most commonly used and is mainly taken from [7].

Short specimens (pull-out failure, see figure 1.5)
In this test a reinforcing bar with a very short bonded length is pulled out of a concrete body. Because of the short bonded length there exists an almost uniform distribution of the bond stress. Therefore this type of test can be done to obtain a bond-slip curve. The concrete cover is chosen large enough as to prevent splitting failure.

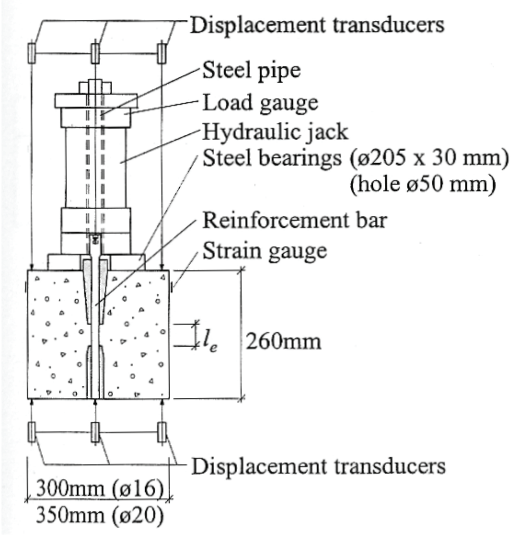


Figure 1.5: Example of short specimen pull-out test setup

Long specimens (pull-out and splitting failure)
For obtaining information about the bond-stress distribution (among other things) along a bar or splice[2], tests on long specimens have also been adopted. These tests can also serve as a reference for FEM modeling. Sometimes in this type of testing strain gages are inserted inside the reinforcing bar. This can be done by splitting the bar into two, channelling out for strain gage placement and then gluing the two halves back together. By varying the concrete cover this type of test can be used to investigate pull-out failure as well as bond splitting failure.

[2] Dutch: lasverbinding

1.2.4 Bond-slip relations

There are several ways for obtaining a bond stress-slip relation. For practical reasons one way is to use simple parametric formulae. This approach is for instance used in the CEB-FIB MC90 model code [7]. A bond-slip curve calculated with this model code is shown in figure 1.6 for confined concrete (i.e. pull-out failure) and unconfined concrete (i.e. splitting failure), both for a concrete compressive strength of $f'ck$=35 N/mm^2.

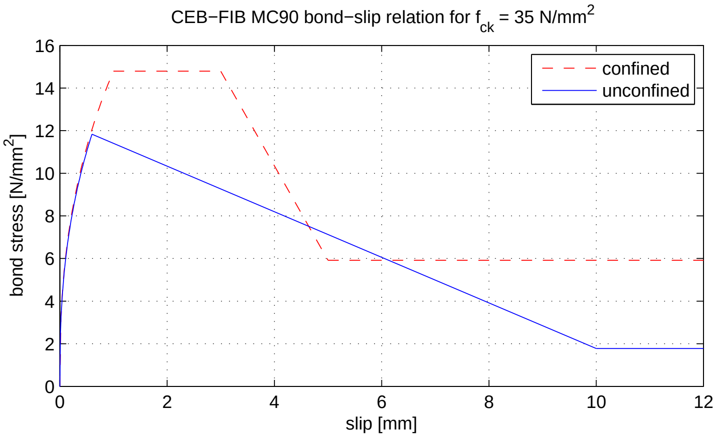


Figure 1.6: Example of a direct bond-slip relation (CEB-FIB MC90)

The parameters determining the CEB-FIB MC90 bond stress-slip curve are:

- confined/unconfined concrete;
- concrete compressive strength $f'ck$;
- clear rib spacing c_s (between 0.5Ø and 1.0Ø; Ø is the nominal bar diameter, for confined concrete only).

Some more recent improved versions of the CEB-FIB curve have also been derived showing for instance a descending branch in the residual stage [9]. Many other formulae and approaches exist in literature; a list of these will not be given here. Several approaches are also discussed in the CEB-FIB MC90 model code itself [7].

Instead of using a parametric formula a more sophisticated possibility is the calculation of a bond stress-slip curve using a bond model. A bond model is a way to calculate a bond stress-slip relation taking into account a large number of parameters depending on the type of model used (see also section 1.2.2). A bond model can incorporate material as well as structural properties to derive a bond-slip relation. An example of such a bond model is the one developed by Den Uijl and Bigaj [8]. This model will be discussed in detail in chapter 2.

1.3 Computational modeling of bond-slip

In general there are several ways to approach a bond-slip problem numerically using the finite elements method and non-linear fracture mechanics. This depends on the scale of the approach:

- **micro scale**
  With a micro scale approach a reinforcing bar (sometimes including the ribs) and the surrounding concrete is modelled. This approach is mainly directed to theoretical research and is not practical for large problems. With a micro scale approach it is possible to get a bond-slip relation as a result of the calculation see figure 1.7 and [2].

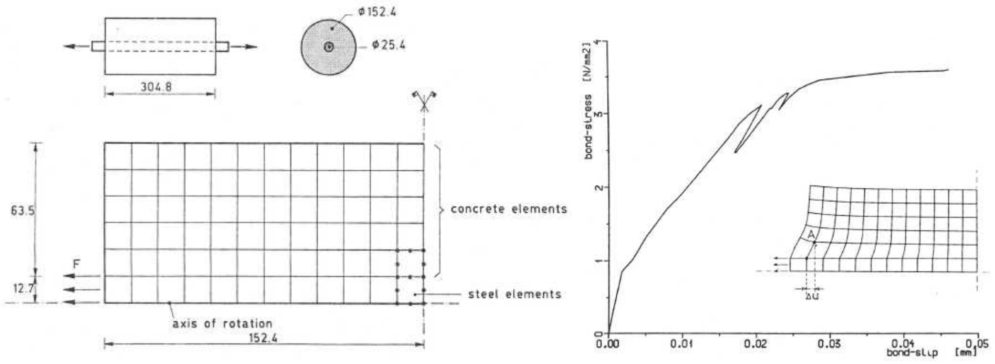


Figure 1.7: Example of micro approach, axi-symmetric finite element discretization (left) and derived bond-slip curve (right), taken from [2]

- **medium scale**
  With a medium scale approach all bond-slip behaviour is lumped into an interface layer thus making use of interface elements. These elements usually connect a truss element (i.e. the rebar) to a concrete continuum element (e.g. an axi-symmetric, plane stress or solid element). This approach can be applied to larger scale models like tension-pull experiments or beams.

- **large scale**
  For even larger structures also using interface elements becomes too cumbersome. At this scale bond-slip can be incorporated indirectly by using a tension-stiffening approach.

For the medium scale approach a definition is needed for slip. Figure 1.8 illustrates that slip “Δu” between the concrete and the reinforcement is measured at a certain distance “a” from the rebar. The slip is caused by a combination of elastic deformation of the concrete and the cracking and crushing of the concrete in the vicinity of the ribs [9].

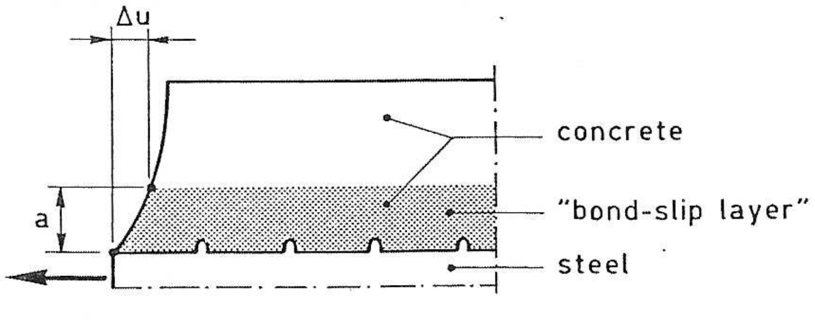


Figure 1.8: Definition of slip Δu (taken from [2])

1.4 Main objectives thesis

As stated in the general introduction section 1.1 this thesis sets out to explore the possibilities of simulating steel-concrete bond-slip behaviour numerically. In the preceding sections the general bond-slip theory was described.

The main objectives for this thesis can now be stated as:

- Research the possibilities of modeling bond-slip using the finite element method and sequentially linear analysis (SLA) in comparison to conventional nonlinear analysis.
- Research the possibilities for obtaining a bond-slip relation that can be used in the simulations.

The following restriction is used for this thesis:

- Research is directed to medium scale bond-slip simulation only (see section 1.3). This implies the bond-slip behaviour to be lumped into a bond-slip layer and interface elements will be used.

Chapter 2

The bond-slip model by Den Uijl and Bigaj

2.1 Introduction

In section 1.2.4 several possibilities for obtaining a bond-slip relation were mentioned. One way was the use of simple parametric formulae (e.g. with the CEB-FIB MC90 model code). The MC90 model code has some drawbacks since only a few parameters that influence the bond-slip relation are taken into account. Another disadvantage is the fact that there is no well defined criterion between splitting failure and pull-out failure (you have to decide for yourself which failure mode is governing and then use the appropriate relation). As was already mentioned in section 1.2.4 another possibility is the use of a so called bond model. In this chapter the bond model that was developed by Den Uijl & Bigaj in 1996 [8] will, to some extent, be described in detail. Besides a brief general description the emphasis will be put on the inner workings of parts of the model that were difficult to understand from [8]. In most literature that make use of this bond model a description of these parts is also lacking (e.g. [12],[13],[16]).

Although several programs of the bond model exist (e.g. an Excel version and a German program called “BATS bond” used in [12]) these programs are either not fully documented or do not generate a bond-slip relation numerically. For this reason and to get a better understanding of the workings of the bond model it is programmed in Matlab (for code see appendix A). This also has the advantage that it can be expanded with a sawtooth generator, which will be explained in section 4.3. After it was programmed the results were verified against results from [14], see appendix B.

The reason for choosing this bond model is the inclusion of all mayor influencing parameters on the local bond-slip relation. Also this model will be used as a basis for the new CEB-FIB MC90 model code that will be released later this year. It should be noted however that because of the nature of the concrete material even with sophisticated bond models in reality there will always be a great scatter in the actual bond strength along a reinforcing bar and under different conditions. Deviations in bond strength by as much as 25% are therefore a real possibility. For numerical simulations however the model is well suited to compare results of using different configurations (e.g. bar diameter, concrete strength etc.) to simulate there effect. For several tension-pull experiments this will be done in chapter 7.

2.2 General description

The starting point for the bond model is the stress state in the concrete surrounding the rebar. For this the concrete is modeled as a thick-walled cylinder where the concrete cover in 3 or 4 directions is taken into account to calculate an effective cover. The effective cover is then taken as the radius of the thick-walled cylinder, see figure 2.1 right. If a bar is pulled relative to the concrete the rebar ribs will be pushed against the concrete (figure 2.1 left). In the model this wedging effect is taken into account by conceiving the interface between the boundary layer and the surrounding concrete as conical (instead of discrete ribs), see figure 2.1 middle. The radial stress, acting perpendicularly to the interface, is the response of the surrounding concrete to the radial displacement at the interface. By assuming dry friction the bond stress is directly proportional to the radial stress.

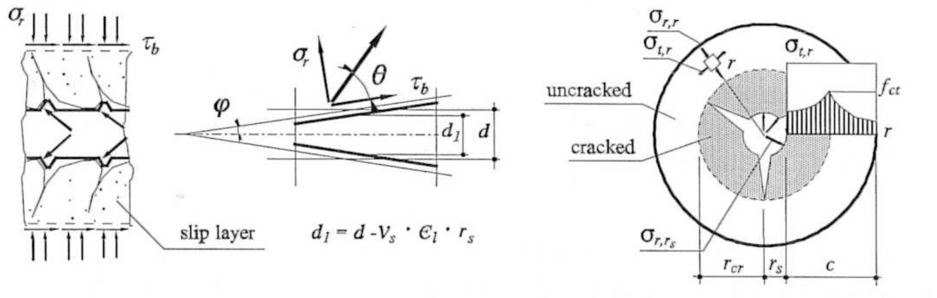


Figure 2.1: Definition of slip layer (left), assumption of conical interface and friction (middle) and concrete response to radial displacement of interface (right) taken from [14]

The stress state and the radial displacement (both at the interface) are calculated by assuming the thick-walled cylinder as having an internal pressure. Three stages are defined: stage I is the (linear elastic) uncracked stage. Here the radial displacement at the interface does not yet result in cracking of the cylinder. Stage II is the partly cracked stage meaning that the concrete will crack gradually from the center of the cylinder through the effective cover (figure 2.1 right). If the effective cover (i.e. the radius of the thick-walled cylinder) is large enough shearing off of the concrete between the rebar and the ribs will occur before the cracks are able to grow through the entire cover (see also section 1.2.1 figure 1.3). In this case the failure mechanism will be pull-out. If the effective cover is too small, the cracks can grow all the way through before an ultimate shear strength is reached at the interface. In this case the failure mechanism will be splitting (see also section 1.2.1 figure 1.3). Only with splitting failure there is also a stage III which represents the entirely cracked stage in which the crack width in the cylinder will continue to increase.

2.3 Parameters used

In the introduction (section 1.2.2) a general list of parameters that influence a bond-slip relation is given. The bond model by Den Uijl & Bigaj uses the following parameters to calculate a bond-slip relation:

- Concrete compressive strength;
- Concrete tensile strength (calculated from the concrete compressive strength);
- Concrete Young's modulus (calculated from the concrete compressive strength);
- Concrete Poisson's ratio;
- Bi-linear concrete softening curve parameters;
- Concrete cover (in 3 or 4 directions);
- Rebar diameter;
- Rebar strain;
- Coefficient of friction;
- Number of splitting cracks.

Besides the above-mentioned input parameters there is also a table of model parameters that were derived empirically (see [8] table 4.5). Among other things in this table the critical shear stress is for instance given as: τ_{b1}=5f_{ct}.

It should be noted that the bond model description and implementation for this thesis is based on the original bond model from 1996 [8]. Since then a further development of the model has taken place. The basics of the model have however not been changed. The later developments primarily focus on:

- More precise formulation of the bi-linear softening curve parameters (using more experimental data);
- Introduction of the effective rib area as an additional parameter [13];
- Using different parameters for rebar that is casted vertically or horizontally [13];
- Effective concrete cover for closely spaced rebar's and/or multiple layers [12].

The field of application for the bond model in this thesis is restricted to:

- one bar specimens;
- ribbed bars;
- no transverse reinforcement (the additional confinement is not taken into account);
- NSC or HSC.

2.4 Calculation of stages I, II and III

The figure below shows the response of the concrete cylinder in terms of the radial stress called σ_{r,r_s} and the radial strain ϵ_{r,r_s} at the interface (r_s is the radius of the rebar).[1] The latter is defined as: $\epsilon_{r,r_s} = \frac{u_{r,r_s}}{r_s}$ (the radial displacement at the interface is normalized with respect to the bar radius).

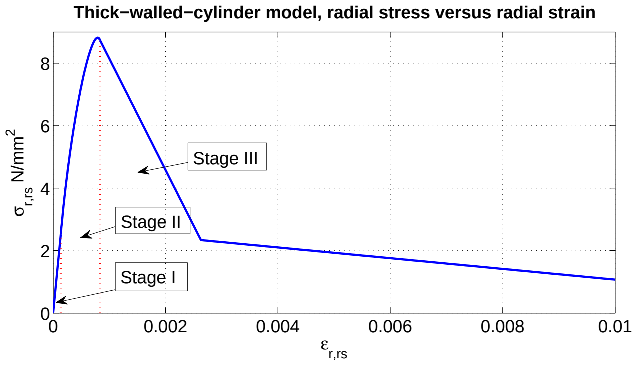


Figure 2.2: Confining capacity thick-walled-cylinder (d_s=12 mm, c=20 mm, f_{cc}=35 N/mm^2)

The maximum radial stress for stage I is equal to the concrete tensile strength. In stage II a parameter called the crack radius r_{cr} (which runs from the bar radius to the effective cover) is stepwise increased. For each crack radius (i.e. crack depth) the response of the thick-walled-cylinder is then calculated, where a summation is used of the radial stress (at the interface) caused by the cracked as well as the uncracked part of the cylinder. The maximum radial stress at the interface is reached after a crack radius equal to about 0.7 times the effective cover.
The calculation of the last stage, stage III, is fairly complicated. From the formulas stated in [8] it proved difficult to know exactly how the radial stress σ^{III}_{r,r_s} and strain ϵ^{III}_{r,r_s} are computed. For this stage a previously developed Excel sheet of the bond model is consulted and its derivations are transferred to the Matlab model. For a complete description of this stage see appendix C.

2.5 Calculation of splitting failure

As was mentioned in section 2.2 the bond stress is directly proportional to the radial stress:

$$\tau_b = \sigma_r cot(\phi) \tag{2.1}$$

where $cot(\phi)$ is the coefficient of friction (constant see [8] table 4.5).
For splitting failure the radial strain at the interface is directly proportional to the slip:

$$\epsilon_{r,r_s} r_s = \delta tan(\varphi) \Leftrightarrow \delta = \frac{\epsilon_{r,r_s} r_s}{tan(\varphi)} \tag{2.2}$$

[1] in the bond model the variables are designated in the subscript by the distance to the primary axis, i.e. the center of the cylinder

where φ is the cone angle between the cone surface and the bar axis (see figure 2.1 middle). An example of a bond-slip curve for splitting failure is given in figure 2.3. Because of the linear relationships the τ-δ curve for splitting failure is similarly shaped as the σ_{r,r_s}-ϵ_{r,r_s} curve.

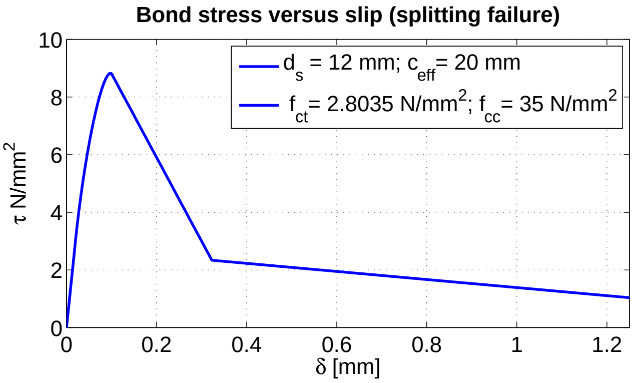


Figure 2.3: Example of bond-slip relation for splitting failure calculated with Matlab bond model

2.6 Calculation of pull-out failure

For pull-out failure the calculation of the bond stress from the radial stress is similar to splitting failure (see section 2.5). The calculation of the slip from the radial strain is more complicated. Instead of a linear relationship now a nonlinear relationship is derived. The assumption is that with pull-out failure the cone angle (φ) will decrease with increasing slip because of the wear of the interface layer. Figure 2.4 shows this relationship which also depends on the steel strain ϵ_s. The curve is described by the points a, b, c and d (for formulas see [8] and appendix D). Between a-b and b-c the function is parabolic and between c-d the function is exponential. Explicit formulations were not given in [8] therefore they are given here (for a complete derivation of these functions see appendix D).

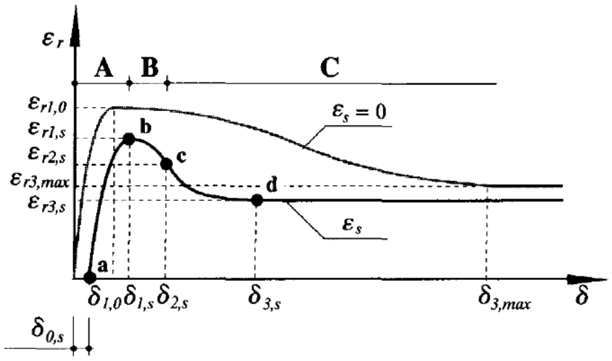


Figure 2.4: Radial strain - slip relation for pull-out failure, taken from [8]

Parabolic function between points a and b:

$$\epsilon_r(\delta) = -(\frac{\epsilon_{r1,0}}{\delta_{1,0}^2})\delta^2 + 2(\frac{\epsilon_{r1,0}}{\delta_{1,0}})\delta \tag{2.3}$$

Parabolic function between points b and c:

$$\begin{aligned}\epsilon_r(\delta) = &-\frac{(\epsilon_{r1}-\epsilon_{r2})}{(\delta_2^2+\delta_{1,0}^2-2\delta_{1,0}\delta_2)}\delta^2 + 2\frac{\delta_{1,0}(\epsilon_{r1}-\epsilon_{r2})}{(\delta_2^2+\delta_{1,0}^2-2\delta_{1,0}\delta_2)}\delta \ldots \\ &+\epsilon_{r1} - \delta_{1,0}^2\frac{(\epsilon_{r1}-\epsilon_{r2})}{(\delta_2^2+\delta_{1,0}^2-2\delta_{1,0}\delta_2)}\end{aligned} \tag{2.4}$$

The derivation of the exponential function between c and d proved to be difficult. The exact format was not given in [8] therefore an assumed function description is used: $\epsilon(\delta) = b\delta e^{-a\delta} + c$ (with a, b, c unknown constants). Ideally the curve should be calculated with an equal slope at point c. Solving the unknowns with this condition analytically however was not possible (see appendix D). Although the unknowns could be calculated numerically the choice is made to use a zero value for the slope at point d instead. This resulted in the following function between points c and d:

$$\epsilon_r(\delta) = \frac{(\epsilon_{r3max}-\epsilon_{r2})}{(\delta_{3max}e^{-1}-\delta_2 e^{-\frac{\delta_2}{\delta_{3max}}})}\delta e^{-\frac{\delta}{\delta_{3max}}} + \epsilon_{r2} - \frac{\delta_2 e^{-\frac{\delta_2}{\delta_{3max}}}(\epsilon_{r3max}-\epsilon_{r2})}{(\delta_{3max}e^{-1}-\delta_2 e^{-\frac{\delta_2}{\delta_{3max}}})} \tag{2.5}$$

Using these formulations an example of a bond-slip curve for pull-out failure is given in figure 2.5. The letters designate the areas described by the various functions (see also figure 2.4). The used assumption of a zero value for the slope at point d worked quite nicely. Unless very high steel strains are taken (>10‰) the slope at point c is almost continuous going from part B to part C.[2]

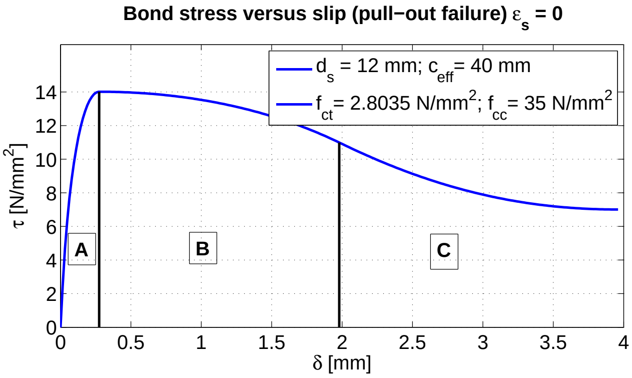


Figure 2.5: Example of bond-slip relation for pull-out failure calculated with Matlab bond model

[2] it should be noted that the calculations for this thesis generally do not involve large steel strains or slip values, therefore the function description for part C is less relevant

2.7 Results for splitting failure

In this section some results of the Matlab bond model will be presented when splitting failure is governing. Figure 2.6 shows the influence of the effective cover (from zero to about 33 mm) on the bond-slip relation. With the used parameters an effective cover larger then about 33 mm will result in pull-out failure. Figure 2.7 shows the influence of the concrete compressive strength (from 25 to 110 N/mm^2) on the bond-slip relation. Indirectly changing the concrete compressive strength also changes the concrete tensile strength and the concrete Young's modulus (see section 2.3). With an increasing strength the maximum bond stress increases but also the behaviour becomes more brittle. Finally figure 2.8 shows the influence of increasing the rebar diameter (from 12 to 40 mm). With equal concrete cover this

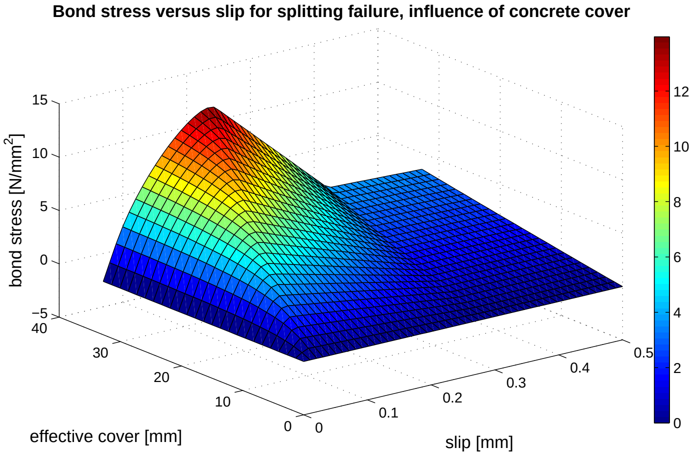


Figure 2.6: Bond-slip versus effective cover (d_s=12 mm, f_{cc}=35 N/mm^2, f_{ct}=2.80 N/mm^2)

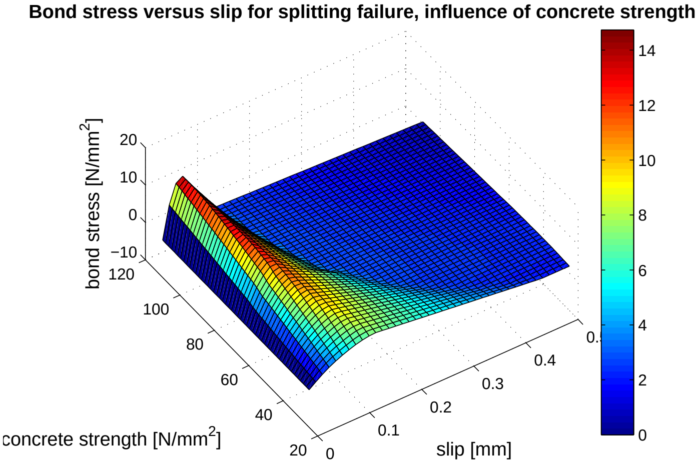


Figure 2.7: Bond-slip versus concrete strength (d_s=12 mm, c=20 mm)

has a reverse effect on the bond-slip relation (note that the y-axis is reversed in the plot).

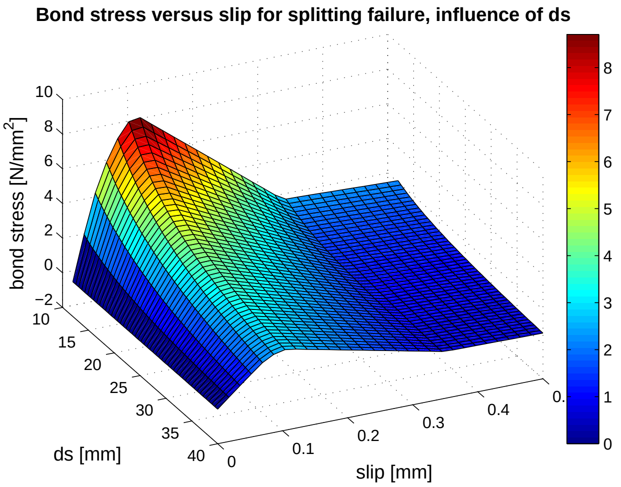


Figure 2.8: Bond-slip versus rebar diameter (c=20 mm, f_{cc}=35 N/mm^2, f_{ct}=2.80 N/mm^2)

2.8 Results for pull-out failure

In this section some results of the Matlab bond model will be presented when pull-out failure is governing. Figure 2.9 shows the influence of the steel strain (from 0.1‰to 100‰) on the bond-slip relation. Implementing this influence was one of the mayor goals when the bond model was conceived. Figure 2.10 shows

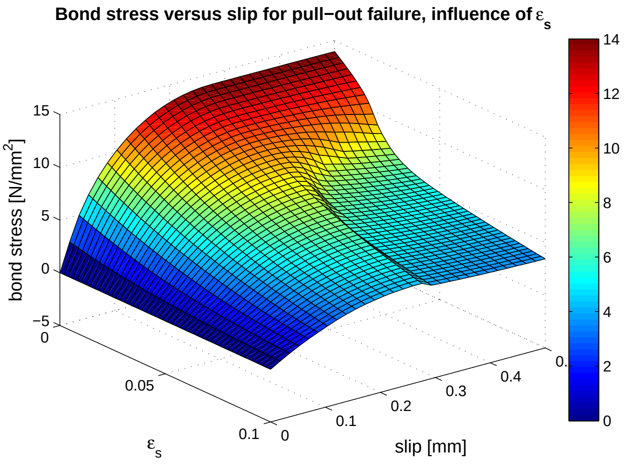


Figure 2.9: Bond-slip versus steel strain ϵ_s (d_s=12 mm, c=40 mm, f_{cc}=35 N/mm^2, f_{ct}=2.80 N/mm^2)

the influence of the concrete compressive strength (from 25 to 110 N/mm^2) on the bond-slip relation. Indirectly changing the concrete compressive strength again also changes the concrete tensile strength and the concrete Young's modulus (see section 2.3). The small jump at a concrete strength of 62.5 N/mm^2 is caused by a difference in the calculation of the concrete tensile strength between NSC and HSC. Finally figure 2.11 shows the influence of increasing the rebar diameter

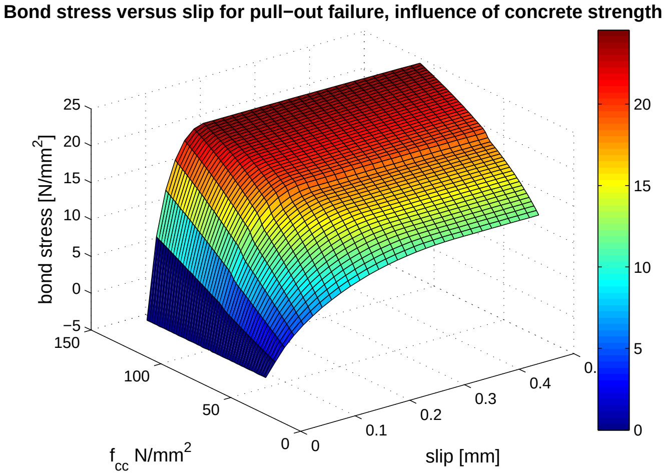


Figure 2.10: Bond-slip versus concrete strength (d_s=12 mm, c=40 mm, ϵ_s=0)

(from 12 to 40 mm). With a concrete cover equal to 4 times the rebar diameter (to prevent splitting failure) this has again a reverse effect on the bond-slip relation (note that the y-axis is reversed in the plot).

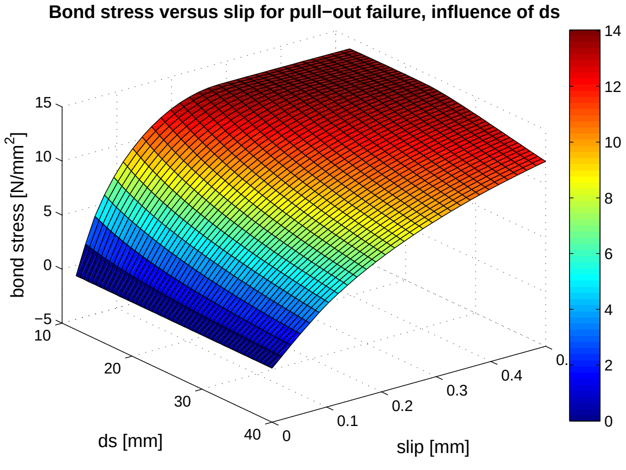


Figure 2.11: Bond-slip versus d_s (c=4d_s mm, ϵ_s=0, f_{cc}=35 N/mm^2, f_{ct}=2.80 N/mm^2)

Chapter 3

Extensions to an SLA software implementation

3.1 Introduction

Since 2001 an extension of the finite element program *DIANA* for the use of sequentially linear analysis (SLA) is under development at Delft University. Until now the models created use plane stress and/or truss elements only. With these element types already a great variety of problems can be solved (e.g. [10], [11], [15]).

One of the main objectives of this thesis is to extend the possibilities of modeling in SLA by expanding the library with interface elements and axi-symmetric elements. These extensions will then be used to analyze bond-slip related problems in chapters 6, 7 and 8.

SLA is especially used for brittle materials like concrete or masonry. These materials have a steep softening behaviour which can lead to many problems when using conventional nonlinear analysis (e.g. bifurcations, non-convergence). To circumvent these problems the negative slope of the softening diagram is replaced by a sawtooth diagram of positive slopes and the incremental-iterative method used in nonlinear analysis is replaced by a series of linear analyses (see [10]). Some specific sawtooth modeling will also be dealt with in chapter 4.

3.2 Program overview

The SLA programming is incorporated as an extension of the nonlin module of *DIANA* (version 9.2). The programming language for *DIANA* and the SLA extension is *Fortran77*. The basic steps performed in a SLA analysis are:

1. Initialization.
2. Perform linear analysis.
3. Identify critical integration point (called CRITIP) and critical load multiplier λ.
4. Multiply loads with the critical load multiplier.
5. Apply a strength and stiffness reduction of the critical integration point.[1]
6. Repeat steps 2 to 5 until the given number of steps is calculated or until no more damage (i.e. strength and stiffness reduction) is possible.

An exhausted description of the general SLA routines and workings will not be given here. The whole procedure is described in detail in an in-house developer's manual [17]. Regarding the programming for this thesis in the previous mentioned overview steps 3 and 5 are the most relevant.

[1] depending on the nonlinear relation it is also possible to apply an increase in strength together with a stiffness reduction, see chapter 4

3.3 Axi-symmetric elements

The programming of the axi-symmetric elements could be partially based on the existing code for plane stress elements. The difference is the addition of a fourth stress component for the circumferential direction: σ_{zz}. Currently in SLA the only option regarding concrete cracking is to use an orthogonal fixed crack total strain model.

The stress-strain relation in SLA (constitutive matrix) is based upon (reduced) Young's modulus in the orthogonal (crack) directions: N, T and Z. In this local crack axes system the relation between the stresses and the strains for an axi-symmetric element is given by:

$$\begin{bmatrix}\sigma_{nn}\\ \sigma_{tt}\\ \sigma_{zz}\\ \sigma_{nt}\end{bmatrix} = \mathbf{D}\begin{bmatrix}\epsilon_{nn}\\ \epsilon_{tt}\\ \epsilon_{zz}\\ \gamma_{nt}\end{bmatrix} \tag{3.1}$$

With $\mathbf{D}$ denoted as:

$$F\begin{bmatrix} (\nu_{tz}\nu_{zt}-1)E_n & -(\nu_{nz}\nu_{zt}+\nu_{nt})E_n & -(\nu_{nt}\nu_{tz}+\nu_{nz})E_n & 0\\ -(\nu_{zn}\nu_{tz}+\nu_{tn})E_t & (\nu_{zn}\nu_{nz}-1)E_t & -(\nu_{tn}\nu_{nz}+\nu_{tz})E_t & 0\\ -(\nu_{tn}\nu_{zt}+\nu_{zn})E_z & -(\nu_{zn}\nu_{nt}+\nu_{zt})E_z & (\nu_{tn}\nu_{nt}-1)E_z & 0\\ 0 & 0 & 0 & \frac{G}{F}\end{bmatrix} \tag{3.2}$$

In which:

$$G = \frac{E}{2(1+\nu)} \tag{3.3}$$

and:

$$F = \frac{1}{(\nu_{zn}\nu_{nt}\nu_{tz}+\nu_{tn}\nu_{nz}\nu_{zt}+\nu_{zn}\nu_{nz}+\nu_{tn}\nu_{nt}+\nu_{tz}\nu_{zt}-1)} \tag{3.4}$$

The D-matrix was derived by first considering a strain-stress relation and then applying separate uniaxial stress states which can then be combined using the principle of superposition. Inverting the final result yields the wanted stress-strain relation (for a complete derivation see appendix E). The two subscripts used are related to the stress direction which is being evaluated (first subscript) and the source stress direction (second subscript). The Poisson's ratios are determined with:

$$\nu_{ij} = \nu_0\left(\frac{E_j}{E_0}\right) \tag{3.5}$$

with i,j={N,T,Z}. Using equation 3.5 the six Poisson's ratios are:

$$\begin{aligned} \nu_{tn} &= \nu_0\left(\tfrac{E_n}{E_0}\right) & \nu_{tz} &= \nu_0\left(\tfrac{E_z}{E_0}\right)\\ \nu_{nt} &= \nu_0\left(\tfrac{E_t}{E_0}\right) & \nu_{zn} &= \nu_0\left(\tfrac{E_n}{E_0}\right)\\ \nu_{zt} &= \nu_0\left(\tfrac{E_t}{E_0}\right) & \nu_{nz} &= \nu_0\left(\tfrac{E_z}{E_0}\right) \end{aligned} \tag{3.6}$$

As can be seen from equation 3.6 the Poisson's ratios are determined by the source stress direction, therefore only three unique values exist. Because of this the constitutive matrix will become fully symmetrical.
The shear modulus G depends on the adopted shear retention model. Within SLA three options are available:

1. Full shear retention (no reduction):

$$G = \frac{E_0}{2(1+\nu_0)} \tag{3.7}$$

2. Constant shear retention:

$$G = \beta \frac{E_0}{2(1+\nu_0)} \tag{3.8}$$

   With $0 < \beta < 1$.

3. Stepwise reduction of shear modulus:

$$E_{min} = \text{min}\{E_n, E_t, E_z\} \tag{3.9}$$

$$\nu_{red} = \nu_0 \left(\frac{E_{min}}{E_0}\right) \tag{3.10}$$

$$G = \frac{E_{min}}{2(1+\nu_{red})} \tag{3.11}$$

Besides the D-matrix which is used to update the most critical integration point the largest part of the programming surrounded the calculation of the admissible load factor. To calculate an admissible load factor the stress vector at integration point level is evaluated in three directions (NTZ) and compared to the current tensile strength. The current tensile strength may differ per direction (NTZ) and depends on the damage history up until the moment of evaluation. For axi-symmetric elements the necessary steps in the routine used to calculate the admissible load factor (λ) is visualized in a flowchart see figure 3.1. To check the code several one-element tests with various load conditions have been analyzed, see appendix F. In all cases the code worked as was intended. The routines that were modified or created are given in appendix G table G.1.

3.4 Interface elements

Because the axis system is fixed the SLA programming involved for interface elements is much less complicated compared to axi-symmetric elements. Also only the shear direction (i.e. bond-slip) is able to have a nonlinearity. Therefore for the calculation of the admissible load factor only one stress component needs to be evaluated and for the critical integration point only one stiffness term needs to be updated.

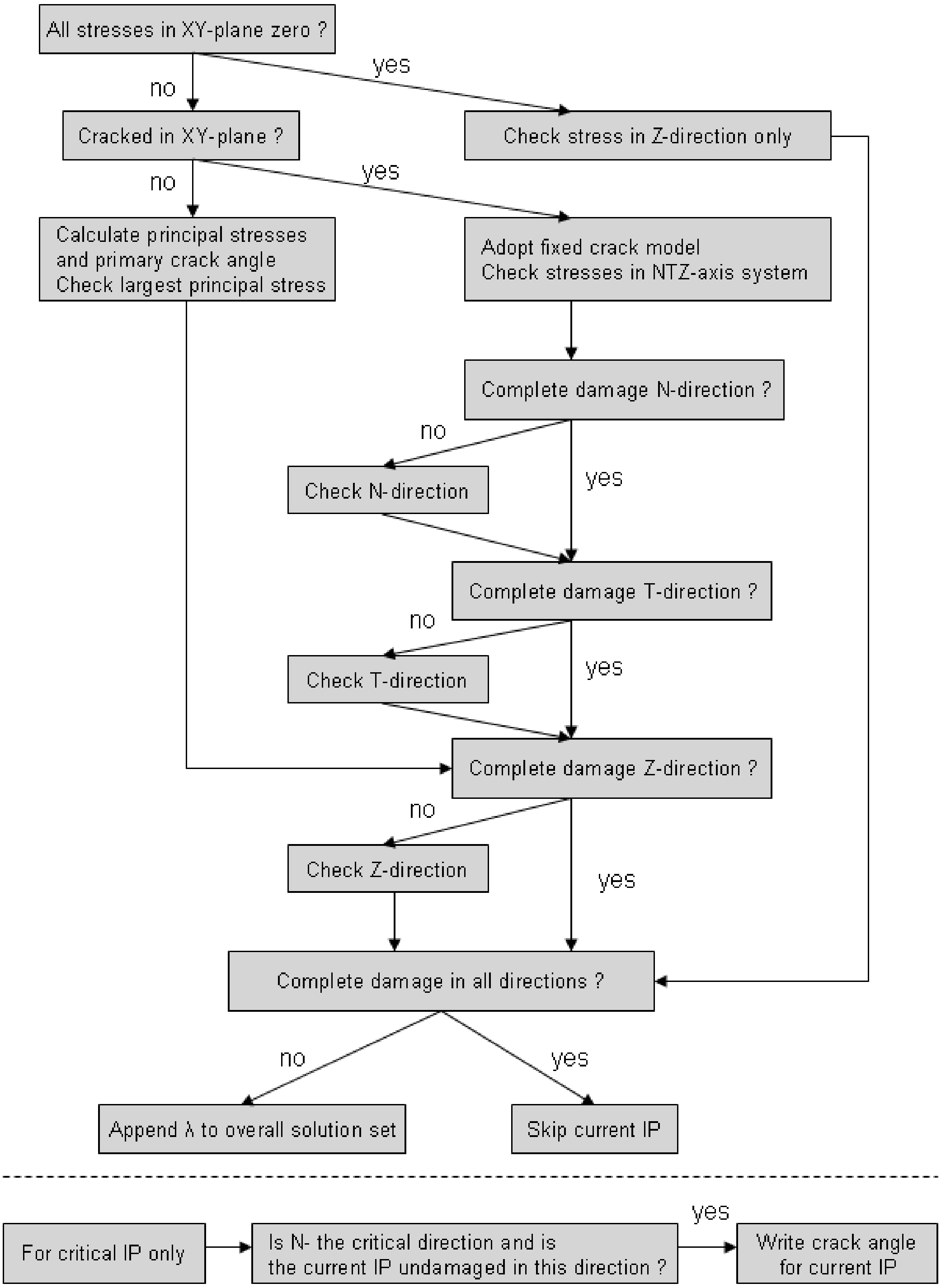

All stresses in XY-plane zero ?
yes
no
Cracked in XY-plane ?
Check stress in Z-direction only
no
yes
Calculate principal stresses and primary crack angle Check largest principal stress
Adopt fixed crack model Check stresses in NTZ-axis system
Complete damage N-direction ?
no
Check N-direction
yes
Complete damage T-direction ?
no
Check T-direction
yes
Complete damage Z-direction ?
no
Check Z-direction
yes
Complete damage in all directions ?
no
yes
Append λ to overall solution set
Skip current IP
For critical IP only
Is N- the critical direction and is the current IP undamaged in this direction ?
yes
Write crack angle for current IP

Figure 3.1: Flow chart to calculate admissible load factor for axi-symmetric element

Because bond-slip is not dependent on slip direction the same bond stress-slip diagram is valid for negative as well as for positive slip values. Therefore with calculating an admissible load factor there is always an equal lower bound and an upper bound value with different signs.

Because the programming was straightforward no one element tests have been performed. The routines that were modified or created are given in appendix G table G.2.

3.5 Cable (truss) elements

For quadratic truss elements (in *DIANA* called cable elements) a minor adaptation had to be made for use with SLA. In the routine called ISCLTR.F an if-statement is added to write the constitutive matrix to integration point level in case of a SLA calculation.

3.6 Post processing

For post processing the calculation of the crack strains in the NTZ-axis system is programmed for the axisymmetric elements. These crack strains are then written to an additional item list already used in SLA. The crack strains are calculated in a similar fashion as with plane stress elements. For the N-direction the crack strain is defined as:

$$\epsilon_{nn}^{(cr)} = \epsilon_{nn}^{(tot)} - \epsilon_{nn}^{(el)} = \epsilon_{nn}^{(tot)} - \frac{\sigma_{nn}}{E_0} \tag{3.12}$$

This means the crack strain is calculated as the difference between the total strain and the elastic strain if the total strain is less then the ultimate crack strain. In case the total strain is larger then the ultimate crack strain, the elastic strain becomes zero and the crack strain equals the total strain (figure 3.2). For the T- and Z-directions the crack strains are calculated analogously. The crack strains are calculated in routine EPSCRK.F with: $\epsilon_{nn}^{(cr)}$: ITEM04, $\epsilon_{tt}^{(cr)}$: ITEM07 and $\epsilon_{zz}^{(cr)}$: ITEM08. For post-processing also the damage indicators are written to the item list. For interface elements this is ITEM09. For axi-symmetric elements these are: ITEM10 (damage indicator N-direction), ITEM11 (damage indicator T-direction) and ITEM12 (damage indicator Z-direction).

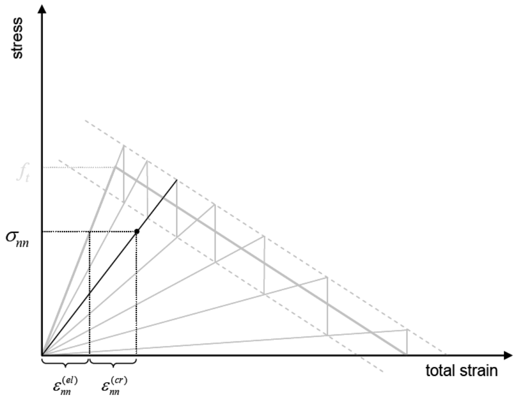


Figure 3.2: Definition of crack strain in SLA (taken from SLA users manual)

Chapter 4

Sawtooth approximation and sawtooth generators

4.1 Introduction

Sawtooth diagrams (or sawtooth approximations) are used in SLA to replace nonlinear curves of constitutive material behaviour. To approximate a nonlinear curve in SLA a sawtooth around the “mother” curve is constructed (see also section 3.1 and [10]). An overview of sawtooth approximations can also be found in [18]. Examples of sawtooth approximations for material behaviour that have been used with SLA are: softening of concrete in tension, crushing of concrete in compression and steel plasticity. In this chapter some extensions to the current theories and applications will be presented that will be used in this thesis.

4.2 Sawtooth modeling of Hordijk tension softening

For concrete different softening curves exits that can be used as a basis for a sawtooth approximation. For a linear softening curve the calculation of such a sawtooth approximation is straightforward. However with a nonlinear curve it becomes more complicated. Currently automatic creation of a sawtooth within the SLA environment is only implemented for a linear softening curve. In this thesis the nonlinear Hordijk softening curve will be used to model concrete in tension. For this an automatic sawtooth generator is programmed using Matlab. This section describes how this model is constructed. For concrete in tension the area underneath the softening curve is kept invariant and is equal to the fracture energy G_f divided by the crack band width h to get mesh size independent results [10].

4.2.1 Nonlinear formulation

The Hordijk tension softening curve is described in the *DIANA* manual section 18.1.1.5. As a reference the formulas are also given below. Some equations in the manual were found to be incorrect; the corrected ones are also given below.

$$\frac{\sigma_{nn}^{cr}(\epsilon_{nn}^{cr})}{f_t} = \begin{cases} \left(1+\left(c_1\frac{\epsilon_{nn}^{cr}}{\epsilon_{nn.ult}^{cr}}\right)^3\right)e^{\left(-c_2\frac{\epsilon_{nn}^{cr}}{\epsilon_{nn.ult}^{cr}}\right)}\dots & \\ \qquad -\frac{\epsilon_{nn}^{cr}}{\epsilon_{nn.ult}^{cr}}(1+c_1^3)e^{-c_2} & 0<\epsilon_{nn}^{cr}<\epsilon_{nn.ult}^{cr} \\ & \\ 0 & \epsilon_{nn.ult}^{cr}<\epsilon_{nn}^{cr}<\infty \end{cases} \tag{4.1}$$

$$\epsilon_{nn.ult}^{cr} = 5.136\frac{G_f^I}{hf_t} \tag{4.2}$$

Equation (18.48) in manual incorrect, correct equation:

$$\epsilon_{nn.ult.min}^{cr} = 6.90457\frac{f_t}{E} \tag{4.3}$$

Equation (18.49) in manual incorrect, correct equation:

$$f_t = \left(0.743 \frac{G_f^I E}{h}\right)^{\frac{1}{2}} \tag{4.4}$$

For a total strain approach the elastic branch also represents fracture energy. The area underneath the σ-ϵ_{tot} curve equals to $\frac{G_f}{h}$ (fracture energy divided by the crack band width). Since $\epsilon_{nn.ult}^{cr}$ in equation 4.2 is based on a decomposed strain model, for a total strain approach the fracture energy G_f is first reduced by the elastic part:

$$G_{f,red}^I = G_f^I - \frac{1}{2}\frac{f_{ct}^2 h}{E} \tag{4.5}$$

Then $\epsilon_{nn.ult}^{cr}$ is calculated using equation 4.2 with the reduced fracture energy.
To prevent a snap-back in the constitutive relation a maximum value for the crack band width h can be calculated by equating 4.2 and 4.3:

$$h_{max} = 0.743 \frac{G_f^I E}{f_t^2} \tag{4.6}$$

If h exceeds h_{max} the tensile strength will be reduced using equation 4.4. For a total strain model the assumption is that the ultimate total strain equals to the ultimate crack strain for a decomposed model. For a linear softening this is correct and the prove of this is straightforward. With the aforementioned formulas however the reduction of the fracture energy due to the elastic part, and the consequently reduction of the ultimate crack strain using equation 4.2 causes a reduction of the crack strain of:

$$\begin{aligned}\Delta\epsilon_{cr} &= \epsilon_{nn.ult}^{cr}(G_f^I) - \epsilon_{nn.ult}^{cr}(G_{f,red}^I) \\ &= 5.136\frac{G_f^I}{hf_t} - 5.136\frac{G_f^I}{hf_t} + \frac{5.136 f_t^2 h}{2Ehf_t} = \frac{5.136}{2}\frac{f_t}{E} = 2.568\epsilon_{el}\end{aligned} \tag{4.7}$$

This means the total strain is reduced with a constant of $1.568\epsilon_{el}$ more then by the elastic strain only. For a linear softening curve the factor in the numerator in equation 4.7 equals to 2. Therefore it can be easily verified that in that case $\Delta\epsilon_{cr}$ is indeed equal to ϵ_{el}. The extra reduction of the total strain for a nonlinear softening curve implies that the material will become more brittle.

4.2.2 Automatic sawtooth generator

Figure 4.1 shows an example of a sawtooth approximation for the Hordijk softening curve calculated with Matlab (for code see appendix H). As can be seen from the figure for the sawtooth approximation a constant, but unequal, vertical uplifting and downshifting of the original curve is used. For the sawtooth approximation two criteria have to be met:

1. The strain of the last vertical tooth should be equal to the ultimate total strain $\epsilon_{tot,max}$ ($\epsilon_{tot,max} = \epsilon_{el} + \epsilon_{cr,ult}$).

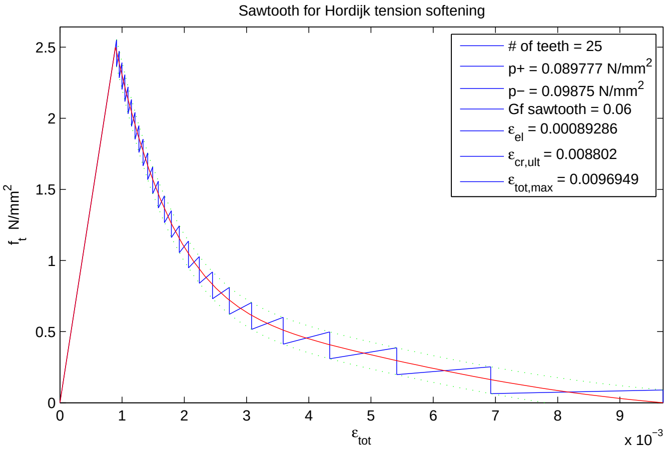


Figure 4.1: Calculated sawtooth of nonlinear softening curve using Matlab

2. The area underneath the sawtooth approximation ($\frac{G_f}{h}$) should be equal to the area underneath the mother curve.

For a fixed number of teeth and a fixed uplifting value (called $p+$ in N/mm^2) condition one determines the value for downshifting (called $p-$ in N/mm^2). The second condition can then be considered as a condition to determine $p+$. In Matlab the following steps are taken:

1. Check if h does not exceed h_{max} (using equation 4.6);
2. Calculate the reduced fracture energy (equation 4.5) and the ultimate crack strain (equation 4.2);
3. Calculate $p+$ with the above mentioned criteria;
4. Plot graph of mother curve and sawtooth approximation;
5. Write sawtooth table (Young's modules and tensile strength for all teeth) to a text file for direct use in *DIANA*.

Step 3 is the most complicated step and involves a lot of zero finding. The program flow of this part is:

- Choose a start value for $p+$: p_0^+;
- Calculate p_0^- using condition one;
- Calculate the fracture energy of the sawtooth approximation;
- Repeat the first 3 steps until condition two is met.

To calculate p^- the sawtooth approximation is calculated beginning on the left hand side with the extended elastic branch until the uplifted curve is reached. Then all teeth are determined in succession. For this a zero finding of the intersection points of the uplifted mother curve and the elastic branches of the sawteeth is needed. This zero finding proved to be unreliable at times because for low negative strain values the Hordijk function shows a steep vertical descending asymptote (see figure 4.2 left).[1] This caused intersections with the elastic branches to be found with corresponding negative strain values (especially for low values of the Young's modulus, i.e. with the last few teeth). To circumvent this problem the original curve is modified for negative strain values with a constant function value equal to the sum of the tensile strength and the uplifting value $p+$ (figure 4.2 right).

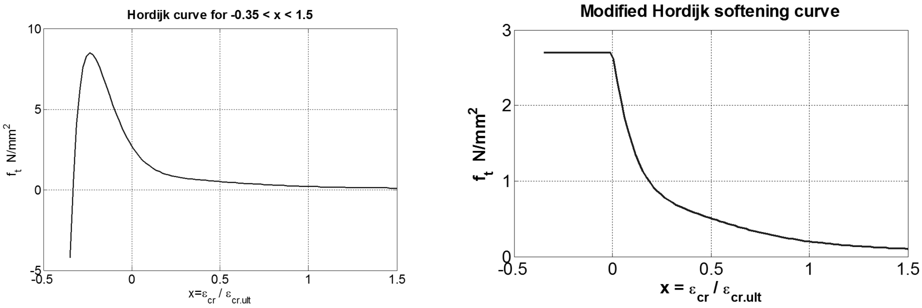


Figure 4.2: Original Hordijk softening curve (left) and modified softening curve (right) with normalized strain axis

For determining $p+$ based on criteria one and two, a zero finding is used that is bounded by a maximum and a minimum value for $p+$ (solution interval). To get results for all possible combinations of parameters this solution interval has to be set wide enough. However this causes a downside as the computing time can become very high. To create a predetermined solution interval for $p+$ the following equations are used:

$$p^+_{min} = \alpha \frac{f_t}{n} \tag{4.8}$$

$$p^+_{max} = \beta \frac{f_t}{n} \tag{4.9}$$

With:

α, β	constants
f_t	concrete tensile strength
n	number of teeth

With the concrete parameters used in chapters 5 and 6 (see section 5.1) good results were obtained using α=1.5 and β=2. The time to compute is usually about 30 seconds unless a lot of teeth are used ($>$40). With the above-mentioned constants to predetermine the solution interval however also a sawtooth approximation with 100 teeth can be calculated within about 50 seconds.

The last tooth calculated should have a stiffness and a tensile strength of zero. Because of the tolerances this will never be the case, therefore the last tooth is omitted from the diagram. Instead for the last tooth a very low stiffness and tensile strength is calculated based on the values of the second last tooth divided by 10000. It was mentioned before that a limitation is used for f_t if h becomes too

[1] the function is only valid between zero and one

large to prevent a snap-back in the softening curve. Although this limitation is also incorporated in the current Matlab model, for SLA this is actually not a problem as a previous study shows [11].

4.3 Sawtooth modeling of Den Uijl and Bigaj bond-slip model

The bond-slip relations from the bond model (see chapter 2) will be used with SLA and therefore the calculated bond-slip relations need to be approximated with a sawtooth diagram. For this the Matlab bond model is extended with an (optional) sawtooth generator. Because the curve is a numerical result consisting of a number of points it is first replaced by a continues curve fit function using the Matlab "spline" command. The spline runs through all data points and has an equal first and second derivative at all intersections. This makes it easier to uplift and downshift the "mother" curve and to calculate the intersections with the elastic branches. Because the bond-slip curves are already nonlinear for low values of slip only a limited elastic part is assumed. This means the sawtooth approximation will start already on the ascending part of the curve. Figures 4.3 and 4.4 show examples of sawtooth approximations for splitting and pull-out failure. For bond-slip an equal uplifting and downshifting of the mother curve is used. Because concrete cracking is involved one could argue that the area underneath the bond-slip curve is somehow related to fracture energy (especially in case of splitting failure). As with concrete in tension this would imply the sawtooth approximation to have an area equal to the mother curve (see section 4.2). Because evidence is lacking (also from the calculation results) and also to prevent a highly complicated calculation of the sawtooth approximation this is not implemented.

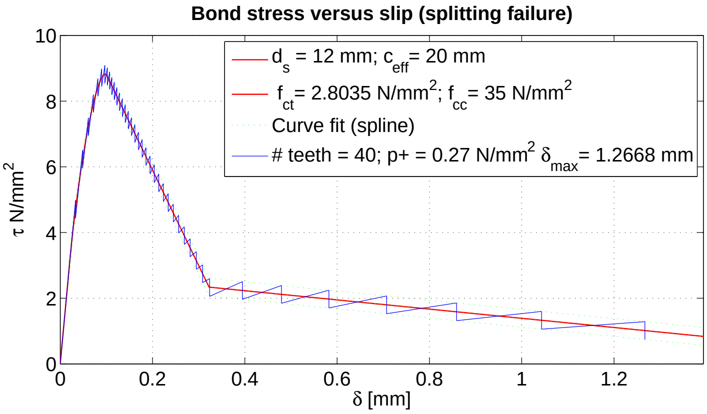


Figure 4.3: Bond-slip relation calculated with the Matlab bond model together with a sawtooth approximation for splitting failure

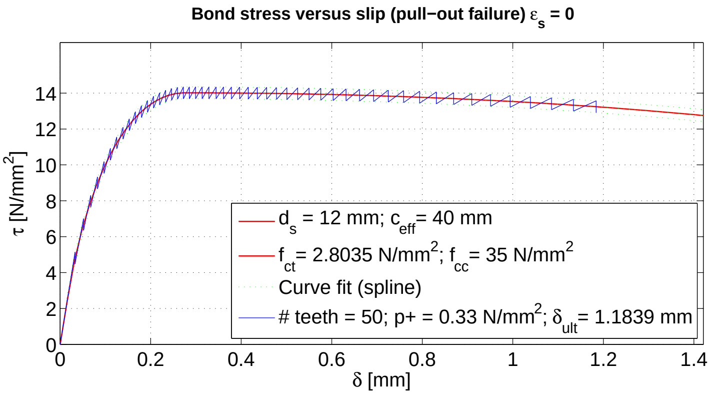


Figure 4.4: Bond-slip relation calculated with the Matlab bond model together with a sawtooth approximation for pull-out failure

4.4 Sawtooth modeling of steel plasticity

In previous studies yielding of steel in SLA is approximated with a sawtooth diagram with an equal uplifted and downshifted value, see figure 4.5 left. However this causes the possibility of steel stresses becoming higher then the yielding stress. Although this is also the case with concrete cracking in tension with steel there is no relation to fracture energy. Therefore to prevent higher then yielding stresses the sawtooth approximation is modified to have a downshifting value only, see figure 4.5 right. To have an equal range the downshifted value $p-$ should be approximately twice as large.

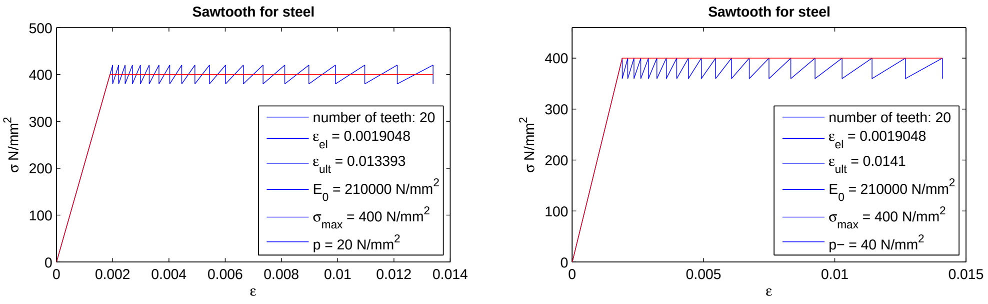


Figure 4.5: Sawtooth for steel used in previous studies (left) and new version (right)

Chapter 5

Nonlinear calculations of the tension-pull experiment by Gijsbers

5.1 Introduction

In the introduction (chapter 1) the possibilities to simulate the bond-slip behaviour at different scales and the description of various bond-slip relations have been presented. In this chapter the aim is focused towards the influence of bond-slip on the general cracking behaviour in a tension-pull experiment. For this purpose as a start first a “reference” calculation will be made of a specific tension-pull test which was already calculated by Rots in 1985 [2]. The aim is to trace back the results by Rots using the finite element program *DIANA*. Some extensions to the original calculation will be presented: a 3D approach using solid element and some investigation into the influence of using different unloading schemes. The nonlinear calculations also serve as a basis for comparison to the SLA calculations of the same model in the next chapter.

The tension-pull specimen is tested by Gijsbers [3]. It consists of a single reinforcing bar with a square concrete cover (figure 5.1). For the FEM discretization the square concrete cover (c=30 mm) is replaced by an equivalent circular cover (c_{eq}=34.4 mm). The material parameters used for the concrete part are: Youngs modulus E=28000 N/mm^2, Poissons ratio ν=0.2, tensile strength f_{ct}=2.5 N/mm^2, fracture energy G_f=0.06 N/mm, crack band width h=11.11 mm and shear retention factor β=0.5. For the reinforcing bar the material parameters are: Youngs modulus E=192300 N/mm^2; Poissons ratio ν=0.2 and yield stress σ_y=400 N/mm^2. For the interface the parameters used are: tangential stiffness S_t=250 N/mm^3, normal stiffness S_n=20000 N/mm^3 and ultimate tangential bond stress τ_u=6.25 N/mm^2. The load-displacement curve found in the experiment (figure 5.2) shows three distinct peaks representing three primary cracks.

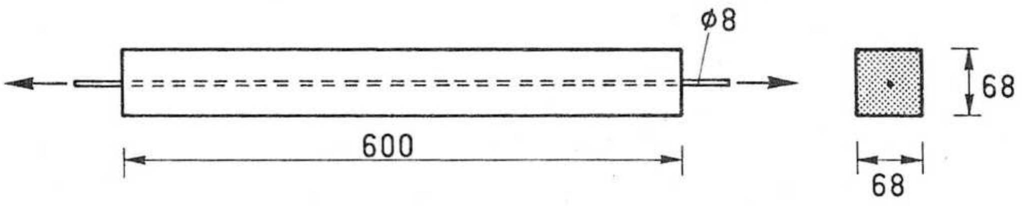


Figure 5.1: Tension-pull experiment by Gijsbers [3]

5.2 Concrete crack model

For concrete cracking there are many possibilities and different crack models available in the *DIANA* FEM package. All calculations will be made using a smeared crack approach. A general distinction can be made between fixed crack models and rotating crack models. Furthermore there is the possibility to use a decomposed crack strain model or a total strain model. For an extensive description of concrete crack models see for instance [6]. Because the aim is to recreate the results by Rots the choice is made to use the same constitutive model. This means an orthogonal fixed crack total strain model is used with nonlinear Hordijk

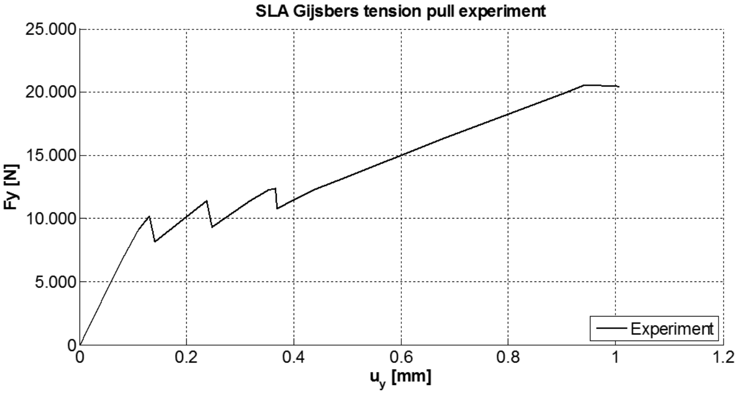


Figure 5.2: Load-displacement curve Gijsbers

tension softening (see section 4.2.1) and the tension cut-off is kept constant (see figure 5.3). This choice is furthermore based on the fact that this is currently the only crack model available when using sequential linear analysis. For all smeared crack models in *DIANA* the unloading is of secant type. The crack band width parameter h is obtained by the width that belongs to a single Gauss integration point.

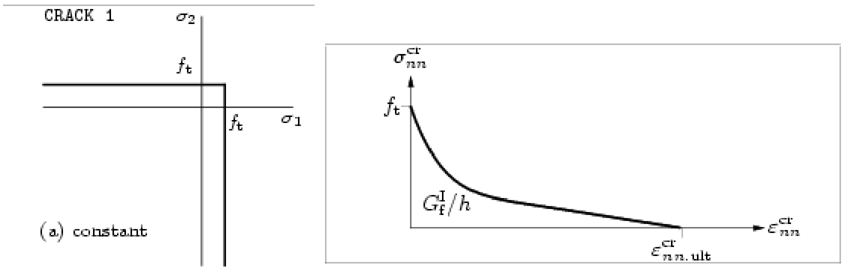


Figure 5.3: Constant concrete tension cut-off (left) and Hordijk tension softening curve (right) taken from DIANA manual

5.3 Interface bond-slip model

For interface elements several options are available in *DIANA* dependent on the element type selected with corresponding parameters (e.g. nonlinear elasticity, bond-slip and user-supplied). When selecting bond-slip one of three standard curves can be selected (figure 5.4). The bond-slip curves are all related to the tangential direction as the normal direction will always remain linear elastic. Also there is no possibility of coupling between the two directions. For the current calculation the multi-linear option is selected (figure 5.4c) with two branches: an elastic part and a perfectly plastic part see figure 5.5 (for numerical values see section 5.1).

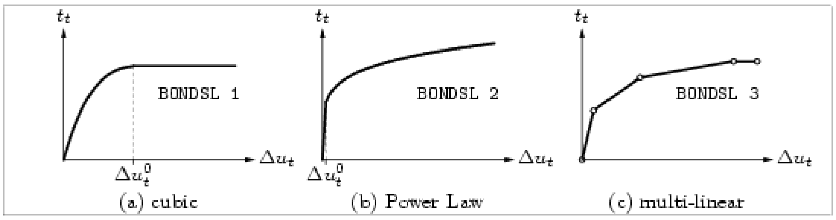


Figure 5.4: Standard DIANA bond-slip curves taken from DIANA manual

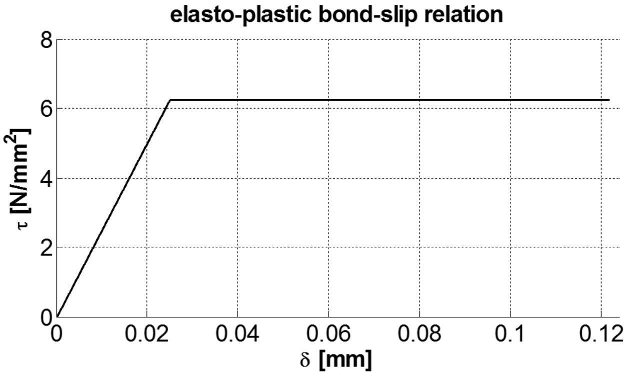


Figure 5.5: Used bond-slip relation (τ_u=6.25 N/mm^2)

5.4 Modeling of imperfections

In the experiment a crack pattern will develop which ultimately consists of several primary cracks. To obtain this crack pattern numerically it is absolutely necessary to incorporate material imperfections in the mesh at specific locations along the length of the model. If no imperfections were to be used a large area will crack simultaneously and no primary crack pattern will be able to develop.

To initiate cracking, three predetermined cross sections along the length of the model are given imperfections using a reduction of the concrete tensile strength (table 5.1). However if the fracture energy and the crack band width remain unchanged, only reducing the tensile strength results in an increase of the ultimate crack strain (figure 5.6). Although the method of creating an imperfection

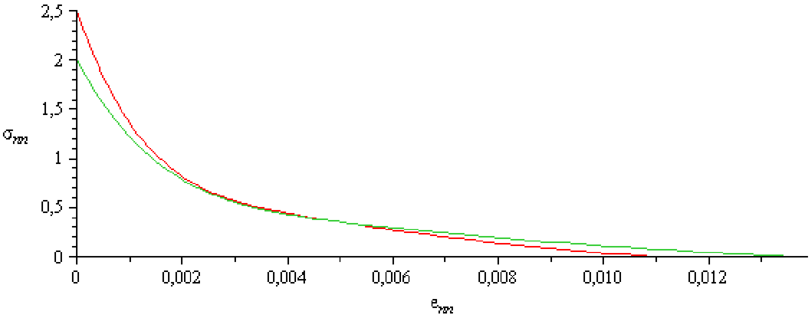


Figure 5.6: Effect of reducing tensile strength with Hordijk softening

is somewhat arbitrary, increased ultimate crack strain at a location of an imperfection seams illogical. To correct this “error” the ultimate crack strain will be reduced by the same coefficient as the tensile strength (figure 5.7). With the crack band width unchanged this can be achieved by reducing the fracture energy with a quadratic reduction factor.

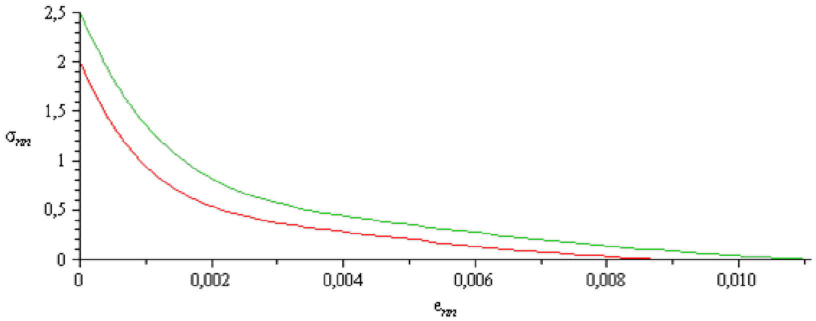


Figure 5.7: Effect of combined reduction of tensile strength and ultimate crack strain with Hordijk softening (reduction factor 0.8)

Table 5.1: Location and magnitude of imperfections

	Standard	Imperfection at $\approx \frac{1}{2}L$	Imperfection at $\approx \frac{3}{4}L$	Imperfection at $\approx \frac{1}{4}L$
f_{ct}[N/mm^2]	2.50	2.00	2.20	2.45
G_f[N/mm]	0.060000	0.038400	0.046464	0.057624

5.5 Axi-symmetric calculation

In this section the tension-pull experiment by Gijsbers will be calculated using axi-symmetric elements.

5.5.1 Model setup

The meshing is similar to the one done by Rots [2]. The mesh (figure 5.8) consists of 36 quadratic axi-symmetric elements for the concrete part and 18 quadratic truss elements for the reinforcing bar. The interface between the concrete and the reinforcing bar is modeled with 18 quadratic interface elements. The concrete imperfections described in section 5.4 are given to a full cross section of two elements. There were some issues with this mesh. First of all the reinforcing bar and therefore also one side of the interface is located in the axis of rotation (i.e. x=0). This caused ill-conditioning of the interface stiffness matrix. To circumvent this problem the truss is shifted slightly out of the axis of rotation. Also it is vital to have a correct stiffness for the interface. The stiffness is determined by the circumference of the reinforcing bar. In *DIANA* an interface can be given any thickness but it is unknown how the area of the interface is calculated exactly especially when using axi-symmetry. Therefore the *DIANA* configuration option "plane stress" is used for the interface since then the circumference can be inputted explicitly.

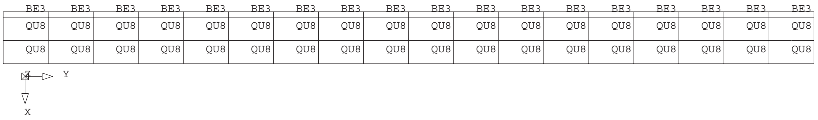

BE3 BE3 BE3 BE3 BE3 BE3 BE3 BE3 BE3 BE3 BE3 BE3 BE3 BE3 BE3 BE3 BE3 BE3
QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8
QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8 QU8
Z
Y
X

Figure 5.8: Finite element mesh

As for boundary conditions in the axis of rotation the truss is supported in the x-direction, the left end point is supported in the y-direction and the load (displacement) is located at the right end point. The imposed deformation is 1 mm (figure 5.9). In *DIANA* one crack band width (see section 5.1) can be set, which is equal for all directions. In this model the crack band width is set for the XY-plane, however for the circumferential direction this would actually depend on the distance to the axis of rotation and on the supposed number of splitting cracks. With three supposed splitting cracks the crack band width for the circumferential direction would vary between 8 and 80 mm as opposed to the 11.11 mm now used. For more details about used element types, integration schemes and program

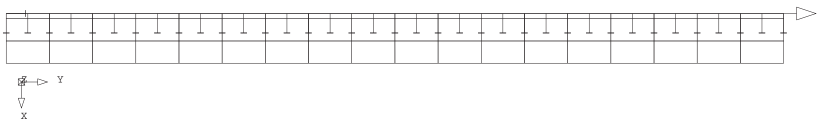


Figure 5.9: Boundary conditions and loading

settings see appendix I. To prevent oscillations of interface tangential strains it is necessary to use a lumped integration scheme for the interface elements [4].

5.5.2 Results

First a preliminary linear elastic calculation is made to check the correctness and general behaviour of the model. Next the nonlinear calculation is performed using displacement control, full Newton-Raphson and an energy based convergence criterion (for details see appendix I). After changing the imperfections several times to the ones given in table 5.1 eventually a crack pattern with three distinct primary cracks (figures 5.11 to 5.14) and a corresponding load-displacement curve (figure 5.10) was found which resembles quite closely the one reported by Rots [2]. Peak values are found at 0.128 mm (9.9 kN), 0.258 mm (11.6 kN), 0.403 mm (13.1 kN) and 0.903 mm (19.9 kN). The disks in figures 5.12 to 5.14 represent splitting cracks. Although primary cracking occurs at an early stage (represented by the three peaks in the load-displacement curve), ultimate crack strains (representing fully open cracks) are reached only after an applied displacement of about u_y=0.6 mm. The ultimate strength is governed by plasticity of the reinforcing bar at midpoint. Just before plasticity starts all three primary cracks have surpassed the ultimate crack strain (fully open cracks). The force concentration at both

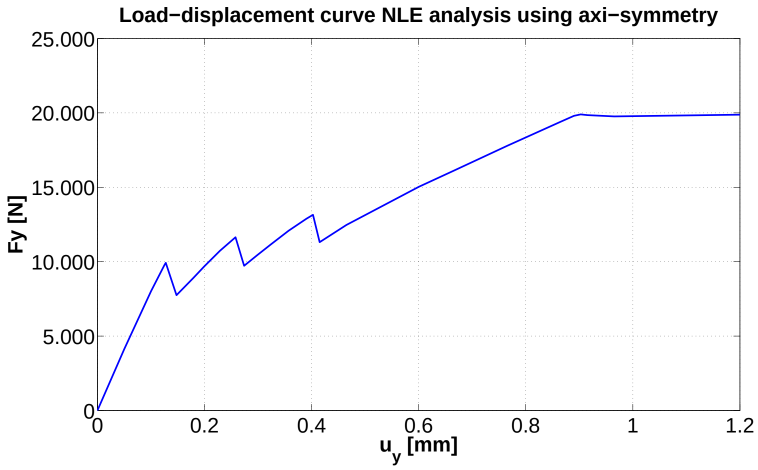


Figure 5.10: Load-displacement curve

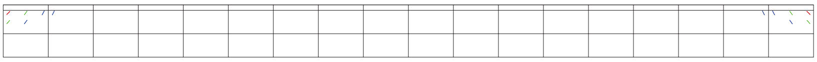


Figure 5.11: Crack pattern (u_y=0.1 mm, ϵ^{cr}_{max}=0.41e-3)

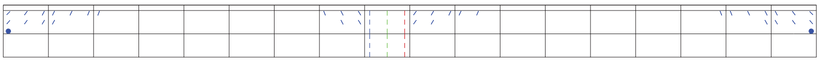


Figure 5.12: Crack pattern (u_y=0.2 mm, ϵ^{cr}_{max}=0.495e-2)

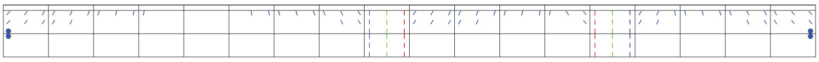


Figure 5.13: Crack pattern (u_y=0.3 mm, ϵ^{cr}_{max}=0.586e-2)

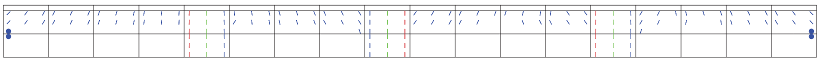


Figure 5.14: Crack pattern (u_y=0.6 mm, ϵ^{cr}_{max}=0.904e-2)

ends introduces the first cracking as can be seen in figure 5.11. If no imperfections were to be used the cracking would continue along the bar from both sides until they would coincide in the middle. However with the used imperfections it is possible to trigger localized primary cracks as can be seen in figures 5.12 to 5.14. With the fixed crack model used it is possible to get three orthogonal cracks in one integration point. Because of the axial loading and consequently uniform stress distribution in this model secondary cracking and splitting are very limited and splitting primarily occurs at the location of the load introduction. The final crack pattern at a load of u_y=1.2 mm is shown in figure 5.15. It is of interest to see the influence of using a nonlinear relation for the bond-slip (i.e. a limitation on the maximum bond stress). Figure 5.16 shows the final crack pattern if the interface is kept linear. It can be seen that now the concrete cracking has increased significantly. This is a logical consequence of the larger bond stresses being transferred to the concrete resulting in higher stresses in the concrete and thus in more cracking. The influence on the global strength of the model (e.g. in a load-displacement curve) however is found to be relatively low. This can most likely be explained by the fact that the global strength and stiffness is governed primarily by the three weakened cross-sections and not so much by the in between cracking.

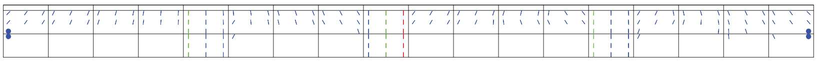


Figure 5.15: Final crack pattern (u_y=1.2 mm, ϵ^{cr}_{max}=0.319e-1)

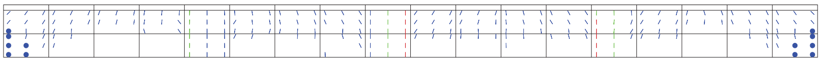


Figure 5.16: Final crack pattern using a linear interface (u_y=1.2 mm, ϵ^{cr}_{max}=0.258e-1)

The steel stresses along the reinforcing bar are in agreement with what one would expect and show peak values at a location of a primary crack (figure 5.17) together with a sign change of the bond stress. Figure 5.18 shows the interface bond stresses along the reinforcing bar at various load levels. In the beginning the largest slip and consequently the highest bond stress occurs near the (loaded or constrained) ends of the model. After a transition zone of about 200 mm the forces from the rebar are completely shared between the rebar and the surrounding concrete resulting in equal displacements and zero bond stresses. When a

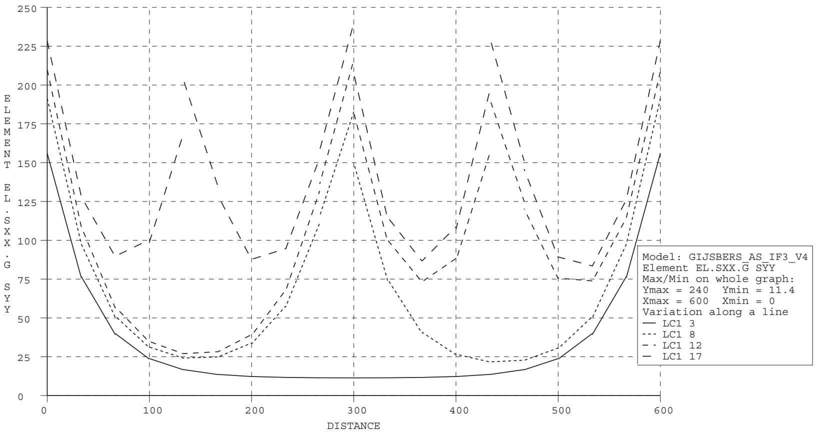


Figure 5.17: Steel stresses along the bar at u_y=0.1 mm (LC1 3), u_y=0.2 mm (LC1 8), u_y=0.3 mm (LC1 12) and u_y=0.42 mm (LC1 17)

primary crack develops the cracked cross section will behave increasingly like a free surface similar to the end faces. As can be seen from figure 5.18 left the bond stress distribution will then split up into two pieces that both have a bond stress distribution similar to the initial one. This behaviour continues until the primary crack pattern is fully developed.

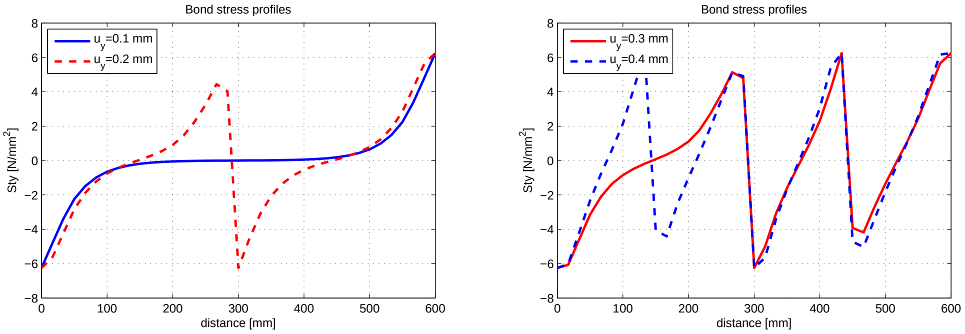


Figure 5.18: Bond stress profiles (u_y=0.1 mm, u_y=0.2 mm) (left) and (u_y=0.3 mm, u_y=0.4 mm) (right)

5.6 3D calculation

In this section the tension-pull experiment will be calculated in 3D. This is done to further explore the possibilities to model a tension-pull experiment and also to see if the results are consistent with the ones obtained with the axi-symmetric calculation.

5.6.1 Model setup

The mesh (figure 5.19) consists of 8x18 quadratic wedge elements and 8x18 quadratic brick elements for the concrete part. The reinforcing bar is modeled with 18 quadratic truss elements. The center axis is the y-axis. The interface between the concrete and the reinforcing bar is modeled with 18 quadratic line-solid connection interface elements. The imperfections described in section 5.4 are given to a full cross section of sixteen elements. In this model the interface has zero thickness and one side is connected to the centerline that is shared by all eight wedge elements. Because of this the cross sectional area of the concrete is slightly enlarged towards the centerline by the area that should actually be the steel bar, the difference is however negligible. For a line-solid interface element the bar diameter is an obligatory parameter and therefore the interface stiffness is calculated automatically by *DIANA*.

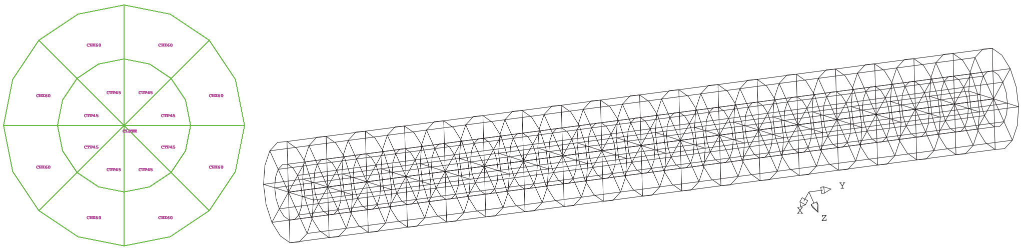


Figure 5.19: Finite element mesh, cross section and 3D view

The boundary conditions consist of supports in three directions for both bar endpoints. To prevent a rigid body rotation around the center axis a tangential support is given to a single node on the outer edge of a brick element. The load (displacement) is located at the right end point. The imposed deformation is 1 mm.

The bond-slip option is not available in *DIANA* when using a line-solid interface element. Therefore instead the nonlinear option is used for the interface elements. This introduces a difference in unloading behaviour. The bond-slip model uses secant unloading whereas with the nonlinear relation there is no difference in the loading and unloading curve (i.e. they are the same). The influence of this will be explored later.

The crack band width is again set to 11.11 mm representing the distance between the integration points in y-direction (i.e. the length direction). For more details about used element types, mesh generation, integration schemes and program settings see appendix K. To prevent oscillations of interface tangential strains it is necessary to use a lumped integration scheme for the interface elements [4].

5.6.2 Results

First a preliminary linear elastic calculation is made to check the correctness and general behaviour of the model. Next the nonlinear calculation is performed using displacement control, full Newton-Raphson and an energy based convergence criterion (for details see appendix K). The load-displacement curve found resembles quite closely the one obtained using axi-symmetric elements (figure 5.20 right). However a bifurcation was found (figure 5.20 left) that relates to the first primary crack, i.e. the displacement at which this first primary crack occurs completely determines the rest of the load-displacement behaviour. Although no bifurcations were found in the axi-symmetric model, it is clear that obtaining correct results in a nonlinear analysis using displacement control is not straightforward. Analysis #2 was taken as a basis for comparison to the curve obtained in the axi-symmetric model (figure 5.20 right). Peak values are found at 0.139 mm (10.1 kN), 0.271 mm (11.6 kN), 0.432 mm (13.5 kN) and 0.922 mm (19.9 kN). Figures 5.21 to 5.24 show the crack patterns at different loading stages. The steel stress profiles and the interface bond stresses also show great similarity to the axi-symmetric model.

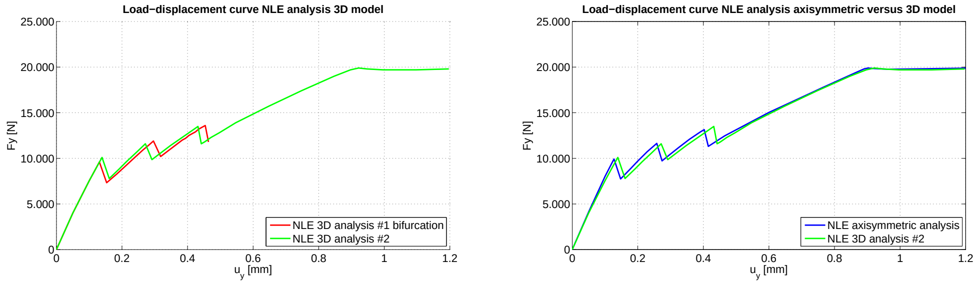


Figure 5.20: Bifurcation in load-displacement curve (left) and comparison of load-displacement curve between axi-symmetric model and 3D model (right)

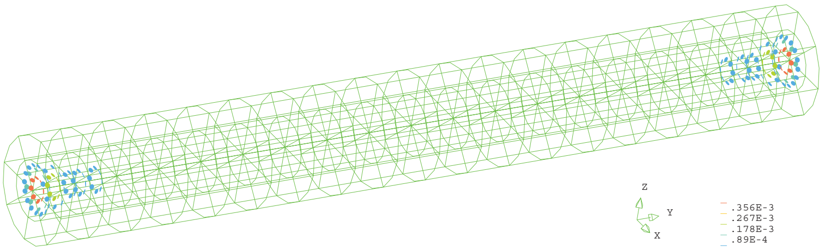


Figure 5.21: Crack pattern (u_y=0.1 mm, ϵ^{cr}_{max}=0.445e-3)

Figure 5.25 shows the splitting cracks that occur on the end faces. Because of the use of solid elements the crack band width in circumferential direction is better estimated then in the case of axi-symmetric elements and the crack pattern is more

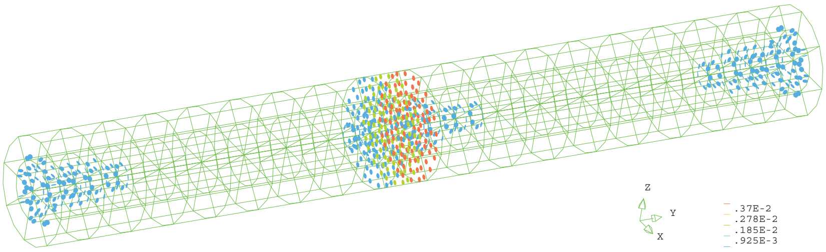


Figure 5.22: Crack pattern (u_y=0.2 mm, ϵ^{cr}_{max}=0.463e-2)

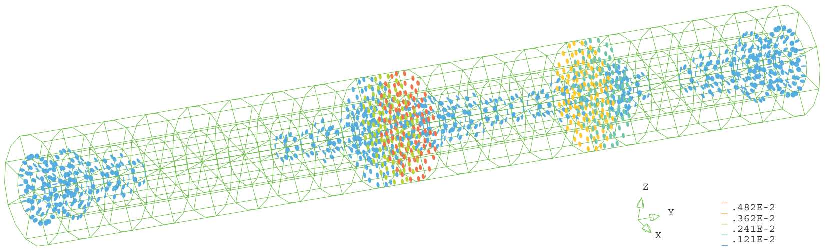


Figure 5.23: Crack pattern (u_y=0.3 mm, ϵ^{cr}_{max}=0.603e-2)

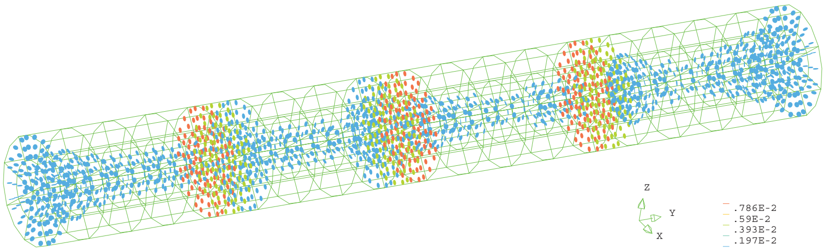


Figure 5.24: Crack pattern (u_y=0.6 mm, ϵ^{cr}_{max}=0.983e-2)

realistic. In comparison to the axi-symmetric calculation now splitting also occurs closer to the reinforcing bar.

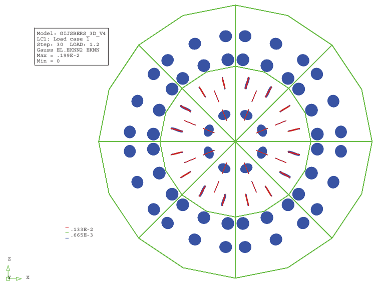


Figure 5.25: Radial view of crack strains (u_y=1.2 mm)

5.7 Modeling and implementation of alternative bond-slip unloading schemes

In the previous sections two calculations were presented with different unloading behaviour for the interface elements. In the axi-symmetric calculation the unloading behaviour was of secant type. In the 3D model there was no difference in the loading or unloading curve. In this section an investigation into the influence of the unloading behaviour of the interface elements in a tension-pull experiment will be conducted. Of first importance is to know how much unloading actually takes place in the tension-pull model. For this purpose some of the results of the axi-symmetric calculation of section 5.5 will be examined in more detail.

5.7.1 Unloading in the axi-symmetric calculation

The interface elements located at the outer edges show a similarly shaped load-relative displacement (load-slip) curve as the outer points themselves see e.g. figure 5.28 element 36. Figure 5.26 shows the load-slip curves for all interface elements (mean integration points per element). From figure 5.26 it can be seen that between primary cracking all interface elements show continuously increasing slip values (in absolute sense). So only when primary cracking occurs does unloading take place. Figure 5.28 shows only those elements with slip values that exceed the elastic branch of the bond-slip curve since only then does the unloading behaviour become relevant. The plastic branch starts at a slip value of ± 0.025 mm. For reference figure 5.27 shows the interface element numbering together with the final crack pattern. Elements 22, 26 and 33 show only continuous loading and no unloading. Elements 19, 28 and 36 show the largest reversal of slip values, although it should be noted that the first unloading peak of these elements is very close to the elastic branch. It is interesting to see that the largest unloading takes place near the load introduction on the outer edges (elements 19 & 36) or

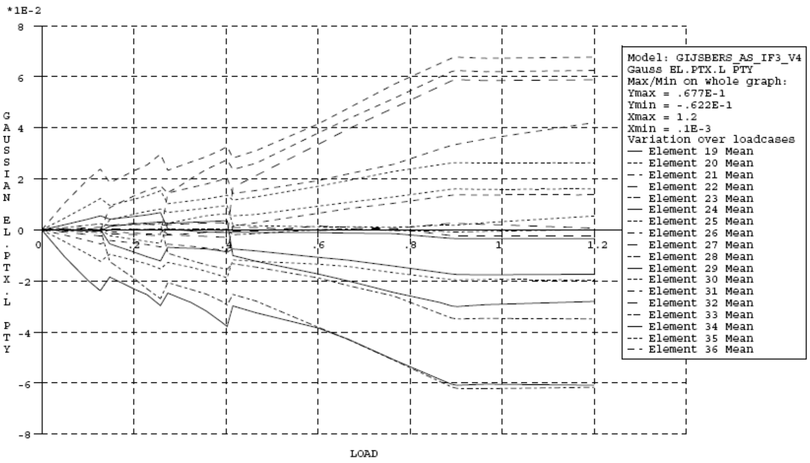


Figure 5.26: Load-slip curve for all interface elements (mean integration points per element)

close to the largest primary crack (element 28). From figure 5.28 it can be concluded that the amount of unloading on the plastic branch is very limited in this model. It is to be expected that changing the unloading behaviour will not have a significant effect on the results of the calculation.

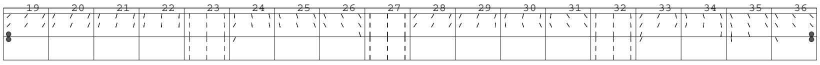


Figure 5.27: Interface element numbering and final crack pattern (u_y=1.2 mm)

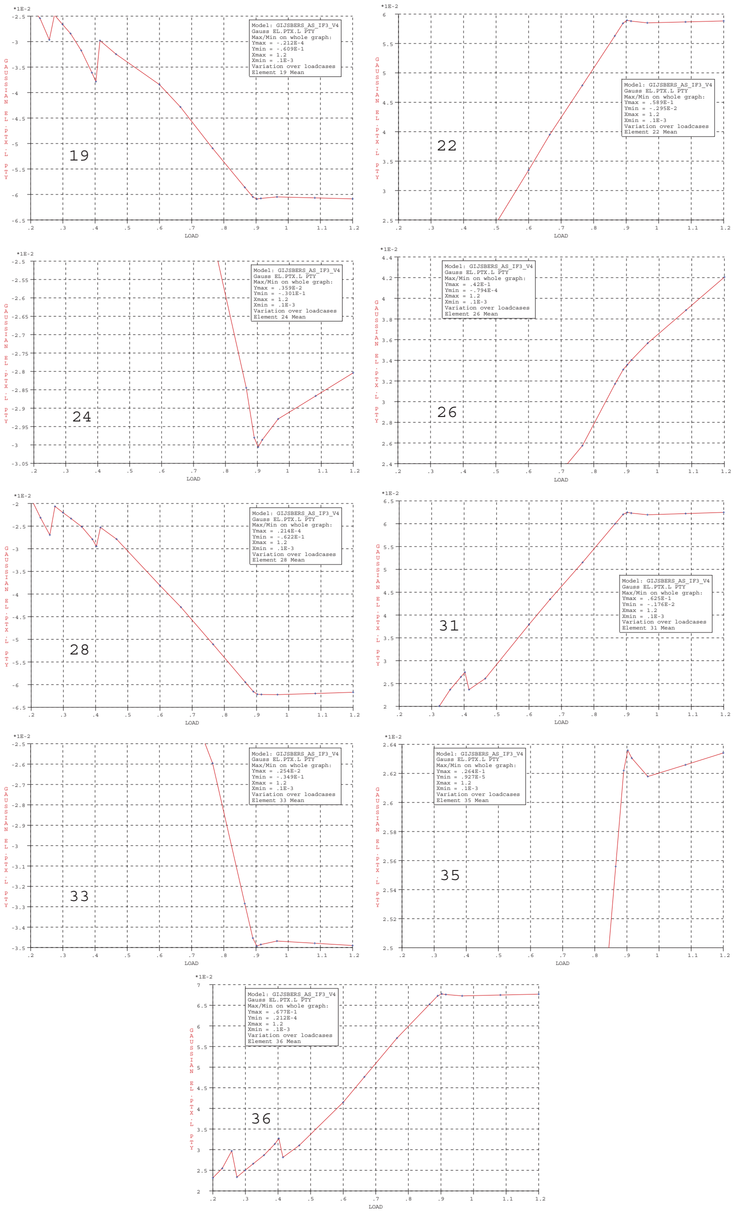


Figure 5.28: Load-slip curves of elements with plastic slip (mean integration points per element)

5.7.2 Programming user interface

From the previous section it is clear that reversal of slip takes place in a tension-pull experiment but is very limited. In *DIANA* it is possible to use a self programmed relation between the relative displacements, and the bond stresses (in *DIANA* called tractions). This section describes the programming of a user supplied subroutine that will be used to explore different unloading options even though the influence is expected to be small. In matrix notation and using *DIANA* notation this relation can be denoted as:

$$\begin{bmatrix} t_n \\ t_t \end{bmatrix} = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \begin{bmatrix} \Delta u_n \\ \Delta u_t \end{bmatrix} \tag{5.1}$$

With:

t_n normal traction in N/mm^2
t_t shear traction in N/mm^2
Δu_n relative normal displacement between concrete and reinforcement
Δu_t relative shear displacement (slip) between concrete and reinforcement

When using the standard bond-slip relations provided by *DIANA* only d_{11} and d_{22} have non-zero values. Also by default the normal direction is kept linear elastic whereas the shear direction can be described with a nonlinear relation (see also section 5.3). To use a user supplied interface model a file called "USRIFC.F" is programmed in *Fortran77* code. The file must update the tractions and the tangent stiffness, i.e. the matrix containing d_{11}, d_{12} etc. depending on the type of relation you want to use. Bond-slip relations with three different types of unloading behaviour have been programmed (see also figure 5.29):

1. Nonlinear unloading (model name: *BOND3N*);

2. Secant unloading (model name: *BOND3S*);

3. Elastic unloading (model name: *BOND3E*).

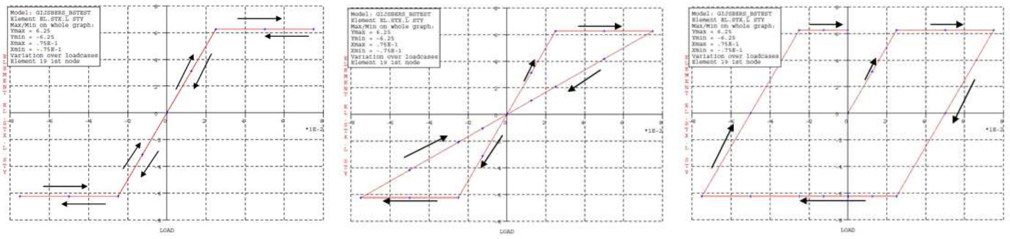


Figure 5.29: Visual representation of different unloading models, from left to right: BOND3N, BOND3S and BOND3E

The three models are programmed in a single file using a "switch" activated by the model name. The model name is therefore a required input in the *DIANA* .dat file. The bond-slip relation used is the same for all three models, i.e. a relation consisting of two branches: an elastic part and a perfectly plastic part (similar to figure 5.5). Because of the shifting of the elastic branch for the *BOND3E* model a so called state variable is used to indicate the location (in terms of slip) of the origin of the (shifted) elastic branch.

After the programming of the *Fortran* code and the necessary modifications to the *DIANA* .dat and .com files a calculation with the user supplied subroutine can be executed. In the calculation the *Fortran* code is automatically compiled and a new executable of "nl41" is created and used instead of the standard *DIANA* version. For details about user supplied subroutines see [5]. For the USRIFC.F code see appendix J. To check the correctness of the code first a small scale one element test is performed on all three models. For the results of these tests see appendix J.

5.7.3 Experimental results from cyclic tests

Before continuing to the results of the calculations with the various bond unloading models some literature that describes unloading behaviour for bond-slip [1] is consulted. In this section a short description of the typical unloading behaviour will be presented to see which bond model would describe the actual behaviour the best. Figure 5.30 shows a typical experimental result for bond-slip for cyclic loading. Upon slip reversal a steep unloading is observed (from point A to B), followed by a low bond stress until the original monotonic curve is reached (point C). Because of the slip the steel-concrete interface between slip values that have already been reached, in the positive as well as the negative direction is damaged. Therefore in the next cycle a larger slip value is needed to obtain the same bond stress.[1] Another example is given in figure 5.31. From the aforementioned it can be stated that only changing the unloading behaviour is far from sufficient to describe bond behaviour in real cyclic loading conditions. However when only focusing on the initial unloading behaviour the steep unloading upon slip reversal is best described with the elastic unloading model. It can also be stated that using the nonlinear unloading (where the loading and unloading curve are the same) is the worst approach as the stiffness upon slip reversal initially equals to zero. Secant unloading can be considered second best. In any case the damaged slip

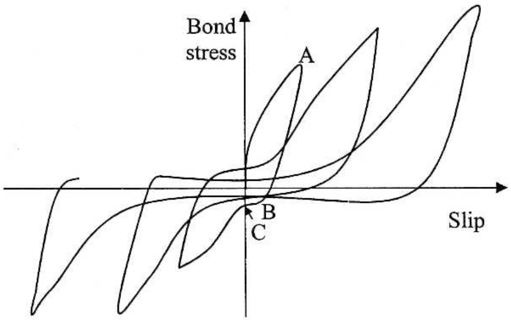


Figure 5.30: Typical bond-slip curve for cyclic loading

[1] note that the increase in peak values for next cycles in figure 5.30 is not typical and depends on the testing, see figure 5.31

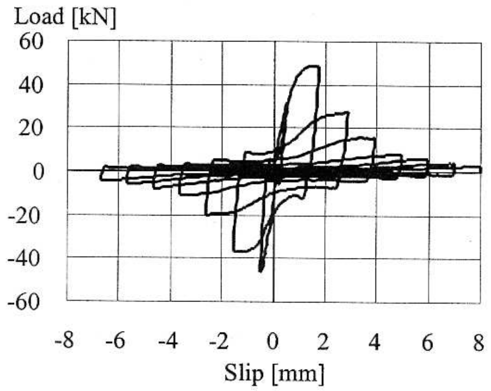


Figure 5.31: Example of load-slip from a cyclic loading test

region (for next cycles) and the low bond stress region (between points B and C, figure 5.30) are not taken into account.

5.7.4 Results of alternative unloading schemes

Although one element tests are performed on the three user supplied interface models an additional check can be easily made for the *BOND3S* model since this model should give exactly the same results as the standard *DIANA* “BONDSL 3” option. This is verified by comparing both results in a load-displacement curve. Figure 5.32 shows this comparison which fortunately shows an exact match. Next

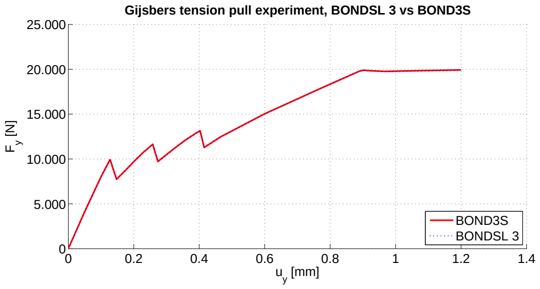


Figure 5.32: DIANA “BONDSL 3” versus User supplied BOND3S

the axi-symmetric model of section 5.5 is calculated using the three different unloading schemes. Figure 5.33 shows the results of these calculations in a load-displacement curve. At close observation it can be seen that initially (up until the second primary crack) the three models give similar results indicating that the influence of unloading is small. This was to be expected since unloading from the plastic branch does not happen before an applied displacement of around 0.2 mm (figure 5.26).

Around the second and third primary crack some differences appear mainly between the *BOND3S*/*BOND3E* and the *BOND3N* model. Figure 5.34 shows this is more detail. The differences are extremely small and hardly of any real consequence as was already mentioned to be expected in section 5.7.1. The same conclusion holds for other results such as the final crack pattern.

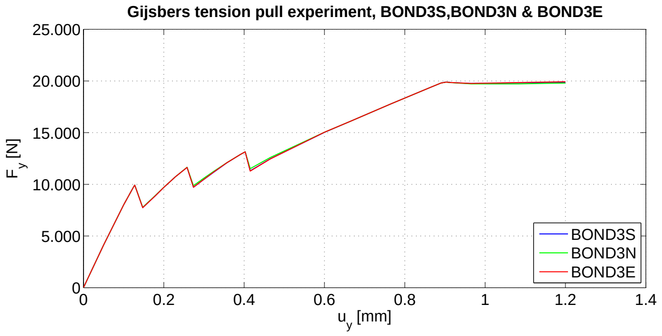


Figure 5.33: Load-displacement curves for axi-symmetric model using different unloading schemes

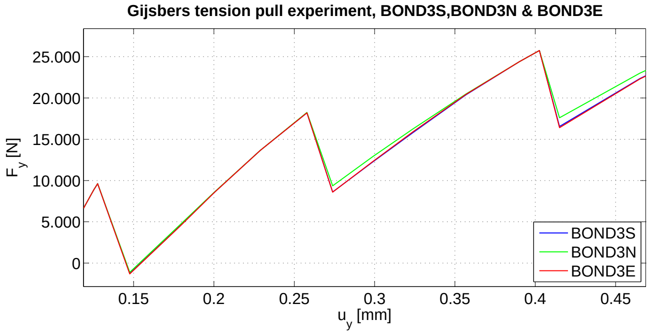


Figure 5.34: Detail of figure 5.33

5.8 Summary and conclusions chapter 5

In this chapter a tension-pull experiment was calculated using interface elements to model the bond-slip between the reinforcement and the concrete. A reference was taken to a previous calculation by Rots [2]. For the concrete an orthogonal fixed crack model was used. The use of axi-symmetric elements and 3D elements was examined. To investigate the influence of the unloading behaviour for the elastic-plastic bond-slip relation, three different unloading schemes have been investigated using an user defined interface model. The following conclusions regarding the tension-pull simulations can be stated:

Axi-symmetric model

1. The results by Rots [2] could be recreated with a large enough accuracy. The load-displacements curves are almost identical and also the crack patterns are very similar. Using displacement control it was possible to "jump" over the primary cracks.

2. It has been found that the needed imperfections to trigger a reasonable primary crack pattern can be quite large. The tensile strength and the ultimate crack strain had to be reduced by as much as 20%.

3. Oscillations of the bond stresses were found in the calculations if the integration scheme of the interface elements was not set to lumped. From literature also with experiments these oscillations do not occur.

4. Using a linear interface (or even a rigid connection) the amount of cracking in the surrounding concrete will increase significantly. However with a tension-pull experiment the load-displacement curve is much less effected.

5. Splitting cracks are not well calculated in an axi-symmetric model. The crack band parameter should in fact vary for the circumferential direction depending on the distance to the axis of rotation and on the number of supposed splitting cracks. *DIANA* does not allow for different crack band widths for different directions.

6. Connecting an interface in *DIANA* to an axis of rotation when using an axi-symmetric model results in ill-conditioning of the interface stiffness matrix. The workaround is to shift it slightly out of the axis of rotation. Also with *DIANA* it is unknown how the interface stiffness in relation to the circumferential direction in an axi-symmetric model is taken into account. To be sure the circumference is correct it is best to use the "plane stress" option for the interface elements in this case.

3D model

1. The results in terms of the load-displacement curve and the crack patterns are very comparable to the ones obtained with the axi-symmetric elements. Furthermore conclusions 2 and 3 of the axi-symmetric model also hold for the 3D model.

2. A bifurcation in the load-displacement was found indicating that obtaining consistent results in a nonlinear calculation using displacement control is not straightforward. No attempts have been made to use arc-length control which would perhaps yield better results.

3. In the model a line-solid interface element is used to connect the truss elements to the surrounding solid elements. In this configuration it is not possible to give the interface a thickness and therefore the concrete needs to be expanded to the center axis resulting is a slightly enlarged cross section. For large diameters of the reinforcing bar this can become a problem. The solution would be an interface element that is able to connect a line to a cylinder and to use brick elements only instead of wedge elements.

Bond-slip unloading behaviour

1. In the axi-symmetric model unloading (i.e. reversal of slip) only takes place when a primary crack occurs. Between primary cracks the slip is monotonic.

2. In the axi-symmetric calculation and with the used bond-slip relation the amount of unloading of the interface elements is very limited. The largest reversal of slip takes place at the end faces or close to a primary crack.

3. Regarding the unloading schemes investigated a comparison to experimental results of cyclic loading test shows that, albeit limited, elastic unloading is the best approximation of the actual unloading behaviour. Nonlinear unloading (i.e. the loading and unloading curves are the same) is the worst approach. Secant unloading can be considered second best.

4. Because the amount of unloading is so limited in the axi-symmetric calculation changing it's behaviour has very little effect. The load-displacement curves are all almost identical and the same holds for the crack patterns. In a larger model with more cracking the type of unloading behaviour could become more relevant.

Chapter 6

SLA calculations of the tension-pull experiment by Gijsbers

6.1 Introduction

In this chapter the tension pull-experiment with axi-symmetric elements that was calculated in section 5.5 will be recalculated now with SLA. Therefore the programmed axi-symmetric elements (see section 3.3) as well as the interface elements (see section 3.4) for SLA will be adopted in the model. Equal to the non-linear calculation Hordijk tension softening will be used. To create the sawtooth approximation the Matlab sawtooth generator described in section 4.2.2 will be used.

In a previous study the tension-pull experiment was already calculated with SLA by Belletti [10]. In this calculation the concrete was modeled with plane stress elements, the reinforcing bar with truss elements and a perfect bond between the concrete and the reinforcing bar was assumed. The calculation results showed that it was possible to get a cracking behaviour that involved the formation of several primary cross-sectional cracks along the length of the model. It also showed that snap-backs are captured realistically with SLA. The main disadvantage of this model was the assumption of perfect bond.

6.2 Adopted sawtooth models

In the NLE calculation imperfections were used at predetermined locations to get a reasonable primary crack pattern. To be able to compare results first also with the SLA calculation the same imperfections are given in the model (see section 5.4). This means for the concrete four separate sawtooth diagrams are needed. One diagram for the unmodified concrete (figure 6.1 left) and three for the various imperfections (figure 6.1 right and figure 6.2). For the interface the bi-linear relation (see also figure 5.5 section 5.3) can be easily casted in a sawtooth diagram. However it is important to know the maximum slip that occurs in the calculation beforehand because this also approximately determines the maximum slip value in the sawtooth diagram. For this the results of the NLE calculation are consulted

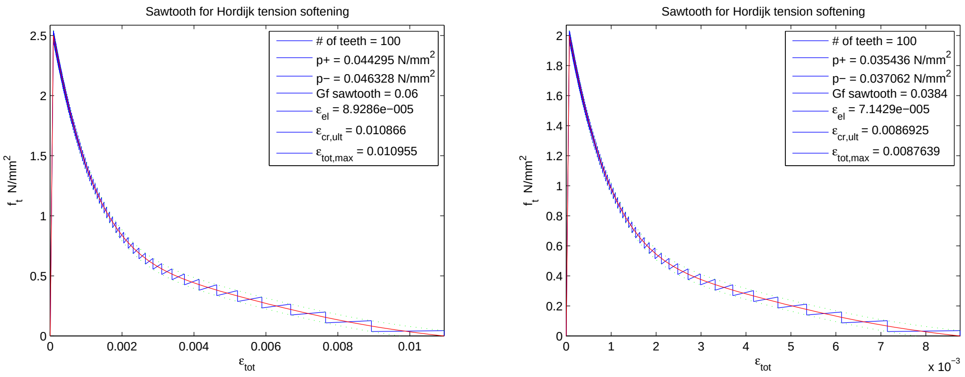


Figure 6.1: Sawtooth approximation for concrete using 100 teeth, unmodified (left), imperfection at midsection (right)

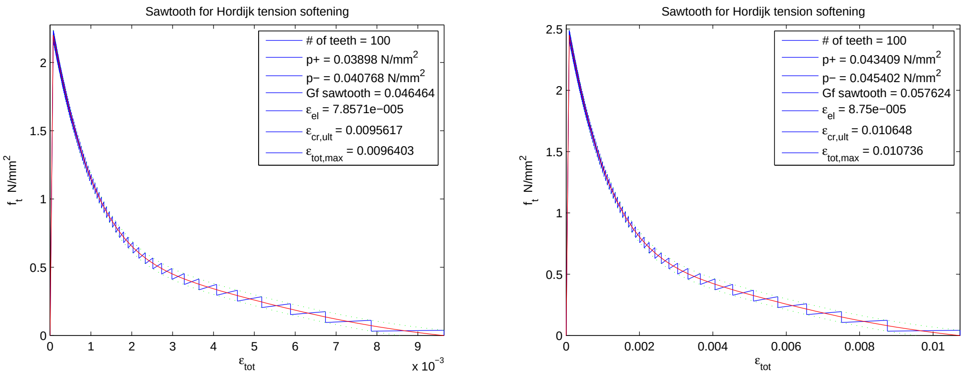


Figure 6.2: Sawtooth approximation for concrete using 100 teeth, imperfection at $\frac{3}{4}L$ (left), imperfection at imperfection at $\frac{1}{4}L$ (right)

and a maximum slip value of about δ=0.09 mm is found. To create a safe margin the sawtooth diagram is somewhat extended, see figure 6.3 left. Also for the steel the results of the NLE calculation are consulted and a maximum strain of about ϵ_s=0.0263 is found. The adopted sawtooth diagram for steel is given in figure 6.3 right. Contrary to earlier SLA calculations (e.g. [10], [11]) only a constant downshifting for the plastic region is used and no uplifting (see section 4.4). The accuracy of the sawtooth diagrams can be calculated as the ratio between the average uplifted and downshifted value and the maximum elastic value. The current sawtooth diagrams have been tailored to have an approximately equal accuracy as can be seen in table 6.1.

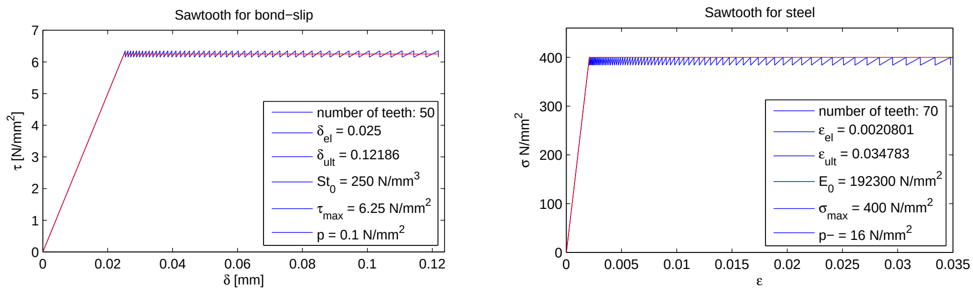


Figure 6.3: Sawtooth approximation for bond-slip (left) and steel (right)

Table 6.1: Accuracy of used sawtooth diagrams

	concrete	bond-slip	steel
average of $p+$ and $p-$ [N/mm^2]	(0.044+0.046)/2	(0.1+0.1)/2	(0+16)/2
maximum elastic value [N/mm^2]	2.5	6.25	400
ratio	1.8%	1.6%	2%

6.3 Results using imperfections

First a comparison is made between the NLE calculation and the SLA calculation with the predetermined imperfections, figure 6.4 shows both results in a load-displacement curve. For the SLA calculation about 6800 load steps were used. The graph shows that the SLA result is in excellent agreement with the NLE analysis. The load-displacement path is near identical except with SLA also snap-backs are captured after each primary crack, figure 6.5 shows this in more detail. The first complete damage of the concrete (i.e. the first integration point on the last sawtooth) occurs at an applied displacement u_y=0.64 mm at midpoint in the element with the strongest imperfection (this is about 5 times the displacement at first primary crack initiation). This is in excellent agreement with the NLE analysis (see section 5.5.2). Table 6.2 shows a comparison between the NLE and the SLA analysis for load level of plastic slip initiation of the interface elements at relevant locations. Also these results are in excellent agreement. Next figure 6.6 shows the interface bond stresses along the reinforcing bar at various load

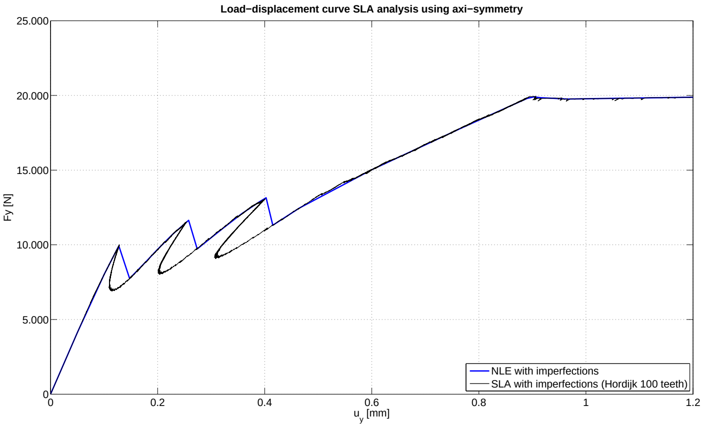


Figure 6.4: Load-displacement curve NLE versus SLA

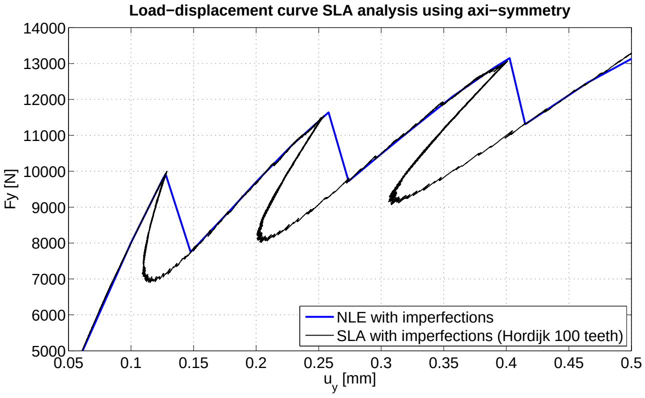


Figure 6.5: Load-displacement curve, detail of figure 6.4 (snap-backs)

Table 6.2: Comparison of plastic slip initiation (δ=δ_{el}=0.025 mm)

	NLE analysis	SLA analysis
end faces	u_y=0.094 mm	u_y=0.097 mm
near first primary crack	u_y=0.19 mm	u_y=0.20 mm
near second primary crack	u_y=0.28 mm	u_y=0.30 mm
near third primary crack	u_y=0.41 mm	u_y=0.40 mm

levels which shows similar results to the NLE calculation (see figure 5.18). Figures 6.7 to 6.11 show the crack strains at various load levels, these figures can be compared to figures 5.11 to 5.15. Compared to the NLE analysis the maximum crack strains show a difference of about +15% for u_y=0.1 mm and <1% for u_y=0.2 mm, u_y=0.3 mm and u_y=0.6 mm, although the load level for SLA can not be exactly determined because of the somewhat irregular shape of the load-displacement curves. The difference at u_y=0.1 mm could be explained by the fact

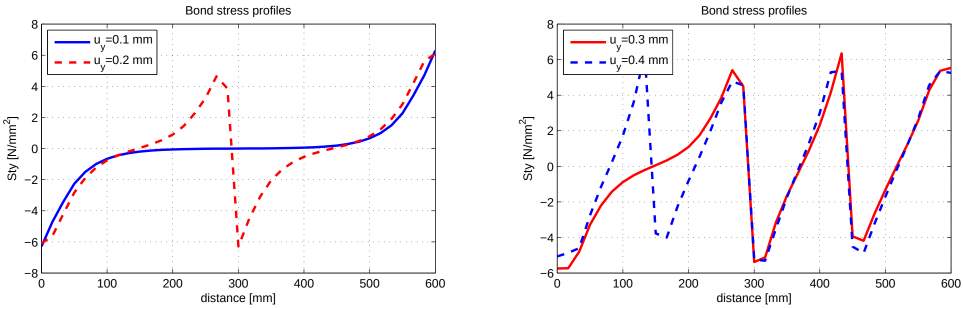
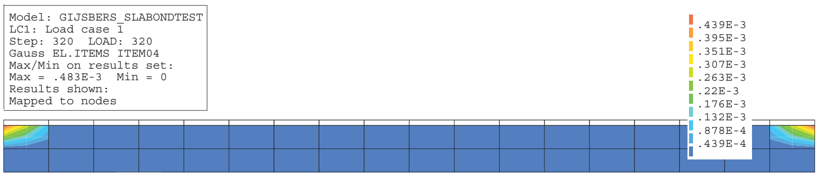
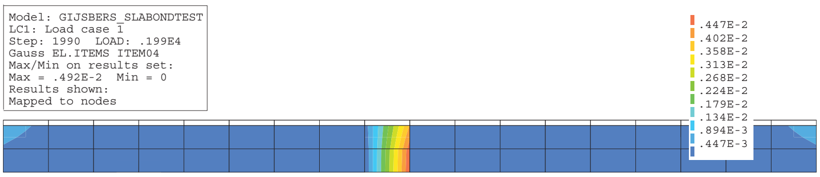


Figure 6.6: Bond stress profiles (u_y=0.1 mm, u_y=0.2 mm) (left) and (u_y=0.3 mm, u_y=0.4 mm) (right)



```
Model: GIJSBERS_SLABONDTEST
LC1: Load case 1
Step: 320  LOAD: 320
Gauss EL.ITEMS ITEM04
Max/Min on results set:
Max = .483E-3  Min = 0
Results shown:
Mapped to nodes
```

Figure 6.7: Crack strain ϵ_{cr}^{nn} at u_y=0.1 mm



```
Model: GIJSBERS_SLABONDTEST
LC1: Load case 1
Step: 1990  LOAD: .199E4
Gauss EL.ITEMS ITEM04
Max/Min on results set:
Max = .492E-2  Min = 0
Results shown:
Mapped to nodes
```

Figure 6.8: Crack strain ϵ_{cr}^{nn} at u_y=0.2 mm

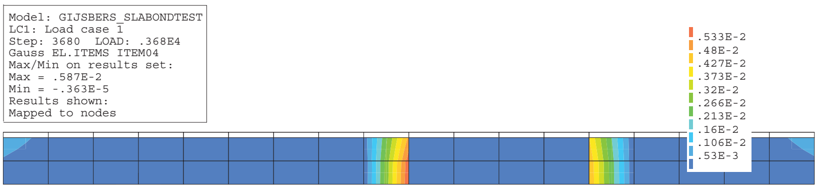


Figure 6.9: Crack strain ϵ_{cr}^{nn} at u_y=0.3 mm

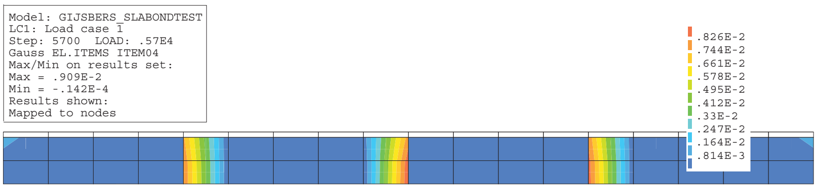


Figure 6.10: Crack strain ϵ_{cr}^{nn} at u_y=0.6 mm

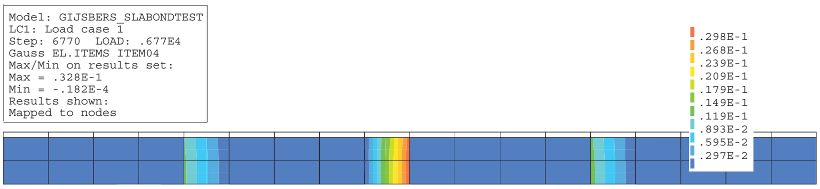


Figure 6.11: Crack strain ϵ_{cr}^{nn} at u_y=1.2 mm

that prior to the primary crack development the cracks at either end near the load introduction are more localized with SLA compared to the NLE analysis despite the imperfections (compare figure 6.7 to 5.11). Figure 6.12 shows the splitting crack strains with a concentration at both ends (caused by the load introduction and the support). Finally figure 6.13 shows the steel yielding localization which occurs at midpoint near the largest imperfection.

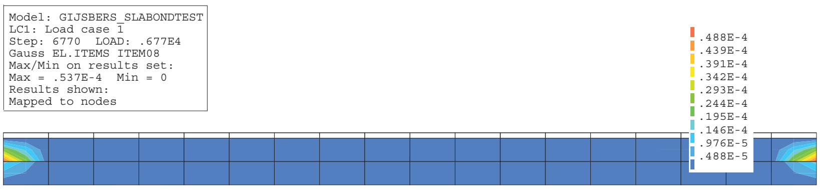


Figure 6.12: Crack strain ϵ_{cr}^{zz} at u_y=1.2 mm

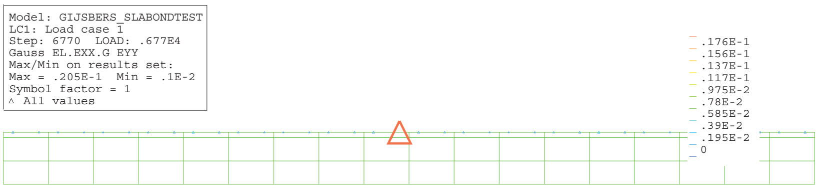


Figure 6.13: Steel strain ϵ_s at u_y=1.2 mm

6.4 Results without imperfections

Now a second calculation will be executed in SLA without the use of imperfections. The sawtooth models for bond-slip and steel are equal to the ones mentioned in section 6.2 except now for the concrete only figure 6.1 left applies. For the SLA calculation about 6550 load steps were used, somewhat less then used in the calculation with imperfections. As can be seen from the graph in figure 6.14 in SLA

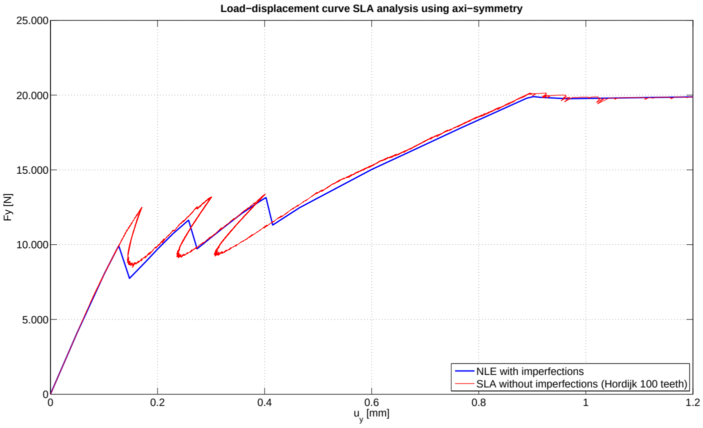


Figure 6.14: Load-displacement curve NLE versus SLA analysis (SLA without imperfections)

there is no need to use imperfections because there will always be a most critical integration point found even though several points might have a near identical stress level. As was to be expected without imperfections the model is somewhat stronger resulting in higher peak values especially for the first two peaks and a very slight uplifting in the post cracking region (the third peak is almost identical because the third imperfection used in the previous calculation was also very limited). Because of the large number of teeth used the SLA curve is again quite smooth. The first complete damage of the concrete (i.e. the first integration point on the last tooth) occurs at u_y=0.74 mm at midpoint (this is about 4.5 times the displacement at first primary crack initiation). Figures 6.15 to 6.19 show the calculated crack strains at various load levels. The crack locations are shifted only one or two integration points compared to the previous calculation with the prede-

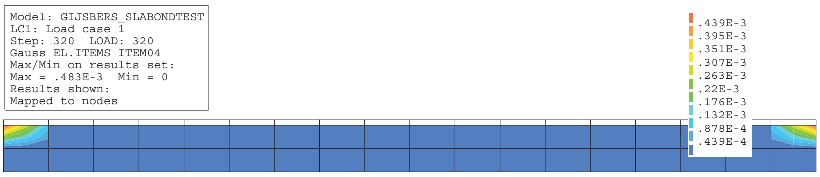


Figure 6.15: Crack strain ϵ_{cr}^{nn} at u_y=0.1 mm

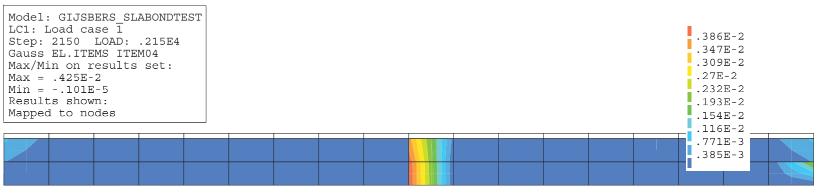


Figure 6.16: Crack strain ϵ_{cr}^{nn} at u_y=0.2 mm

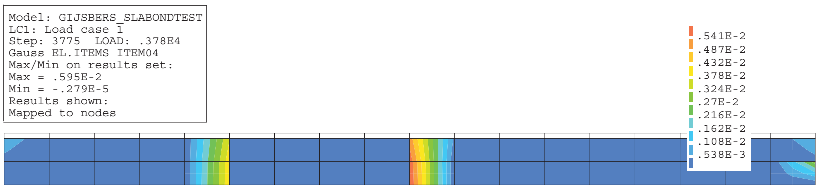


Figure 6.17: Crack strain ϵ_{cr}^{nn} at u_y=0.3 mm

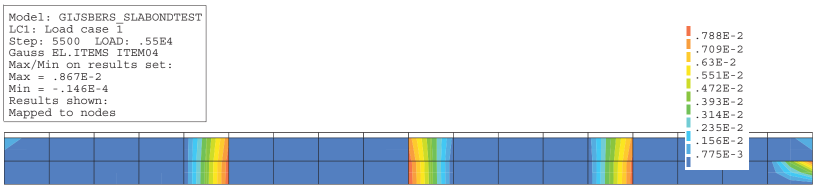


Figure 6.18: Crack strain ϵ_{cr}^{nn} at u_y=0.6 mm

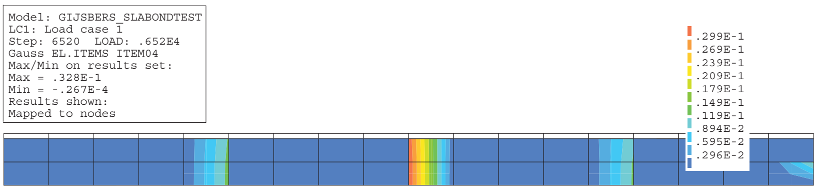


Figure 6.19: Crack strain ϵ_{cr}^{nn} at u_y=1.2 mm

termined imperfections. In all cases the crack distances are consistently between four and five elements (∼133 - 167 mm).

To gain some insight into the initiation of the first primary crack figure 6.20 shows a plot of the principal stresses along the first row of integration points in the concrete along the rebar for various load levels (up until the first damage occurs at the location of the first primary crack, load step 955). These integration points are of interest because cracks always start to originate from the rebar. The global

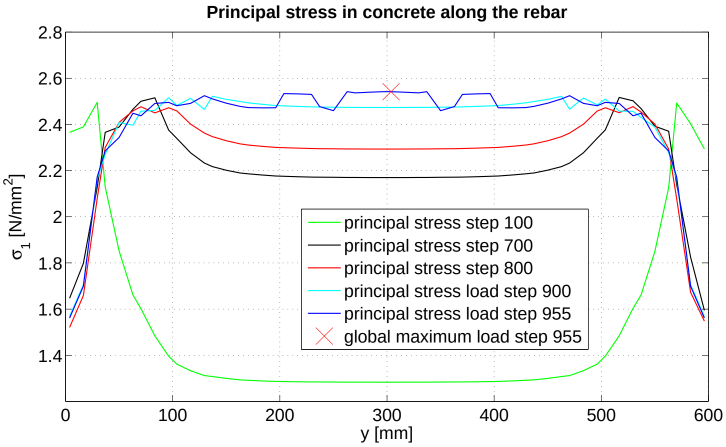


Figure 6.20: Principal stress development in the concrete along the rebar up until the start of the first primary crack localization

maximum in figure 6.20 is equal to the (uplifted) tensile strength of the first saw-tooth, this is a natural result from the SLA load scaling process. From figure 6.20 it can be seen that along the length some stress oscillation occurs. At midsection figure 6.21 shows this is more detail. It is also interesting to see the amount of

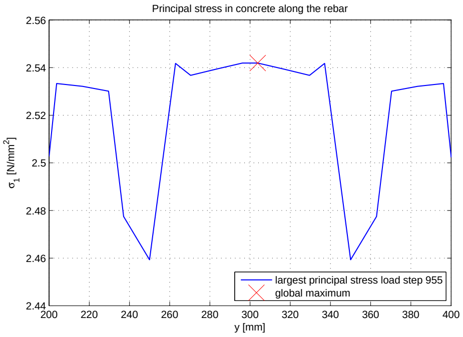


Figure 6.21: Principal stress oscillations in the vicinity of the first primary crack localization

damage that has occurred prior to the formation of the first primary crack, this is illustrated in figure 6.22. As can be seen the damage progresses from both loaded ends along the rebar. Also in this load step near the center part 4 integration points close to the rebar have already been damaged once. The current damage in figure 6.22 occurs in the centerline (for location see figure 6.21).

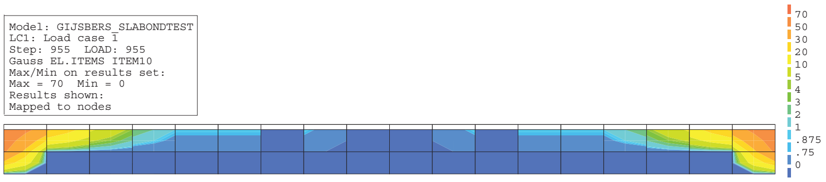


Figure 6.22: Occurred damage prior to first primary crack localization (damage indicator N-direction, 100 equals complete damage)

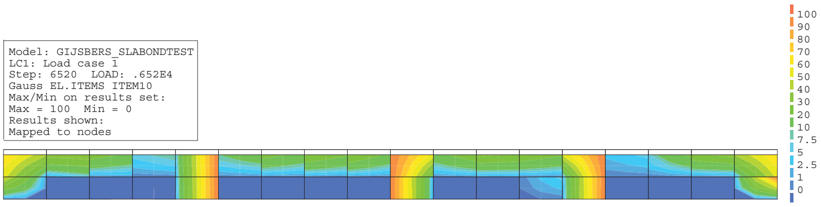


Figure 6.23: Damage at u_y=1.2 mm (damage indicator N-direction, 100 equals complete damage)

The final damage at u_y=1.2 mm (load step 6520) is plotted in figure 6.23. This reveals that large areas between the primary cracks remain completely undamaged. Figure 6.24 shows the splitting crack strains whereas figure 6.25 shows the steel yielding localization. To gain some insight into slip occurrence figure 6.26 shows the load steps with plastic slipping of elements 19 (blue) and 36 (cyan) located at both end faces (results from models with and without imperfections) whereas figure 6.27 shows the load steps with plastic slipping of elements 27 (green) and 28 (red) located on either side of the first primary crack. As can be seen from figure 6.26 because of the higher strength of the model with no imperfections, there is more slipping on the end faces prior to the development of the

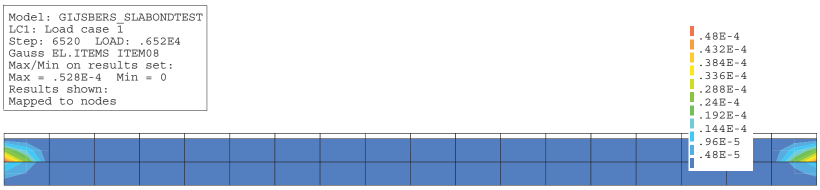


Figure 6.24: Crack strain ϵ_{cr}^{zz} at u_y=1.2 mm

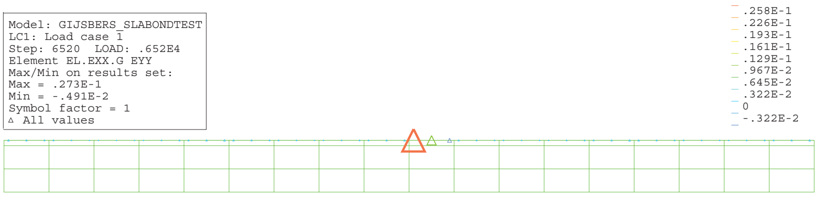


Figure 6.25: Steel strain ϵ_s at u_y=1.2 mm

first primary crack. The same holds for the slipping prior to the development of the second primary crack (figure 6.27). After a primary crack has occurred concrete cracking is governing until the previous reaction force level has been recovered at which point plastic slip will continue.

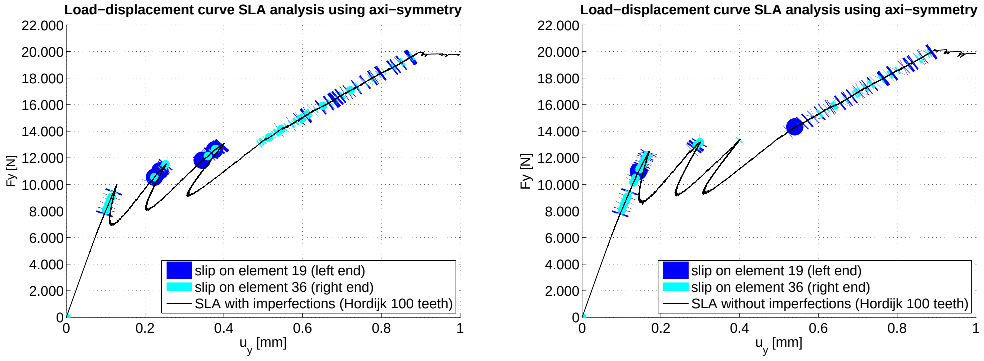


Figure 6.26: Slip occurrence at end faces, with imperfections (left), no imperfections (right)

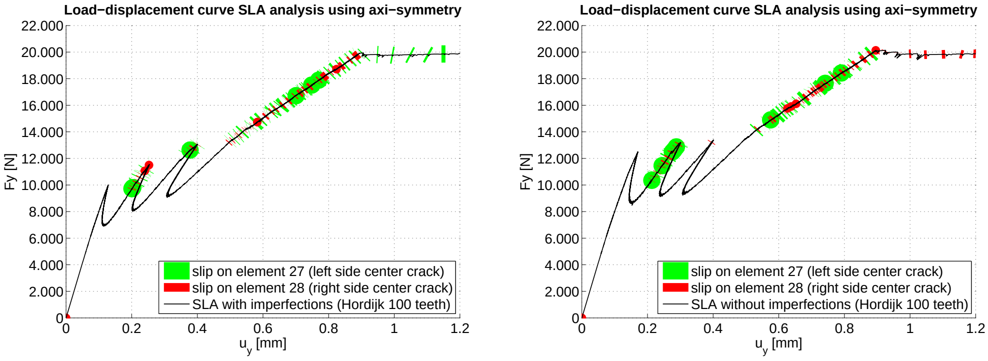


Figure 6.27: Slip occurrence near first primary crack, with imperfections (left), no imperfections (right)

6.5 Reducing number of teeth

In the previous sections calculations were presented using a great number of teeth to approximate the nonlinear relations. Recall that the accuracy of a sawtooth diagram could be calculated as the ratio between the average uplifted and downshifted value and the maximum elastic value (section 6.2). In the previous sections the ratio used was about 1.8%.

In this section the influence of reducing the number of teeth will be investigated. Using fewer teeth has the obvious advantage of less computing time. This section will investigate the influence this has on the results and in particular on the load-displacement curve and the crack locations. For this purpose the previous accuracy will be reduced stepwise by a factor of two, i.e. accuracies of 3.6%, 7.4%

and 15.4% will be used. In all calculations imperfections will no longer be used. The number of teeth used for the various accuracies as well as the sawtooth diagrams are given in appendix L. The last tooth of the concrete sawtooth diagrams is kept constant with E=4e-04 N/mm^2 and f_t=4.4e-06 N/mm^2.[1] The results of the reduced number of teeth are plotted in a load-displacement curve together with the plot of the original calculation with 100 teeth which is used as a reference, see figure 6.28.

When using less then 50 teeth for the concrete the graphs begin to show more then three local peaks indicating the primary cracks. Figures 6.29 to 6.31 show the corresponding crack locations. The calculations with 25 or 12 teeth deviate quite a lot with respect to the previous primary crack locations. From the results the primary crack distances vary between ∼100 mm and ∼167 mm (one element

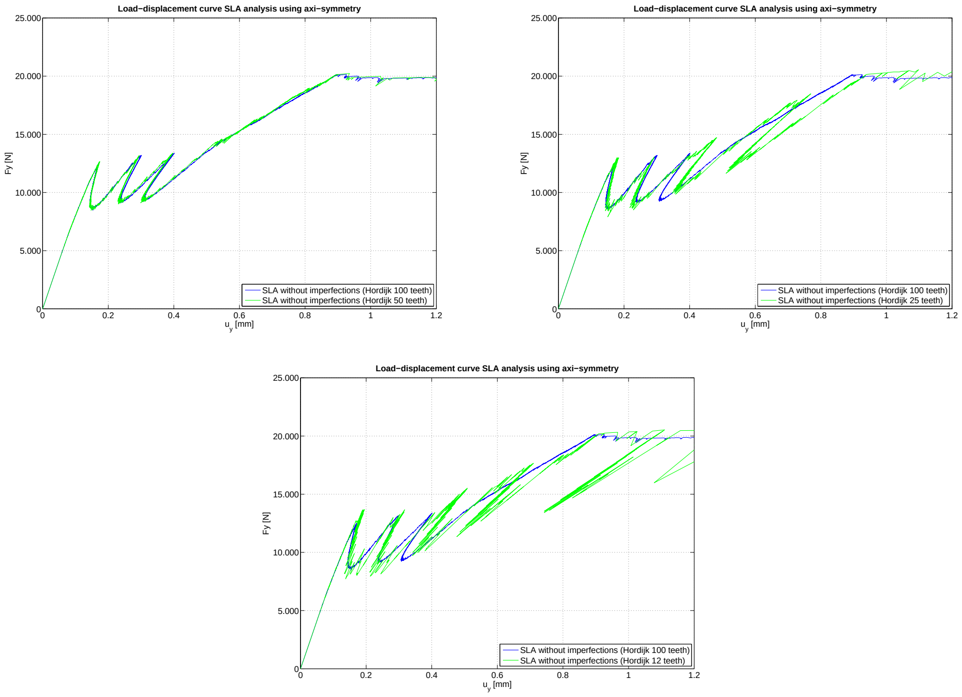


Figure 6.28: Load-displacement curves showing influence of reducing number of teeth, 50 (top left), 25 (top right), 12 (bottom) compared to 100 teeth

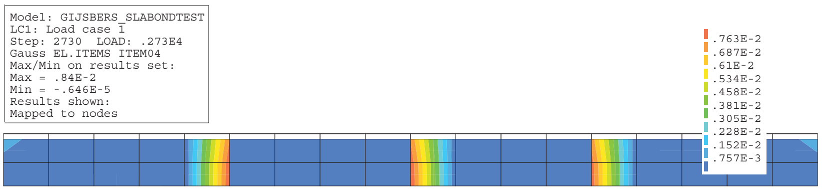

```
Model: GIJSBERS_SLABONDTEST
LC1: Load case 1
Step: 2730  LOAD: .273E4
Gauss EL.ITEMS ITEM04
Max/Min on results set:
Max = .84E-2
Min = -.646E-5
Results shown:
Mapped to nodes
```

Figure 6.29: Crack strain ϵ_{cr}^{nn} at u_y=0.6 mm (Hordijk 50 teeth)

[1] these values are equal to the last tooth of the sawtooth diagram with 100 teeth

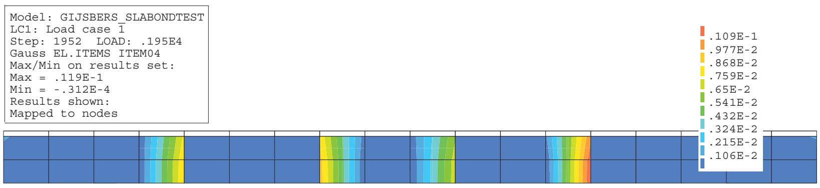


Figure 6.30: Crack strain ϵ_{cr}^{nn} at u_y=0.8 mm (Hordijk 25 teeth)

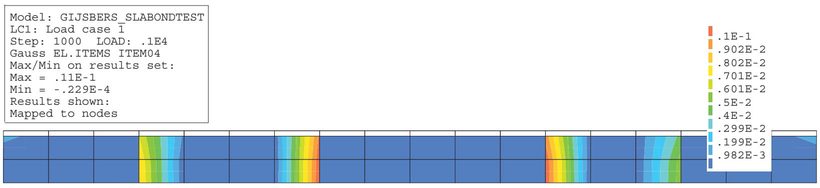


Figure 6.31: Crack strain ϵ_{cr}^{nn} at u_y=0.8 mm (Hordijk 12 teeth)

is 33.33 mm wide). The eventual crack pattern is foremost determined by the location and therefore the initiation of the first primary crack. A crack is initiated from the highest stress point in a particular cross section. The highest stress points occur near the center of the model and close to the rebar. Similar to figure 6.20, figure 6.32 shows again the principal concrete stresses in the first row of integration points along the rebar. The load steps plotted in figure 6.32 coincide with the first damage at the location of the first primary crack. The crosses indicate the global maximum (using 10 digits). The results show a great scatter in location for the various sawtooth accuracies. Also plotted in figure 6.32 is a result of using an extreme number of sawteeth for the concrete (250) as well as for the interface (250) resulting in a very smooth curve that can be used as a reference. From figure 6.32 it becomes clear that with reduced accuracy the stress oscillations increase and a larger area shows stresses close to the tensile strength. This causes the first primary crack locations to become more and more scattered. With

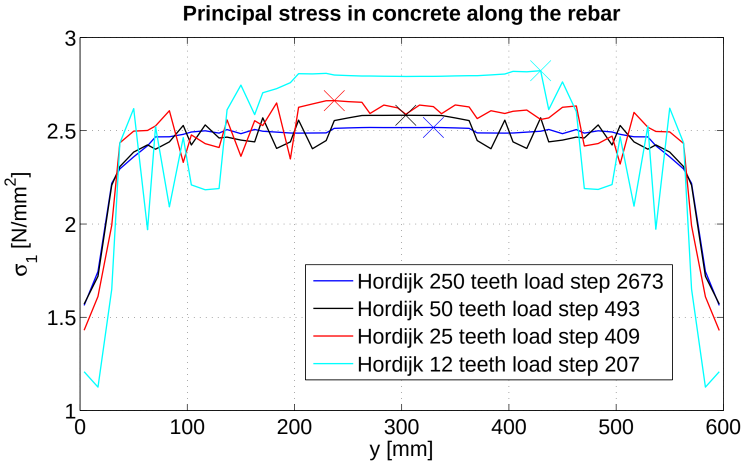


Figure 6.32: Principal concrete stresses along the rebar for various sawtooth accuracies

higher accuracy the high stress area becomes more and more confined resulting in a more consistent first primary crack location. For lower accuracies the first primary crack location is more or less a coincidence. The damage in the model prior to the initiation of the first primary crack is plotted in figures 6.33 to 6.35. From these figures is appears that with decreasing accuracies larger areas of the model become damaged prior to the first crack localization. With the used accuracies

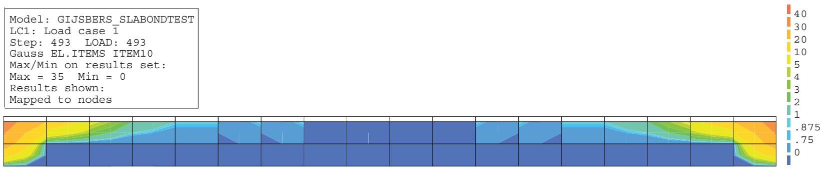


Figure 6.33: Occurred damage prior to first primary crack localization (Hordijk 50 teeth, damage indicator N-direction, 50 equals complete damage)

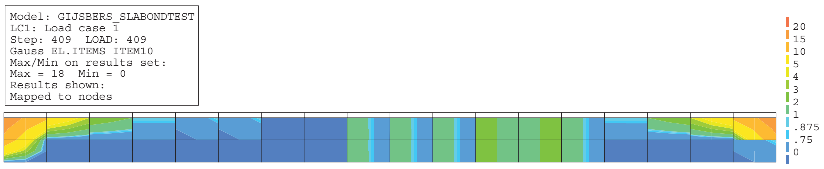


Figure 6.34: Occurred damage prior to first primary crack localization (Hordijk 25 teeth, damage indicator N-direction, 25 equals complete damage)

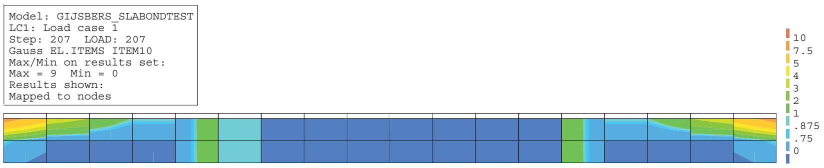


Figure 6.35: Occurred damage prior to first primary crack loacalization (Hordijk 12 teeth, damage indicator N-direction, 12 equals complete damage)

the computing time is reported in table 6.3. The time depends on the number of load steps needed as well as on the number of integration points. The relation of the number of teeth used to the number of loadsteps or the cpu time is almost linear.

Table 6.3: Influence of reducing number of teeth on number of needed steps and computing time

	100 teeth	50 teeth	25 teeth	12 teeth
number of loadsteps (u_y(max)=1.2 mm)	6550	3300	2060	1200
CPU time	326	138	81	42

6.6 Changing the shear retention behaviour

Until now a constant shear reduction factor of β=0.5 has been used (see sections 3.3 and 5.1) which is considered to be quite high. The disadvantage of using a constant factor is the severe shear stiffness reduction immediately upon cracking and the possibility of shear locking for high crack strains. An alternative method is to use a stepwise reduction (section 3.3). With this method the shear stiffness becomes a function of the normal crack strain which has a more physical basis [15], see figure 6.36. In comparison to the previous calculations using a stepwise reduction causes a stiffer initial response and a softer ultimate response. In the previous section the number of teeth was reduced stepwise. In this section the same sawtooth diagrams will be used (see tables L.1 to L.3) but recalculated with a stepwise shear reduction. Figure 6.37 shows the results of the calculations in various load-displacement curves. Compared to the previous section (figure 6.28) the results are much more consistent. With all sawtooth accuracies three primary cracks develop (see figures 6.38 to 6.41). Also the primary cracks are more curved and localized near the free edge. Figure 6.42 shows again the principal concrete stresses in the first row of integration points along the rebar. The crosses again indicate the global maximum (using 10 digits).[2] Compared to the results of a constant shear retention the oscillations of the principal stresses have somewhat diminished. Also the values of the principal stresses at both ends have dropped considerably. For comparison the damage in the model prior to the initiation of the first primary crack for the various sawtooth accuracies is plotted in figures 6.43

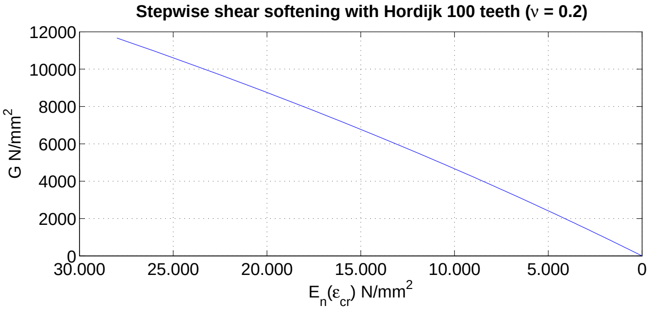


Figure 6.36: Stepwise shear reduction with Hordijk softening using 100 teeth

[2] the two crosses indicating the global maximum for 50 teeth are indeed equal stress points using 10 digits, with SLA only one will be given a strength and stiffness reduction in this loadstep

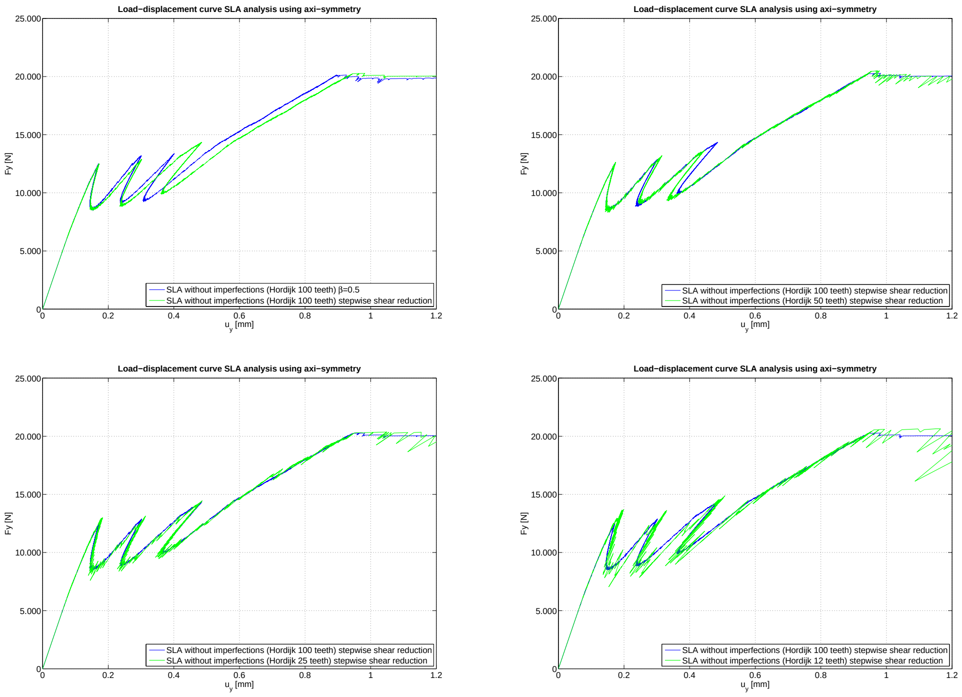
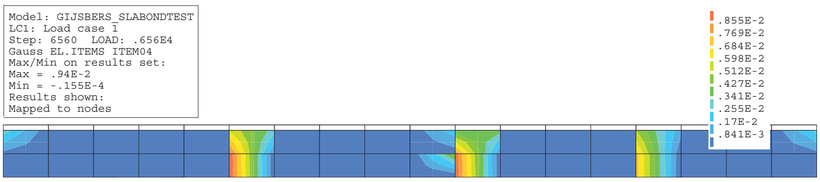
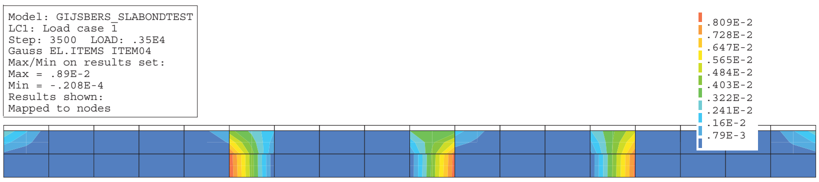


Figure 6.37: Influence of reducing number of teeth when using a stepwise shear reduction, β=0.5 versus stepwise reduction 100 teeth (top left), stepwise reduction 100 versus 50 teeth (top right), stepwise reduction 100 versus 25 teeth (bottom left) and stepwise reduction 100 versus 12 teeth (bottom right)



```
Model: GIJSBERS_SLABONDTEST
LC1: Load case 1
Step: 6560   LOAD: .656E4
Gauss EL.ITEMS ITEM04
Max/Min on results set:
Max = .94E-2
Min = -.155E-4
Results shown:
Mapped to nodes
```

Figure 6.38: Crack strain ϵ_{cr}^{nn} at u_y=0.6 mm (Hordijk 100 teeth)



```
Model: GIJSBERS_SLABONDTEST
LC1: Load case 1
Step: 3500   LOAD: .35E4
Gauss EL.ITEMS ITEM04
Max/Min on results set:
Max = .89E-2
Min = -.208E-4
Results shown:
Mapped to nodes
```

Figure 6.39: Crack strain ϵ_{cr}^{nn} at u_y=0.6 mm (Hordijk 50 teeth)

to 6.46. Especially with 25 teeth the localization has improved considerably (compare figure 6.45 to 6.34). With 12 teeth the localization is again quite poor indicating that the accuracy of this sawtooth diagram is not sufficient to get good results. With 50 or a 100 teeth the localizations are similar to the ones using a constant shear retention factor (compare figures 6.43 and 6.44 to 6.22 and 6.33) but the amount of damage at both ends is much higher.

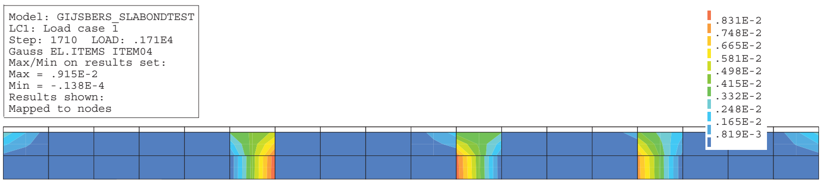


Figure 6.40: Crack strain ϵ_{cr}^{nn} at u_y=0.6 mm (Hordijk 25 teeth)

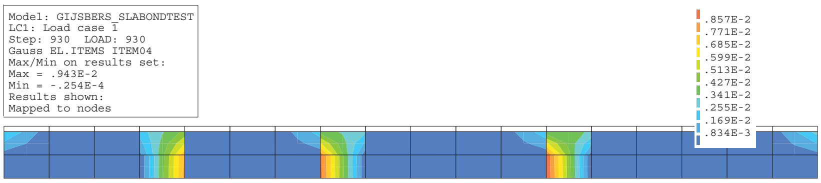


Figure 6.41: Crack strain ϵ_{cr}^{nn} at u_y=0.6 mm (Hordijk 12 teeth)

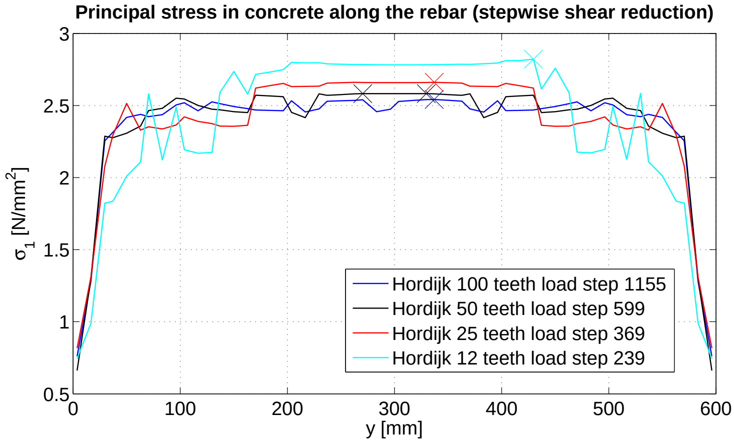


Figure 6.42: Principal concrete stresses along the rebar for various sawtooth accuracies

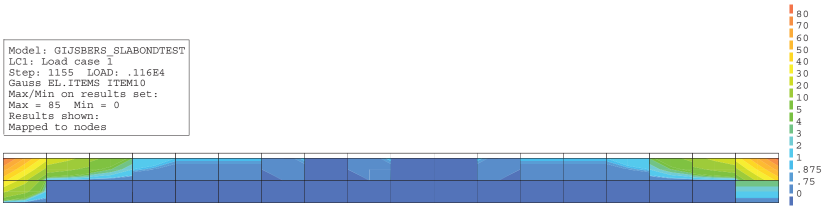


Figure 6.43: Occurred damage prior to first primary crack localization (Hordijk 100 teeth, damage indicator N-direction, 100 equals complete damage)

Table 6.4 gives the number of needed steps for the calculations compared to the previous section. In comparison the calculations with a lot of teeth use much more load steps when using a stepwise shear reduction. Strangely the calculations with 25 or 12 teeth are hardly effected in terms of needed load steps.

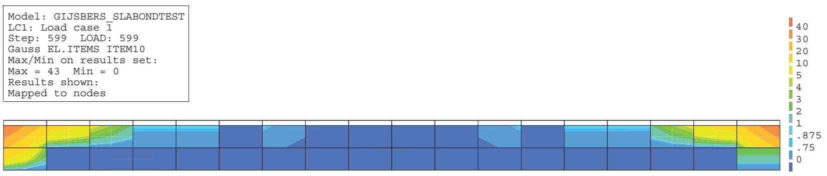


Figure 6.44: Occurred damage prior to first primary crack localization (Hordijk 50 teeth, damage indicator N-direction, 50 equals complete damage)

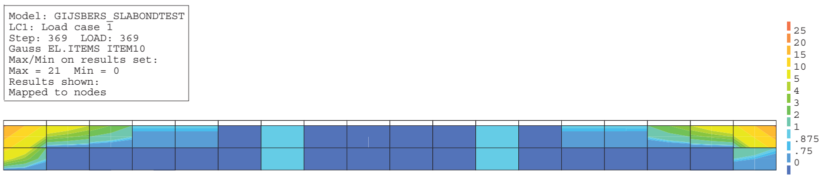


Figure 6.45: Occurred damage prior to first primary crack localization (Hordijk 25 teeth, damage indicator N-direction, 25 equals complete damage)

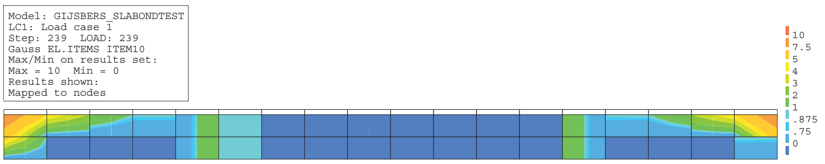


Figure 6.46: Occurred damage prior to fist primary crack localization (Hordijk 12 teeth, damage indicator N-direction, 12 equals complete damage)

Table 6.4: Influence of stepwise shear retention and reducing number of teeth on number of needed steps compared to constant shear retention

	100 teeth	50 teeth	25 teeth	12 teeth
number of loadsteps	8400	4600	2230	1220
(u_y(max)=1.2 mm)	(+28%)	(+39%)	(+8%)	(+2%)

6.7 Using mesh refinements

From the previous two sections the best results were obtained using at least 25 teeth for the concrete and when using a stepwise shear retention. In this section the influence of mesh refinements will be examined. The mesh used until now can be considered to be quite coarse. It is possible that a finer mesh will give better results especially when using fewer teeth. In this section two mesh refinements will be examined, in each case the number of elements will be doubled.

Another aspect is mesh alignment. However this is not expected to play an important role in the crack development because of the uniaxial stress distribution. Also a random mesh would raise some questions regarding the crack band width (as it would differ for each element) and without an automated sawtooth generator this aspect can currently not be explored in a practical sense. Some investigation into mesh alignment with SLA can be found in [15].

6.7.1 Used refinements

With the first mesh refinement the number of elements is doubled in x- and y-direction to 36 and 4 respectively, see figure 6.47. For the investigation again Hordijk softening will be used with various accuracies (i.e. number of teeth), see appendix M. For the interface and the steel bar fixed sawtooth diagrams will be used (see section 6.2 figure 6.3). This is a mere practical choice and will not have a significant effect on the results. With all calculations stepwise shear retention will be used (see section 6.6).

The crack band width is calculated as the spacing between the integration points: h=600/(36x3)=5.5556 mm (a 3x3 integration scheme is used). The second mesh refinement is analogous to the first now with 72 and 8 elements in x- and y-direction respectively, see figure 6.48. The crack band width for this mesh is: h=600/(72x3)=2.7778 mm.

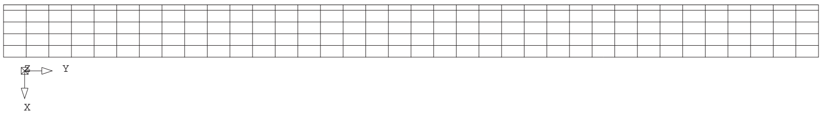


Figure 6.47: First mesh refinement

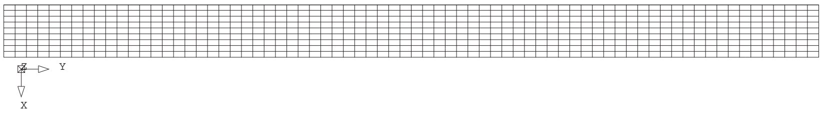


Figure 6.48: Second mesh refinement

6.7.2 Results

Contrary to the expectation the first mesh refinement does not give better results compared to the previous mesh when fewer teeth are used for the concrete sawtooth diagram. The problem of increased damage with decreasing accuracy for the concrete sawtooth approximation prior to the development of the first primary crack, which was already witnessed in the previous two sections, is again prevailing. However it is now already playing an important role even when a 100 teeth are used. Figure 6.49 shows the damaged areas (indicated in red) prior to the first primary crack which clearly shows that with all accuracies used there is now more damage in the center part of the model in comparison to the previous mesh. The excessive damage especially with 50, 25 and 12 teeth also has the disadvantage of needing a relatively large number of load steps. To check if increasing the number of teeth even further then 100 would yield better results another calculation is

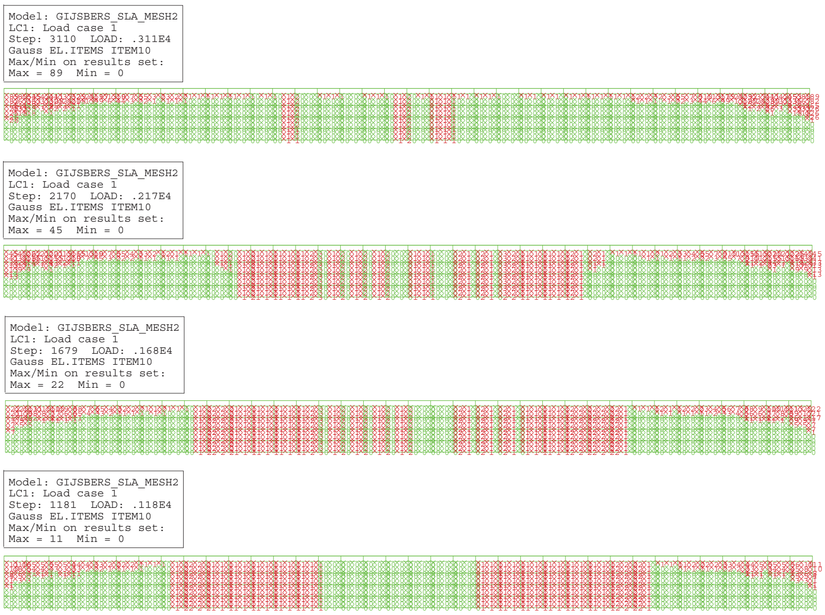


Figure 6.49: Damaged integration points (indicated in red) prior to first primary crack, from top to bottom: Hordijk 100, 50, 25 and 12 teeth

performed now using 200 teeth. Figure 6.50 shows the result in terms of damaged areas prior to the first primary crack localization. Clearly the result has improved considerably. The obvious downside of using so many teeth in conjunction with the use of a fine mesh is the needed number of load steps (34700). Figure 6.50 again demonstrates the previous encountered behaviour of the cracking sequence in a tension-pull experiment, meaning that cracking progresses from both ends along the rebar followed by a swift and clear primary crack localization where ultimately large areas of the model remain completely undamaged (see also figure 6.23). The problem of localization becomes worse in terms of damaged areas with the second mesh refinement, see figure 6.51.[3] The results of both mesh refinements are also plotted in load-displacement curves together with the results of the previous mesh for various sawtooth accuracies, see figure 6.52. At this time it should be noted that despite the aforementioned localization problem eventually with SLA localization does always take place, albeit at a higher load level. This is especially visible with the second mesh refinement when using only 12

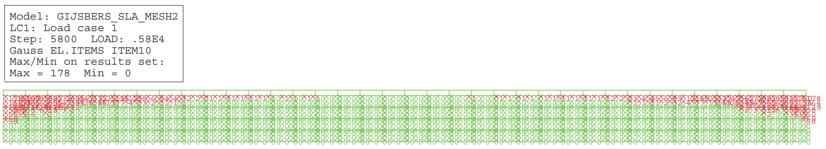


Figure 6.50: Damaged integration points (indicated in red) prior to first primary crack with Hordijk 200 teeth

[3] with the second mesh refinement no calculation with a 100 teeth is performed due to needed computing time

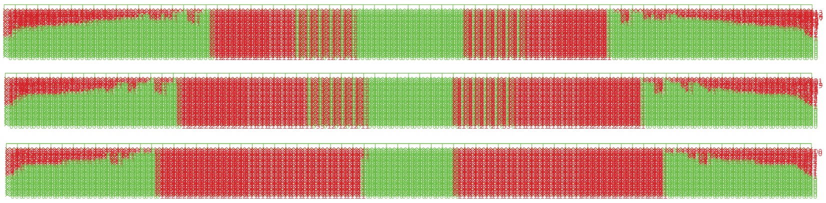


Figure 6.51: Damaged integration points (indicated in red) prior to first primary crack, from top to bottom: Hordijk 50, 25 and 12 teeth

teeth (see figure 6.52 bottom right). Instead of a quick snap-back the graph shows increasing load on the first and second peak and is very spiked. The principal stress oscillation that was encountered previously (section 6.6) is shown in figure 6.53 for the second mesh refinement with 50 and 12 teeth. The load steps plotted again coincide with the first damage at the location of the first primary crack. The periods of the stress oscillations compared to the previous mesh have decreased considerably. The graph with 12 teeth shows an increase in amplitude compared to the graph with 50 teeth. Changing the *DIANA* solver to a so called direct solution method (i.e. prohibiting the use of numeral methods) did not resolve this problem.

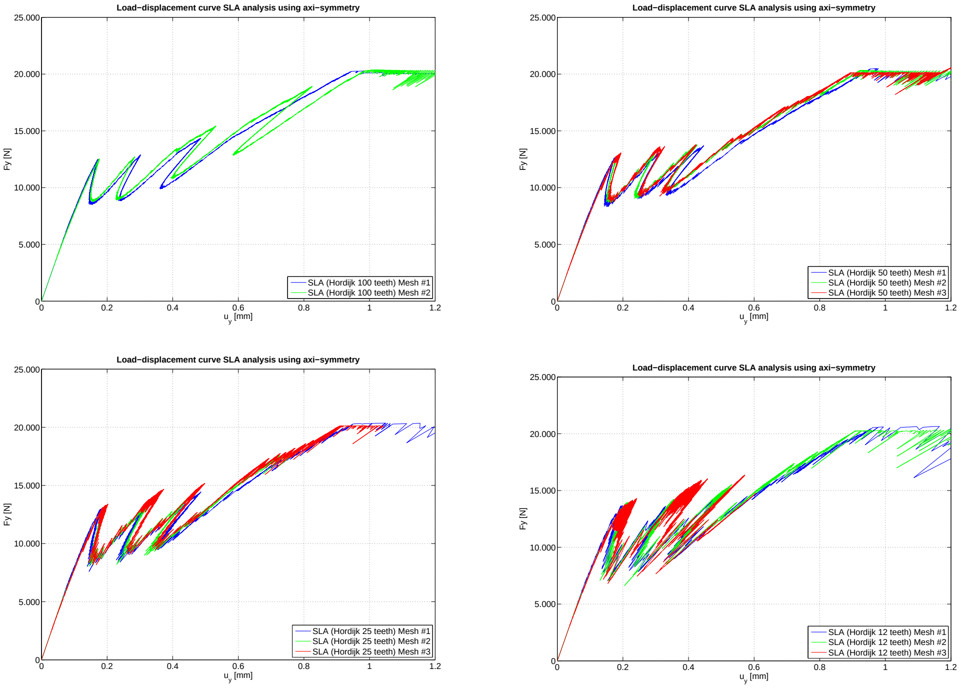


Figure 6.52: Influence of number of teeth and mesh refinements, 100 teeth (top left), 50 teeth (top right), 25 teeth (bottom left) and 12 teeth (bottom right)

The cause of the observed behaviour of large damaged areas when using fewer teeth and with mesh refinements is believed to be the ratio between the stiffness

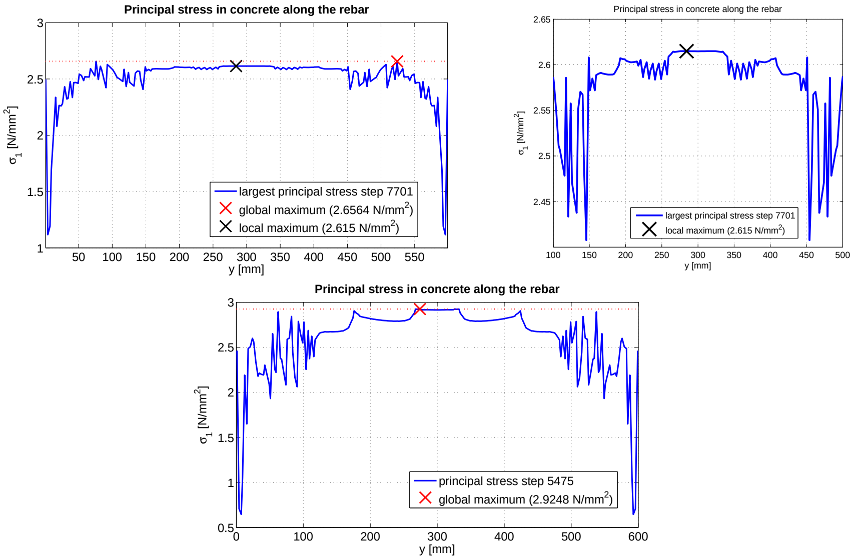


Figure 6.53: Principal concrete stresses along the rebar with second mesh refinement, Hordijk 50 teeth (top) versus Hordijk 12 teeth (bottom)

of the interface and the concrete. In all cases the damaged areas progress from the ends towards the middle (like a crack "front"). With a low accuracy sawtooth diagram the local stiffness reduction upon cracking is higher compared to a high accuracy sawtooth diagram. This changes the local stiffness ratio introducing a stress concentration (i.e. making the interface stiffness relatively high). This stress concentration will subsequently initiate the next crack creating the observed "chain effect". An indication of this was also observed with the calculations in the next chapter were different stiffness values for the interface were used. With a high stiffness interface stress concentrations are found near the loaded ends resulting in the formation of the first primary cracks at these locations instead of in the middle.

Finally the number of loadsteps needed for all meshes used is reported in table 6.5. The relation of the number of concrete elements to the number of loadsteps needed is almost linear.

Table 6.5: Influence of mesh refinement and number of teeth on number of needed loadsteps

number of loadsteps (u_y(max)=1.2 mm	100 teeth	50 teeth	25 teeth	12 teeth
original mesh (36 concrete elements)	8400	4600	2230	1220
first mesh refinement (144 concrete elements)	24400	10300	6000	4100
second mesh refinement (576 concrete elements)	X	32200	23300	13000

6.8 Summary and conclusions chapter 6

In this chapter the tension-pull experiment that was calculated in the previous chapter using axi-symmetric elements and using a non-linear analysis is recalculated with SLA (sequentially linear analysis). For this the previously derived extensions for SLA, i.e. the axi-symmetric and interface elements, are used. For the calculations the non-linear Hordijk softening curve of concrete in tension is approximated with a sawtooth calculated with the previously derived sawtooth generator. At first a calculation is performed using the same imperfections that were used in the non-linear analysis to be able to compare results. Next a calculation is performed without these imperfections to investigate the possibilities in SLA. A number of other aspects is also investigated including the use of various accuracies for the sawtooth approximations, changing the shear retention behavior and the use of mesh refinements. The following conclusions regarding the SLA calculations can be stated:

SLA calculation using imperfections

1. Using high accuracy sawtooth approximations all results are in excellent agreement with the non-linear analysis. There are hardly any differences found except in the load-displacement curve with SLA also snap-backs are captured after each primary crack. This indicates SLA is a good alternative to non-linear analysis when using axi-symmetric and interface elements.

2. Despite the imperfections used the SLA analysis still shows somewhat better crack localization, especially at the loaded ends, compared to the non-linear analysis.

3. The bond stress profiles found in the SLA analysis are in good agreement with the non-linear analysis indicating that the adopted sawtooth approximation for bond-slip and the SLA code for interface elements work correctly.

4. The ultimate load is governed almost completely by the yielding of the rebar. The results show an ultimate load in the load-displacement curve equal to the one found with the non-linear analysis. This indicates it is better to use downshifting only and no uplifting of the “mother” curve for the sawtooth approximation of steel plasticity.

SLA calculation without imperfections

1. Despite the uniform stress distribution in case of a tension-pull experiment and no imperfections in the model the SLA analysis again shows excellent localization resulting in the formation of three primary cracks. This result would not be possible with a non-linear analysis as cracking would appear everywhere simultaneously in case of a uniform stress distribution. This justifies the conclusion that SLA is superior to non-linear analysis in case of a tension-pull experiment.

2. The primary cracks always originate from the integration points close to the rebar. A principal stress plot along the rebar at the moment of primary crack initiation shows that some stress oscillation occurs.

3. Large areas between the primary cracks remain completely uncracked similar to the result of the non-linear analysis where imperfections are used. This again indicates the excellent localization capability of SLA.

4. It has been found that slip of the interfaces occurs just prior to the formation, and close to, each primary crack and the end faces. The snap-backs in the load-displacement curve are completely governed by concrete cracking only. After a fully developed primary crack pattern and at a higher load level slip of the interfaces will continue.

Reducing number of teeth

1. The aforementioned principal stress oscillation along the rebar amplifies with decreasing accuracy for the sawtooth approximations. Also the cracked area in the model prior to the development of the first primary crack increases. Although primary crack localization does always take place, with decreasing accuracy the locations of the primary cracks become more and more scattered.

2. Reducing the accuracy for the steel sawtooth diagram hardly affects the ultimate load capacity. Using a lower accuracy only results in a more spiked load-displacement curve.

3. With the used elasto-plastic bond-slip relation the accuracy for the sawtooth approximation does not have a large effect on the overall results, especially the load-displacement curve. Detailed results however, specifically the bond stress profiles, are affected and become less smooth with decreasing accuracy.

4. It has been found that the relation between the number of teeth used for the sawtooth approximation of the concrete and the CPU time is almost linear. Since many more loadsteps involve concrete cracking compared to bond-slip or steel yielding the sawtooth accuracy for the latter two are of less importance with respect to computing time.

Changing the shear retention behavior

1. Using a stepwise shear retention instead of a constant shear retention factor, apart from having a more physical basis, improves the results. With reduced number of teeth the damaged areas prior to the development of the first primary crack have somewhat diminished. The results for the various accuracies of the sawtooth approximations are now more comparable to each other (i.e. more consistent).

2. The aforementioned principal stress oscillations along the rebar have also diminished with the use of stepwise shear retention compared to a constant shear retention factor. This also results in more consistent primary crack locations for the various sawtooth accuracies.

3. The number of loadsteps needed when using a stepwise shear retention instead of a constant shear retention factor has increased considerably when using high accuracy sawtooth approximations (+30-40%). For lower accuracies the increase in needed load steps is much less.

Using mesh refinements

1. To get comparable results with a finer mesh the number of sawteeth used for the concrete sawtooth approximation has to be increased instead of decreased. Both the principal stress oscillations as well as the damaged areas prior to the development of the first primary crack reported earlier have increased when a finer mesh is used.

2. The cause of the increased cracking in case of a mesh refinement appears to be related to the stiffness ratio between the interface and the concrete. With a low accuracy sawtooth diagram the local stiffness reduction upon cracking is higher compared to a high accuracy diagram. This change in the local stiffness ratio appears to introduce a stress concentration which will subsequently initiate the next crack ultimately creating the large cracked areas.

3. From the results it becomes clear that using more teeth for the concrete sawtooth approximation is much more effective and will give better results then using a finer mesh in case of a tension-pull experiment.

Chapter 7

SLA calculations of the tension-pull experiments by Mayer

7.1 Introduction

In this chapter several tension-pull experiments that were conducted by U. Mayer [12] will be calculated with SLA. The experiments all consist of long specimens (2.3 to 2.9 m) and have various (square) cross sections, various concrete qualities (B25 and B45) and also different reinforcement ratios as well as rebar diameters (from 6 to 25 mm). Figure 7.1 shows the test setup and the the placement of the strain gauges used with the experiments. With all these variations these experiments are perfectly suited for using the bond model described in chapter 2 to calculate the bond-slip curves and the necessary sawtooth approximations (see section 4.3). The report by Mayer does not contain load-displacement curves or complete crack patterns. Therefore direct comparisons of the results of the calculations in this chapter to the experiments is not possible. From the long strain gauges of 2 meters placed in the center (“WA 2000” see figure 7.1) average strain-stress curves have been reported. Therefore these will also be reported with the SLA calculations. However no variations of parameters will be explored to fit the experimental data. Instead the emphasis will be put on the general (cracking) behaviour under the influence of the various configurations and making qualitative comparisons where possible.

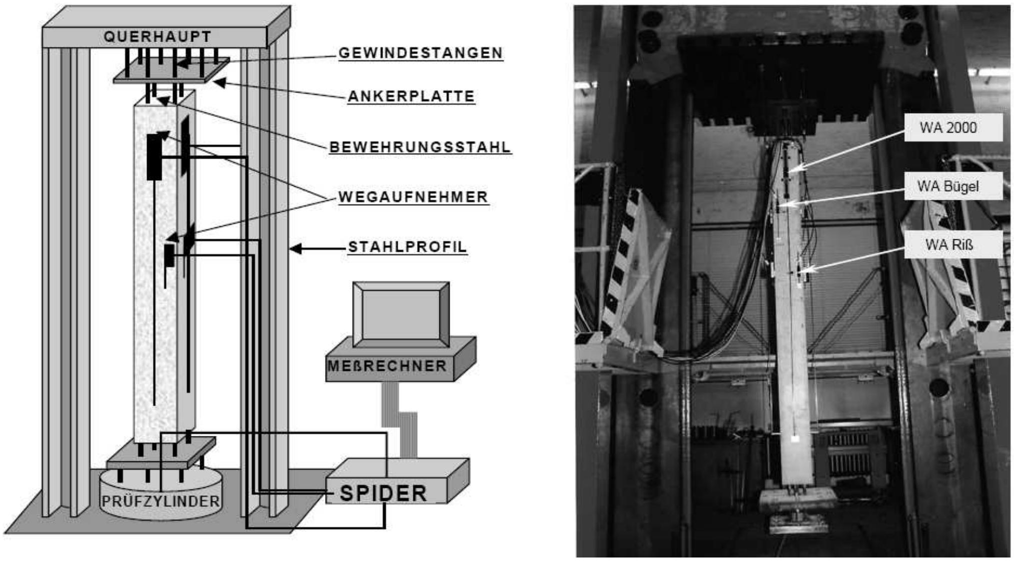
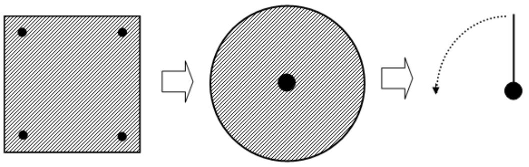


Figure 7.1: Mayer test setup and placements of strain gauges

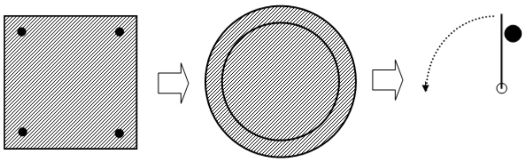
7.2 Discretization options

The square reinforced concrete cross-section with rebars located in the corners and/or near the sides used in the experiments can be discretizised to a finite element mesh in various ways:

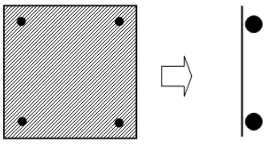
1. Transform into an equivalent circular cross section and then use axi symmetry with rebars lumped to the axis of rotation:



2. Transform into an equivalent circular cross section and then use axi symmetry with rebars lumped to a position with equal concrete cover:



3. Use plane stress for the square concrete cross section with rebars lumped to actual positions:



Since this thesis is directed to axi-symmetric elements in SLA the second option is chosen to be most appropriate. This will also simulate the cracking behaviour in the vicinity of the concrete cover. The mesh will therefore again consist of axi-symmetric elements for the concrete, truss elements for the rebar and interface elements for bond-slip. To have a correct stiffness for the interface again the *DIANA* configuration option “plane stress” is used (see also section 5.5.1) but now the sum of the circumferences of all rebars is taken as the interface area.

7.3 Overview of calculated experiments

In table 7.1 an overview is given of the different tension-pull experiments that will be calculated in this chapter (see also [12] table 4.1). As stated previously the location of the lumped rebar (radius from the axis of symmetry) is determined with the condition of an equal concrete cover, e.g. for S4D12AB25: r_{rebar}=169.26-30-12/2=133.3 mm. The mesh is chosen to be quite coarse (see also section 6.7).

code	reinforcement and concrete cover [mm]	reinforcement ratio [%]	steel class	concrete type	dimensions bxdxL [mm]	equivalent concrete radius [mm]
S4D12AB25	4Ø 12 (c=30)	0.50	A	B25	300x300x2500	169.26
S8D12AB25	8Ø 12 (c=30)	1.00	A	B25	300x300x2500	169.26
S8D12AB45	8Ø 12 (c=30)	1.00	A	B45	300x300x2500	169.26
S4D16AB25	4Ø 16 (c=35)	0.50	A	B25	400x400x2700	225.68
S6D16AB25	6Ø 16 (c=35)	0.75	A	B25	400x400x2700	225.68
S2D25AB25	2Ø 25 (c=35)	0.60	S	B25	400x400x2900	225.68
S4D25AB25	4Ø 25 (c=35)	1.20	S	B25	400x400x2900	225.68

Table 7.1: Overview and codes of calculated experiments

Since the actual concrete dimensions are a lot bigger compared to previous calculations a somewhat larger number of elements is used. The mesh is also determined by the condition to have nodes at the edges of the two meter strain gauges. Figures 7.2 to 7.4 show the mesh used for the various dimensions together with the crack band width h. For the axi-symmetric elements again a 3x3 integration scheme is used. In thickness direction one element is used to represent the concrete cover whereas 4 elements are used to model the distance from the axis of rotation to the reinforcement.

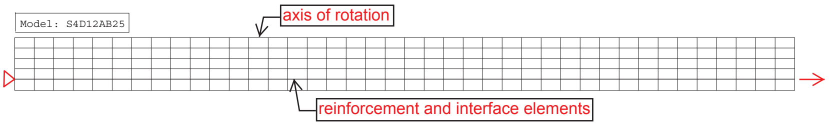


Figure 7.2: Mesh for dimensions 300x300x2500 mm (40 elements in longitudinal direction), crack band width: h=2500/(40x3)=20.83 mm

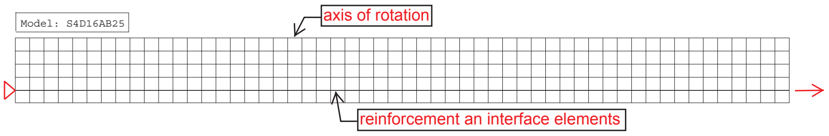


Figure 7.3: Mesh for dimensions 400x400x2700 mm (54 elements in longitudinal direction), crack band width: h=2700/(54x3)=16.67 mm

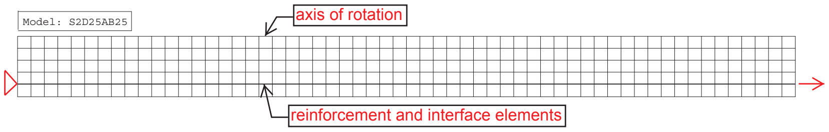


Figure 7.4: Mesh for dimensions 400x400x2900 mm (58 elements in longitudinal direction), crack band width: h=2900/(58x3)=16.67 mm

7.4 Material properties

The concrete material properties that will be used in the calculations are given in table 7.2. The fracture energy was not reported in [12], therefore it is calculated with the CEB-FIB MC90 model code. G_{f0} is determined with a maximum aggregate size of 16 mm which was reported by Mayer being used in the experiments.

Table 7.2: Concrete material properties

concrete type	average cube compressive strength f_{ccm} [N/mm^2]	average tensile strength f_{ctm} [N/mm^2]	average modulus of elasticity [N/mm^2]	fracture energy [N/mm]
B25	28.4	2.1	30000	0.053
B45	64.7	3.3	37400	0.099

$$\text{B25: } G_f = G_{f0}\left(\frac{f_{cm}}{10}\right)^{0.7} = 0.030\left(\frac{0.80\cdot 28.4}{10}\right)^{0.7} = 0.053\ N/mm \qquad (7.1)$$

$$\text{B45: } G_f = G_{f0}\left(\frac{f_{cm}}{10}\right)^{0.7} = 0.030\left(\frac{0.85\cdot 64.7}{10}\right)^{0.7} = 0.099\ N/mm \qquad (7.2)$$

The material properties for steel are given in the table 7.3. The Poisson's ratio used for the concrete as well as the steel is 0.2.

Table 7.3: Steel material properties

diameter [mm]	steel class	A_s [mm^2]	f_y [N/mm^2]	E_s [N/mm^2]
12	A	115.9	532	196000
16	A	199.8	519	203000
25	A	478.5	585	204000

7.5 Bond-slip sawtooth diagrams

The sawtooth diagrams used for the concrete and steel will not be given here, instead see appendix N. As mentioned in the beginning of this chapter the bond-slip relations for the various simulations will be calculated with the Matlab bond model. In most cases, but not all, pull-out is the governing failure mode. Since the bond-slip relation for pull-out failure is strongly dependent on the steel strain (see section 2.8), this influence will be taken into account. In fact the best way would be to update the bond-slip relation continuously during the calculation as the steel strains changes. Since this is not possible instead a fixed steel strain (i.e. an average steel strain) is taken into account with a value equal to half the maximum elastic strain: $\epsilon_s \approx$0.001. Figure 7.5 shows the various calculated bond-slip curves as well as the value for the initial elastic stiffness.[1] The effective concrete cover is calculated from the cover to the outside and a maximum cover to the inside of 4xds. As can be seen from these figures the various bond-slip relations are very different in terms of strength and stiffness.

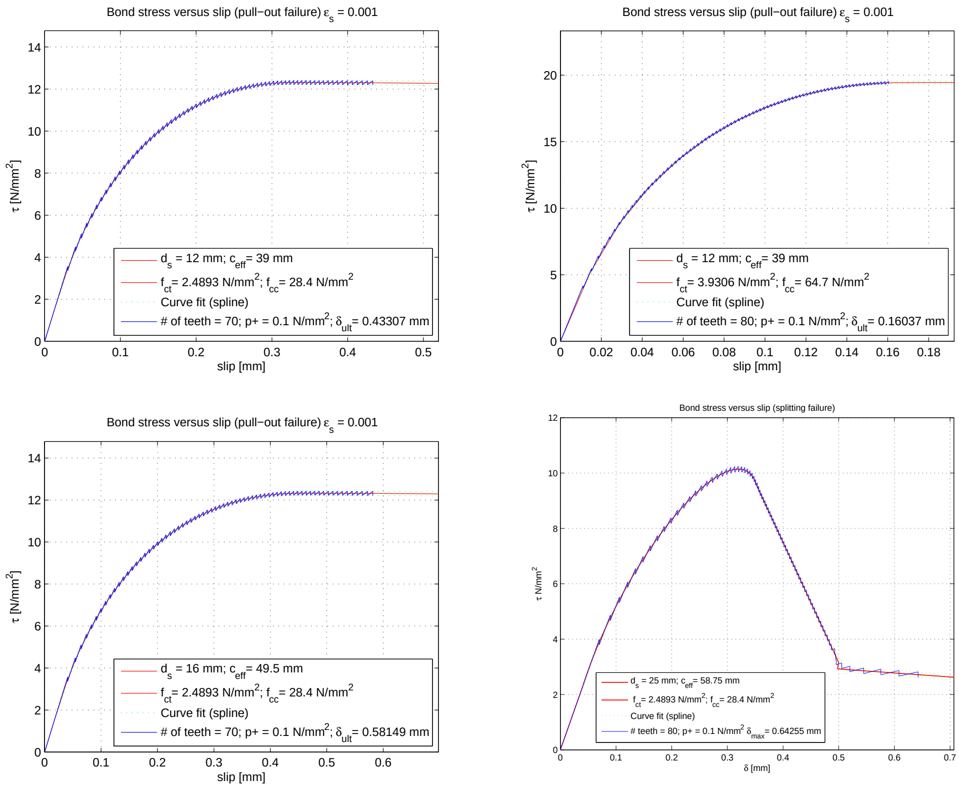


Figure 7.5: Bond-slip sawtooth used for simulations: S4D12AB25 and S8D12AB25 ($S_{t,el}$=116.7 N/mm^3) (top right), S8D12AB45 ($S_{t,el}$=377.3 N/mm^3) (top left), S4D16AB25 and S6D16AB25 ($S_{t,el}$=87.4 N/mm^3) (bottom left), S2D25AB25 and S4D25B25 ($S_{t,el}$=57.1 N/mm^3) (bottom right)

[1] in the bond model the concrete tensile strength is calculated from the compressive strength (see section 2.3), therefore the values for the tensile strength in figure 7.5 deviate from the ones given in section 7.4

7.6 Results tension-pull experiments

In this section the influence of several parameters on the cracking behaviour in a tension-pull experiment will be investigated using the results of the various calculations. The calculations are refered to by code, see table 7.1 section 7.3. Some detailed results can also be found in appendices O to U.

7.6.1 Influence of reinforcement ratio

In this section the influence of the reinforcement ratio on the cracking behaviour of the tension-pull experiments will be examined. For this two simulations will be compared: S4D12A and S8D12A. All material parameters are equal except with S8D12A the reinforcement ratio is 1.00% whereas with S4D12A it is 0.50%. The mesh used in both calculations is given in figure 7.2 section 7.3. Both use the same bond-slip relation (see figure 7.5). From a modelling perspective only the circumference for the interface elements and the truss cross section, representing the lumped rebars, differs. Figure 7.6 shows the results of the calculations in a load-displacement curve. For S4D12A and S8D12A about 15500 and 37000 load steps are needed respectively. Both graphs show extremely sharp drops every time a primary crack occurs. It should be noted however that the concrete dimensions are very different from the calculations in the previous chapters. The final crack patterns are shown in figure 7.7. The complete crack development is given in appendices O and P. No principal stress oscillations occur, indicating that the used sawtooth approximation for concrete is sufficiently accurate, and in both cases the cracking starts at midpoint (see figure 7.8). The influence of increased reinforcement is in good agreement with what one would expect. In simulation S8D12A secondary cracks occur between all primary cracks due to the increased reinforcement and the primary crack distance is cut in halve. The damage after the formation of the primary crack pattern is shown in figure 7.9. This shows that with increased reinforcement the cracks are more dispersed and the crack width decreases (note that the scaling is equal in both figures).

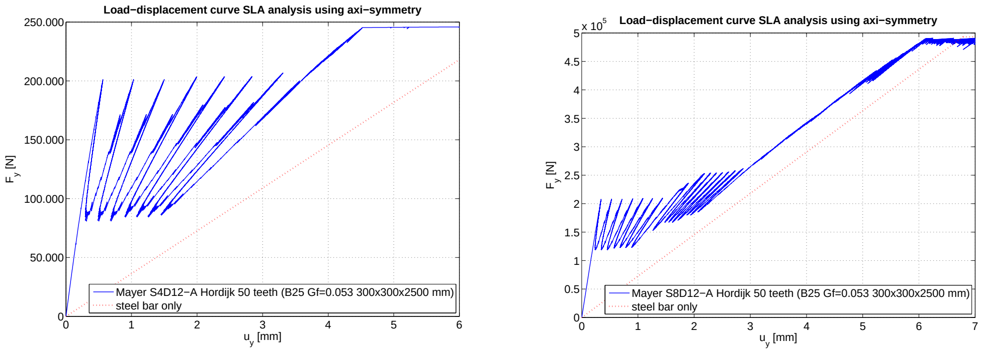


Figure 7.6: Load-displacement curve simulation S4D12A (left) versus S8D12A (right)

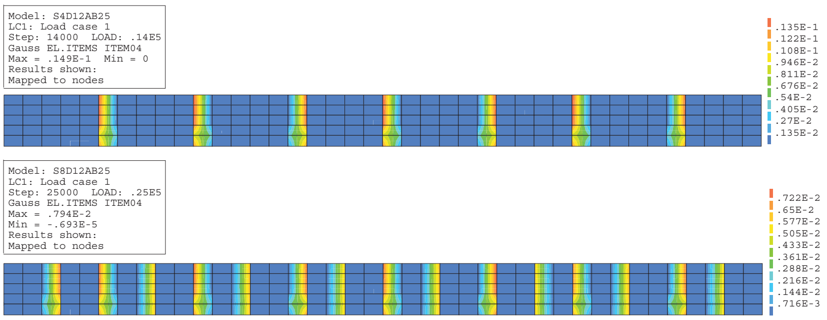


Figure 7.7: Final crack pattern simulation S4D12AB25 (top) and S8D12AB25 (bottom)

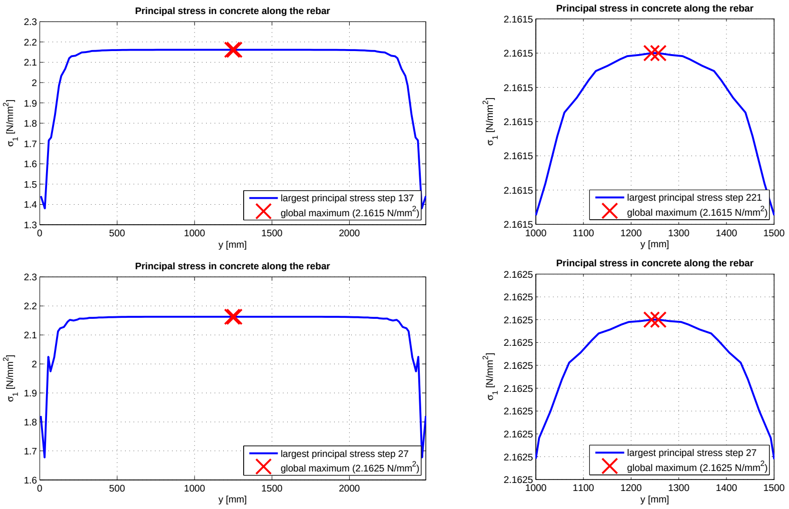


Figure 7.8: Principal concrete stresses along inner side of rebar at first primary crack initiation in simulation S4D12A (top) and S8D12A (bottom)

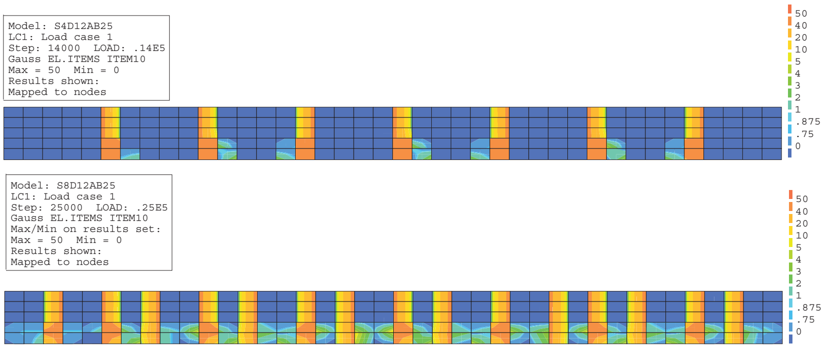


Figure 7.9: Damage simulation S4D12AB25 at u_y=3.65 mm and S8D12AB25 at u_y=3.40 mm (damage indicator N-direction, 50 equals complete damage)

7.6.2 Influence of concrete type

In this section the influence of concrete type on the cracking behaviour of the tension-pull experiments will be examined. For this two simulations will be compared: S8D12A with B25 and B45. All other material parameters are equal. The mesh used in both calculations is given in figure 7.2 section 7.3. The bond-slip relation of course now differs (see figure 7.5). From a modelling perspective apart from the bond-slip relation for the interface elements also the concrete material properties differ. Figure 7.10 shows the results of the calculations in a load-displacement curve. For S8D12AB25 and S8D12AB45 about 37000 and 47000 load steps are needed respectively. Since the ultimate load level is governed almost entirely by steel yielding it is equal in both simulations. The final crack patterns are shown in figure 7.11. The complete crack development is given in appendices P and Q. It can be seen that the average crack distance differs, with B25 the average crack distance is about 156 mm whereas with B45 it is about 132 mm.

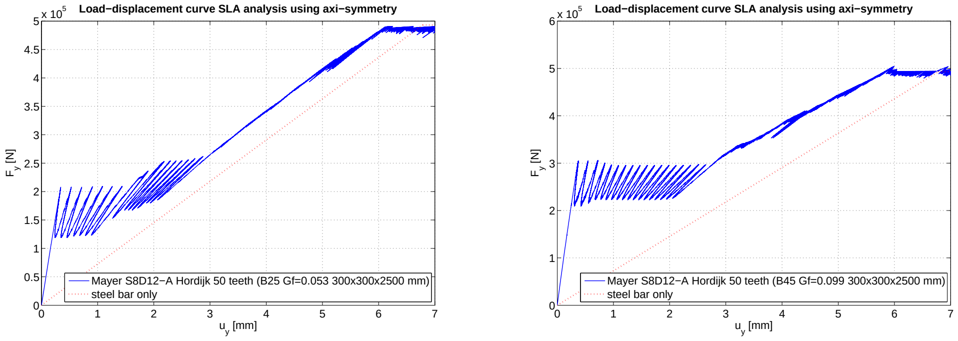


Figure 7.10: Load-displacement curve simulation S8D12AB25 (left) versus S8D12AB45 (right)

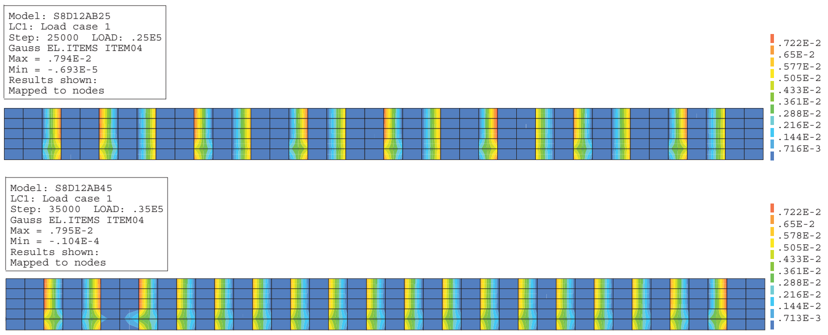


Figure 7.11: Final crack pattern simulation S8D12AB25 (top) and S8D12AB45 (bottom)

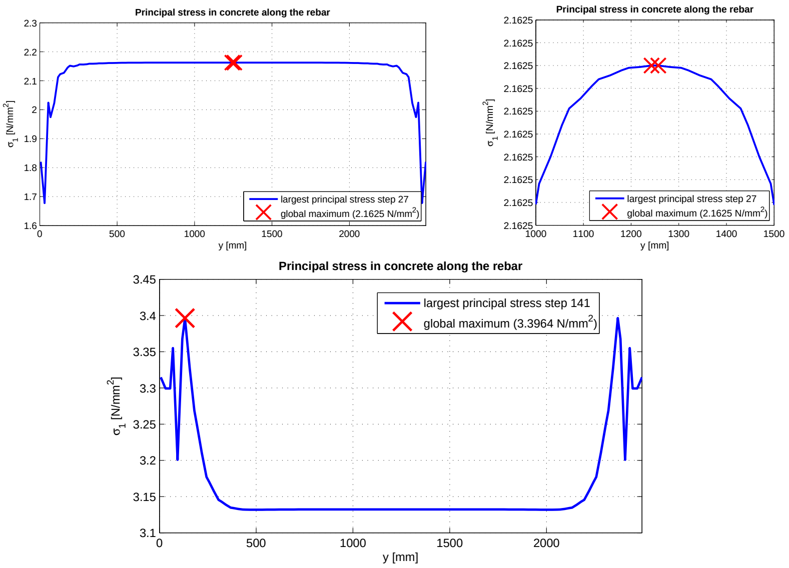


Figure 7.12: Principal concrete stresses along inner side of rebar at first primary crack initiation in simulation S8D12AB25 (top) and S8D12AB45 (bottom)

The concrete principal stresses along the rebar at first primary crack initiation (figure 7.12) show how the cracking starts for both simulations. From this, and from the complete crack development for S8D12AB45 in appendix Q, it can be seen that with B45 cracking starts near a loaded end instead of in the middle. This behaviour is due to the increased relative bond-slip stiffness (see equations 7.3 and 7.4) which causes a much quicker load transfer from the rebar into the surrounding concrete and therefore creating a local high stress point (the amount of concrete surrounding the rebar, i.e. the concrete cover, is insufficient to disperse this local high stress).

$$\text{S8D12AB25: ratio} = \frac{OS_{t,el}}{E_c} = \frac{150.796\text{x}116.7}{30000} = 0.587 \tag{7.3}$$

$$\text{S8D12AB45: ratio} = \frac{OS_{t,el}}{E_c} = \frac{150.796\text{x}377.3}{37400} = 1.521 \tag{7.4}$$

With:

O	total circumference of all rebars
$S_{t,el}$	initial elastic stiffness of interface (see also figure 7.5)
E_c	concrete elastic stiffness

Finally figure 7.13 shows the damage for both simulations after the formation of the primary crack pattern. This shows that with B45 the cracks are more dispersed and the crack width slightly decreases (note that the scaling is equal in both figures). Also with B45 cracks appear near the axis of rotation.

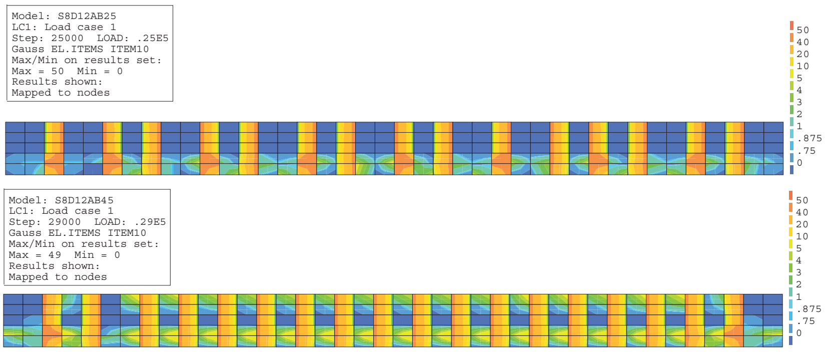


Figure 7.13: Damage simulation S8D12AB25 at u_y=3.40 mm and S8D12AB45 at u_y=2.60 mm (damage indicator N-direction, 50 equals complete damage)

7.6.3 Influence of rebar diameter

In this section the influence of rebar diameter on the cracking behaviour of the tension-pull experiments will be examined. For this three simulations will be compared: S4D12A, S4D16A and S4D25A all with concrete quality B25. Model S4D25A has a slightly larger reinforcement ratio of 1.20% compared to the other two (1.00%). The mesh used in the calculations is given in figures 7.2 to 7.4 section 7.3. The bond-slip relation of course differs in each simulation (see figure 7.5). Figure 7.14 shows the results of the calculations in a load-displacement curve. For S4D12AB25, S4D16AB25 and S4D25AB25 about 15500, 17000 and 39000 load steps are needed respectively. With increasing rebar diameter the lin-

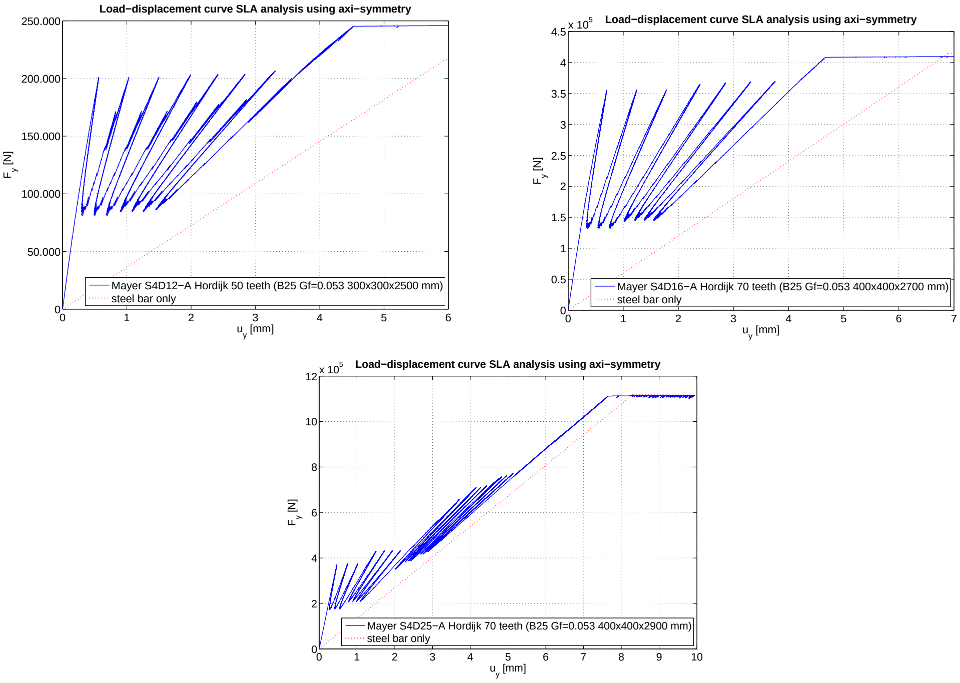


Figure 7.14: Load-displacement curve simulation S4D12A (top left) versus S4D16A (top right) and S4D25A (bottom)

ear elastic stiffness increases but this is also due to the increased concrete cross-section with S4D16AB25 and S4D25AB25. The cracking of S4D12AB25 and S4D16AB25 are very similar although the ultimate load level is much higher for S4D16AB25. In simulation S4D25AB25 secondary cracks occur between all primary cracks. The final crack patterns are shown in figure 7.15. The complete crack development is given in appendices O, R and U. The concrete principal stresses along the rebar at first primary crack initiation for S4D16AB25 and S4D25AB25 are very similar to the stress profile of S4D12AB25 (see figure 7.8) so they are not give here. Finally figure 7.16 shows the damage for the three simulations after the formation of the primary crack pattern. As could be seen from the load-displacement curve and appendix U in simulation S4D25AB25 it appears that the interface stiffness is borderline sufficient to get secondary cracks. The stiffness ratio between the interface and the concrete for the three simulations are calculated as:

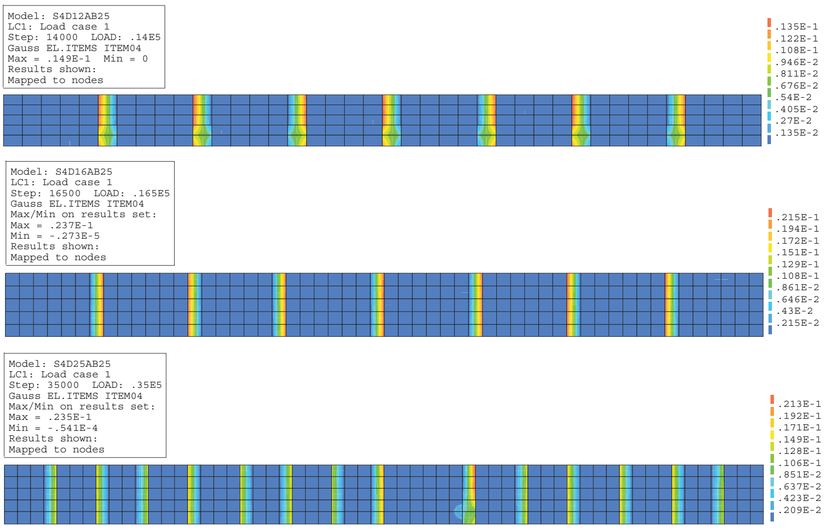


Figure 7.15: Final crack pattern simulations S4D12AB25 (top), S4D16AB25 (middle) and S4D25AB25 (bottom)

$$\text{S4D12AB25: ratio} = \frac{OS_{t,el}}{E_c} = \frac{150.796\text{x}116.7}{30000} = 0.587 \tag{7.5}$$

$$\text{S4D16AB25: ratio} = \frac{OS_{t,el}}{E_c} = \frac{201.062\text{x}87.4}{30000} = 0.586 \tag{7.6}$$

$$\text{S4D25AB25: ratio} = \frac{OS_{t,el}}{E_c} = \frac{314.160\text{x}57.1}{30000} = 0.598 \tag{7.7}$$

With:

O	total circumference of all rebars
$S_{t,el}$	initial elastic stiffness of interface (see also figure 7.5)
E_c	concrete elastic stiffness

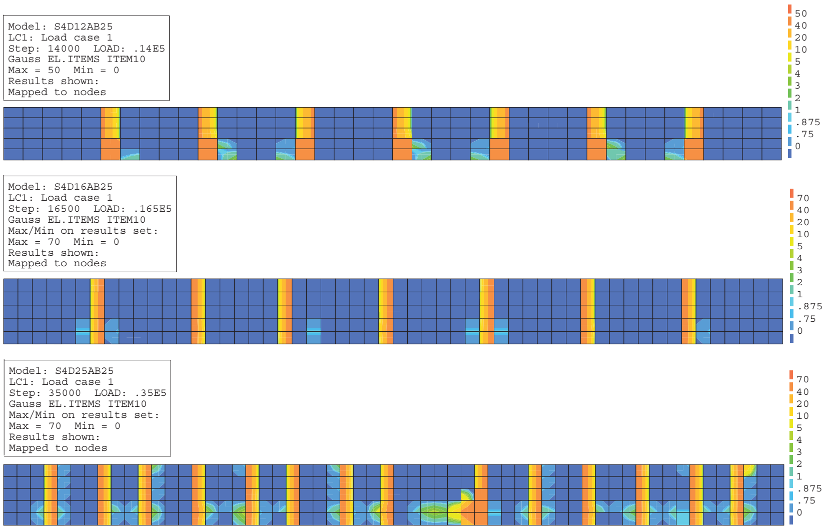


Figure 7.16: Damage simulation S4D12AB25 at u_y=3.65 mm, S4D16AB25 at u_y=4.60 mm and S4D25AB25 at u_y=6.15 mm (damage indicator N-direction, 50/70 equals complete damage)

The decreasing stiffness $S_{t,el}$ for bond-slip with increasing rebar diameter is a nice demonstration of the bond model (see also figure 2.11 section 2.8). It can be seen that the stiffness ratio for S4D12AB25 and S4D16AB25 are identical whereas the ratio for S4D25AB25 is slightly larger due to the increase in circumference. This is the explanation for the development of a secondary crack pattern in case of S4D25AB25.

7.6.4 Stress-average strain plots

The following figures show comparisons between the SLA calculations and individual experimental results in a stress-average strain curve. To compute the average strain the difference is taken between the displacements at the two end positions of the two meter strain gauges divided by the distance between them. The nodes taken are located on the outside of the equivalent circular cross-section. The stress is calculated from the reaction force at the lumped rebar divided by the area of the lumped rebar.

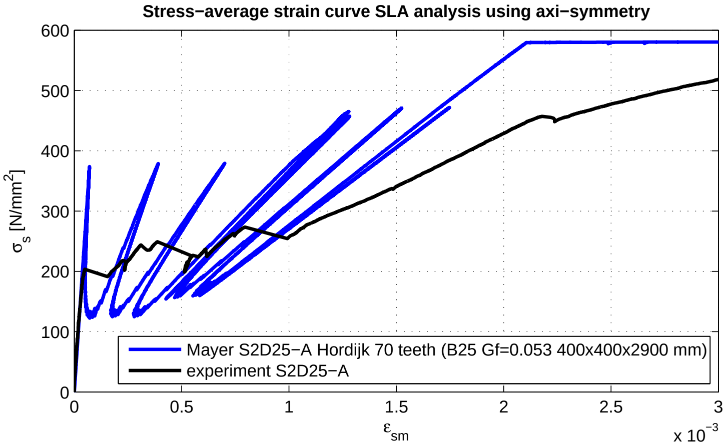


Figure 7.17: Stress-average strain curve from simulation S2D25AB25, SLA versus experimental result

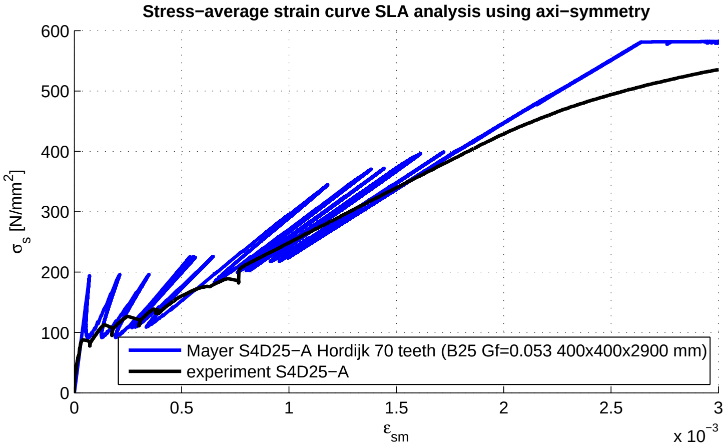


Figure 7.18: Stress-average strain curve from simulation S4D25AB25, SLA versus experimental result

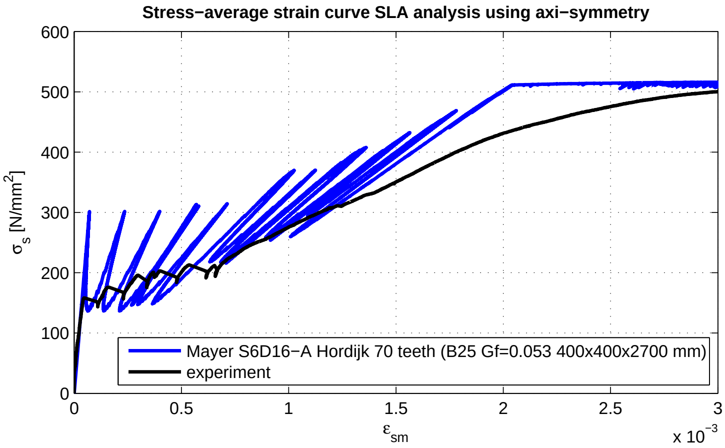


Figure 7.19: Stress-average strain curve from simulation S6D16AB25, SLA versus experimental result

The experiments are executed using displacement control and do not show snap-backs. The cause of the differences between the experiments and the SLA calculations can be a number of things:

- bending stresses occuring in the experiment due to handeling or non-straightness;
- pre-cracking due to shrinkage;
- non-uniform concrete strenght;
- deviations in dimensions.

Regarding the SLA calculations especially the fracture energy and the crack band width are parameters that could be changed to get closer to the experimental results.

7.6.5 Overview of results, number of cracks and crack spacing

Table 7.4: Results of calculated experiments

code	number of primary cracks	average crack spacing [mm]
S4D12AB25	7	312.50
S8D12AB25	15	156.25
S8D12AB45	18	131.58
S4D16AB25	7	337.50
S6D16AB25	14	180.00
S2D25AB25	7	362.50
S4D25AB25	14	193.33

7.7 Summary and conclusions chapter 7

In this chapter several tension-pull experiments by Mayer [12] are calculated with SLA using the Matlab bond model derived in chapter 2 to calculate the bond-slip relations. These bond-slip relations are then approximated with a sawtooth diagram using the sawtooth generator described in section 4.3. A variety of configurations is calculated, i.e. different reinforcement ratios, rebar diameter as well as concrete strength. The influence of these parameters in particular on the cracking behaviour is then analyzed. Because the report by Mayer does not contain load-displacement curves or complete crack patterns only limited comparisons to the experiments can be made. The following conclusions regarding the SLA calculations can be stated:

Use of the bond model

1. For the different configurations the bond model has demonstrated a great variety in bond-slip curves. The linear elastic stiffness's vary from $S_{t,el}$=57.1 N/mm^3 to $S_{t,el}$=377.3 N/mm^3. Also with the parameters used the results show both failure mechanisms, i.e. pull-out failure and splitting failure. Since the cracking in the tension-pull simulations is very dependent on the (interface) stiffness the great variety in crack patterns is a direct result of the various computed bond-slip curves. This therefore demonstrates the importance of the use of the bond model.

2. For the sawtooth approximations in the simulations a large number of teeth is used. Since many more load steps involve concrete cracking this has only a minor effect on the computing time. Because the computed bond-slip curves only have a limited elastic part it is important to start the sawtooth approximation already on the ascending part.

Influence of reinforcement ratio

1. The results are in good agreement with the expectation. With increased reinforcement the number of cracks increases. The results also show that with increased reinforcement the primary cracks show lower crack strains indicating a lower crack width. Also, apart from the primary cracks, the area close to and around the rebar is much more cracked.

Influence of concrete type

1. The cracking behaviour in simulation S8D12A with concrete quality B45 is very different from the one with B25. Because of the much higher relative interface stiffness with B45 (3x) there is a much shorter load-transfer zone and the first primary crack starts close to the loaded end. The principal stress plot along the rebar also shows a peak stress at this location. Successive primary cracks then evolve about every 132 mm from this end. Also this is the only simulation that shows cracking near to the axis of rotation. The cause of this latter behaviour is unclear.

Influence of rebar diameter

1. From the results the influence of rebar diameter with equal reinforcement ratio appears to be limited. The calculation results from simulation S4D25A are however interesting since in this case the development of a secondary crack pattern is only possible after a much higher applied load. This suggests the interface stiffness is borderline sufficient in this case to initiate secondary cracking prior to the steel yielding.

Comparison to experimental results

1. The comparison to the experimental results is somewhat inconclusive. The sharp snapbacks found in the simulations do not appear in the experiments. This is most likely due to the use of displacement control in the experiments. All experimental results show a weaker response in the stress-average strain plots. In the experimental testing this could be the result of bending stresses, pre-cracking due to shrinkage, non-uniform concrete strength or deviations in dimensions.

Chapter 8

SLA calculation of a beam in bending experiment by Walraven

8.1 Introduction

In this chapter a short demonstration will be given of including bond-slip in the calculation of a bending beam with SLA. For this a reinforced concrete beam that was tested by Walraven will be simulated, see also [18], [19]. Two calculations will be presented, one using full bond (i.e. no interface elements) and one with a bond-slip relation calculated with the Matlab bond model. Only a short comparison of the two models will be made. In the experiment the beam fails in bending, see figure 8.1. The beam has a thickness of 200 mm, a height of 150 mm and a main span of 1750 mm. The reinforcing consists of 2Ø10 and 1Ø8, see also figure 8.2. The material parameters used for the concrete part are: Young's modulus E=25000 N/mm^2, Poisson's ratio ν=0.2, tensile strength f_{ct}=2.5 N/mm^2, fracture energy G_f=0.06 N/mm, crack band width h=25 mm and shear retention factor β=0.2. For the reinforcing bars the material parameters are: Young's modulus E=210000, Poisson's ratio ν=0.2, and yield stress σ_y=400 N/mm^2. The mesh is given in figure 8.3. The red line indicates the position of the reinforcement. For the concrete part plane stress elements are now used. The sawtooth diagrams used for the steel and the concrete are given in figure 8.4. For bond-slip the sawtooth approximation is calculated using a weighted diameter of 9.33 mm, see figure 8.5. For the concrete a compressive strength of 34.2 N/mm^2 is used (see [19]).[1] For the calculation of the effective cover three distances are used: 2x 25 mm and 1x

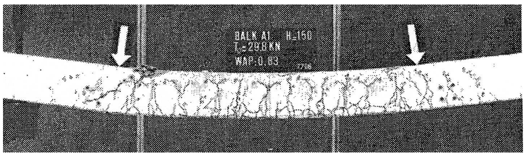


Figure 8.1: Bending failure in experiment

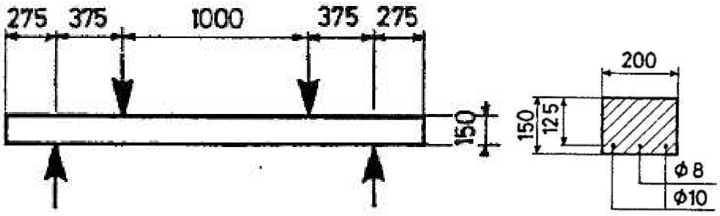


Figure 8.2: Dimensions and reinforcement bending beam tested by Walraven

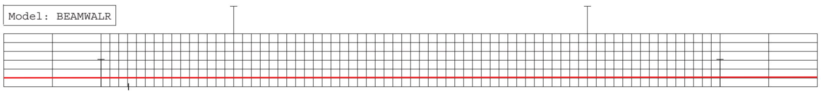


Figure 8.3: Finite element mesh, boundary conditions and loading

[1] in the bond model the concrete Young's modulus and tensile strength are both calculated from the compressive strength see section 2.3

40 mm. This is based on the corner bars that have a concrete cover of 25 mm in two directions. The distance of 40 mm refers to the distance to the inner side of the beam. This is chosen equal to four times the largest rebar of Ø10.[2]

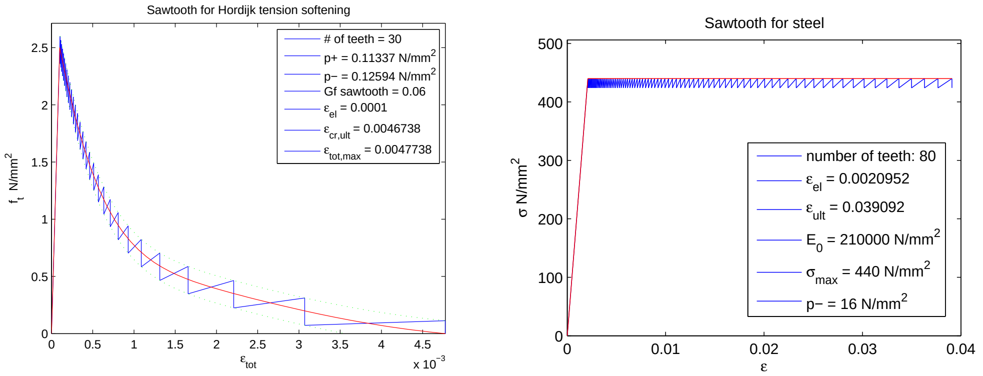


Figure 8.4: Sawtooth approximation for concrete (left) and steel (right)

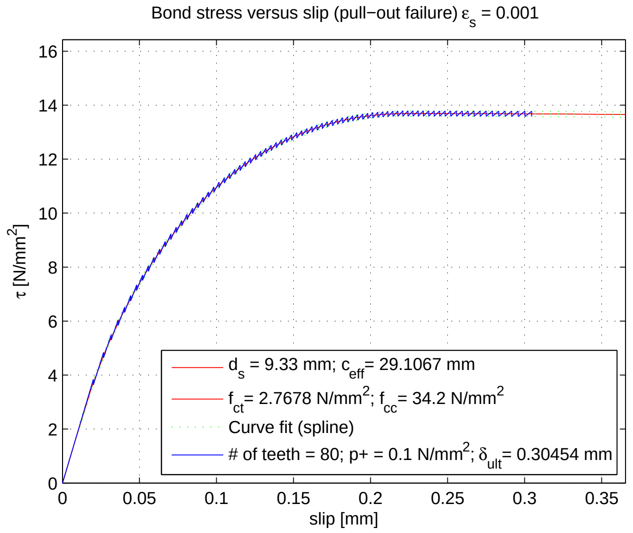


Figure 8.5: Sawtooth approximation for bond-slip using 80 teeth

[2] this is the maximum value that is taken into account with the bond model, higher values have almost no effect on the response of the thick-walled cylinder

8.2 Results using full bond

The damage evolution is given in figure 8.6.

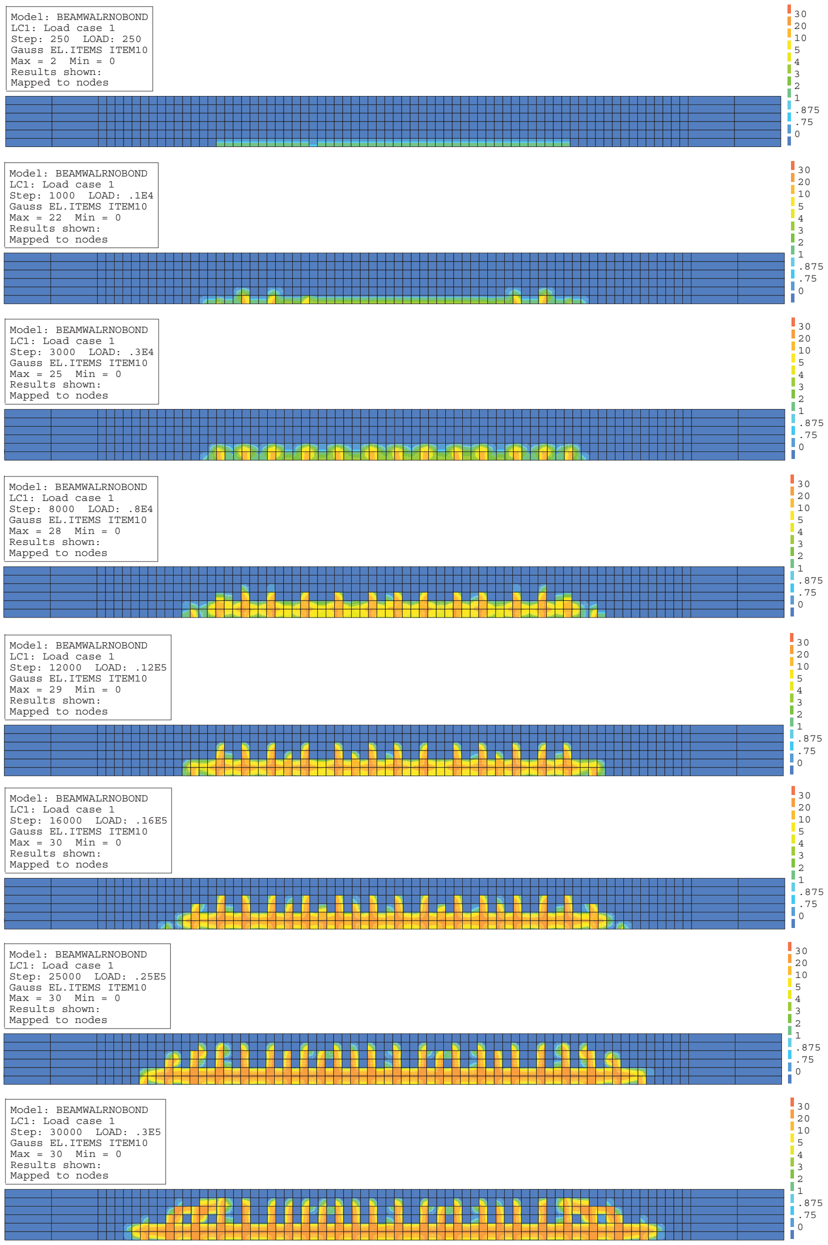


Figure 8.6: Damage evolution using full bond (damage indicator N-direction, 30 equals complete damage)

8.3 Results with bond-slip

The damage evolution is given in figure 8.7.

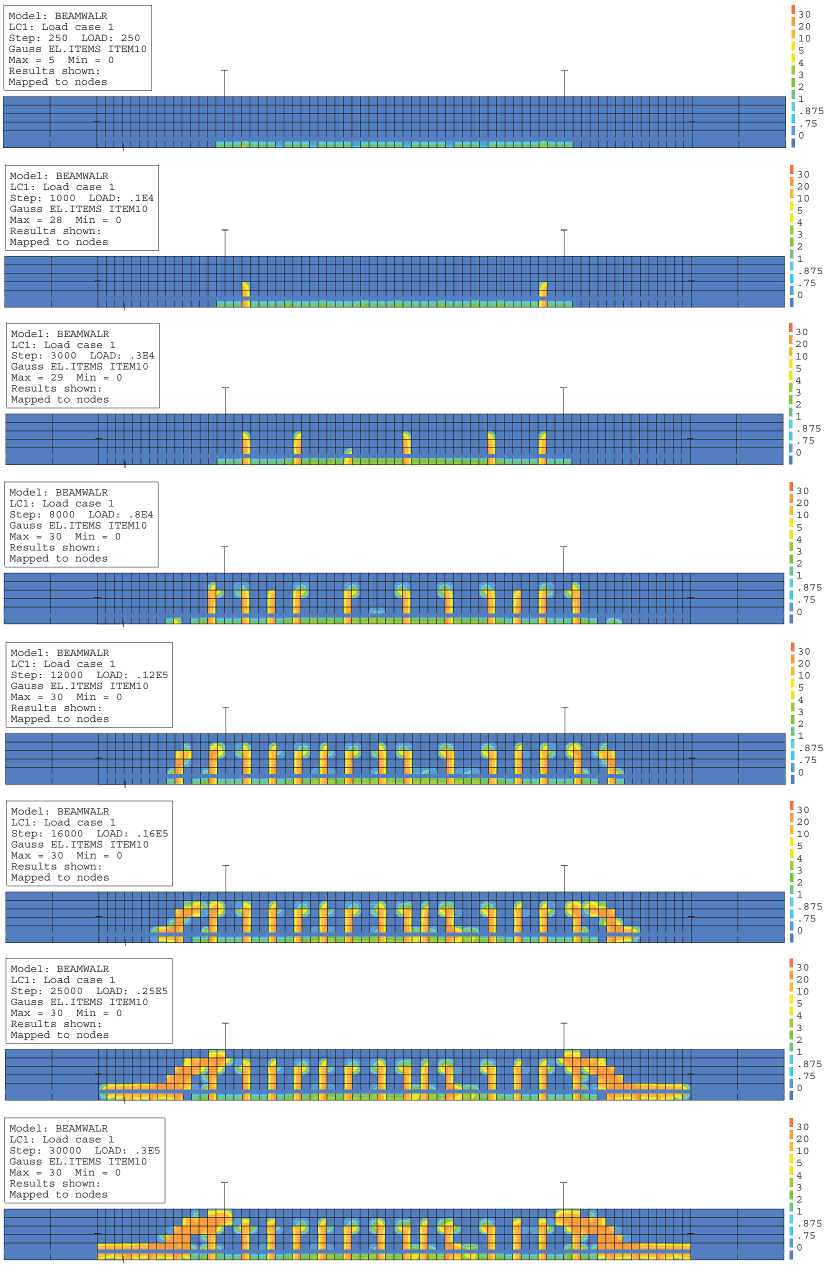


Figure 8.7: Damage evolution using bond-slip (damage indicator N-direction, 30 equals complete damage)

8.4 Load-displacement curve

The results of both calculations are plotted in a load-displacement curve together with the experimental result and a previous SLA result by Belletti, see figure 8.8. The deviation from the curve by Belletti cannot be explained at the moment of writing, both use full bond. A noted difference is the use of linear elements by Belletti instead of quadratic. The use of bond-slip is especially visible in the early cracking stage, see figure 8.9.

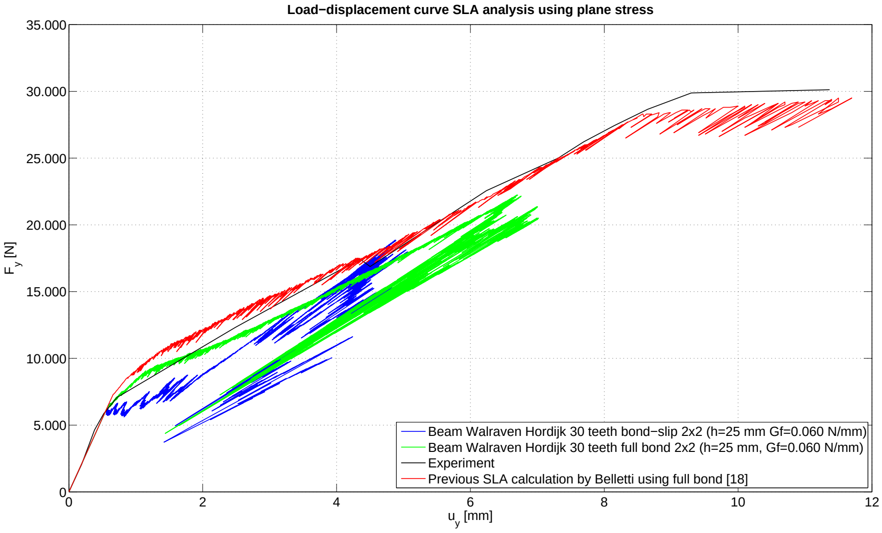


Figure 8.8: Load displacement curve, full bond versus bond-slip

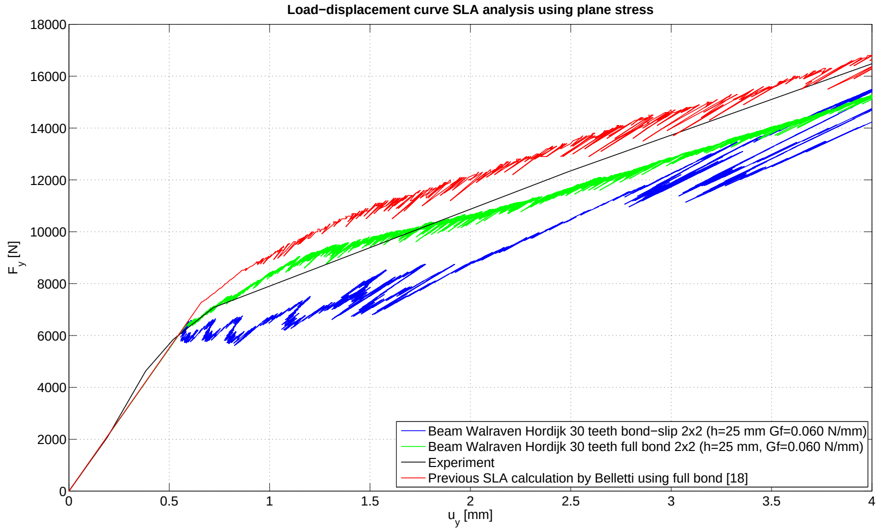


Figure 8.9: Detail of figure 8.8

Chapter 9

Conclusions and recommendations

9.1 Conclusions

From the non-linear calculations of the tension-pull experiment by Gijsbers the following conclusions can be drawn:

1. To obtain a primary crack pattern consisting of several cross-sectional cracks in a tension-pull experiment it is absolutely necessary to use material imperfections. A calculation with no imperfections will result in a very diffuse crack pattern without localized (primary) cracks.

2. The degree of the material imperfections needed to get a physically plausible crack pattern is unclear. In this specific calculation imperfections were used up until a strength and ultimate crack strain reduction of 20%. Using imperfections however in general will cause an underestimation of the actual strength of the model.

3. Using displacement control it is possible to “jump” over primary cracks in the load-displacement curve.

4. The use of a non-linear interface with a limitation of the ultimate bond stress will limit the amount of cracking in the surrounding concrete. The influence on the general stiffness of the model, i.e. in a load-displacement curve, is however much less. Most likely the general strength and stiffness of a tension-pull model is more governed by the primary cross-sectional cracks and much less by the in-between cracking.

5. Good results were obtained with quadratic elements and a relatively coarse mesh. The plots of the concrete principal stresses, steel stresses and interface bond stresses are all quite smooth. Possibly a criterion to obtain smooth stress profiles can be determined in terms of the minimum number of elements needed between primary cracks. This all depends on the strength and stiffness values of the concrete as well as the interface and the amount of reinforcement.

6. The added value of using a full 3D model for a tension-pull experiment is limited. The only advantage is the somewhat better calculation of splitting cracks. However to simulate this correctly also circumferential imperfections are needed to enforce localized primary splitting cracks.

7. The amount of unloading of the interfaces (i.e. reversal of slip) that takes place in the tension-pull model is very limited. Therefore changing the unloading behaviour (between elastic, secant or non-linear) has almost no effect. The largest unloading takes place near the end faces or near a primary crack. From literature it was found that the bond stress-slip relation under cyclic loading conditions is highly complicated and therefore cannot be expressed in a general unloading type. Despite this, although limited, elastic unloading appears to be the best approximation.

From the SLA calculations of the tension-pull experiment by Gijsbers the following conclusions can be drawn:

1. The results of the model with imperfections are in excellent agreement with the non-linear analysis. Also with SLA snap-backs in the load-displacement curve are captured after each primary crack. This indicates SLA is a good alternative to non-linear analysis when using axi-symmetric and interface elements. This was one of the primary objectives of this thesis (see section 1.4). The comparison to the non-linear analysis justifies the conclusion that the new code for axi-symmetric and interface elements as well as the sawtooth generator for Hordijk softening are working correctly.

2. Despite the imperfections used the SLA analysis still shows somewhat better crack localization, compared to the non-linear analysis.

3. The adopted sawtooth approximation for bond-slip using an equal uplifting and downshifting of the "mother" curve shows good results in terms of the calculated bond stress profiles.

4. The calculation in SLA without imperfections shows the formation of a primary crack pattern consisting of several primary cracks. This demonstrates SLA to be superior to non-linear analysis since obtaining this result without imperfections in a non-linear analysis is not possible.

5. In general with the tension-pull calculations cracks originate from stress points close to the rebar. Prior to the formation of the first primary crack the principal stresses along the rebar show some stress oscillations. These oscillations become more apparent when a sawtooth diagram for concrete is used with less teeth (i.e. using a lower accuracy). This also results in differences in primary crack locations. The same oscillations are witnessed when using a finer mesh. However with a finer mesh and equal accuracy for the sawtooth approximation the stress oscillations increase. It is believed that this phenomenon is related to the stiffness ratio between the concrete and the interface (see section 6.7.2). Using an extreme number of teeth for the concrete as well as the interface has shown that these oscillations disappear.

6. Using stepwise shear retention instead of a constant shear retention factor improves the results. The stress oscillations diminish and the primary crack locations are more consistent when the number of teeth for the concrete sawtooth diagram is reduced.

From the SLA calculations of the tension-pull experiments by Mayer the following conclusions can be drawn:

1. The various calculations with the bond model have demonstrated a great variety in bond-slip curves in terms of initial stiffness and failure mode. Since this has a great effect on the primary crack pattern, in case of a tension-pull experiment simulation, this emphasizes the importance and value of the use of a bond model.

2. The results of the simulations have demonstrated the influence of the various parameters (reinforcement ratio, concrete type and rebar diameter) on the cracking behaviour. In general using a higher bond-slip stiffness results in an increase in primary cracks. When the stiffness ratio between the interface and the surrounding concrete has reached a certain level, primary cracking will initiate close to the loaded end (instead of in the center) because of the occurrence of a local peak stress.

3. The comparison to the experiments, in terms of a stress-average strain plot, all show a weaker response. It is likely that this is the result of additional stresses occurring in the experiment (due to bending, non-straightness, handling etc.) and a non-uniform concrete strength. The latter will cause cracking to initiate in the weakest link. It is my opinion that it will prove difficult to simulate a tension-pull experiment exactly since the material properties along the whole length of a specimen, which are usually unknown, influence the response in terms of load-displacement and development of primary cracks. A possibility could be to use a random strength for all elements.

9.2 Guidelines for simulating tension-pull experiments

The following guidelines can be given for simulating a tension-pull experiment with non-linear analysis:

1. Good results are obtained using a relatively coarse mesh.

2. The use of material imperfections (or random strength) is vital to get a primary crack pattern.

3. Using a limitation of the bond stress, in the bond-slip curve, has a great influence on the cracking but not on the load-displacement response.

4. The use of a 3D model does not have any significant advantage over the use of axi-symmetry.

5. In case of continues loading until failure, the bond-slip unloading type is not relevant.

The following guidelines can be given for simulating a tension-pull experiment with SLA:

1. With SLA material imperfections are not needed.

2. Best results are obtained using a relatively coarse mesh.

3. A fine mesh can ultimately result in large areas cracking between the primary cracks.

4. Best results are obtained using a stepwise shear retention instead of a constant shear retention.

5. It is recommended to use as many teeth as possible for the concrete sawtooth approximation.

6. Since many more loadsteps involve concrete cracking, it is recommended to use an accurate sawtooth approximation for bond-slip as well as steel since this has only a minor effect on computing time and can only improve results.

7. It is recommended to check the principal stresses along the reinforcing bar for oscillations prior to the initiation of the first primary crack. If strong oscillations occur it is recommended to use more teeth for the concrete sawtooth approximation and/or a coarser mesh.

8. In case of actual experiment simulations the use of a bond model is vital to get a reliable bond-slip relation.

9. For the cracking behaviour the following stiffness ratio is of importance:

   $$\frac{OS_{t,el}}{E_c} \tag{9.1}$$

   With:

O	total circumference of all rebars
$S_{t,el}$	initial elastic stiffness of interface
E_c	concrete elastic stiffness

9.3 Recommendations

1. Creating an automated sawtooth generator for Hordijk tension softening within the current SLA environment.

2. The use of random concrete strength for tension-pull simulations.

3. Possibly investigation into the new formulae for bond-slip that will be presented in the new CEB-FIB MC model code that will be released later this year. The formulae will be based on the bond model by Den Uijl & Bigaj and it might be possible to create a material model for use with *DIANA* and/or SLA.

4. Further investigation into the cause of the concrete principal stress oscillations occurring with SLA.

5. The influence of including bond-slip in SLA in larger structures, such as beams. A review could be performed to compare previous SLA results.

6. Comparing the average crack spacing obtained with SLA to experiments or known formulae.

Summary

In non-linear concrete fracture mechanics computational problems can arise such as non-convergence or bifurcations. These types of problems become of more importance when dealing with a structure that has a very uniform stress distribution. Most often the non-linear analysis fails in such cases because cracking will start simultaneously in large parts of the structure when the applied stress equals the tensile strength. This results in a very diffuse crack pattern instead of a pattern with localized (primary) cracks. In real life concrete structures because of the non-uniform tensile strength cracking will occur in the weakest link.

An alternative method to conventional non-linear analysis is sequentially linear analysis or SLA for short. The aforementioned problems that can arise with non-linear analysis are non-existent with SLA. In SLA the negative slope of the stress-strain curve for concrete in tension (the softening curve) is replaced by a sawtooth diagram of positive slopes and the incremental-iterative method used in non-linear analysis is replaced by a series of linear analyses. In every loadstep in the analysis a local damage is given to the integration point that has a stress closest to its local strength. This procedure ensures damage to become localized since in every loadstep only one integration point is given a local damage.

In this thesis the possibilities to simulate bond-slip behaviour between steel and concrete with SLA and with the use of interface elements are explored. For this the existing SLA software implementation of the finite element program *DIANA* at Delft University is extended with new (*Fortran77*) code to incorporate axi-symmetric and interface elements. Because in all calculations the non-linear Hordijk softening curve is used for concrete in tension also a so called sawtooth generator is written in Matlab. This will create the needed sawtooth diagram used in SLA.

To examine the local bond-slip mechanism for this thesis a literature study is carried out. From this study it becomes clear that bond-slip is not only dependent on material properties but also on structural properties. Furthermore two distinct failure mechanisms exist, namely pull-out failure and splitting failure, each having there own unique bond-slip relation. The possibilities to create such a bond-slip relation are explored. One such possibility is the use of the CEB-FIB MC90 model code where simple formulae are used dependent on pure material properties. Another possibility is the use of a more sophisticated bond model. A bond model can

incorporate material as well as structural properties to derive a bond-slip relation. One such bond model is the one that was developed by Den Uijl and Bigaj [8]. This model has many advantages one of them being that it will identify pull-out or splitting failure automatically. For this thesis the model is completely written in Matlab code using the formulas of the dissertation by Bigaj [8] and a previously developed Excel sheet. Also the parts and formulations of the model that were somewhat hard to understand from [8] are reported. For the use with SLA the Matlab bond model is also extended with a sawtooth generator to approximate the bond-slip curve.

Before SLA calculations are performed first the possibilities of modeling a tension-pull experiment using standard non-linear analysis are explored. This also serves as a reference calculation to compare to the SLA results. Imperfections are used to enforce the development of a primary crack pattern. For the FEM discretization axi-symmetry is used. The same tension-pull experiment is then also modelled and analyzed using 3D elements. For the bond-slip relation at first a simple bi-linear elasto-plastic relation is taken. For the axisymmetric calculation the unloading of the interfaces (i.e. the bond-slip) is explored and the use of three different types of unloading behaviour is then analyzed using a user defined material model with *DIANA* that is written in *Fortran77* code. This shows that only limited unloading takes place and changing it's behaviour has no significant effect on the results.

The non-linear calculation with axi-symmetric elements and with material imperfections is recalculated with SLA to compare both results. The results show that the SLA calculation is in excellent agreement with the non-linear analysis. The comparison to the non-linear analysis justifies the conclusion that the new code for axi-symmetric and interface elements as well as the sawtooth generator for Hordijk softening are working correctly.

Next the model is analyzed in SLA but now without the use of imperfections. The results show that it is indeed possible to retrieve a primary crack pattern without the use of imperfections. The damage observed in the model also demonstrates excellent localization similar to the non-linear analysis where imperfections are used. Several specific aspects of the SLA analysis are explored further. For an SLA analysis an important aspect is the computing time. This aspect is examined by reducing the number of teeth of the sawtooth diagrams used for the various non-linearitys in material behaviour. Also the influence of using different shear retention relations is examined. Finally with this model the use and influence of mesh refinements is explored. The results show that principal stress oscillations occur in the concrete close to the rebar that effect the cracking behaviour especially when using a fine mesh or a low number of teeth for the concrete sawtooth diagram.

For the use of the Matlab bond model and the sawtooth generator a variety of tension-pull experiments that were conducted by Mayer [12] are simulated. In all models axi-symmetry is used. The experiments all consist of long reinforced columns with various reinforcement ratios, concrete qualities, rebar diameters etc. The calculated bond-slip curves using the bond model show a great variety in bond-slip relations. The results of the calculations show the influence the different parameters have on especially the fully developed crack pattern.

Finally a calculation is made of a simply supported reinforced concrete beam with and without the inclusion of bond-slip. The results show that with the inclusion of bond-slip cracks are much more localized.

Bibliography

[1] K. Lundgren, Three-Dimensional Modelling of Bond in Reinforced Concrete. Theoretical Model, Experiments and Applications. Ph.D Dissertation. Chalmers University of Technology (1999)

[2] J.G. Rots, Bond-slip simulations using smeared cracks and/or interface elements. Delft University of Technology (1985)

[3] F.B.J. Gijsbers, A.A. Hehemann, Some tensile tests on reinforced concrete. TNO-IBBC Report BI-77-61, Rijswijk (1977)

[4] J.G. Rots, J.C.J. Schellekens, Interface elements in concrete mechanics. Delft University of Technology (1990)

[5] J. Manie, J. Jansen, *DIANA* User's Course. Working with User-supplied Subroutines. Release 9.3. TNO *DIANA* (2008)

[6] J.G. Rots, Computational modeling of concrete fracture. Ph.D Dissertation. Delft University of Technology (1988)

[7] Fédération internationale du béton (fib). Bond of reinforcement in concrete: state-of-art report. (2000)

[8] A.J. Bigaj, Structural Dependence of Rotation Capacity of Plastic Hinges in RC Beams and Slabs. Ph.D Dissertation. Delft University of Technology (1999)

[9] J. Magnusson, Bond and Anchorage of Ribbed Bars in High-Strength Concrete. Ph.D Dissertation. Chalmers University of Technology (2000)

[10] J.G. Rots, B. Belletti, S. Invernizzi, Robust modeling of RC structures with an “event-by-event” strategy. Engineering Fracture Mechanics 75 (2008)

[11] S. Invernizzi, D. Trovato, M.A.N. Hendriks, A.V. van de Graaf, Sequentially linear modelling of local snap-back in extremely brittle structures. (2010)

[12] U. Mayer, Influence of the rib pattern of ribbed reinforcement on the structural behavior of reinforced concrete members. Ph.D Dissertation. University of Stuttgart (2001)

[13] S. Lettow, Mitwirkung des Betons zwischen Rissen im inelastischen Stahldehnungsbereich unter besonderer Berücksichtigung der Materialeigenschaften und der Oberflächengestalt des Stahls. University of Stuttgart (1998)

[14] L. Elfgren, K. Noghabai, Tension of Reinforced Concrete Prisms - Round Robin Analysis and tests on Bond - Report of RILEM TC 147-FMBA. “Bigaj, J. den Uijl, Tension Stiffening Simulation with Confinement Based Bond Model.” (2001)

[15] M.J. de Jong, M.A.N. Hendriks, J.G. Rots, Shear retention and mesh alignment during fracture using sequentially linear analysis. (2009)

[16] P. Schumacher, Rotation Capacity of Self-Compacting Steel Fiber Reinforced Concrete. Ph.D Dissertation. Delft University of Technology (2006)

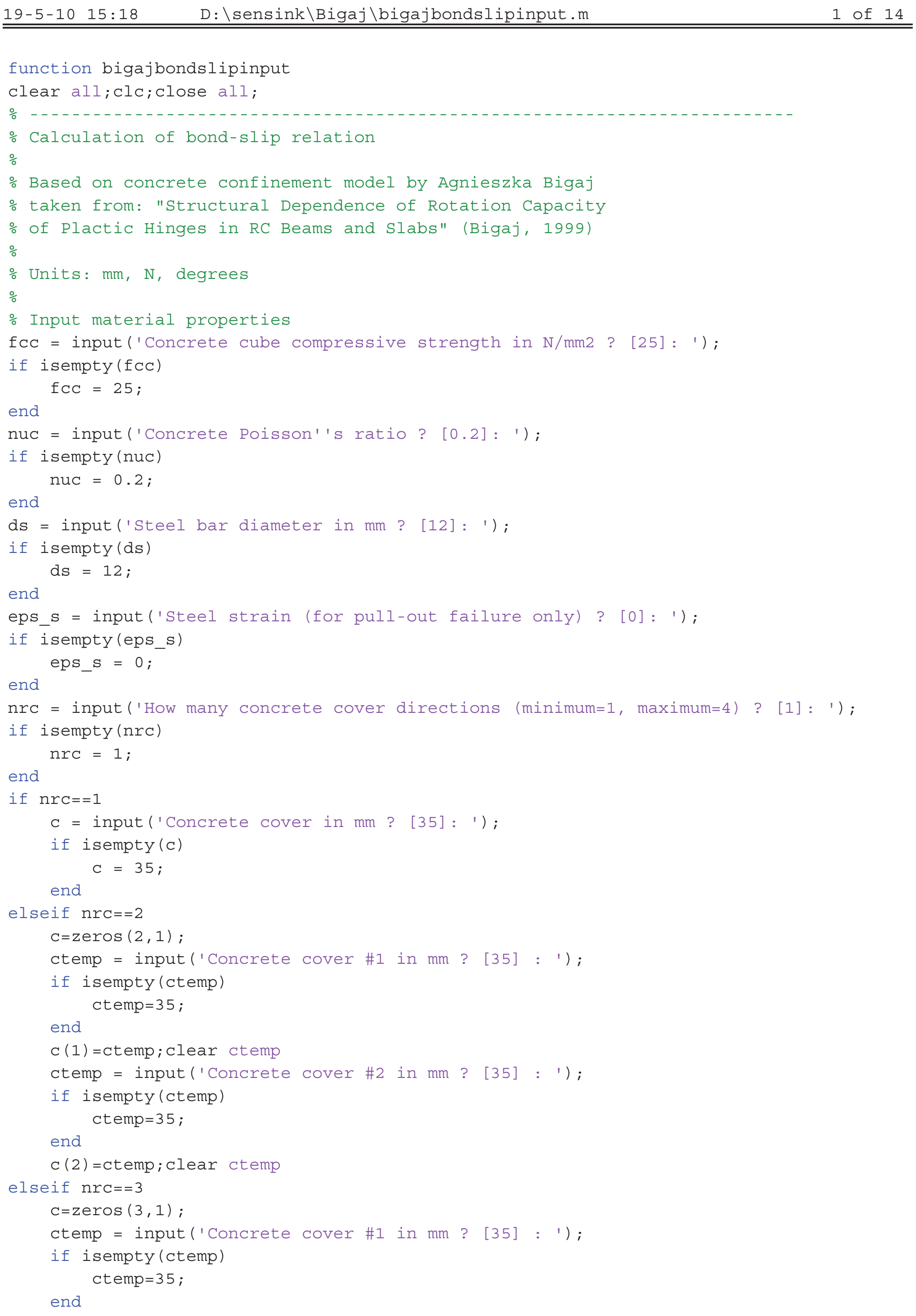
[17] A. van de Graaf, Sequentially Linear Analysis technique implementation in *DIANA* 9.2, Developers manual, version 0.1.02 (2009)

[18] J.G. Rots, B. Belletti, S. Invernizzi, Simplified Saw-Tooth Softening Model for Reinforced Concrete Structures. Fédération internationale du béton, Proceedings of the 2^{nd} International Congress (2006)

[19] J.C. Walraven, The influence of depth on the shear strength of lightweight concrete beams without shear reinforcement. Report 5-78-4. Stevin Laboratory Delft University of Technology (1978)

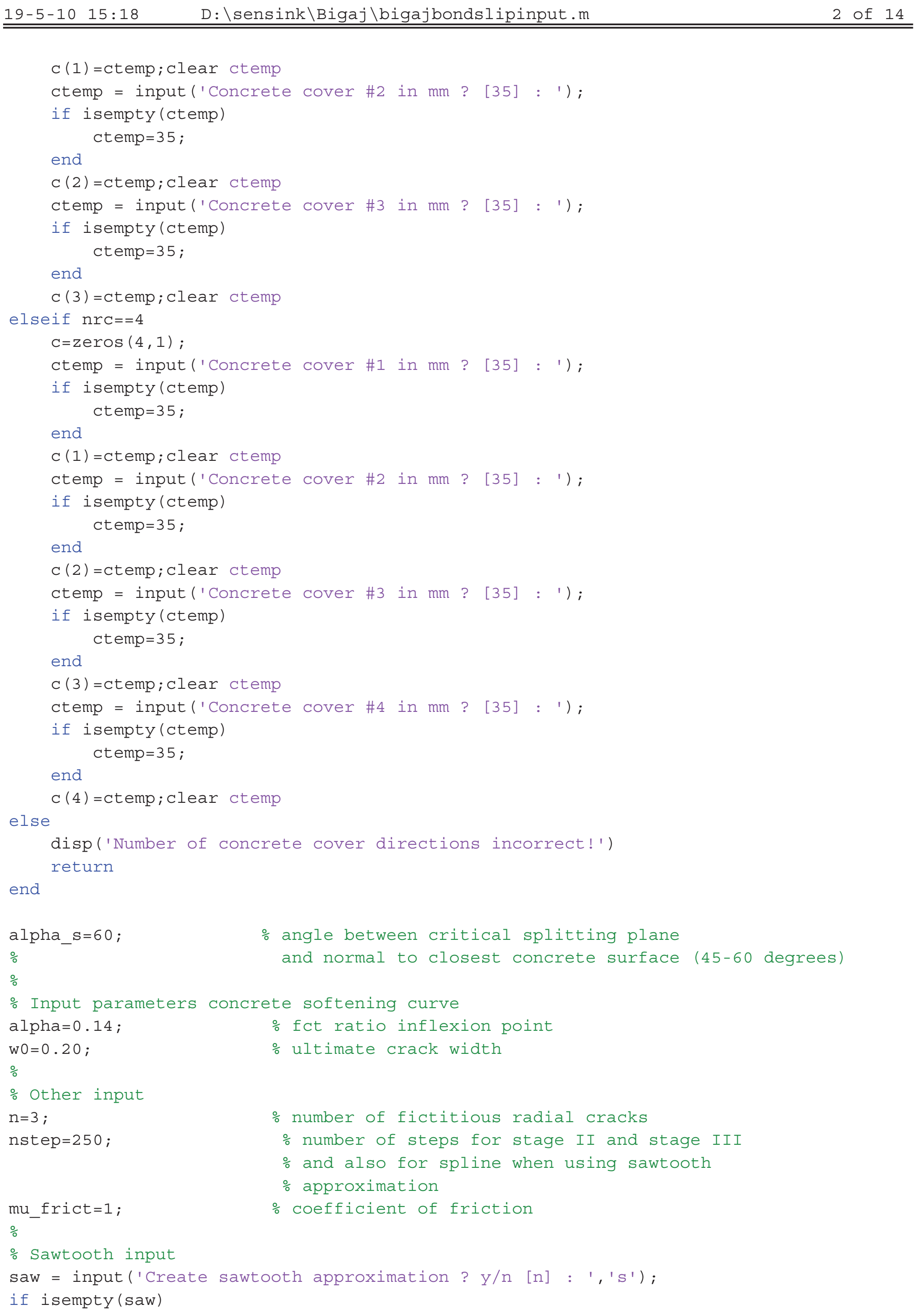
Appendix A

Matlab bond model: code

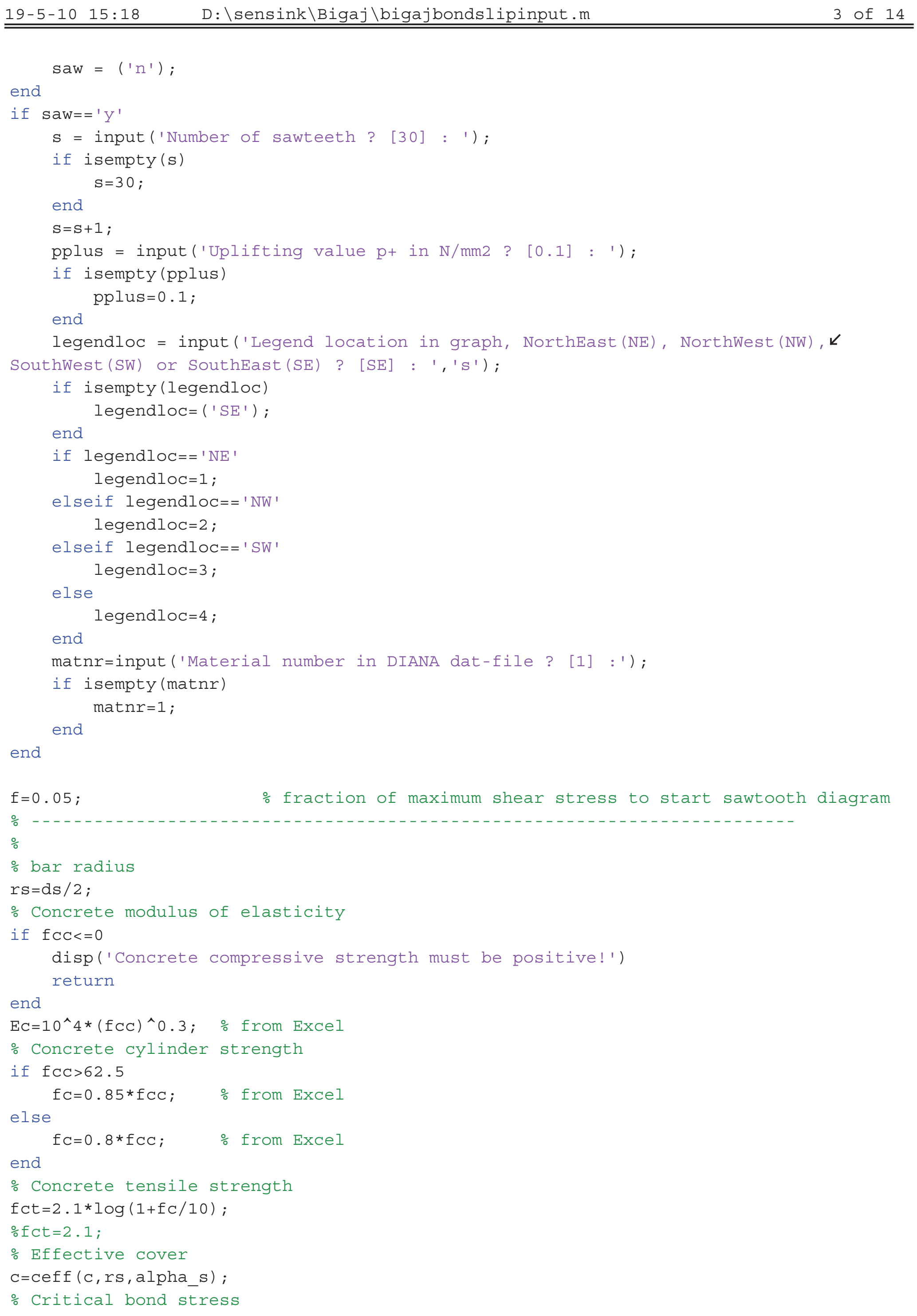


19-5-10 15:18   D:\sensink\Bigaj\bigajbondslipinput.m   1 of 14

```
function bigajbondslipinput
clear all;clc;close all;
% ------------------------------------------------------------------------
% Calculation of bond-slip relation
%
% Based on concrete confinement model by Agnieszka Bigaj
% taken from: "Structural Dependence of Rotation Capacity
% of Plactic Hinges in RC Beams and Slabs" (Bigaj, 1999)
%
% Units: mm, N, degrees
%
% Input material properties
fcc = input('Concrete cube compressive strength in N/mm2 ? [25]: ');
if isempty(fcc)
    fcc = 25;
end
nuc = input('Concrete Poisson''s ratio ? [0.2]: ');
if isempty(nuc)
    nuc = 0.2;
end
ds = input('Steel bar diameter in mm ? [12]: ');
if isempty(ds)
    ds = 12;
end
eps_s = input('Steel strain (for pull-out failure only) ? [0]: ');
if isempty(eps_s)
    eps_s = 0;
end
nrc = input('How many concrete cover directions (minimum=1, maximum=4) ? [1]: ');
if isempty(nrc)
    nrc = 1;
end
if nrc==1
    c = input('Concrete cover in mm ? [35]: ');
    if isempty(c)
        c = 35;
    end
elseif nrc==2
    c=zeros(2,1);
    ctemp = input('Concrete cover #1 in mm ? [35] : ');
    if isempty(ctemp)
        ctemp=35;
    end
    c(1)=ctemp;clear ctemp
    ctemp = input('Concrete cover #2 in mm ? [35] : ');
    if isempty(ctemp)
        ctemp=35;
    end
    c(2)=ctemp;clear ctemp
elseif nrc==3
    c=zeros(3,1);
    ctemp = input('Concrete cover #1 in mm ? [35] : ');
    if isempty(ctemp)
        ctemp=35;
    end
```



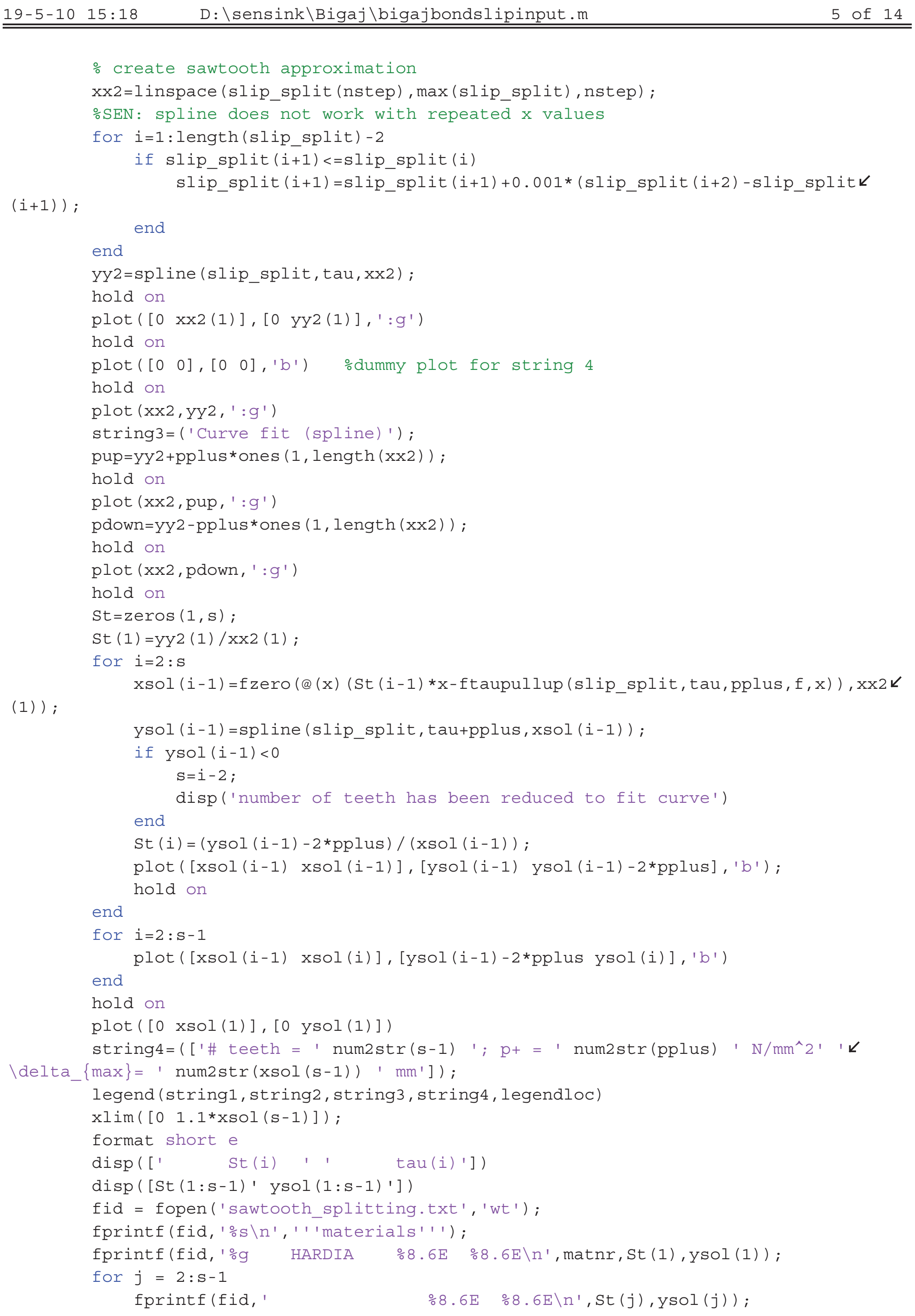
```
19-5-10 15:18      D:\sensink\Bigaj\bigajbondslipinput.m                  2 of 14

    c(1)=ctemp;clear ctemp
    ctemp = input('Concrete cover #2 in mm ? [35] : ');
    if isempty(ctemp)
        ctemp=35;
    end
    c(2)=ctemp;clear ctemp
    ctemp = input('Concrete cover #3 in mm ? [35] : ');
    if isempty(ctemp)
        ctemp=35;
    end
    c(3)=ctemp;clear ctemp
elseif nrc==4
    c=zeros(4,1);
    ctemp = input('Concrete cover #1 in mm ? [35] : ');
    if isempty(ctemp)
        ctemp=35;
    end
    c(1)=ctemp;clear ctemp
    ctemp = input('Concrete cover #2 in mm ? [35] : ');
    if isempty(ctemp)
        ctemp=35;
    end
    c(2)=ctemp;clear ctemp
    ctemp = input('Concrete cover #3 in mm ? [35] : ');
    if isempty(ctemp)
        ctemp=35;
    end
    c(3)=ctemp;clear ctemp
    ctemp = input('Concrete cover #4 in mm ? [35] : ');
    if isempty(ctemp)
        ctemp=35;
    end
    c(4)=ctemp;clear ctemp
else
    disp('Number of concrete cover directions incorrect!')
    return
end

alpha_s=60;             % angle between critical splitting plane
%                         and normal to closest concrete surface (45-60 degrees)
%
% Input parameters concrete softening curve
alpha=0.14;              % fct ratio inflexion point
w0=0.20;                 % ultimate crack width
%
% Other input
n=3;                     % number of fictitious radial cracks
nstep=250;                % number of steps for stage II and stage III
                          % and also for spline when using sawtooth
                          % approximation
mu_frict=1;              % coefficient of friction
%
% Sawtooth input
saw = input('Create sawtooth approximation ? y/n [n] : ','s');
if isempty(saw)
```



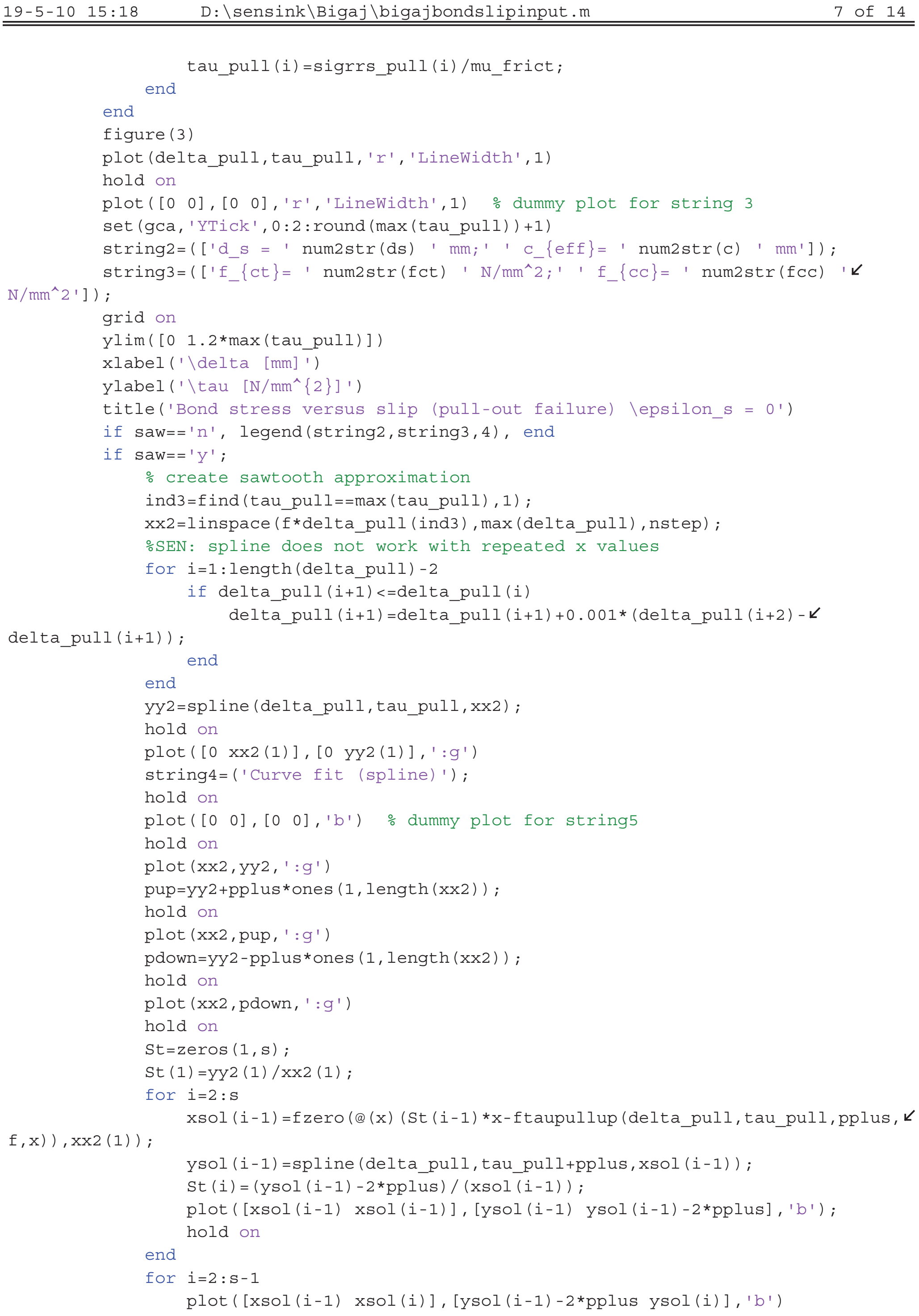
19-5-10 15:18 D:\sensink\Bigaj\bigajbondslipinput.m 3 of 14

```
    saw = ('n');
end
if saw=='y'
    s = input('Number of sawteeth ? [30] : ');
    if isempty(s)
        s=30;
    end
    s=s+1;
    pplus = input('Uplifting value p+ in N/mm2 ? [0.1] : ');
    if isempty(pplus)
        pplus=0.1;
    end
    legendloc = input('Legend location in graph, NorthEast(NE), NorthWest(NW),↙
SouthWest(SW) or SouthEast(SE) ? [SE] : ','s');
    if isempty(legendloc)
        legendloc=('SE');
    end
    if legendloc=='NE'
        legendloc=1;
    elseif legendloc=='NW'
        legendloc=2;
    elseif legendloc=='SW'
        legendloc=3;
    else
        legendloc=4;
    end
    matnr=input('Material number in DIANA dat-file ? [1] :');
    if isempty(matnr)
        matnr=1;
    end
end

f=0.05;                 % fraction of maximum shear stress to start sawtooth diagram
% ------------------------------------------------------------------------
%
% bar radius
rs=ds/2;
% Concrete modulus of elasticity
if fcc<=0
    disp('Concrete compressive strength must be positive!')
    return
end
Ec=10^4*(fcc)^0.3;  % from Excel
% Concrete cylinder strength
if fcc>62.5
    fc=0.85*fcc;    % from Excel
else
    fc=0.8*fcc;     % from Excel
end
% Concrete tensile strength
fct=2.1*log(1+fc/10);
%fct=2.1;
% Effective cover
c=ceff(c,rs,alpha_s);
% Critical bond stress
```

```
tau_b1=5*fct;
c1=c+rs;
phi=0.1*fc;
eps_cr=1.2*fct/Ec;  % factor 1.2 from Excel
% factors a1, a2, b1 and b2 bi-linear softening diagram (from Excel)
if fc<30
    beta=0.25;
else
    beta=0.25-0.0015*(fc-30);
end
a1=-(1-beta)/alpha;a2=-beta/(1-alpha);b1=1;b2=beta/(1-alpha);
% --------------------- calculate stage I and stage II --------------------
rcr=linspace(rs,c1,nstep+1);
[sigrrsII epsrrsII]=fstageII(rcr,rs,fct,c1,eps_cr,n,w0,a1,b1,nuc);
% plot stage I
figure(1)
plot([0 epsrrsII(1)],[0 sigrrsII(1)])
hold on
% plot stage II
plot(epsrrsII,sigrrsII)
xlabel('\epsilon_{r,rs}')
ylabel('\sigma_{r,rs} N/mm^{2}')
xlim([0 0.01])
% --------------------- calculate stage III ------------------------------
[sigrrsIII epsrrsIII C3]=fstageIII(a1,a2,eps_cr,n,w0,c1,b1,b2,rs,fct,nstep);
hold on
plot(epsrrsIII,sigrrsIII)
grid on
set(gca,'YTick',0:2:4+round(max(sigrrsII)))
title('Thick-walled-cylinder model, radial stress versus radial strain')
% --------------------- total sigma and epsilon --------------------------
% total sigma [r,rs] vector
sigrrs=[linspace(0,sigrrsII(1),nstep)';sigrrsII(2:nstep+1)';sigrrsIII];
% total epsilon [r,rs] vector
epsrrs=[linspace(0,epsrrsII(1),nstep)';epsrrsII(2:nstep+1)';epsrrsIII];

if max(sigrrs)<tau_b1/mu_frict
    % splitting failure
    % slip when splitting failure
    slip_split=(epsrrs*rs/(tand(phi)))';    % Eq (4.30)
    figure(2)
    tau=(sigrrs*mu_frict)';                 % Eq (4.29)
    plot(slip_split,tau,'r','LineWidth',1)
    string1=(['d_s = ' num2str(ds) ' mm;' ' c_{eff}= ' num2str(c) ' mm']);
    hold on
    plot([0 0],[0 0],'r','LineWidth',1)   % dummy plot for string2
    string2=([' f_{ct}= ' num2str(fct) ' N/mm^2;' ' f_{cc}= ' num2str(fcc) '↙
N/mm^2']);
    grid on
    xlim([0 1.25])
    xlabel('\delta [mm]')
    ylabel('\tau N/mm^{2}')
    title('Bond stress versus slip (splitting failure)')
    if saw=='n', legend(string1,string2,1), end
    if saw=='y';
```

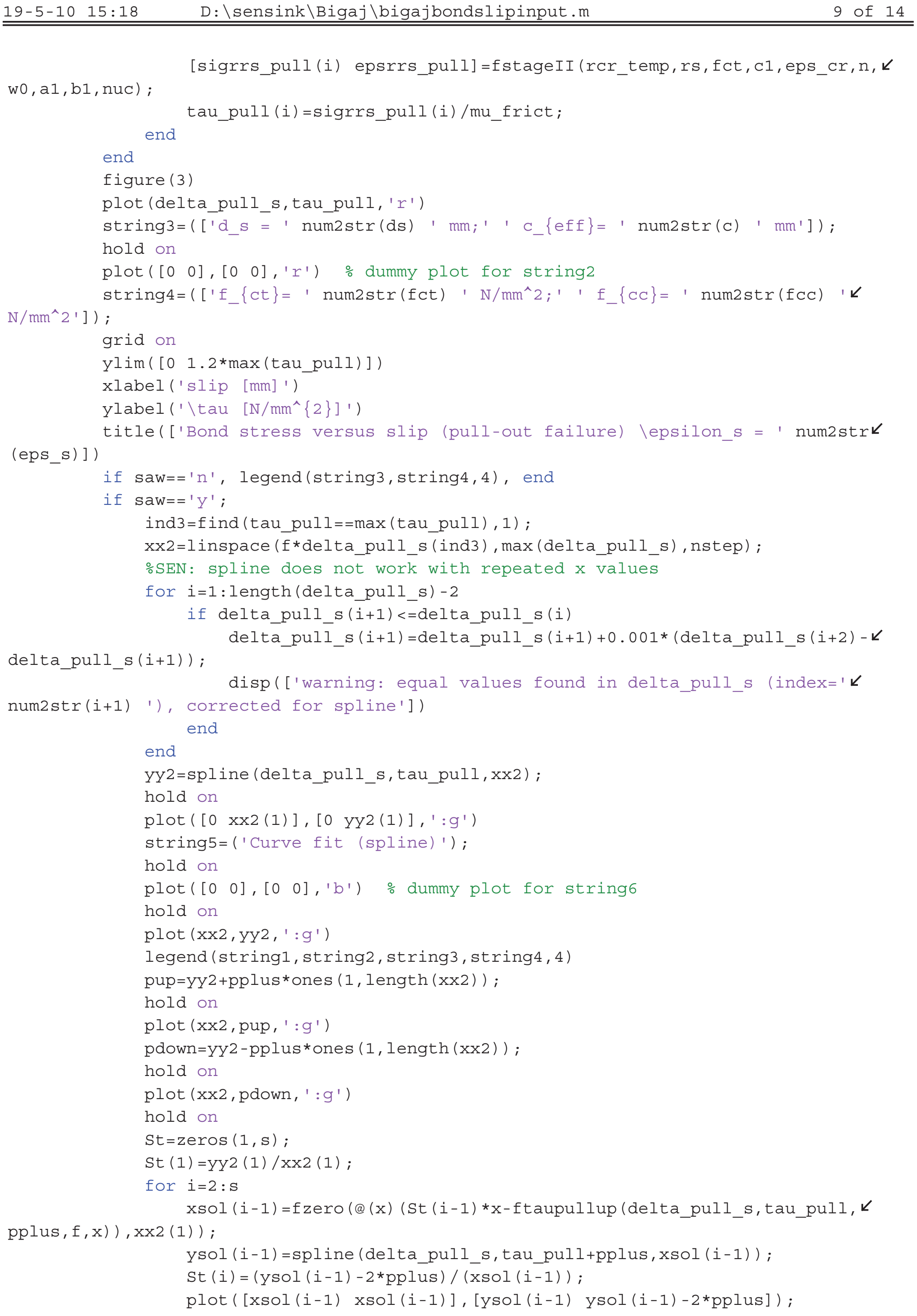


```
19-5-10 15:18      D:\sensink\Bigaj\bigajbondslipinput.m                    5 of 14

        % create sawtooth approximation
        xx2=linspace(slip_split(nstep),max(slip_split),nstep);
        %SEN: spline does not work with repeated x values
        for i=1:length(slip_split)-2
            if slip_split(i+1)<=slip_split(i)
                slip_split(i+1)=slip_split(i+1)+0.001*(slip_split(i+2)-slip_split↙
(i+1));
            end
        end
        yy2=spline(slip_split,tau,xx2);
        hold on
        plot([0 xx2(1)],[0 yy2(1)],':g')
        hold on
        plot([0 0],[0 0],'b')   %dummy plot for string 4
        hold on
        plot(xx2,yy2,':g')
        string3=('Curve fit (spline)');
        pup=yy2+pplus*ones(1,length(xx2));
        hold on
        plot(xx2,pup,':g')
        pdown=yy2-pplus*ones(1,length(xx2));
        hold on
        plot(xx2,pdown,':g')
        hold on
        St=zeros(1,s);
        St(1)=yy2(1)/xx2(1);
        for i=2:s
            xsol(i-1)=fzero(@(x)(St(i-1)*x-ftaupullup(slip_split,tau,pplus,f,x)),xx2↙
(1));
            ysol(i-1)=spline(slip_split,tau+pplus,xsol(i-1));
            if ysol(i-1)<0
                s=i-2;
                disp('number of teeth has been reduced to fit curve')
            end
            St(i)=(ysol(i-1)-2*pplus)/(xsol(i-1));
            plot([xsol(i-1) xsol(i-1)],[ysol(i-1) ysol(i-1)-2*pplus],'b');
            hold on
        end
        for i=2:s-1
            plot([xsol(i-1) xsol(i)],[ysol(i-1)-2*pplus ysol(i)],'b')
        end
        hold on
        plot([0 xsol(1)],[0 ysol(1)])
        string4=(['# teeth = ' num2str(s-1) '; p+ = ' num2str(pplus) ' N/mm^2' '↙
\delta_{max}= ' num2str(xsol(s-1)) ' mm']);
        legend(string1,string2,string3,string4,legendloc)
        xlim([0 1.1*xsol(s-1)]);
        format short e
        disp(['      St(i)  ' '      tau(i)'])
        disp([St(1:s-1)' ysol(1:s-1)'])
        fid = fopen('sawtooth_splitting.txt','wt');
        fprintf(fid,'%s\n','''materials''');
        fprintf(fid,'%g   HARDIA   %8.6E  %8.6E\n',matnr,St(1),ysol(1));
        for j = 2:s-1
            fprintf(fid,'               %8.6E  %8.6E\n',St(j),ysol(j));
        end
        fclose(fid);
        disp('hardia table written to file:   sawtooth_splitting.txt')
    end
else
    % pull-out failure
    disp(['Maximum bond stress found: ' num2str(max(sigrrs)) ' N/mm^2'...
        ' (critical bond stress = ' num2str(tau_b1) ' N/mm^2)'])
    % epsilon_r1_0, find epsilon_r1_0 by linear interpolation
    % SEN: kan ook beter met rcr bepalen in fstageII met fzero waarbij
    % sigrrsII=tau_b1/mu_frict, hieruit volgt eps_r1_0
    ind=find(sigrrs>tau_b1/mu_frict,1);
    eps_r1_0=epsrrs(ind-1)+...
        ((tau_b1-sigrrs(ind-1))/(sigrrs(ind)-sigrrs(ind-1)))*...
        (epsrrs(ind)-epsrrs(ind-1));
    % slip delta_1,0
    delta_1_0=2*eps_r1_0*rs/(tand(phi));    % Eq (4.40)
    % slip delta_3max
    delta_3max=0.33*ds;                     % Table 4.5
    % max. residual bond stress tau_b3max
    tau_b3max=2.5*fct;                      % Table 4.5
    % max. radial strain eps_3max
    sigrrs_b3max=tau_b3max/mu_frict;        % Eq (4.29)
    % epsilon_r3_max, find epsilon_r3_max by linear interpolation
    ind2=find(sigrrs>sigrrs_b3max,1);
    eps_r3max=epsrrs(ind2-1)+...
        ((sigrrs_b3max-sigrrs(ind2-1))/(sigrrs(ind2)-sigrrs(ind2-1)))*...
         (epsrrs(ind2)-epsrrs(ind2-1));
     % calculate slip delta_2
     delta_2_0=delta_3max/2;                % Eq (4.36)
     % calculate epsilon_r2
     eps_r2_0=(eps_r3max+eps_r1_0)/2;       % Eq (4.37)
     delta_pull=linspace(0,delta_3max,250);
     eps_pull=feps_pull_0(eps_r1_0,eps_r2_0,eps_r3max,delta_1_0,delta_2_0,delta_3max,↙
delta_pull);
     figure(2)
     plot(delta_pull,eps_pull,'r')
     grid on
     xlabel('\delta [mm]')
     ylabel('\epsilon_r')
     title('\epsilon_r versus slip (pull-out failure)')
     string1=('\epsilon_s = 0');
     if eps_s==0
         legend(string1,4)
         clear rcr
         tau_pull(1)=0;
         for i=2:length(delta_pull)
             if eps_pull(i)<epsrrsII(1)
                 sigrrs_pull(i)=(eps_pull(i)/epsrrsII(1))*sigrrsII(1);
                 tau_pull(i)=sigrrs_pull(i)/mu_frict;
             else
                 rcr_temp=fminbnd(@(rcr_temp)(fstageIIpull(rcr_temp,rs,fct,c1,eps_cr,↙
n,w0,a1,b1,nuc,eps_pull(i))),rs,c1);
                 [sigrrs_pull(i) epsrrs_pull]=fstageII(rcr_temp,rs,fct,c1,eps_cr,n,↙
w0,a1,b1,nuc);
```



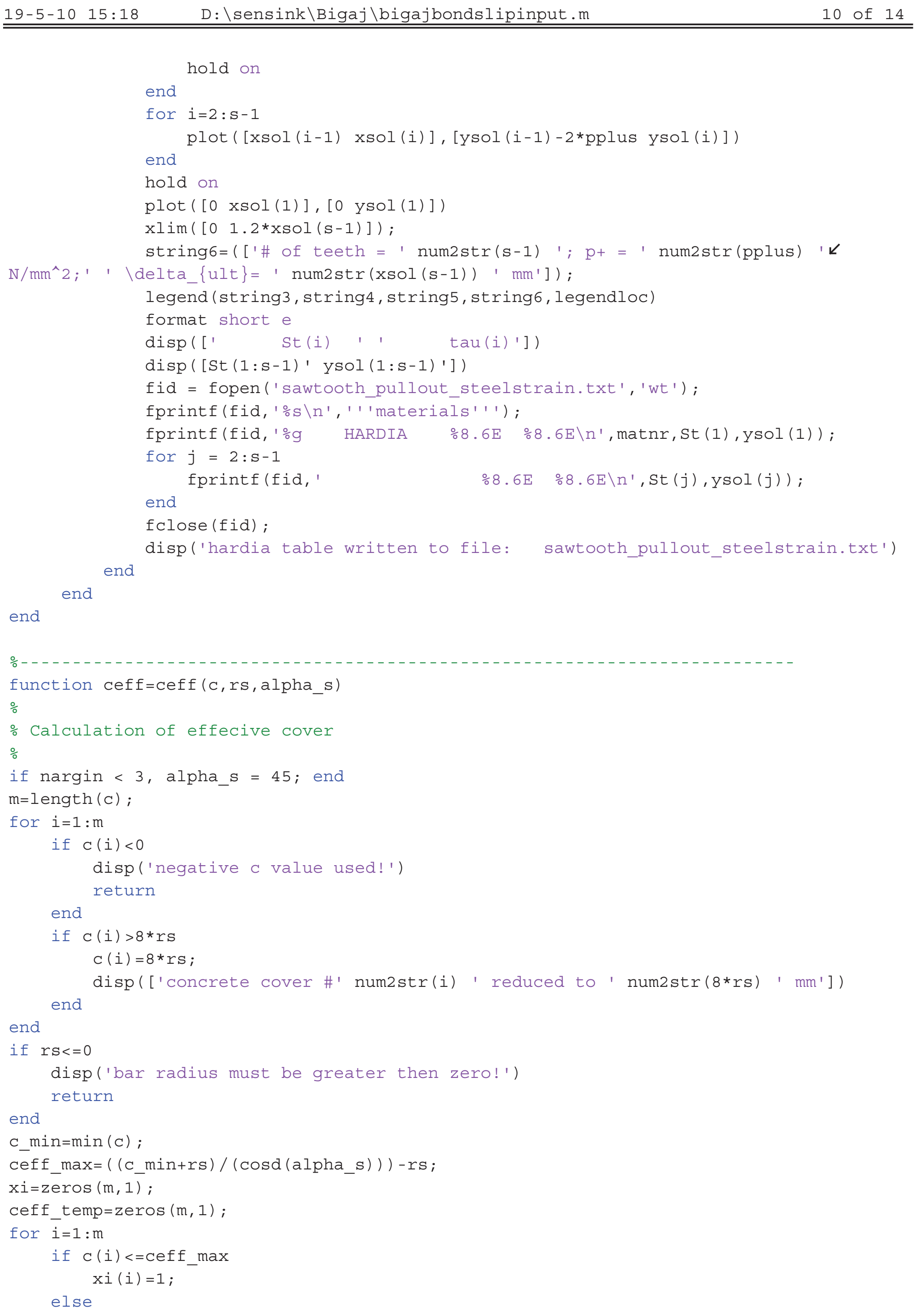
```
19-5-10 15:18      D:\sensink\Bigaj\bigajbondslipinput.m                  7 of 14

                tau_pull(i)=sigrrs_pull(i)/mu_frict;
            end
        end
        figure(3)
        plot(delta_pull,tau_pull,'r','LineWidth',1)
        hold on
        plot([0 0],[0 0],'r','LineWidth',1)  % dummy plot for string 3
        set(gca,'YTick',0:2:round(max(tau_pull))+1)
        string2=(['d_s = ' num2str(ds) ' mm;' ' c_{eff}= ' num2str(c) ' mm']);
        string3=(['f_{ct}= ' num2str(fct) ' N/mm^2;' ' f_{cc}= ' num2str(fcc) '↙
N/mm^2']);
        grid on
        ylim([0 1.2*max(tau_pull)])
        xlabel('\delta [mm]')
        ylabel('\tau [N/mm^{2}]')
        title('Bond stress versus slip (pull-out failure) \epsilon_s = 0')
        if saw=='n', legend(string2,string3,4), end
        if saw=='y';
            % create sawtooth approximation
            ind3=find(tau_pull==max(tau_pull),1);
            xx2=linspace(f*delta_pull(ind3),max(delta_pull),nstep);
            %SEN: spline does not work with repeated x values
            for i=1:length(delta_pull)-2
                if delta_pull(i+1)<=delta_pull(i)
                    delta_pull(i+1)=delta_pull(i+1)+0.001*(delta_pull(i+2)-↙
delta_pull(i+1));
                end
            end
            yy2=spline(delta_pull,tau_pull,xx2);
            hold on
            plot([0 xx2(1)],[0 yy2(1)],':g')
            string4=('Curve fit (spline)');
            hold on
            plot([0 0],[0 0],'b')  % dummy plot for string5
            hold on
            plot(xx2,yy2,':g')
            pup=yy2+pplus*ones(1,length(xx2));
            hold on
            plot(xx2,pup,':g')
            pdown=yy2-pplus*ones(1,length(xx2));
            hold on
            plot(xx2,pdown,':g')
            hold on
            St=zeros(1,s);
            St(1)=yy2(1)/xx2(1);
            for i=2:s
                xsol(i-1)=fzero(@(x)(St(i-1)*x-ftaupullup(delta_pull,tau_pull,pplus,↙
f,x)),xx2(1));
                ysol(i-1)=spline(delta_pull,tau_pull+pplus,xsol(i-1));
                St(i)=(ysol(i-1)-2*pplus)/(xsol(i-1));
                plot([xsol(i-1) xsol(i-1)],[ysol(i-1) ysol(i-1)-2*pplus],'b');
                hold on
            end
            for i=2:s-1
                plot([xsol(i-1) xsol(i)],[ysol(i-1)-2*pplus ysol(i)],'b')
```

```
                 hold on
             end
             plot([0 xsol(1)],[0 ysol(1)],'b')
             xlim([0 1.2*xsol(s-1)]);
             string5=(['# teeth = ' num2str(s-1) '; p+ = ' num2str(pplus) ' N/mm^2;'↙
' \delta_{ult}= ' num2str(xsol(s-1)) ' mm']);
             legend(string2,string3,string4,string5,legendloc)
             format short e
             disp(['      St(i)  ' '      tau(i)'])
             disp([St(1:s-1)' ysol(1:s-1)'])
             fid = fopen('sawtooth_pullout.txt','wt');
             fprintf(fid,'%s\n','''materials''');
             fprintf(fid,'%g    HARDIA    %8.6E  %8.6E\n',matnr,St(1),ysol(1));
             for j = 2:s-1
                 fprintf(fid,'               %8.6E  %8.6E\n',St(j),ysol(j));
             end
             fclose(fid);
             disp('hardia table written to file:   sawtooth_pullout.txt')
         end
     end
     % --------------------------------------------------------------------
     if eps_s>0
         % slip delta_1_s Eq (4.34)
         delta_1_s=delta_1_0;
         % epsilon_r3
         eps_r3=eps_r3max*exp(-8.5*eps_s);
         % epsilon_r1_s Eq (4.35)
         eps_r1_s=(eps_r1_0-eps_r3)*exp(-30*eps_s)+eps_r3;
         % delta_3min
         delta_3min=2.1*delta_1_0;
         % delta_3_s Eq (4.38)
         delta_3_s=(delta_3max-delta_3min)*exp(-100*eps_s)+delta_3min;
         % epsilon_r3_s Eq (4.39)
         eps_r3_s=eps_r3max*exp(-8.5*eps_s);
         % delta_2_s (Eq 4.36)
         delta_2_s=delta_3_s/2;
         % epsilon_r2_s (Eq 4.37)
         eps_r2_s=(eps_r1_s+eps_r3_s)/2;
         delta_pull_s=linspace(0,delta_3_s,250);
         eps_pull_s=feps_pull_s(eps_r1_s,eps_r2_s,eps_r3_s,delta_1_s,delta_2_s,↙
delta_3_s,delta_pull_s);
         hold on
         plot(delta_pull_s,eps_pull_s,'b')
         string2=(['\epsilon_s = ' num2str(eps_s)]);
         legend(string1,string2,4)
         %-----
         clear rcr
         tau_pull(1)=0;
         for i=2:length(delta_pull_s)
             if eps_pull_s(i)<epsrrsII(1)
                 sigrrs_pull(i)=(eps_pull_s(i)/epsrrsII(1))*sigrrsII(1);
                 tau_pull(i)=sigrrs_pull(i)/mu_frict;
             else
                 rcr_temp=fminbnd(@(rcr_temp)(fstageIIpull(rcr_temp,rs,fct,c1,eps_cr,↙
n,w0,a1,b1,nuc,eps_pull_s(i))),rs,c1);
```

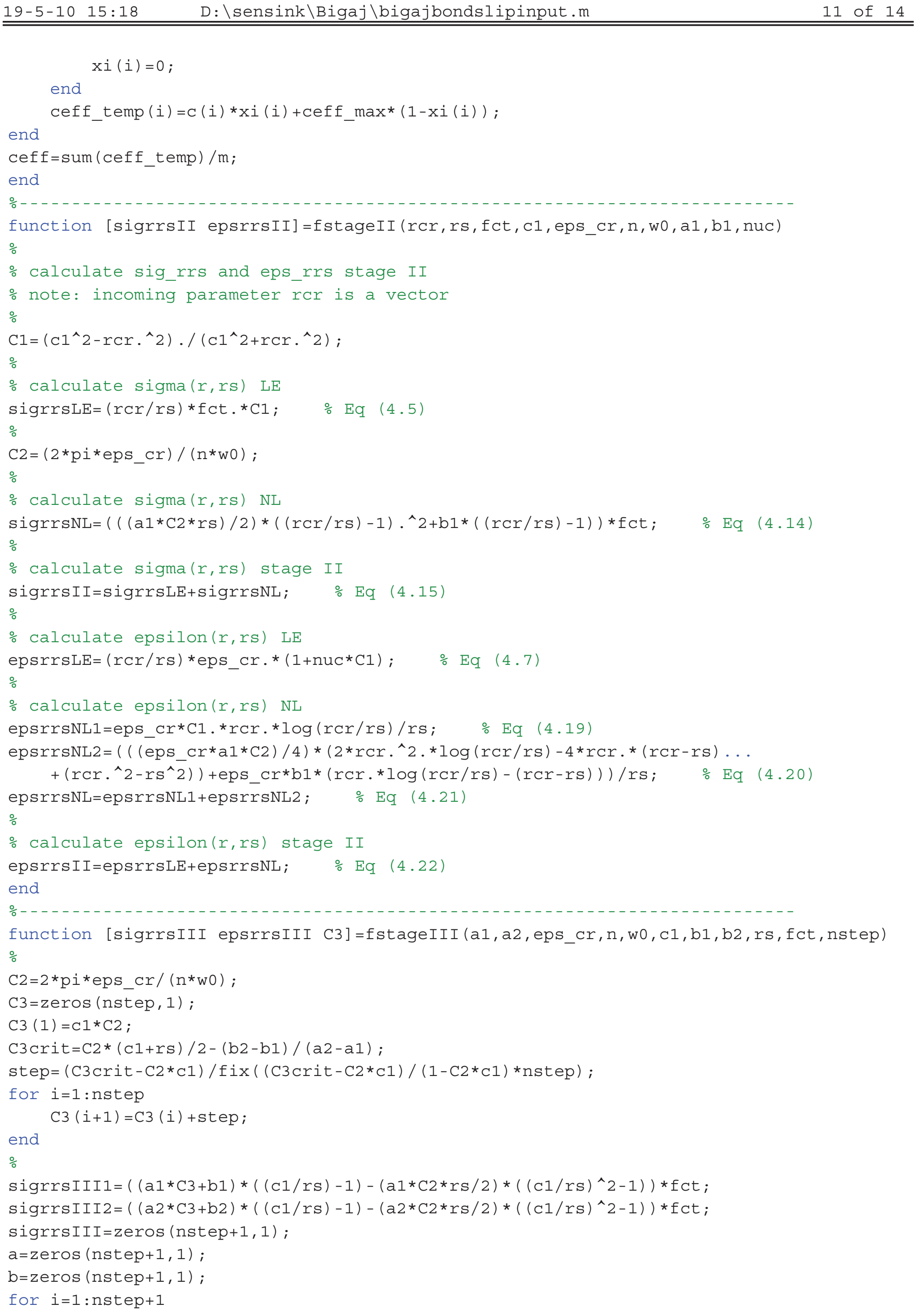


19-5-10 15:18 D:\sensink\Bigaj\bigajbondslipinput.m 9 of 14

```
                [sigrrs_pull(i) epsrrs_pull]=fstageII(rcr_temp,rs,fct,c1,eps_cr,n,↙
w0,a1,b1,nuc);
                tau_pull(i)=sigrrs_pull(i)/mu_frict;
            end
        end
        figure(3)
        plot(delta_pull_s,tau_pull,'r')
        string3=(['d_s = ' num2str(ds) ' mm;' ' c_{eff}= ' num2str(c) ' mm']);
        hold on
        plot([0 0],[0 0],'r')  % dummy plot for string2
        string4=(['f_{ct}= ' num2str(fct) ' N/mm^2;' ' f_{cc}= ' num2str(fcc) '↙
N/mm^2']);
        grid on
        ylim([0 1.2*max(tau_pull)])
        xlabel('slip [mm]')
        ylabel('\tau [N/mm^{2}]')
        title(['Bond stress versus slip (pull-out failure) \epsilon_s = ' num2str↙
(eps_s)])
        if saw=='n', legend(string3,string4,4), end
        if saw=='y';
            ind3=find(tau_pull==max(tau_pull),1);
            xx2=linspace(f*delta_pull_s(ind3),max(delta_pull_s),nstep);
            %SEN: spline does not work with repeated x values
            for i=1:length(delta_pull_s)-2
                if delta_pull_s(i+1)<=delta_pull_s(i)
                    delta_pull_s(i+1)=delta_pull_s(i+1)+0.001*(delta_pull_s(i+2)-↙
delta_pull_s(i+1));
                    disp(['warning: equal values found in delta_pull_s (index='↙
num2str(i+1) '), corrected for spline'])
                end
            end
            yy2=spline(delta_pull_s,tau_pull,xx2);
            hold on
            plot([0 xx2(1)],[0 yy2(1)],':g')
            string5=('Curve fit (spline)');
            hold on
            plot([0 0],[0 0],'b')  % dummy plot for string6
            hold on
            plot(xx2,yy2,':g')
            legend(string1,string2,string3,string4,4)
            pup=yy2+pplus*ones(1,length(xx2));
            hold on
            plot(xx2,pup,':g')
            pdown=yy2-pplus*ones(1,length(xx2));
            hold on
            plot(xx2,pdown,':g')
            hold on
            St=zeros(1,s);
            St(1)=yy2(1)/xx2(1);
            for i=2:s
                xsol(i-1)=fzero(@(x)(St(i-1)*x-ftaupullup(delta_pull_s,tau_pull,↙
pplus,f,x)),xx2(1));
                ysol(i-1)=spline(delta_pull_s,tau_pull+pplus,xsol(i-1));
                St(i)=(ysol(i-1)-2*pplus)/(xsol(i-1));
                plot([xsol(i-1) xsol(i-1)],[ysol(i-1) ysol(i-1)-2*pplus]);
```



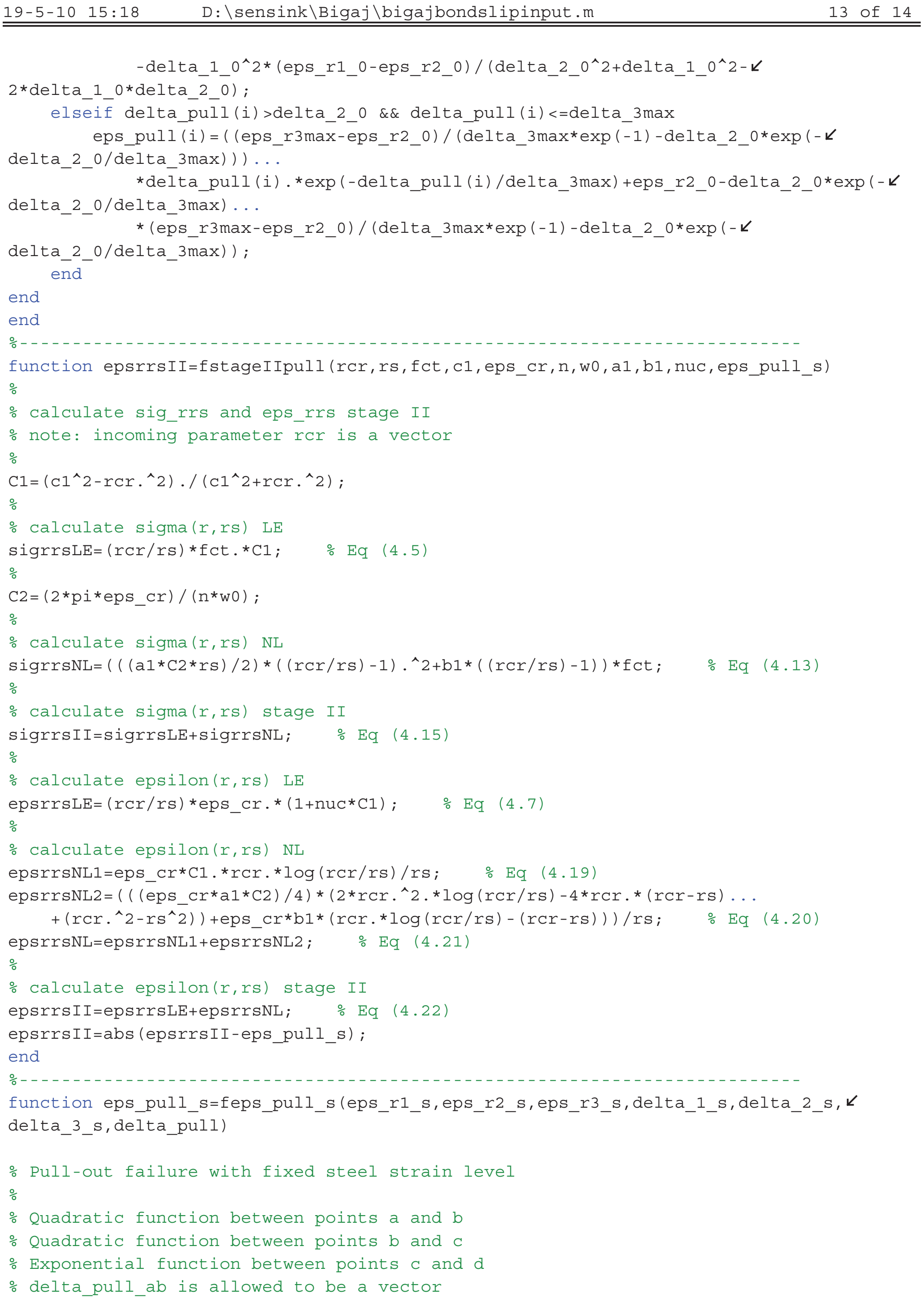
```
                 hold on
             end
             for i=2:s-1
                 plot([xsol(i-1) xsol(i)],[ysol(i-1)-2*pplus ysol(i)])
             end
             hold on
             plot([0 xsol(1)],[0 ysol(1)])
             xlim([0 1.2*xsol(s-1)]);
             string6=(['# of teeth = ' num2str(s-1) '; p+ = ' num2str(pplus) '↙
N/mm^2;' ' \delta_{ult}= ' num2str(xsol(s-1)) ' mm']);
             legend(string3,string4,string5,string6,legendloc)
             format short e
             disp(['      St(i)  ' '      tau(i)'])
             disp([St(1:s-1)' ysol(1:s-1)'])
             fid = fopen('sawtooth_pullout_steelstrain.txt','wt');
             fprintf(fid,'%s\n','''materials''');
             fprintf(fid,'%g    HARDIA    %8.6E  %8.6E\n',matnr,St(1),ysol(1));
             for j = 2:s-1
                 fprintf(fid,'               %8.6E  %8.6E\n',St(j),ysol(j));
             end
             fclose(fid);
             disp('hardia table written to file:   sawtooth_pullout_steelstrain.txt')
         end
     end
end

%--------------------------------------------------------------------------
function ceff=ceff(c,rs,alpha_s)
%
% Calculation of effecive cover
%
if nargin < 3, alpha_s = 45; end
m=length(c);
for i=1:m
    if c(i)<0
        disp('negative c value used!')
        return
    end
    if c(i)>8*rs
        c(i)=8*rs;
        disp(['concrete cover #' num2str(i) ' reduced to ' num2str(8*rs) ' mm'])
    end
end
if rs<=0
    disp('bar radius must be greater then zero!')
    return
end
c_min=min(c);
ceff_max=((c_min+rs)/(cosd(alpha_s)))-rs;
xi=zeros(m,1);
ceff_temp=zeros(m,1);
for i=1:m
    if c(i)<=ceff_max
        xi(i)=1;
    else
```



```
19-5-10 15:18      D:\sensink\Bigaj\bigajbondslipinput.m                    11 of 14
```

```
        xi(i)=0;
    end
    ceff_temp(i)=c(i)*xi(i)+ceff_max*(1-xi(i));
end
ceff=sum(ceff_temp)/m;
end
%-------------------------------------------------------------------------
function [sigrrsII epsrrsII]=fstageII(rcr,rs,fct,c1,eps_cr,n,w0,a1,b1,nuc)
%
% calculate sig_rrs and eps_rrs stage II
% note: incoming parameter rcr is a vector
%
C1=(c1^2-rcr.^2)./(c1^2+rcr.^2);
%
% calculate sigma(r,rs) LE
sigrrsLE=(rcr/rs)*fct.*C1;    % Eq (4.5)
%
C2=(2*pi*eps_cr)/(n*w0);
%
% calculate sigma(r,rs) NL
sigrrsNL=(((a1*C2*rs)/2)*((rcr/rs)-1).^2+b1*((rcr/rs)-1))*fct;    % Eq (4.14)
%
% calculate sigma(r,rs) stage II
sigrrsII=sigrrsLE+sigrrsNL;    % Eq (4.15)
%
% calculate epsilon(r,rs) LE
epsrrsLE=(rcr/rs)*eps_cr.*(1+nuc*C1);    % Eq (4.7)
%
% calculate epsilon(r,rs) NL
epsrrsNL1=eps_cr*C1.*rcr.*log(rcr/rs)/rs;    % Eq (4.19)
epsrrsNL2=(((eps_cr*a1*C2)/4)*(2*rcr.^2.*log(rcr/rs)-4*rcr.*(rcr-rs)...
    +(rcr.^2-rs^2))+eps_cr*b1*(rcr.*log(rcr/rs)-(rcr-rs)))/rs;    % Eq (4.20)
epsrrsNL=epsrrsNL1+epsrrsNL2;    % Eq (4.21)
%
% calculate epsilon(r,rs) stage II
epsrrsII=epsrrsLE+epsrrsNL;    % Eq (4.22)
end
%-------------------------------------------------------------------------
function [sigrrsIII epsrrsIII C3]=fstageIII(a1,a2,eps_cr,n,w0,c1,b1,b2,rs,fct,nstep)
%
C2=2*pi*eps_cr/(n*w0);
C3=zeros(nstep,1);
C3(1)=c1*C2;
C3crit=C2*(c1+rs)/2-(b2-b1)/(a2-a1);
step=(C3crit-C2*c1)/fix((C3crit-C2*c1)/(1-C2*c1)*nstep);
for i=1:nstep
    C3(i+1)=C3(i)+step;
end
%
sigrrsIII1=((a1*C3+b1)*((c1/rs)-1)-(a1*C2*rs/2)*((c1/rs)^2-1))*fct;
sigrrsIII2=((a2*C3+b2)*((c1/rs)-1)-(a2*C2*rs/2)*((c1/rs)^2-1))*fct;
sigrrsIII=zeros(nstep+1,1);
a=zeros(nstep+1,1);
b=zeros(nstep+1,1);
for i=1:nstep+1
```

```
    sigrrsIII(i)=max(sigrrsIII1(i),sigrrsIII2(i));
    if sigrrsIII1(i)>=sigrrsIII2(i)
        a(i)=a1;
        b(i)=b1;
    else
        a(i)=a2;
        b(i)=b2;
    end
end
% calculate epsilon(r,rs) RBM
epsrrsRBM=C3*(n*w0/(2*pi*rs));
% calculate epsilon(r,rs) delta_c
epsrrsdc=zeros(nstep+1,1);
for i=1:nstep+1
    epsrrsdc(i)=eps_cr*(a(i)*C3(i)+b(i))*((c1/rs)*log(c1/rs)-(c1/rs)+1)...
        -(a(i)*C2*eps_cr*rs/4)*(2*(c1/rs)^2*log(c1/rs)-(c1/rs)^2+1);
end
epsrrsIII=epsrrsRBM+epsrrsdc;
end
%--------------------------------------------------------------------------
function taupull=ftaupullup(delta_pull,tau_pull,pplus,f,x)

index=find(tau_pull==max(tau_pull),1);
xmin=f*delta_pull(index);

if x>xmin
    taupull=spline(delta_pull,tau_pull+pplus,x);
else
    taupull=spline(delta_pull,tau_pull+pplus,xmin);
end

end
%--------------------------------------------------------------------------
function eps_pull=feps_pull_0(eps_r1_0,eps_r2_0,eps_r3max,delta_1_0,delta_2_0,↙
delta_3max,delta_pull)

% Pull-out failure with zero steel strain
%
% Quadratic function between points a and b
% Quadratic function between points b and c
% Exponential function between points c and d
% delta_pull_ab is allowed to be a vector

eps_pull=zeros(length(delta_pull),1);
for i=1:length(delta_pull)
    if delta_pull(i)<=delta_1_0
        eps_pull(i)=-(eps_r1_0/(delta_1_0^2))*delta_pull(i).^2....
            +2*(eps_r1_0/delta_1_0)*delta_pull(i);
    elseif delta_pull(i)>delta_1_0 && delta_pull(i)<=delta_2_0
        eps_pull(i)=-((eps_r1_0-eps_r2_0)/(delta_2_0^2+delta_1_0^2-↙
2*delta_1_0*delta_2_0))...
            *delta_pull(i).^2+...
            (2*delta_1_0*(eps_r1_0-eps_r2_0)/(delta_2_0^2+delta_1_0^2-↙
2*delta_1_0*delta_2_0))...
            *delta_pull(i)+eps_r1_0...
```



```
19-5-10 15:18      D:\sensink\Bigaj\bigajbondslipinput.m                 13 of 14

            -delta_1_0^2*(eps_r1_0-eps_r2_0)/(delta_2_0^2+delta_1_0^2-↙
2*delta_1_0*delta_2_0);
    elseif delta_pull(i)>delta_2_0 && delta_pull(i)<=delta_3max
        eps_pull(i)=((eps_r3max-eps_r2_0)/(delta_3max*exp(-1)-delta_2_0*exp(-↙
delta_2_0/delta_3max)))...
            *delta_pull(i).*exp(-delta_pull(i)/delta_3max)+eps_r2_0-delta_2_0*exp(-↙
delta_2_0/delta_3max)...
            *(eps_r3max-eps_r2_0)/(delta_3max*exp(-1)-delta_2_0*exp(-↙
delta_2_0/delta_3max));
    end
end
end
%---------------------------------------------------------------------------
function epsrrsII=fstageIIpull(rcr,rs,fct,c1,eps_cr,n,w0,a1,b1,nuc,eps_pull_s)
%
% calculate sig_rrs and eps_rrs stage II
% note: incoming parameter rcr is a vector
%
C1=(c1^2-rcr.^2)./(c1^2+rcr.^2);
%
% calculate sigma(r,rs) LE
sigrrsLE=(rcr/rs)*fct.*C1;    % Eq (4.5)
%
C2=(2*pi*eps_cr)/(n*w0);
%
% calculate sigma(r,rs) NL
sigrrsNL=(((a1*C2*rs)/2)*((rcr/rs)-1).^2+b1*((rcr/rs)-1))*fct;    % Eq (4.13)
%
% calculate sigma(r,rs) stage II
sigrrsII=sigrrsLE+sigrrsNL;    % Eq (4.15)
%
% calculate epsilon(r,rs) LE
epsrrsLE=(rcr/rs)*eps_cr.*(1+nuc*C1);    % Eq (4.7)
%
% calculate epsilon(r,rs) NL
epsrrsNL1=eps_cr*C1.*rcr.*log(rcr/rs)/rs;    % Eq (4.19)
epsrrsNL2=(((eps_cr*a1*C2)/4)*(2*rcr.^2.*log(rcr/rs)-4*rcr.*(rcr-rs)...
    +(rcr.^2-rs^2))+eps_cr*b1*(rcr.*log(rcr/rs)-(rcr-rs)))/rs;    % Eq (4.20)
epsrrsNL=epsrrsNL1+epsrrsNL2;    % Eq (4.21)
%
% calculate epsilon(r,rs) stage II
epsrrsII=epsrrsLE+epsrrsNL;    % Eq (4.22)
epsrrsII=abs(epsrrsII-eps_pull_s);
end
%---------------------------------------------------------------------------
function eps_pull_s=feps_pull_s(eps_r1_s,eps_r2_s,eps_r3_s,delta_1_s,delta_2_s,↙
delta_3_s,delta_pull)

% Pull-out failure with fixed steel strain level
%
% Quadratic function between points a and b
% Quadratic function between points b and c
% Exponential function between points c and d
% delta_pull_ab is allowed to be a vector
```

```
eps_pull_s=zeros(length(delta_pull),1);
for i=1:length(delta_pull)
    if delta_pull(i)<=delta_1_s
        eps_pull_s(i)=-(eps_r1_s/(delta_1_s^2))*delta_pull(i).^2+2*↙
(eps_r1_s/delta_1_s)...
         *delta_pull(i);
    elseif delta_pull(i)>delta_1_s && delta_pull(i)<=delta_2_s
        eps_pull_s(i)=-((eps_r1_s-eps_r2_s)/(delta_2_s^2+delta_1_s^2-↙
2*delta_1_s*delta_2_s))...
         *delta_pull(i).^2+...
         (2*delta_1_s*(eps_r1_s-eps_r2_s)/(delta_2_s^2+delta_1_s^2-↙
2*delta_1_s*delta_2_s))...
         *delta_pull(i)+eps_r1_s...
         -delta_1_s^2*(eps_r1_s-eps_r2_s)/(delta_2_s^2+delta_1_s^2-↙
2*delta_1_s*delta_2_s);
    elseif delta_pull(i)>delta_2_s && delta_pull(i)<=delta_3_s
        eps_pull_s(i)=((eps_r3_s-eps_r2_s)/(delta_3_s*exp(-1)-delta_2_s*exp(-↙
delta_2_s/delta_3_s)))...
         *delta_pull(i).*exp(-delta_pull(i)/delta_3_s)+eps_r2_s-delta_2_s*exp(-↙
delta_2_s/delta_3_s)...
         *(eps_r3_s-eps_r2_s)/(delta_3_s*exp(-1)-delta_2_s*exp(-↙
delta_2_s/delta_3_s));
    end
end
end
%--------------------------------------------------------------------------
end
```

Appendix B

Matlab bond model: verification

In this appendix a description will be given of the verifications performed on the Matlab bond model. The calculated bond-slip curves use the same parameters as the ones reported in [14]. Three concrete qualities are used: B30, B60 and B100. Also three rebar diameters are used: 12, 16 and 20 mm.

Figure B.1 shows a comparison of the Matlab bond model with a previous Excel version in terms of the radial stress-radial strain response of the thick-walled cylinder for the various rebar diameters for B30.

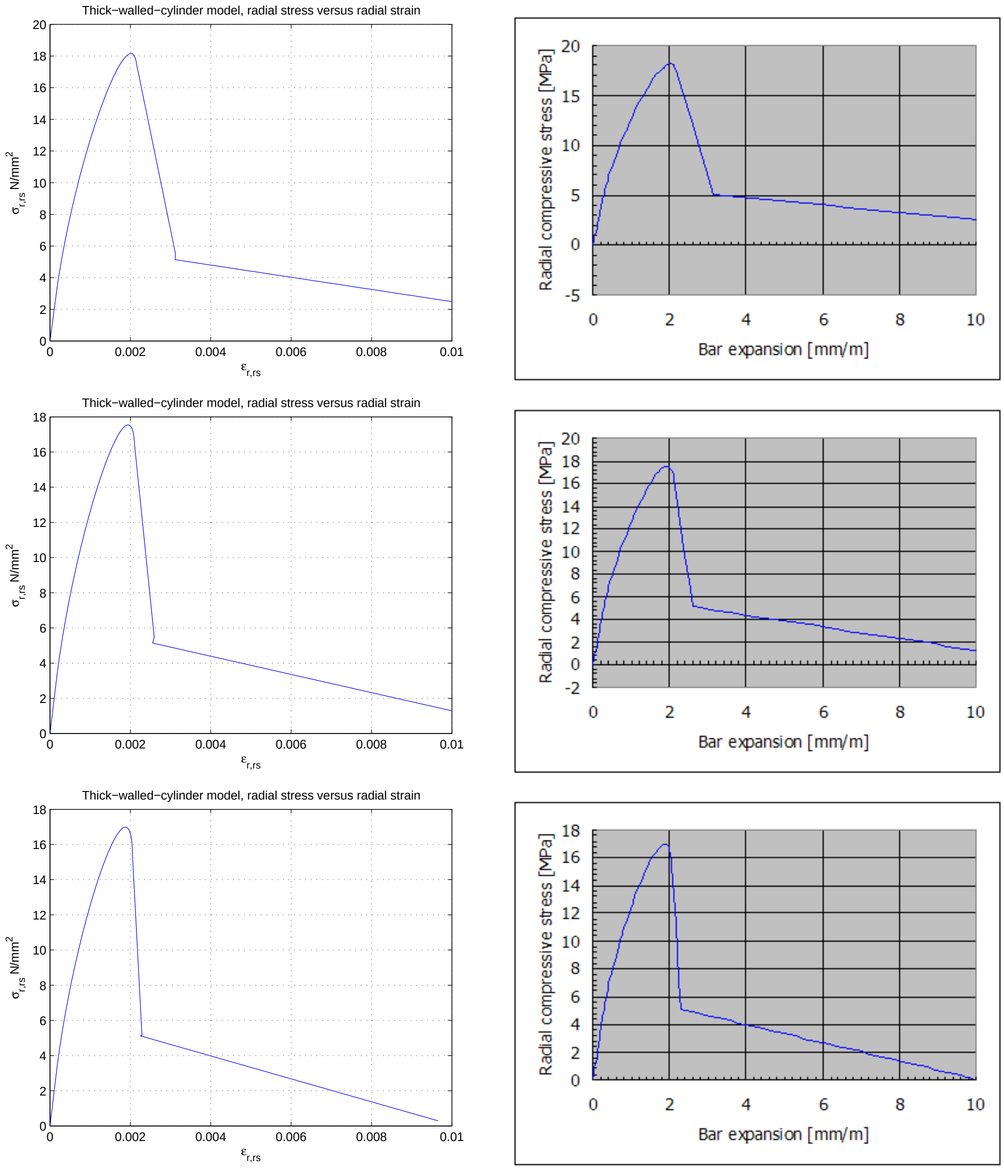


Figure B.1: *Radial stress-strain response thick-walled cylinder, Matlab bond model (left) versus Excel version (right) for B30, from top to bottom: Ø12, Ø16 and Ø20 mm (f_{cc}=30 N/mm^2, f_{ct}=2.57 N/mm^2, E_c=27742 N/mm^2, c_{eff}=4xd_s, ν=0.2)*

Figure B.2 shows a comparison of the Matlab bond model with a previous Excel version in terms of the radial stress-radial strain response of the thick-walled cylinder for the various rebar diameters for B60.

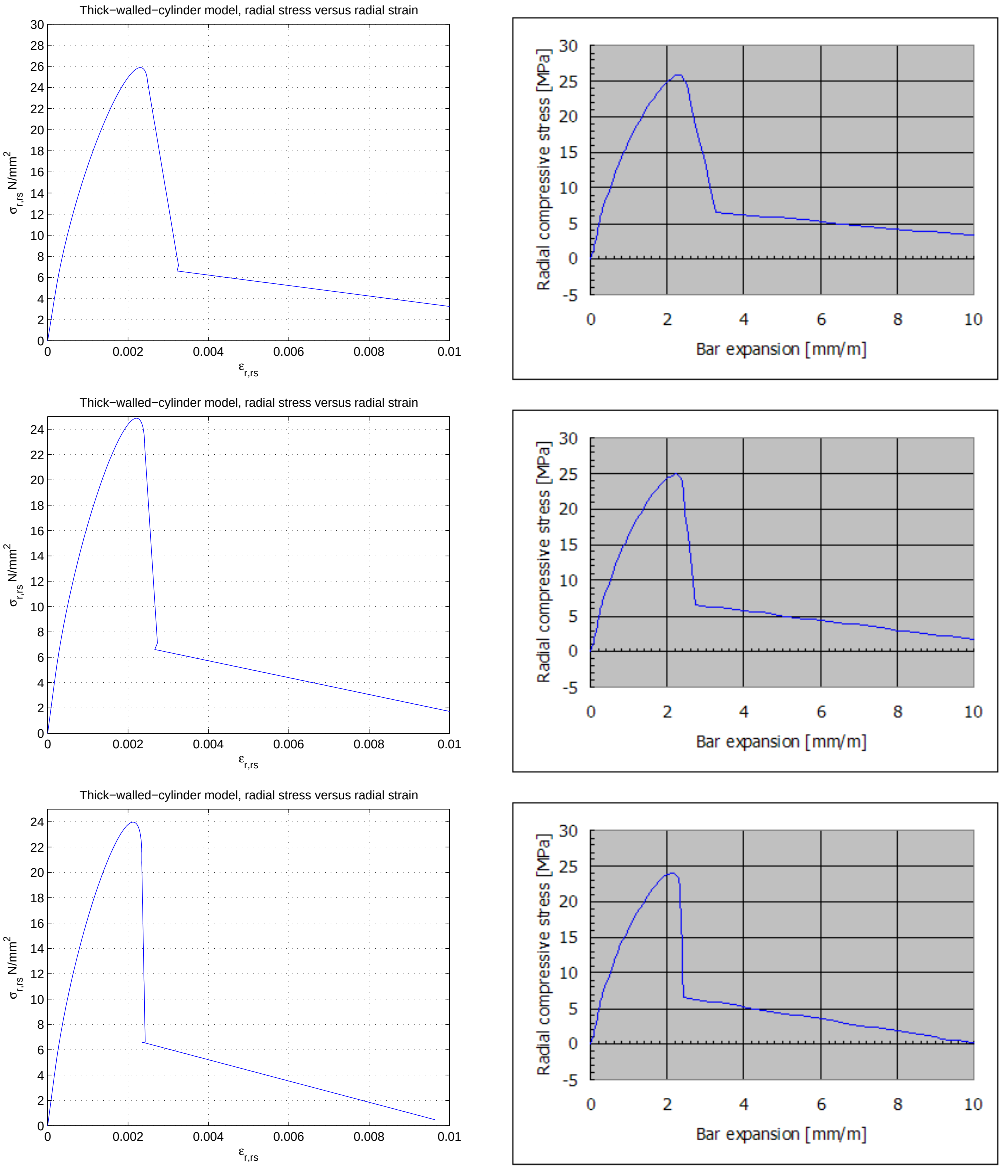


Figure B.2: Radial stress-strain response thick-walled cylinder, Matlab bond model (left) versus Excel version (right) for B60, from top to bottom: Ø12, Ø16 and Ø20 mm (f_{cc}=62 N/mm^2, f_{ct}=3.75 N/mm^2, E_c=34492 N/mm^2, c_{eff}=4xd_s, ν=0.2)

Figure B.3 shows a comparison of the Matlab bond model with a previous Excel version in terms of the radial stress-radial strain response of the thick-walled cylinder for the various rebar diameters for B100.

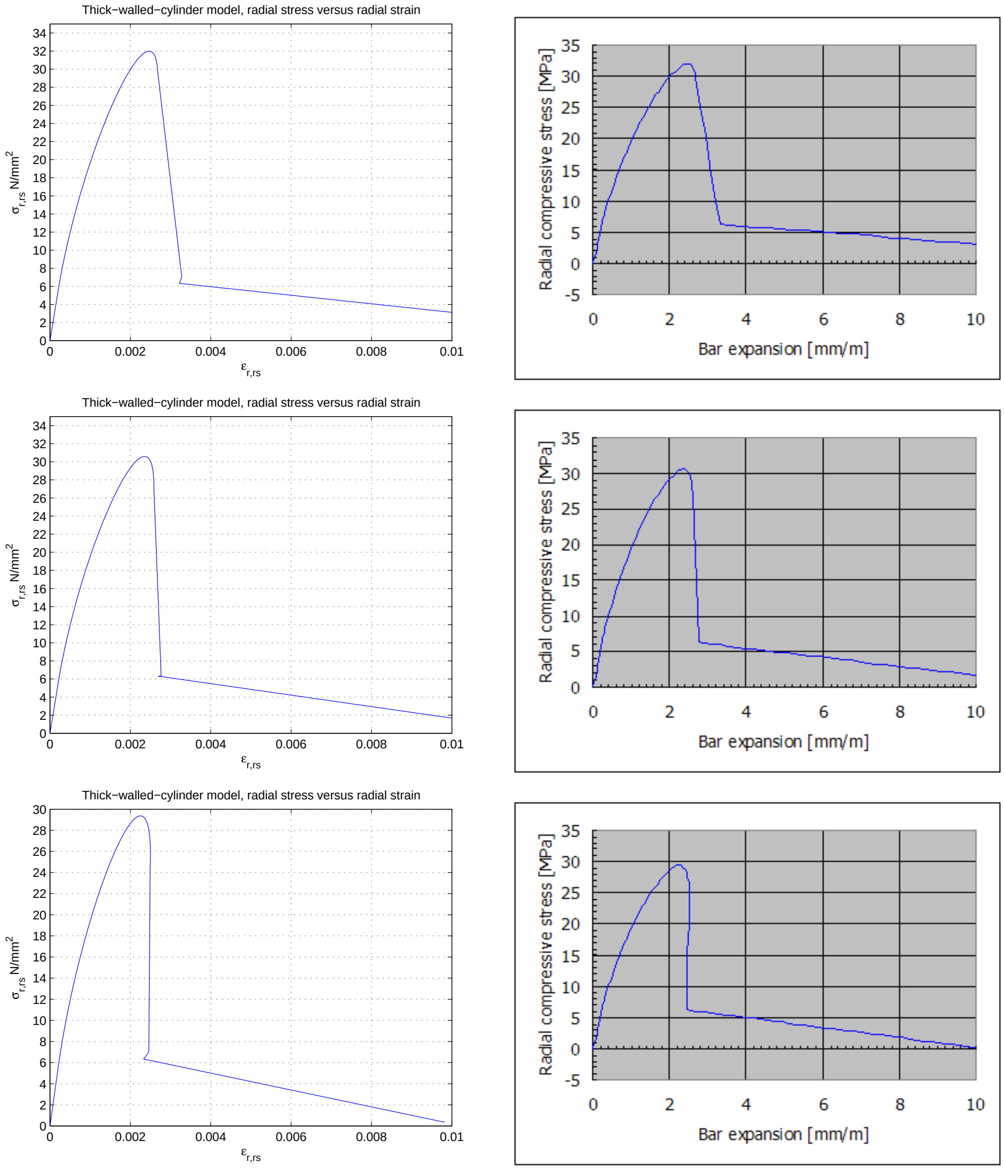


Figure B.3: Radial stress-strain response thick-walled cylinder, Matlab bond model (left) versus Excel version (right) for B100, from top to bottom: Ø12, Ø16 and Ø20 mm (f_{cc}=100 N/mm^2, f_{ct}=4.73 N/mm^2, E_c=39811 N/mm^2, c_{eff}=4xd_s, ν=0.2)

In figure B.4 bond-slip relations are given for B30. These have been compared to [14]. The initial elastic stiffness is given in the caption for each rebar diameter.

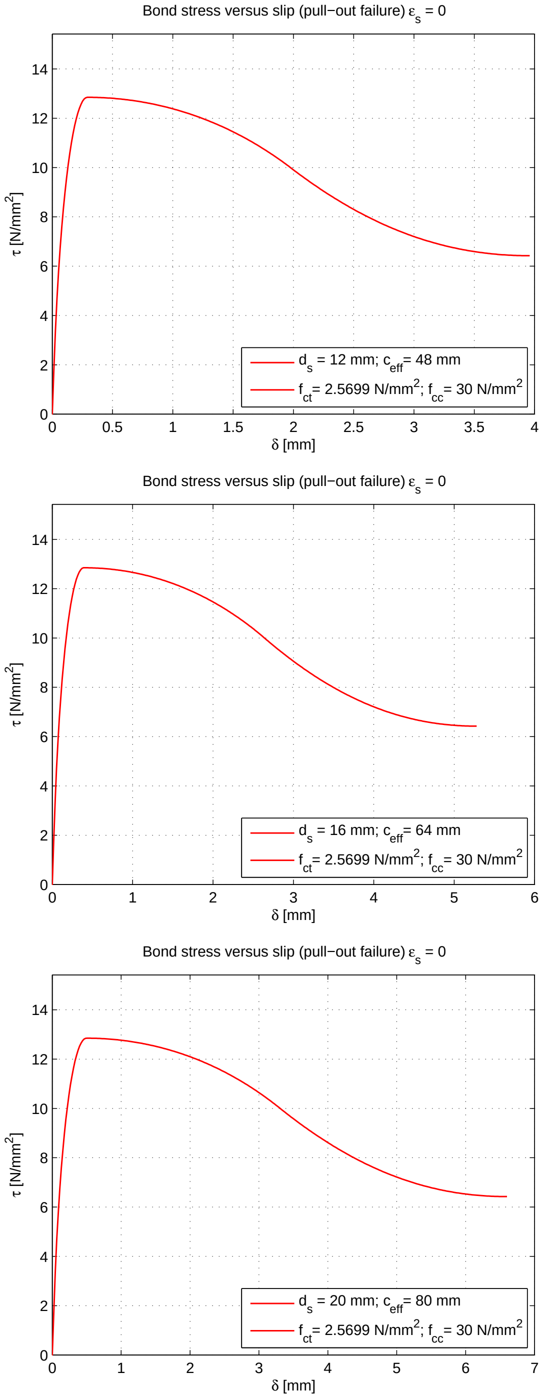


Figure B.4: Bond-slip calculated with Matlab bond model for B30 and ϵ_s=0, from top to bottom: Ø12 ($S_{t,el}$=129 N/mm^3), Ø16 ($S_{t,el}$=97 N/mm^3) and Ø20 mm ($S_{t,el}$=77 N/mm^3)(f_{cc}=30 N/mm^2, f_{ct}=2.57 N/mm^2, E_c=27742 N/mm^2, c_{eff}=4xd_s, ν=0.2)

In figure B.5 bond-slip relations are given for B60. These have been compared to [14]. The initial elastic stiffness is given in the caption for each rebar diameter.

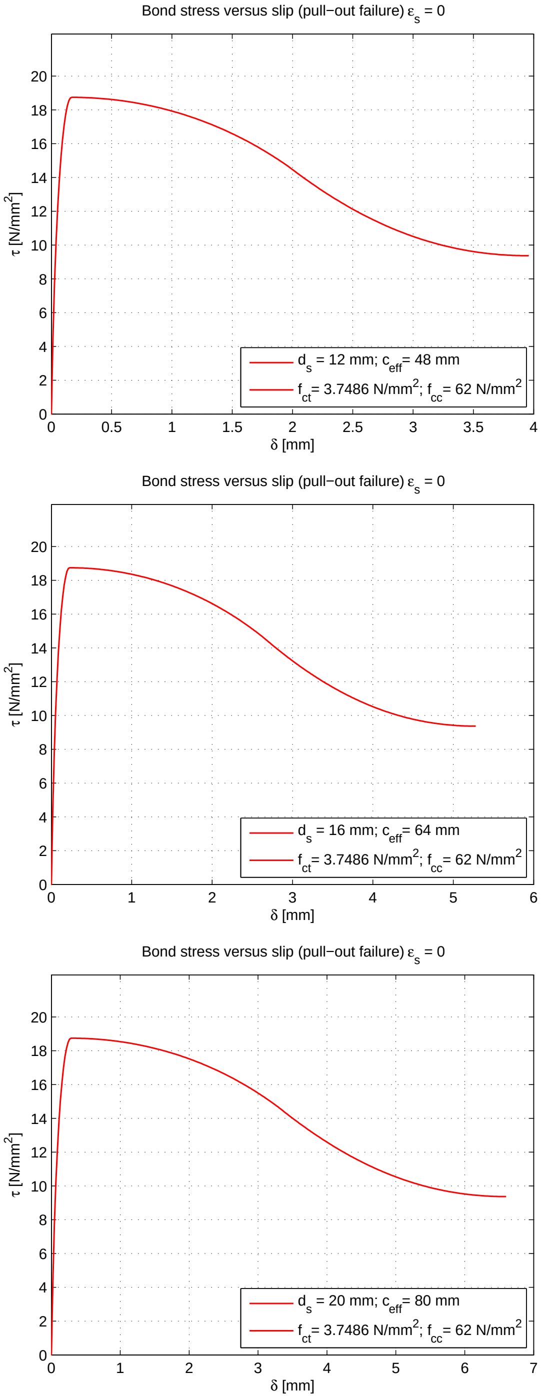


Figure B.5: Bond-slip calculated with Matlab bond model for B60 and ϵ_s=0, from top to bottom: Ø12 ($S_{t,el}$=341 N/mm^3), Ø16 ($S_{t,el}$=256 N/mm^3) and Ø20 mm ($S_{t,el}$=205 N/mm^3)(f_{cc}=62 N/mm^2, f_{ct}=3.75 N/mm^2, E_c=34492 N/mm^2, c_{eff}=4xd_s, ν=0.2)

In figure B.6 bond-slip relations are given for B100. These have been compared to [14]. The initial elastic stiffness is given in the caption for each rebar diameter.

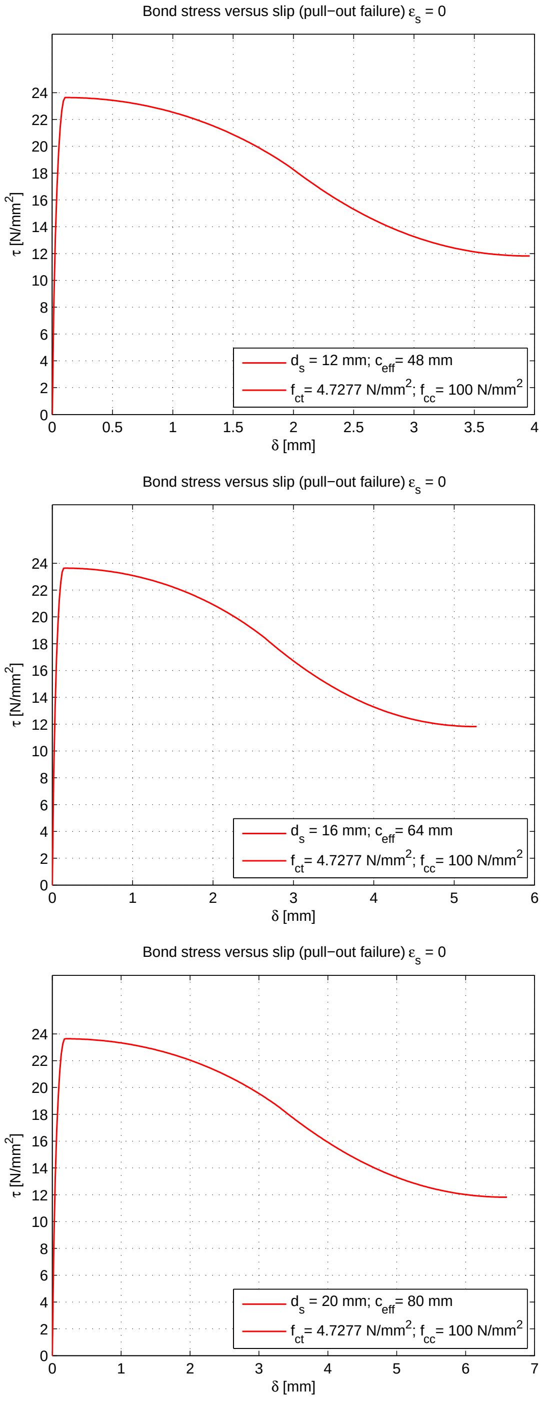


Figure B.6: Bond-slip calculated with Matlab bond model for B100 and ϵ_s=0, from top to bottom: Ø12 ($S_{t,el}$=700 N/mm^3), Ø16 ($S_{t,el}$=519 N/mm^3) and Ø20 mm ($S_{t,el}$=416 N/mm^3)(f_{cc}=100 N/mm^2, f_{ct}=4.73 N/mm^2, E_c=39811 N/mm^2, c_{eff}=4xd_s, ν=0.2)

Appendix C

Bond model: calculation of stage III

The calculation of $\boldsymbol{\sigma}^{III}_{r,r_s}$ and $\boldsymbol{\epsilon}^{III}_{r,r_s}$ in the entirely cracked stage (stage III) of the bond model is not well documented in [8]. Here a complete derivation of this stage will be given. The formulas used are partially taken from an Excel version of the bond model. The formulations will be given in a stepwise fashion similar to the programming done in Matlab. The parameters needed to calculate stage III are given in table C.1. The first step is to calculate the constant C_2 (see [8] equation (4.11)):

a_1, a_2, b_1, b_2, w_0	parameters bi-linear softening curve (constants)
f_{ct}	concrete tensile strength
ϵ_{cr}	crack strain ($\epsilon_{cr} = 1.2\frac{f_{ct}}{E}$)
n	number of fictitious radial cracks (constant)
r_s	bar radius
c_1	sum of effective cover and bar radius: $c_1 = c_{eff} + r_s$
n_{step}	discretization parameter (constant) controlling parameter C_3

Table C.1: Parameters stress stage III

$$C_2 = \frac{2\pi\epsilon_{cr}}{nw_0} \tag{C.1}$$

Stage III is guided by the parameter C_3 which is a parameter for the total crack width. Therefore C_3 is stepwise increased, the first (initial) value of C_3 is calculated as:

$$C_3^1 = c_1C_2 = (c_{eff} + r_s)\frac{2\pi\epsilon_{cr}}{nw_0} \tag{C.2}$$

Next a critical value for C_3 is defined:

$$C_3^{crit} = \frac{C_2(c_1 + r_s)}{2} - \frac{(b_2 - b_1)}{(a_2 - a_1)} \tag{C.3}$$

Now the values for $C_3^{2:nstep+1}$ are consecutively calculated from the previous value as:

$$C_3^{i+1} = C_3^i + \frac{(C_3^{crit} - C_2c_1)}{\text{fix}((C_3^{crit} - C_2c_1)/(1 - C_2c_1)n_{step})} \tag{C.4}$$

Fix is a Matlab command that rounds values towards zero creating integers (similar to the Excel command: TRUNC). Now $\boldsymbol{\sigma}^{III}_{r,r_s}$ is calculated with equation (4.25) from [8] for both parts of the bi-linear softening curve (i.e. using parameter set a_1, b_1 as well as parameter set a_2, b_2[1]:

$$\begin{aligned}\boldsymbol{\sigma}^{III,a_1b_1}_{r,r_s} &= \left((a_1\mathbf{C}_3 + b_1)\left(\frac{c_1}{r_s} - 1\right) - \left(a_1C_2\frac{r_s}{2}\right)\left(\left(\frac{c_1}{r_s}\right)^2 - 1\right)\right)f_{ct} \\ \boldsymbol{\sigma}^{III,a_2b_2}_{r,r_s} &= \left((a_2\mathbf{C}_3 + b_2)\left(\frac{c_1}{r_s} - 1\right) - \left(a_2C_2\frac{r_s}{2}\right)\left(\left(\frac{c_1}{r_s}\right)^2 - 1\right)\right)f_{ct}\end{aligned} \tag{C.5}$$

[1] note that bold font is used to designate vectors

Finally the radial stress at the interface $\boldsymbol{\sigma}_{r,r_s}^{III}$ is taken as the maximum value from both parameter sets:

$$\boldsymbol{\sigma}_{r,r_s}^{III} = \max\{\boldsymbol{\sigma}_{r,r_s}^{III,a_1 b_1}, \boldsymbol{\sigma}_{r,r_s}^{III,a_2 b_2}\} \tag{C.6}$$

Also the corresponding softening parameters either a_1 or a_2 and b_1 or b_2 are stored in vectors: $\mathbf{a}$ and $\mathbf{b}$. These will be used to calculate the strain $\boldsymbol{\epsilon}_{r,r_s}^{\Delta c}$. The rigid body movement part of the radial strain is calculated using equation (4.27) from [8]:

$$\boldsymbol{\epsilon}_{r,r_s}^{RBM} = \mathbf{C}_3 \frac{n w_0}{2\pi r_s} \tag{C.7}$$

Next the change in wall thickness (Δc) part of the radial strain is calculated using equation (4.28) from [8]:

$$\begin{aligned}\boldsymbol{\epsilon}_{r,r_s}^{\Delta c} = \epsilon_{cr} \left(\mathbf{a}\mathbf{C}_3 + \mathbf{b}\right) \left(\frac{c_1}{r_s} \ln\left(\frac{c_1}{r_s}\right) - \frac{c_1}{r_s} + 1\right) \dots \\ - \frac{\mathbf{a} C_2 \epsilon_{cr} r_s}{4} \left(2 \left(\frac{c_1}{r_s}\right)^2 \ln\left(\frac{c_1}{r_s}\right) - \left(\frac{c_1}{r_s}\right)^2 + 1\right)\end{aligned} \tag{C.8}$$

Finally the radial strain at the interface is calculated using equation (4.26) from [8]:

$$\boldsymbol{\epsilon}_{r,r_s}^{III} = \boldsymbol{\epsilon}_{r,r_s}^{RBM} + \boldsymbol{\epsilon}_{r,r_s}^{\Delta c} \tag{C.9}$$

Appendix D

Bond model: pull-out failure, derivation of radial strain - slip relations

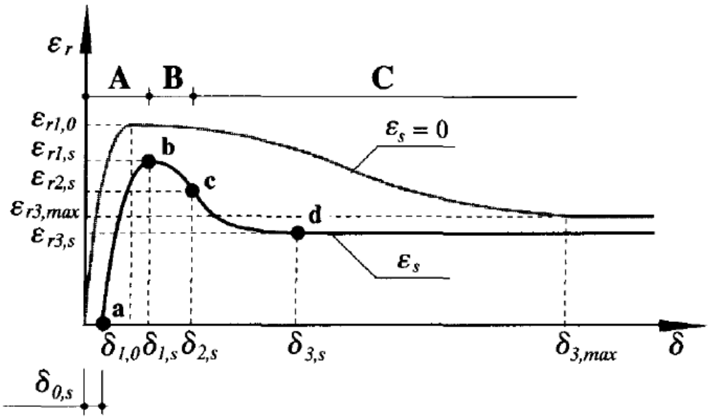


Figure D.1: Radial strain - slip relation for pull-out bond failure, taken form [8]

Used formulations for points a, b, c and d when ϵ_s=0 (boundary curve)

Point a has coordinates (0,0). Point b has coordinates ($\delta_{1,0}$,$\epsilon_{r1,0}$). $\epsilon_{r1,0}$ is the strain corresponding to the radial stress for which holds (see also section 2.4):

$$\sigma_{r,r_s} = \frac{\tau_{b1}}{cot(\phi)} \tag{D.1}$$

Where τ_{b1} is the critical bond stress, see [8] table 4.5. $\delta_{1,0}$ is given by equation (4.40) in [8]:

$$\delta_{1,0} = \frac{2\epsilon_{r1,0} r_s}{tan(\varphi)} \tag{D.2}$$

Point c has coordinates ($\delta_{2,0}$,$\epsilon_{r2,0}$). First the coordinates of point d ($\delta_{3,max}$,$\epsilon_{r3,max}$) are needed, $\delta_{3,max}$ is given in [8] table 4.5:

$$\delta_{3,max} = 0.33 d_s \tag{D.3}$$

$\epsilon_{r3,max}$ is the strain corresponding to the radial stress for which holds (see also section 2.4):

$$\sigma_{r,r_s} = \frac{\tau_{b3,max}}{cot(\phi)} \tag{D.4}$$

Where $\tau_{b3,max}$ is the maximum residual bond stress, see [8] table 4.5. Finally the coordinates of point c are determined by equations (4.36) and (4.37) from [8]:

$$\delta_{2,0} = \frac{\delta_{3,max}}{2} \tag{D.5}$$

$$\epsilon_{r2,0} = \frac{(\epsilon_{r3,max} + \epsilon_{r1,0})}{2} \tag{D.6}$$

Used formulations for points a, b, c and d when ϵ_s >0

$\delta_{0,s}$ is chosen to be zero, therefore point a has coordinates (0,0). Point b has coordinates ($\delta_{1,s}$,$\epsilon_{r1,s}$). $\delta_{1,s}$ is given by equation (4.34) in [8]:

$$\delta_{1,s} = \delta_{0,s} + \delta_{1,0} = \frac{2\epsilon_{r1,0} r_s}{tan(\varphi)} \tag{D.7}$$

To calculate $\epsilon_{r1,s}$ first ϵ_{r3} is needed:

$$\epsilon_{r3} = \epsilon_{r3,max} e^{-K(\epsilon_{r3})\epsilon_s} \tag{D.8}$$

Where $K(\epsilon_{r3})$ is a reduction constant, see [8] table 4.5. Now $\epsilon_{r1,s}$ is calculated with equation (4.35) from [8]:

$$\epsilon_{r1,s} = (\epsilon_{r1,0} - \epsilon_{r3}) e^{-K(\epsilon_{r1})\epsilon_s} + \epsilon_{r3} \tag{D.9}$$

Where $K(\epsilon_{r1})$ is a reduction constant, see [8] table 4.5. Now first the coordinates of point d are needed. Point d has coordinates ($\delta_{3,s}$,$\epsilon_{r3,s}$). To calculate $\delta_{3,s}$, first $\delta_{3,min}$ is needed:

$$\delta_{3,min} = 2.1\delta_{1,0} \tag{D.10}$$

Now $\delta_{3,s}$ is calculated with equation (4.38) from [8]:

$$\delta_{3,s} = (\delta_{3,max} - \delta_{3,min}) e^{-K(\delta_3)\epsilon_s} + \delta_{3,min} \tag{D.11}$$

Where $K(\delta_3)$ is a reduction constant, see [8] table 4.5. $\epsilon_{r3,s}$ is calculated with equation (4.39) from [8]:

$$\epsilon_{r3,s} = \epsilon_{r3,max} e^{-K(\epsilon_{r3})\epsilon_s} \tag{D.12}$$

Where $K(\epsilon_{r3})$ is a reduction constant, see [8] table 4.5. Finally the coordinates of point c ($\delta_{2,s}$,$\epsilon_{r2,s}$) are calculated using equations (4.36) and (4.37) from [8]:

$$\delta_{2,s} = \frac{\delta_{3,s}}{2} \tag{D.13}$$

$$\epsilon_{r2,s} = \frac{(\epsilon_{r1,s} + \epsilon_{r3,s})}{2} \tag{D.14}$$

Derivation op parabolic function between points a and b:

$$\epsilon_r(\delta) = a\delta^2 + b\delta + c \tag{D.15}$$

Boundary conditions:

$$P1 = (0;0) \tag{D.16}$$

$$P2 = (\delta_{1,0}; \epsilon_{r1,0}) \tag{D.17}$$

$$\frac{d\epsilon_r}{d\delta}|_{\delta=\delta_{1,0}} = 0 \tag{D.18}$$

From equation D.16 it follows:

$$c = 0 \tag{D.19}$$

From equation D.18 it follows:

$$\frac{d\epsilon_r}{d\delta}|_{\delta=\delta_{1,0}} = 2a\delta_{1,0} + b = 0 \Leftrightarrow b = -2a\delta_{1,0} \tag{D.20}$$

From equations D.17 and **??** it follows:

$$\epsilon_{r1,0} = a\delta_{1,0}^2 - 2a\delta_{1,0}^2 = -a\delta_{1,0}^2 \Leftrightarrow a = -\frac{\epsilon_{r1,0}}{\delta_{1,0}^2} \tag{D.21}$$

From equations D.20 and D.21 it follows:

$$b = \frac{2\epsilon_{r1,0}}{\delta_{1,0}} \tag{D.22}$$

Wanted parabolic function between points a and b:

$$\epsilon_r(\delta) = -(\frac{\epsilon_{r1,0}}{\delta_{1,0}^2})\delta^2 + 2(\frac{\epsilon_{r1,0}}{\delta_{1,0}})\delta \tag{D.23}$$

Derivation op parabolic function between points b and c:

$$\epsilon_r(\delta) = a\delta^2 + b\delta + c \tag{D.24}$$

Boundary conditions:

$$P1 = (\delta_{1,0}; \epsilon_{r1}) \tag{D.25}$$

$$P2 = (\delta_{2,0}; \epsilon_{r2,0}) \tag{D.26}$$

$$\frac{d\epsilon_r}{d\delta}|_{\delta=\delta_{1,0}} = 0 \tag{D.27}$$

From equation D.27 it follows:

$$\frac{d\epsilon_r}{d\delta}|_{\delta=\delta_{1,0}} = 2a\delta_{1,0} + b = 0 \Leftrightarrow b = -2a\delta_{1,0} \tag{D.28}$$

From equations D.25 and D.28 it follows:

$$\epsilon_{r1} = -a\delta_{1,0}^2 + c \tag{D.29}$$

From equations D.26 and D.28 it follows:

$$\epsilon_{r2} = a(\delta_2^2 - 2\delta_{1,0}\delta_2) + c \tag{D.30}$$

Subtracting equation D.30 from D.29 gives:

$$\epsilon_{r1} - \epsilon_{r2} = -a\delta_{1,0}^2 - a(\delta_2^2 - 2\delta_{1,0}\delta_2) \Leftrightarrow a = -\frac{(\epsilon_{r1} - \epsilon_{r2})}{(\delta_2^2 + \delta_{1,0}^2 - 2\delta_{1,0}\delta_2)} \tag{D.31}$$

From equations D.28 and D.31 it follows:

$$b = \frac{2\delta_{1,0}(\epsilon_{r1} - \epsilon_{r2})}{(\delta_2^2 + \delta_{1,0}^2 - 2\delta_{1,0}\delta_2)} \tag{D.32}$$

From equations D.29 and D.31 it follows:

$$c = \epsilon_{r1} - \delta_{1,0}^2 \frac{(\epsilon_{r1} - \epsilon_{r2})}{(\delta_2^2 + \delta_{1,0}^2 - 2\delta_{1,0}\delta_2)} \tag{D.33}$$

Wanted parabolic function between points b and c:

$$\begin{aligned} \epsilon_r(\delta) = -\frac{(\epsilon_{r1} - \epsilon_{r2})}{(\delta_2^2 + \delta_{1,0}^2 - 2\delta_{1,0}\delta_2)}\delta^2 + \frac{2\delta_{1,0}(\epsilon_{r1} - \epsilon_{r2})}{(\delta_2^2 + \delta_{1,0}^2 - 2\delta_{1,0}\delta_2)}\delta \ldots \\ + \epsilon_{r1} - \delta_{1,0}^2 \frac{(\epsilon_{r1} - \epsilon_{r2})}{(\delta_2^2 + \delta_{1,0}^2 - 2\delta_{1,0}\delta_2)} \end{aligned} \tag{D.34}$$

Derivation of exponential function between points c and d:[1]

$$\epsilon_r(\delta) = b\delta e^{-a\delta} + c \tag{D.35}$$

Boundary conditions:

$$P1 = (\delta_{2,0}; \epsilon_{r2}) \tag{D.36}$$

$$P2 = (\delta_{3max}; \epsilon_{r3max}) \tag{D.37}$$

$$\frac{d\epsilon_r}{d\delta}|_{\delta=\delta_{3max}} = 0 \tag{D.38}$$

From equation D.38 it follows:

$$\frac{d\epsilon_r}{d\delta}|_{\delta=\delta_{3max}} = be^{-a\delta_{3max}} - ab\delta_{3max}e^{-a\delta_{3max}} = 0$$

$$\Leftrightarrow b - ab\delta_{3max} = 0 \Leftrightarrow a = \frac{1}{\delta_{3max}} \tag{D.39}$$

From equations D.36 and D.39 it follows:

$$\epsilon_{r2} = b\delta_{2,0}e^{-\frac{\delta_{2,0}}{\delta_{3max}}} + c \tag{D.40}$$

From equations D.37 and D.39 it follows:

$$\epsilon_{r3max} = b\delta_{3max}e^{-\frac{\delta_{3max}}{\delta_{3max}}} + c = b\delta_{3max}e^{-1} + c \tag{D.41}$$

Subtracting equation D.40 from D.41 gives:

$$\epsilon_{r3max} - \epsilon_{r2} = -b\delta_{2,0}e^{-\frac{\delta_{2,0}}{\delta_{3max}}} + b\delta_{3max}e^{-1}$$

$$\Leftrightarrow b = \frac{(\epsilon_{r3max} - \epsilon_{r2})}{(\delta_{3max}e^{-1} - \delta_{2,0}e^{-\frac{\delta_{2,0}}{\delta_{3max}}})} \tag{D.42}$$

From equations D.40 and D.42 it follows:

$$c = \epsilon_{r2} - \frac{\delta_{2,0}e^{-\frac{\delta_{2,0}}{\delta_{3max}}}(\epsilon_{r3max} - \epsilon_{r2})}{(\delta_{3max}e^{-1} - \delta_{2,0}e^{-\frac{\delta_{2,0}}{\delta_{3max}}})} \tag{D.43}$$

Wanted exponential function between points c and d:

$$\epsilon_r(\delta) = \frac{(\epsilon_{r3max} - \epsilon_{r2})}{(\delta_{3max}e^{-1} - \delta_2 e^{-\frac{\delta_2}{\delta_{3max}}})}\delta e^{-\frac{\delta}{\delta_{3max}}} + \epsilon_{r2} - \frac{\delta_{2,0}e^{-\frac{\delta_{2,0}}{\delta_{3max}}}(\epsilon_{r3max} - \epsilon_{r2})}{(\delta_{3max}e^{-1} - \delta_{2,0}e^{-\frac{\delta_{2,0}}{\delta_{3max}}})} \tag{D.44}$$

[1] the use of $\epsilon_r(\delta) = be^{-a\delta} + c$ leads to equation $-abe^{-a\delta_{3max}} = 0$ when using boundary condition D.38 yielding no nonzero solution for a

Alternative derivation of exponential function between points c and d using equal slope at point c:

Note that this derivation did not yield an analytical solution for the unknowns a,b and c. Therefore it was not used in the Matlab bond model.

Using an equal slope at point c implies boundary condition D.38 now changes to:

$$\frac{d\epsilon_r}{d\delta}|_{\delta=\delta_{2,0}} = Q \tag{D.45}$$

With Q some constant of the slope at $\delta_{2,0}$ from the parabolic function between points b and c. In this case the function can be given as:

$$\epsilon_r(\delta) = be^{-a\delta} + c \tag{D.46}$$

since the new boundary condition now does not necessarily give a zero solution for a. Using all boundary conditions leads to equations:

$$\epsilon_{r2} = be^{-a\delta_{2,0}} + c \Leftrightarrow be^{-a\delta_{2,0}} = \epsilon_{r2} + c \tag{D.47}$$

$$\epsilon_{r3max} = be^{-a\delta_{3max}} + c \tag{D.48}$$

$$-abe^{-a\delta_{2,0}} = Q \tag{D.49}$$

Substituting equation D.47 into D.49 gives:

$$-a(\epsilon_{r2} + c) = Q \Leftrightarrow c = \frac{Q}{a} + \epsilon_{r2} \tag{D.50}$$

Subtracting D.47 from D.48 gives:

$$b(e^{-a\delta_{3max}} - e^{-a\delta_{2,0}}) = \epsilon_{r3max} - \epsilon_{r2} \Leftrightarrow b = \frac{(\epsilon_{r3max} - \epsilon_{r2})}{(e^{-a\delta_{3max}} - e^{-a\delta_{2,0}})} \tag{D.51}$$

Substituting equation D.50 and D.51 into D.47 gives:

$$\frac{(\epsilon_{r3max} - \epsilon_{r2})}{(e^{-a\delta_{3max}} - e^{-a\delta_{2,0}})}e^{-a\delta_{2,0}} + \frac{Q}{a} + \epsilon_{r2} = \epsilon_{r2} \Leftrightarrow \frac{(\epsilon_{r3max} - \epsilon_{r2})}{(e^{-a\delta_{3max}} - e^{-a\delta_{2,0}})}e^{-a\delta_{2,0}} + \frac{Q}{a} = 0 \tag{D.52}$$

This is were the derivation got stuck, I was only able to solve equation D.52 numerically for the unknown constant a using Maple.

Appendix E

Derivation of constitutive matrix for axi-symmetric elements in SLA

General 3D strain-stress relation:

$$\begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ 2\epsilon_{xy} \\ 2\epsilon_{yz} \\ 2\epsilon_{zx} \end{bmatrix} = \frac{1}{E} \begin{bmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu) \end{bmatrix} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{bmatrix} \tag{E.1}$$

For plane strain and axi-symmetry σ_{yz} and σ_{zx} are set equal to zero, therefore equation E.1 reduces to:

$$\begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ 2\epsilon_{xy} \end{bmatrix} = \frac{1}{E} \begin{bmatrix} 1 & -\nu & -\nu & 0 \\ -\nu & 1 & -\nu & 0 \\ -\nu & -\nu & 1 & 0 \\ 0 & 0 & 0 & 2(1+\nu) \end{bmatrix} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \end{bmatrix} \tag{E.2}$$

In the next derivations two indices are used. The first is related to the stress direction which is being evaluated, the second is the source stress direction. Since it is fully uncoupled the shear term is ignored.

Only uniaxial stress σ_{xx}:

$$\epsilon_{xx} = \tfrac{\sigma_{xx}}{E_x} \quad \epsilon_{yy} = -\nu_{yx}\tfrac{\sigma_{xx}}{E_x} \quad \epsilon_{zz} = -\nu_{zx}\tfrac{\sigma_{xx}}{E_x} \tag{E.3}$$

Only uniaxial stress σ_{yy}:

$$\epsilon_{xx} = -\nu_{xy}\tfrac{\sigma_{yy}}{E_y} \quad \epsilon_{yy} = \tfrac{\sigma_{yy}}{E_y} \quad \epsilon_{zz} = -\nu_{zy}\tfrac{\sigma_{yy}}{E_y} \tag{E.4}$$

Only uniaxial stress σ_{zz}:

$$\epsilon_{xx} = -\nu_{xz}\tfrac{\sigma_{zz}}{E_z} \quad \epsilon_{yy} = -\nu_{yz}\tfrac{\sigma_{zz}}{E_z} \quad \epsilon_{zz} = \tfrac{\sigma_{zz}}{E_z} \tag{E.5}$$

Using the principle of superposition combining σ_{xx}, σ_{yy} and σ_{zz} gives:

$$\begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_x} & \frac{-\nu_{xy}}{E_y} & \frac{-\nu_{xz}}{E_z} & 0 \\ \frac{-\nu_{yx}}{E_x} & \frac{1}{E_y} & \frac{-\nu_{yz}}{E_z} & 0 \\ \frac{-\nu_{zx}}{E_x} & \frac{-\nu_{zy}}{E_y} & \frac{1}{E_z} & 0 \\ 0 & 0 & 0 & \frac{2(1+\nu)}{E} \end{bmatrix} \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \end{bmatrix} \tag{E.6}$$

Note that $2\epsilon_{ij} = \gamma_{ij}$. Inverting this matrix (using Maple) gives the stress-strain relation:

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \end{bmatrix} = \mathbf{D} \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{xy} \end{bmatrix} \tag{E.7}$$

With $\mathbf{D}$ denoted as:

$$F\begin{bmatrix} (\nu_{yz}\nu_{zy}-1)E_x & -(\nu_{xz}\nu_{zy}+\nu_{xy})E_x & -(\nu_{xy}\nu_{yz}+\nu_{xz})E_x & 0 \\ -(\nu_{zx}\nu_{yz}+\nu_{yx})E_y & (\nu_{zx}\nu_{xz}-1)E_y & -(\nu_{yx}\nu_{xz}+\nu_{yz})E_y & 0 \\ -(\nu_{yx}\nu_{zy}+\nu_{zx})E_z & -(\nu_{zx}\nu_{xy}+\nu_{zy})E_z & (\nu_{yx}\nu_{xy}-1)E_z & 0 \\ 0 & 0 & 0 & \frac{G}{F} \end{bmatrix} \tag{E.8}$$

In which:

$$G=\frac{E}{2(1+\nu)} \tag{E.9}$$

and:

$$F=\frac{1}{(\nu_{zx}\nu_{xy}\nu_{yz}+\nu_{yx}\nu_{xz}\nu_{zy}+\nu_{zx}\nu_{xz}+\nu_{yx}\nu_{xy}+\nu_{yz}\nu_{zy}-1)} \tag{E.10}$$

For a damaged element the axis in the xy-plane is rotated to a local nt-plane. The z-axis remains unchanged (i.e. fixed). Changing notations to a local damaged axis-system (i.e. ntz-axis) gives the wanted constitutive matrix for an axi-symmetric element in SLA:

$$\begin{bmatrix} \sigma_{nn} \\ \sigma_{tt} \\ \sigma_{zz} \\ \sigma_{nt} \end{bmatrix} = \mathbf{D} \begin{bmatrix} \epsilon_{nn} \\ \epsilon_{tt} \\ \epsilon_{zz} \\ \gamma_{nt} \end{bmatrix} \tag{E.11}$$

With $\mathbf{D}$ denoted as:

$$F\begin{bmatrix} (\nu_{tz}\nu_{zt}-1)E_n & -(\nu_{nz}\nu_{zt}+\nu_{nt})E_n & -(\nu_{nt}\nu_{tz}+\nu_{nz})E_n & 0 \\ -(\nu_{zn}\nu_{tz}+\nu_{tn})E_t & (\nu_{zn}\nu_{nz}-1)E_t & -(\nu_{tn}\nu_{nz}+\nu_{tz})E_t & 0 \\ -(\nu_{tn}\nu_{zt}+\nu_{zn})E_z & -(\nu_{zn}\nu_{nt}+\nu_{zt})E_z & (\nu_{tn}\nu_{nt}-1)E_z & 0 \\ 0 & 0 & 0 & \frac{G}{F} \end{bmatrix} \tag{E.12}$$

In which:

$$G=\frac{E}{2(1+\nu)} \tag{E.13}$$

and:

$$F=\frac{1}{(\nu_{zn}\nu_{nt}\nu_{tz}+\nu_{tn}\nu_{nz}\nu_{zt}+\nu_{zn}\nu_{nz}+\nu_{tn}\nu_{nt}+\nu_{tz}\nu_{zt}-1)} \tag{E.14}$$

For shear reduction options see section 3.3.

Appendix F

Verification of SLA code for axi-symmetric elements

The test model (figure 1) consists of one square linear Q8AXI element (sides of 40 mm) with four point Gauss integration (note that the one point integration scheme does not work in the SLA environment). The element will be tested with different loading schemes (with ν=0 or ν=0.2). For axi-symmetric elements the axis of rotation is always the y-axis. Only proportional loading will be tested.

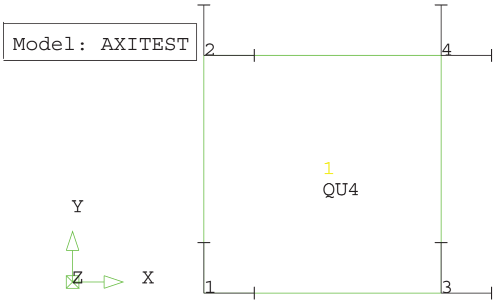


Figure F.1: Mesh for test axi-symmetric element

Loading schemes:

1. All four nodes are given a uniform translation of unity in x-direction. This will cause mainly circumferential stresses σ_{zz}.
2. Nodes 2 and 4 are given a uniform translation of unity in y-direction. Nodes 1 and 3 are constrained in y-direction and nodes 1 and 2 are constrained in x-direction. This will cause mainly stresses in the xy-plane.
3. Nodes 2 and 4 are given a translation of unity in y-direction, nodes 3 and 4 are given a translation of 0.5 in x-direction.

Because the element is linear all stresses in the xy-plane are constant, whereas the stresses σ_{zz} depend on the distance from the axis of rotation ($\epsilon_{zz} = \frac{u_x}{r}$). Integration points 1 and 3 are located nearest to the axis of rotation. The distance ratio r for the two rows of integration points is approximately $r_{2,4}$ / $r_{1,3}$ = 3.734, this ratio can be used to check the stress ratios. The cracking will be initiated with a very small hardia table consisting of just 2 teeth:

2.800000E+04 2.5
2.400000E+04 2.3

For each loading scheme and Poisson's ratio the following things have been checked:

- is the corrent critical load multiplier identified from the stresses
- is the principal stress calculated correctly
- is the constitutive matrix updated correctly
- does the program stop correctly (e.g. when no more damage is possible)

Appendix G

Overview of routines for SLA software extensions

Table G.1: Overview of SLA routines for axi-symmetric elements ($^{(mod)}$: modified original DIANA file; $^{(modex)}$: modified existing SLA file; $^{(new)}$: newly created SLA file)

routine	
DLPEAX$^{(mod)}$	SLA element initialization, store parameters at IP level.
MATAXI$^{(mod)}$	Prevent initially orthotropic materials to be used with SLA.
LAMBIB$^{(modex)}$	Redirect axi-symmetric SLA elements to routine LAMAXI.
LAMAXI$^{(new)}$	Calculate admissible load factor (λ).
DAMAXI$^{(new)}$	Update damage indicator for critical integration point.
MATAX2$^{(new)}$	Get all relevant material model parameters and calculate new stress-strain relation matrix for critical integration point using routine SEAXI2.
SEAXI2$^{(new)}$	Calculate stress-strain relation matrix in local crack axis system (NTZ) and rotate back to general axis system.

Table G.2: Overview of SLA routines for interface elements ($^{(modex)}$: modified existing SLA file; $^{(new)}$: newly created SLA file)

routine	
LAMBIB$^{(modex)}$	Redirect interface SLA elements to routine LAMLIE.
LAMLIE$^{(new)}$	Calculate admissible load factor (λ).
DAMLIE$^{(new)}$	Update damage indicator for critical integration point.
MATIF2$^{(modex)}$	Calculate bond stress-slip relation matrix.

Table G.3: Overview of SLA routines for cable (truss) elements ($^{(mod)}$: modified original DIANA file)

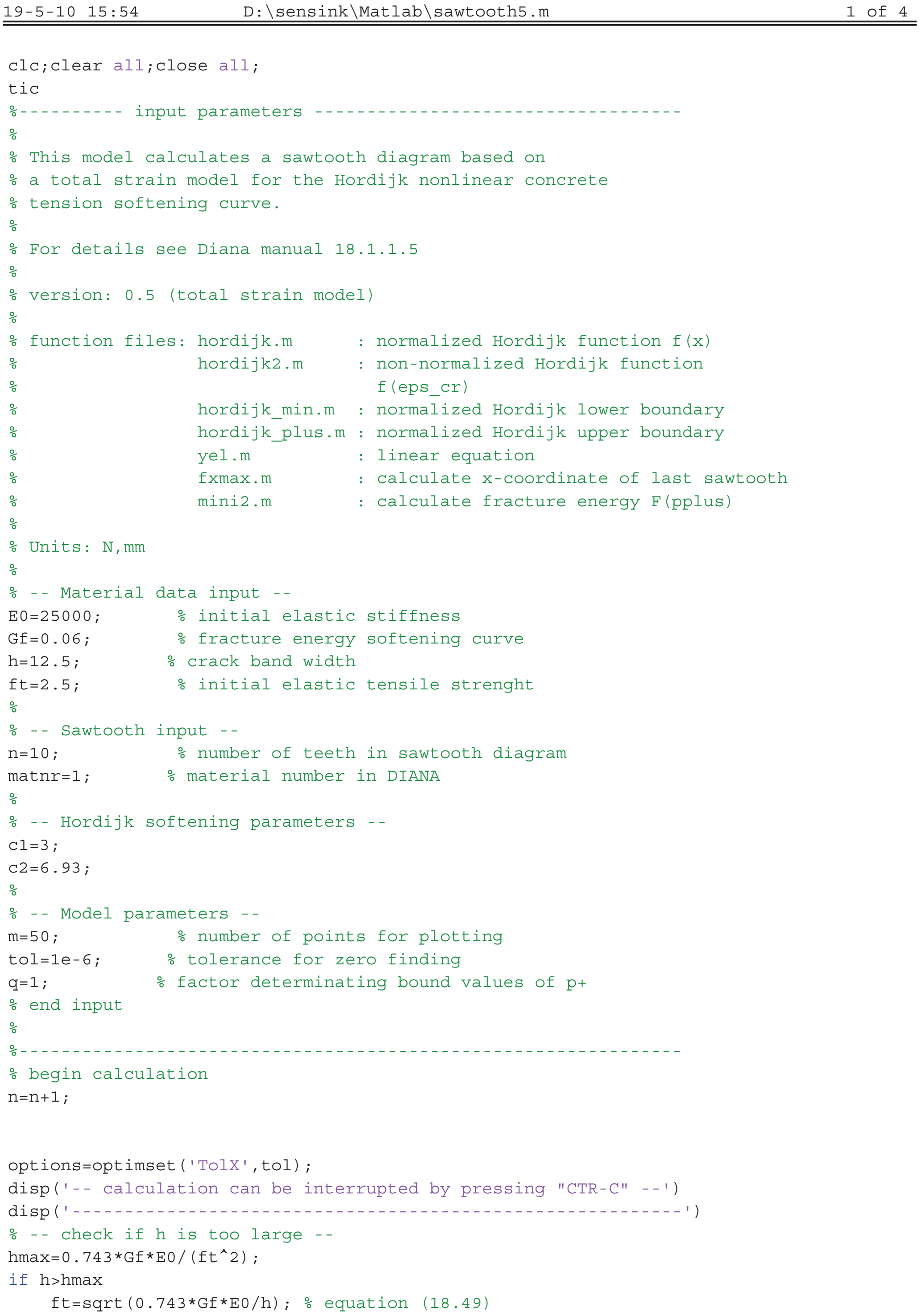
routine	
ISCLTR$^{(mod)}$	Write constitutive matrix to integration point level in case of SLA.

Appendix H

Matlab sawtooth generator for Hordijk tension softening: code

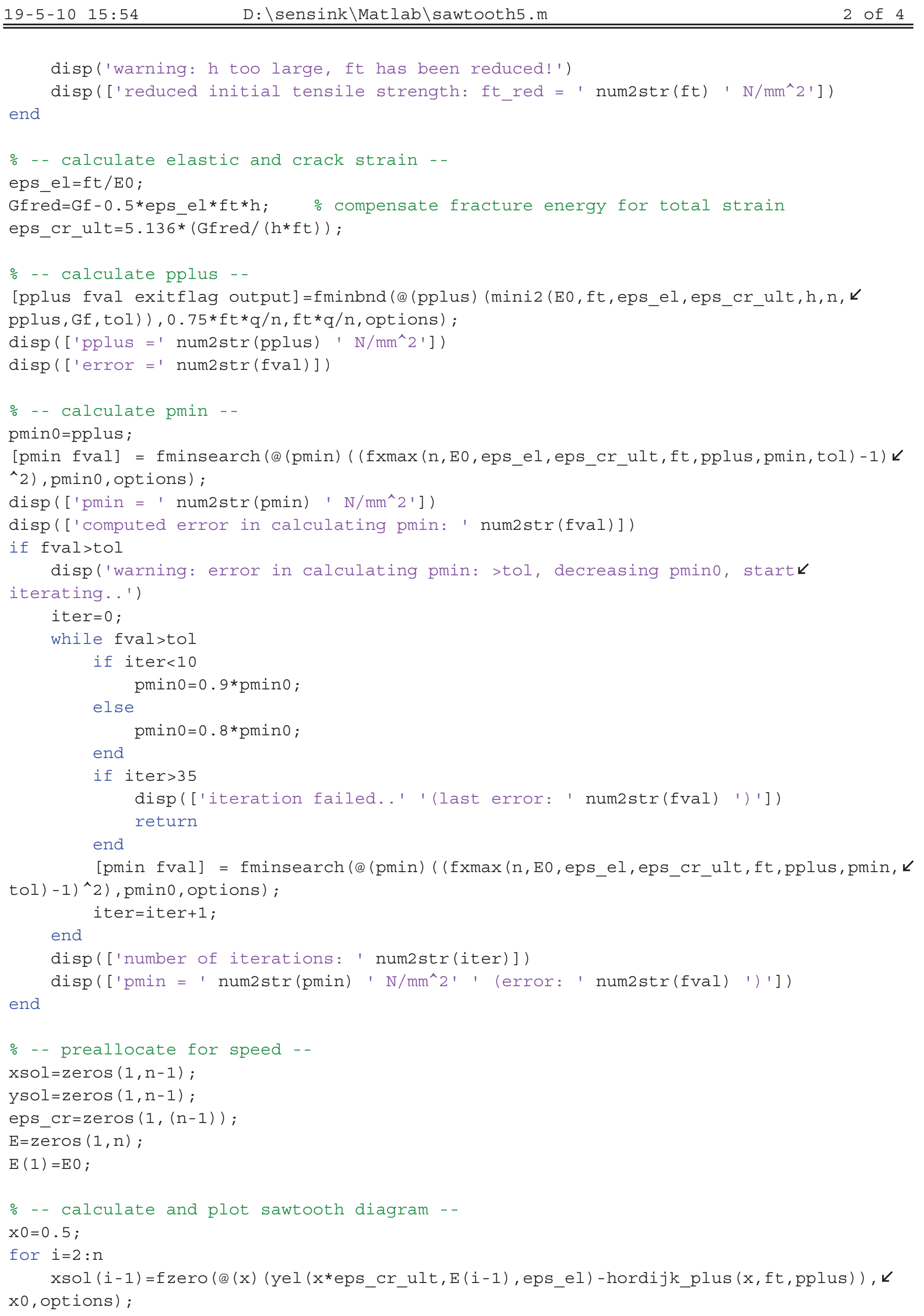
Note: the Matlab sawtooth generator is not full proof, i.e. it will not give a good solution for all possible combinations of parameters automatically. This depends on the user defined parameter (q) that controls the solution interval for p+. See also section 4.2.2 equations 4.8 and 4.9.

Main program:

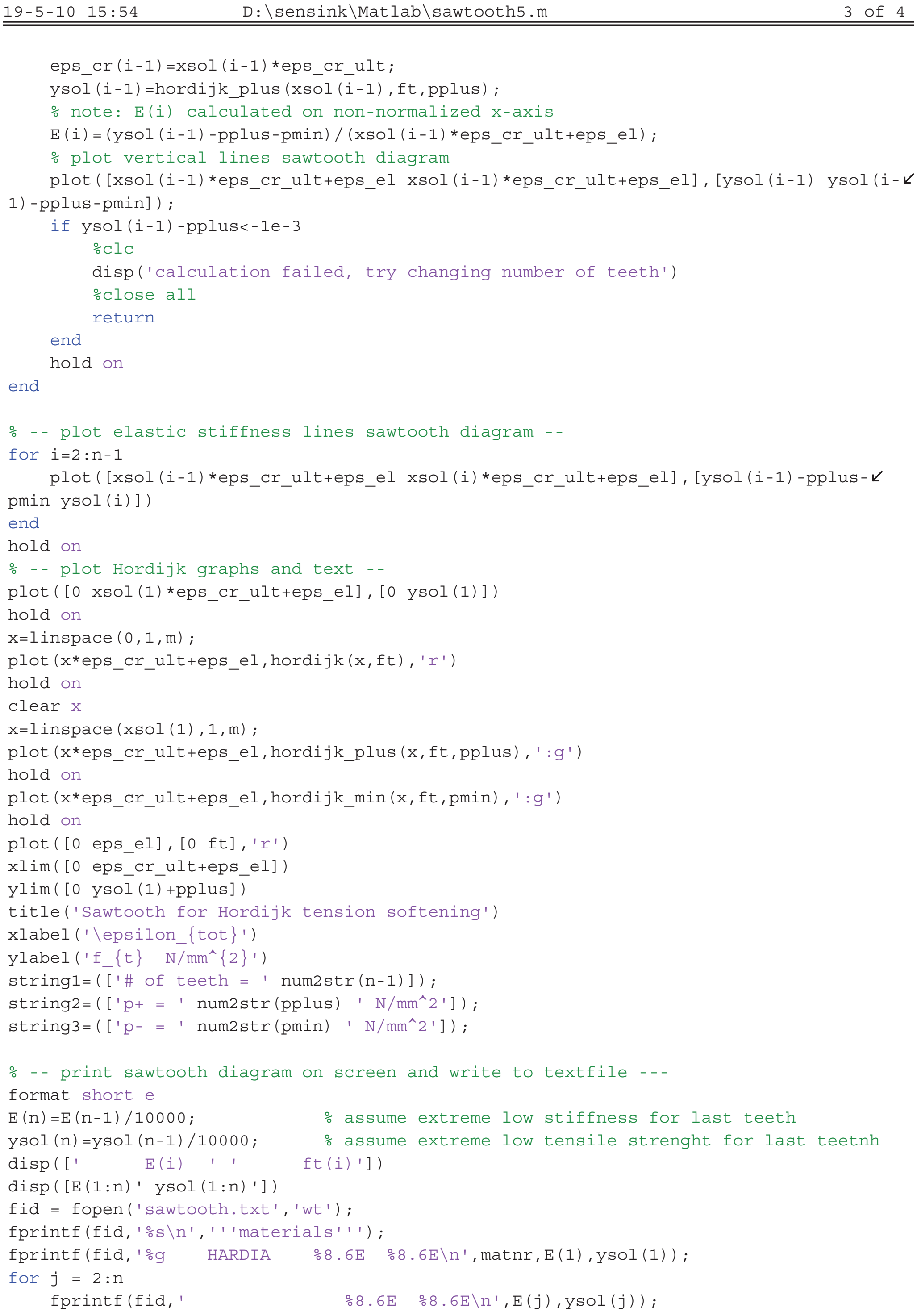


19-5-10 15:54 D:\sensink\Matlab\sawtooth5.m 1 of 4

```
clc;clear all;close all;
tic
%---------- input parameters -----------------------------------
%
% This model calculates a sawtooth diagram based on
% a total strain model for the Hordijk nonlinear concrete
% tension softening curve.
%
% For details see Diana manual 18.1.1.5
%
% version: 0.5 (total strain model)
%
% function files: hordijk.m      : normalized Hordijk function f(x)
%                 hordijk2.m     : non-normalized Hordijk function
%                                  f(eps_cr)
%                 hordijk_min.m  : normalized Hordijk lower boundary
%                 hordijk_plus.m : normalized Hordijk upper boundary
%                 yel.m          : linear equation
%                 fxmax.m        : calculate x-coordinate of last sawtooth
%                 mini2.m        : calculate fracture energy F(pplus)
%
% Units: N,mm
%
% -- Material data input --
E0=25000;       % initial elastic stiffness
Gf=0.06;        % fracture energy softening curve
h=12.5;        % crack band width
ft=2.5;         % initial elastic tensile strenght
%
% -- Sawtooth input --
n=10;           % number of teeth in sawtooth diagram
matnr=1;       % material number in DIANA
%
% -- Hordijk softening parameters --
c1=3;
c2=6.93;
%
% -- Model parameters --
m=50;           % number of points for plotting
tol=1e-6;      % tolerance for zero finding
q=1;          % factor determinating bound values of p+
% end input
%
%---------------------------------------------------------------
% begin calculation
n=n+1;


options=optimset('TolX',tol);
disp('-- calculation can be interrupted by pressing "CTR-C" --')
disp('---------------------------------------------------------')
% -- check if h is too large --
hmax=0.743*Gf*E0/(ft^2);
if h>hmax
    ft=sqrt(0.743*Gf*E0/h); % equation (18.49)
```

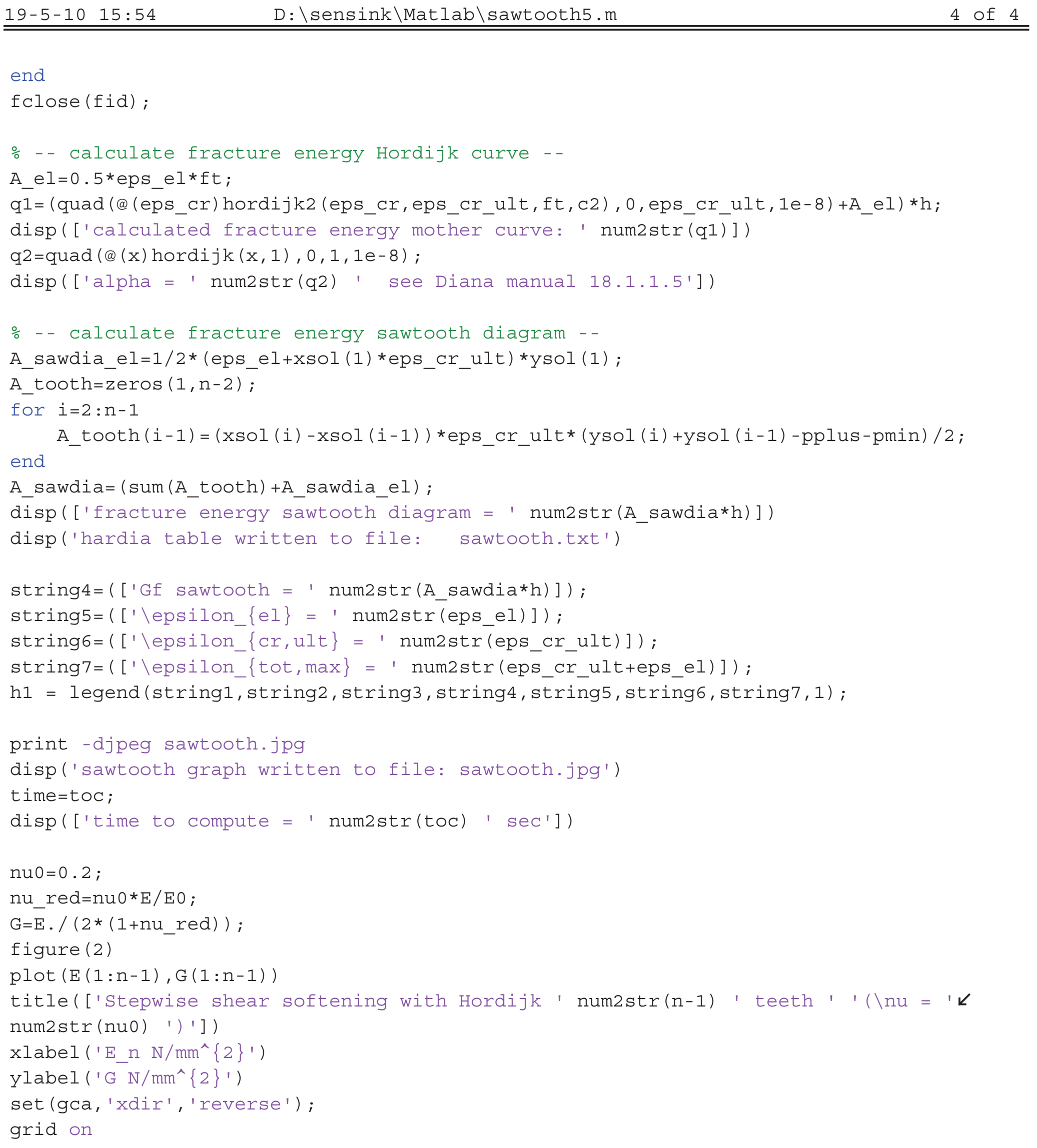


```
    disp('warning: h too large, ft has been reduced!')
    disp(['reduced initial tensile strength: ft_red = ' num2str(ft) ' N/mm^2'])
end

% -- calculate elastic and crack strain --
eps_el=ft/E0;
Gfred=Gf-0.5*eps_el*ft*h;    % compensate fracture energy for total strain
eps_cr_ult=5.136*(Gfred/(h*ft));

% -- calculate pplus --
[pplus fval exitflag output]=fminbnd(@(pplus)(mini2(E0,ft,eps_el,eps_cr_ult,h,n,↙
pplus,Gf,tol)),0.75*ft*q/n,ft*q/n,options);
disp(['pplus =' num2str(pplus) ' N/mm^2'])
disp(['error =' num2str(fval)])

% -- calculate pmin --
pmin0=pplus;
[pmin fval] = fminsearch(@(pmin)((fxmax(n,E0,eps_el,eps_cr_ult,ft,pplus,pmin,tol)-1)↙
^2),pmin0,options);
disp(['pmin = ' num2str(pmin) ' N/mm^2'])
disp(['computed error in calculating pmin: ' num2str(fval)])
if fval>tol
    disp('warning: error in calculating pmin: >tol, decreasing pmin0, start↙
iterating..')
    iter=0;
    while fval>tol
        if iter<10
            pmin0=0.9*pmin0;
        else
            pmin0=0.8*pmin0;
        end
        if iter>35
            disp(['iteration failed..' '(last error: ' num2str(fval) ')'])
            return
        end
        [pmin fval] = fminsearch(@(pmin)((fxmax(n,E0,eps_el,eps_cr_ult,ft,pplus,pmin,↙
tol)-1)^2),pmin0,options);
        iter=iter+1;
    end
    disp(['number of iterations: ' num2str(iter)])
    disp(['pmin = ' num2str(pmin) ' N/mm^2' ' (error: ' num2str(fval) ')'])
end

% -- preallocate for speed --
xsol=zeros(1,n-1);
ysol=zeros(1,n-1);
eps_cr=zeros(1,(n-1));
E=zeros(1,n);
E(1)=E0;

% -- calculate and plot sawtooth diagram --
x0=0.5;
for i=2:n
    xsol(i-1)=fzero(@(x)(yel(x*eps_cr_ult,E(i-1),eps_el)-hordijk_plus(x,ft,pplus)),↙
x0,options);
```



19-5-10 15:54   D:\sensink\Matlab\sawtooth5.m   3 of 4

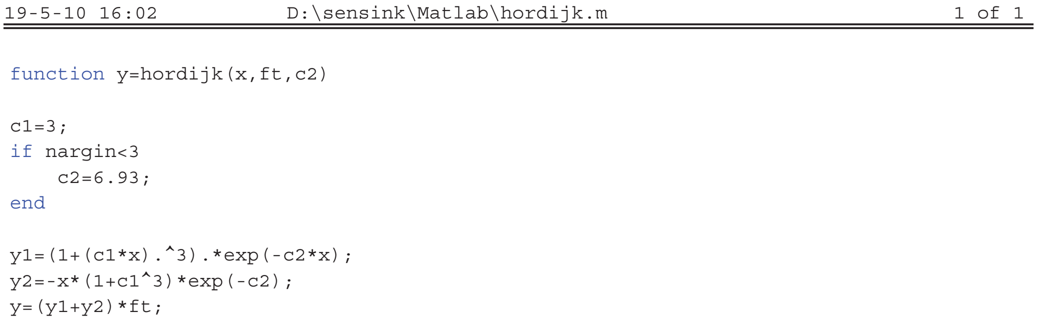
```
    eps_cr(i-1)=xsol(i-1)*eps_cr_ult;
    ysol(i-1)=hordijk_plus(xsol(i-1),ft,pplus);
    % note: E(i) calculated on non-normalized x-axis
    E(i)=(ysol(i-1)-pplus-pmin)/(xsol(i-1)*eps_cr_ult+eps_el);
    % plot vertical lines sawtooth diagram
    plot([xsol(i-1)*eps_cr_ult+eps_el xsol(i-1)*eps_cr_ult+eps_el],[ysol(i-1) ysol(i-↙
1)-pplus-pmin]);
    if ysol(i-1)-pplus<-1e-3
        %clc
        disp('calculation failed, try changing number of teeth')
        %close all
        return
    end
    hold on
end

% -- plot elastic stiffness lines sawtooth diagram --
for i=2:n-1
    plot([xsol(i-1)*eps_cr_ult+eps_el xsol(i)*eps_cr_ult+eps_el],[ysol(i-1)-pplus-↙
pmin ysol(i)])
end
hold on
% -- plot Hordijk graphs and text --
plot([0 xsol(1)*eps_cr_ult+eps_el],[0 ysol(1)])
hold on
x=linspace(0,1,m);
plot(x*eps_cr_ult+eps_el,hordijk(x,ft),'r')
hold on
clear x
x=linspace(xsol(1),1,m);
plot(x*eps_cr_ult+eps_el,hordijk_plus(x,ft,pplus),':g')
hold on
plot(x*eps_cr_ult+eps_el,hordijk_min(x,ft,pmin),':g')
hold on
plot([0 eps_el],[0 ft],'r')
xlim([0 eps_cr_ult+eps_el])
ylim([0 ysol(1)+pplus])
title('Sawtooth for Hordijk tension softening')
xlabel('\epsilon_{tot}')
ylabel('f_{t}  N/mm^{2}')
string1=(['# of teeth = ' num2str(n-1)]);
string2=(['p+ = ' num2str(pplus) ' N/mm^2']);
string3=(['p- = ' num2str(pmin) ' N/mm^2']);

% -- print sawtooth diagram on screen and write to textfile ---
format short e
E(n)=E(n-1)/10000;            % assume extreme low stiffness for last teeth
ysol(n)=ysol(n-1)/10000;      % assume extreme low tensile strenght for last teetnh
disp(['      E(i)  ' '      ft(i)'])
disp([E(1:n)' ysol(1:n)'])
fid = fopen('sawtooth.txt','wt');
fprintf(fid,'%s\n','''materials''');
fprintf(fid,'%g    HARDIA    %8.6E  %8.6E\n',matnr,E(1),ysol(1));
for j = 2:n
    fprintf(fid,'               %8.6E  %8.6E\n',E(j),ysol(j));
```



```
19-5-10 15:54          D:\sensink\Matlab\sawtooth5.m                    4 of 4
```

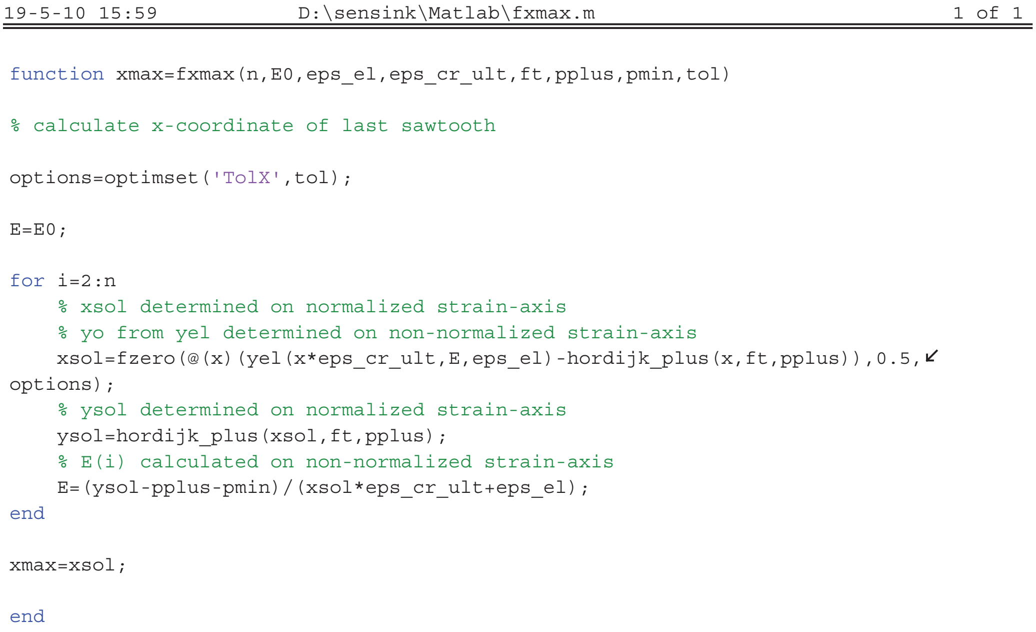
```
end
fclose(fid);

% -- calculate fracture energy Hordijk curve --
A_el=0.5*eps_el*ft;
q1=(quad(@(eps_cr)hordijk2(eps_cr,eps_cr_ult,ft,c2),0,eps_cr_ult,1e-8)+A_el)*h;
disp(['calculated fracture energy mother curve: ' num2str(q1)])
q2=quad(@(x)hordijk(x,1),0,1,1e-8);
disp(['alpha = ' num2str(q2) '  see Diana manual 18.1.1.5'])

% -- calculate fracture energy sawtooth diagram --
A_sawdia_el=1/2*(eps_el+xsol(1)*eps_cr_ult)*ysol(1);
A_tooth=zeros(1,n-2);
for i=2:n-1
    A_tooth(i-1)=(xsol(i)-xsol(i-1))*eps_cr_ult*(ysol(i)+ysol(i-1)-pplus-pmin)/2;
end
A_sawdia=(sum(A_tooth)+A_sawdia_el);
disp(['fracture energy sawtooth diagram = ' num2str(A_sawdia*h)])
disp('hardia table written to file:   sawtooth.txt')

string4=(['Gf sawtooth = ' num2str(A_sawdia*h)]);
string5=(['\epsilon_{el} = ' num2str(eps_el)]);
string6=(['\epsilon_{cr,ult} = ' num2str(eps_cr_ult)]);
string7=(['\epsilon_{tot,max} = ' num2str(eps_cr_ult+eps_el)]);
h1 = legend(string1,string2,string3,string4,string5,string6,string7,1);

print -djpeg sawtooth.jpg
disp('sawtooth graph written to file: sawtooth.jpg')
time=toc;
disp(['time to compute = ' num2str(toc) ' sec'])

nu0=0.2;
nu_red=nu0*E/E0;
G=E./(2*(1+nu_red));
figure(2)
plot(E(1:n-1),G(1:n-1))
title(['Stepwise shear softening with Hordijk ' num2str(n-1) ' teeth ' '(\nu = '↙
num2str(nu0) ')'])
xlabel('E_n N/mm^{2}')
ylabel('G N/mm^{2}')
set(gca,'xdir','reverse');
grid on
```

Function files:

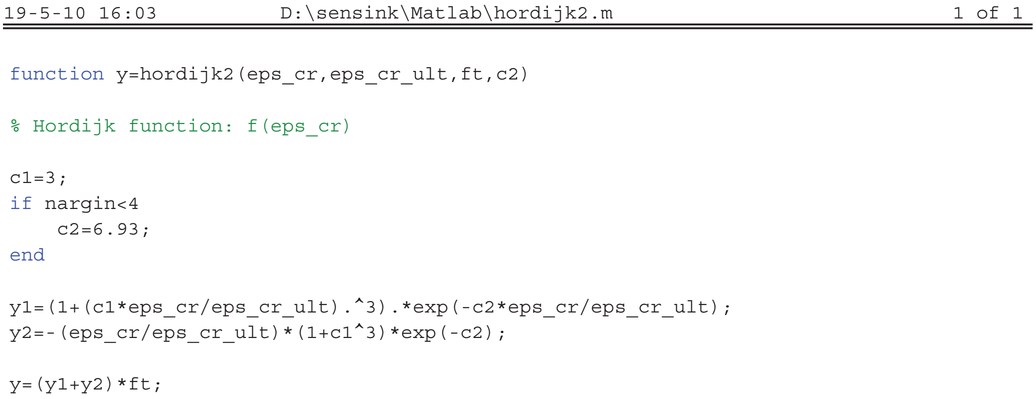


```
19-5-10 16:02          D:\sensink\Matlab\hordijk.m                      1 of 1
```

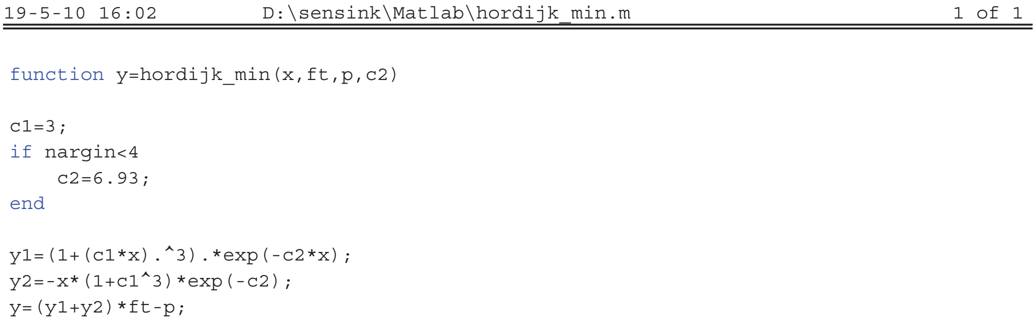
```
function y=hordijk(x,ft,c2)

c1=3;
if nargin<3
    c2=6.93;
end

y1=(1+(c1*x).^3).*exp(-c2*x);
y2=-x*(1+c1^3)*exp(-c2);
y=(y1+y2)*ft;
```



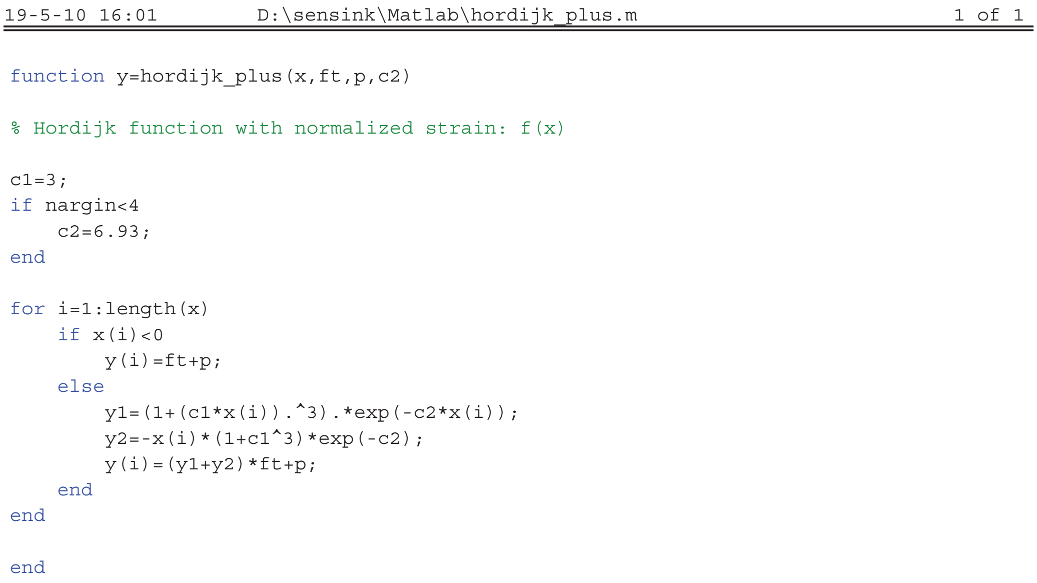
```
19-5-10 15:59          D:\sensink\Matlab\fxmax.m                        1 of 1

function xmax=fxmax(n,E0,eps_el,eps_cr_ult,ft,pplus,pmin,tol)

% calculate x-coordinate of last sawtooth

options=optimset('TolX',tol);

E=E0;

for i=2:n
    % xsol determined on normalized strain-axis
    % yo from yel determined on non-normalized strain-axis
    xsol=fzero(@(x)(yel(x*eps_cr_ult,E,eps_el)-hordijk_plus(x,ft,pplus)),0.5,↙
options);
    % ysol determined on normalized strain-axis
    ysol=hordijk_plus(xsol,ft,pplus);
    % E(i) calculated on non-normalized strain-axis
    E=(ysol-pplus-pmin)/(xsol*eps_cr_ult+eps_el);
end

xmax=xsol;

end
```



```
19-5-10 16:03          D:\sensink\Matlab\hordijk2.m                     1 of 1

function y=hordijk2(eps_cr,eps_cr_ult,ft,c2)

% Hordijk function: f(eps_cr)

c1=3;
if nargin<4
    c2=6.93;
end

y1=(1+(c1*eps_cr/eps_cr_ult).^3).*exp(-c2*eps_cr/eps_cr_ult);
y2=-(eps_cr/eps_cr_ult)*(1+c1^3)*exp(-c2);

y=(y1+y2)*ft;
```

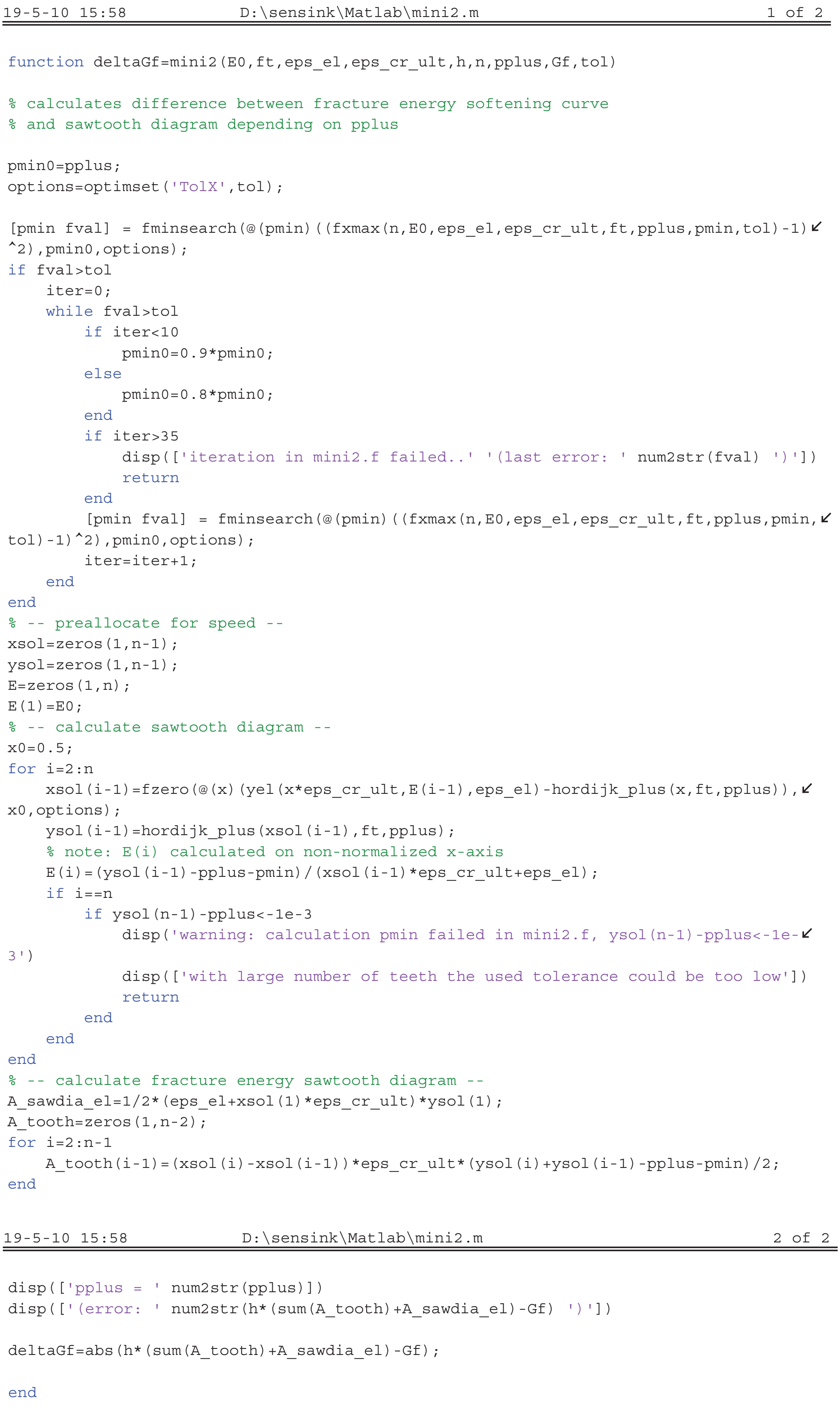


```
19-5-10 16:02          D:\sensink\Matlab\hordijk_min.m                  1 of 1

function y=hordijk_min(x,ft,p,c2)

c1=3;
if nargin<4
    c2=6.93;
end

y1=(1+(c1*x).^3).*exp(-c2*x);
y2=-x*(1+c1^3)*exp(-c2);
y=(y1+y2)*ft-p;
```



19-5-10 16:01 D:\sensink\Matlab\hordijk_plus.m 1 of 1

```
function y=hordijk_plus(x,ft,p,c2)

% Hordijk function with normalized strain: f(x)

c1=3;
if nargin<4
    c2=6.93;
end

for i=1:length(x)
    if x(i)<0
        y(i)=ft+p;
    else
        y1=(1+(c1*x(i)).^3).*exp(-c2*x(i));
        y2=-x(i)*(1+c1^3)*exp(-c2);
        y(i)=(y1+y2)*ft+p;
    end
end

end
```



```
19-5-10 15:58            D:\sensink\Matlab\mini2.m                    1 of 2

function deltaGf=mini2(E0,ft,eps_el,eps_cr_ult,h,n,pplus,Gf,tol)

% calculates difference between fracture energy softening curve
% and sawtooth diagram depending on pplus

pmin0=pplus;
options=optimset('TolX',tol);

[pmin fval] = fminsearch(@(pmin)((fxmax(n,E0,eps_el,eps_cr_ult,ft,pplus,pmin,tol)-1)↙
^2),pmin0,options);
if fval>tol
    iter=0;
    while fval>tol
        if iter<10
            pmin0=0.9*pmin0;
        else
            pmin0=0.8*pmin0;
        end
        if iter>35
            disp(['iteration in mini2.f failed..' '(last error: ' num2str(fval) ')'])
            return
        end
        [pmin fval] = fminsearch(@(pmin)((fxmax(n,E0,eps_el,eps_cr_ult,ft,pplus,pmin,↙
tol)-1)^2),pmin0,options);
        iter=iter+1;
    end
end
% -- preallocate for speed --
xsol=zeros(1,n-1);
ysol=zeros(1,n-1);
E=zeros(1,n);
E(1)=E0;
% -- calculate sawtooth diagram --
x0=0.5;
for i=2:n
    xsol(i-1)=fzero(@(x)(yel(x*eps_cr_ult,E(i-1),eps_el)-hordijk_plus(x,ft,pplus)),↙
x0,options);
    ysol(i-1)=hordijk_plus(xsol(i-1),ft,pplus);
    % note: E(i) calculated on non-normalized x-axis
    E(i)=(ysol(i-1)-pplus-pmin)/(xsol(i-1)*eps_cr_ult+eps_el);
    if i==n
        if ysol(n-1)-pplus<-1e-3
            disp('warning: calculation pmin failed in mini2.f, ysol(n-1)-pplus<-1e-↙
3')

            disp(['with large number of teeth the used tolerance could be too low'])
            return
        end
    end
end
% -- calculate fracture energy sawtooth diagram --
A_sawdia_el=1/2*(eps_el+xsol(1)*eps_cr_ult)*ysol(1);
A_tooth=zeros(1,n-2);
for i=2:n-1
    A_tooth(i-1)=(xsol(i)-xsol(i-1))*eps_cr_ult*(ysol(i)+ysol(i-1)-pplus-pmin)/2;
end

19-5-10 15:58            D:\sensink\Matlab\mini2.m                    2 of 2

disp(['pplus = ' num2str(pplus)])
disp(['(error: ' num2str(h*(sum(A_tooth)+A_sawdia_el)-Gf) ')'])

deltaGf=abs(h*(sum(A_tooth)+A_sawdia_el)-Gf);

end
```

Appendix I

Nonlinear axi-symmetric calculation: *DIANA* program settings

Used element types and integration scheme:

CL6TR Truss element (default integration)
CL12I Interface element (5-point lumped integration)
CQ16A Axi-symmetric element (3x3 Gauss integration)

Other *DIANA* program settings used:

Interface elements in plane stress configuration (MEMBRA)

Non-linear settings:

Energy based convergence criterion (default tolerance of 0.0001)
Maximum number of iterations: 40

Appendix J

DIANA User interface models *BOND3N*, *BOND3S* and *BOND3E* code and verification

File: /home1/sensink/usrifc.f

```
      SUBROUTINE USRIFC(U0, DU, NT, AGE0, DTIME, TEMP0,
     $                  DTEMP, ELEMEN, INTPT, COORD, SE, ITER,
     $                  USRMOD, USRVAL, NUV, USRSTA, NUS,
     $                  USRIND, NUI, TRA, STIFF)
C.......................................Copyright (c) 2009-2010 TU DELFT
C...
C...   PURPOSE:
C...     USER-SUPPLIED ELASTO-PLASTIC (TANGENTIAL DIRECTION) AND
C...     ELASTIC (NORMAL DIRECTION) INTERFACE MODELS WITH VARIOUS
C...     UNLOADING BEHAVIOR.
C...
C...     MODELS AND UNLOADING TYPE:
C...     BOND3N   NONLINEAR UNLOADING
C...     BOND3S   SECANT UNLOADING
C...     BOND3E   ELASTIC UNLOADING
C...
C...   ARGUMENTS:
C...     TRA     double     out  TRACTION VECTOR
C...     STIFF   double     out  STIFFNESS MATRIX
C...     TRA     double     in   TRACTION VECTOR
C...     STIFF   double     in   STIFFNESS MATRIX
C...     USRVAL  double     in   PLASTIC SHEAR STRESS LIMIT VALUE
C...     [USRSTA  double     in   ELASTIC BRANCH ORIGIN SLIP VALUE,
C...                              (NORMALLY ZERO) MODEL BOND3E ONLY]
C...
C...   PROGRAMMED BY S.W.H. ENSINK
C...   Author: S.W.H. ENSINK
C...   Revision: 0.1
C...   Date: 2009/11/02
C.......................................................................
C
      CHARACTER*6      USRMOD
      INTEGER          NT, NUV, ELEMEN, INTPT, ITER, NUS, NUI,
     $                 USRIND(NUI)
      DOUBLE PRECISION U0(NT), DU(NT), STIFF(NT,NT), TRA(NT),
     $                 USRVAL(NUV), AGE0, DTIME, TEMP0, DTEMP,
     $                 COORD(3), SE(NT,NT), USRSTA(NUS), D11, D22,
     $                 DSTIF(2), TAUULT
C
C...    GET MATERIAL DATA
      CALL GTC('../MATERI/DSTIF', DSTIF, 2)
C
C...    GET MATERIAL DATA FROM USRVAL
      TAUULT = USRVAL(1)
C      CALL PRIVAL( TAUULT, 'TAUULT' )
C
C...    MODEL BOND3N: ELASTO-PLASTIC (BI-LINEAR) BONDSLIP TANGENTIAL
C...                  LINEAR NORMAL
C...                  NLE UNLOADING
C...    INPUT FROM DATFILE: USRVAL (SHEAR STRESS LIMIT VALUE)
C...    CALCULATE CURRENT TOTAL TRACTIONS AND TANGENT STIFFNESS
      IF (USRMOD .EQ. 'BOND3N') THEN
         TRA(1) = DSTIF(1) * (U0(1) + DU(1))
         CALL RSET(0.D0, STIFF, NT*NT)
         STIFF(1,1) = DSTIF(1)
         IF (ABS(U0(2) + DU(2)) .GE. TAUULT / DSTIF(2)) THEN
            IF (U0(2) + DU(2) .GE. TAUULT / DSTIF(2)) THEN
               TRA(2) = TAUULT
            ELSE
               TRA(2) = -TAUULT
            END IF
         ELSE
            TRA(2) = DSTIF(2) * (U0(2) + DU(2))
            STIFF(2,2) = DSTIF(2)
         END IF
C
C...    MODEL BOND3S: ELASTO-PLASTIC (BI-LINEAR) BONDSLIP TANGENTIAL
C...                  LINEAR NORMAL
C...                  SECANT UNLOADING
C...    INPUT FROM DATFILE: USRVAL (SHEAR STRESS LIMIT VALUE)
C...    CALCULATE CURRENT TOTAL TRACTIONS AND TANGENT STIFFNESS
      ELSE IF(USRMOD .EQ. 'BOND3S') THEN
100   CONTINUE
         TRA(1) = DSTIF(1) * (U0(1) + DU(1))
C...    BLOCK 1 INITIAL ELASTIC BRANCH
         IF (ABS(U0(2) + DU(2)) .LT. TAUULT / DSTIF(2) .AND.
     $      STIFF(2,2) .EQ. DSTIF(2)) THEN
            TRA(2) = DSTIF(2) * (U0(2) + DU(2))
            CALL RSET(0.D0, STIFF, NT*NT)
            STIFF(1,1) = DSTIF(1)
            STIFF(2,2) = DSTIF(2)
            CALL LOGPRI('USRIFC: INITIAL ELASTIC BRANCH',0)
C...    BLOCK 2 FIRST PLASTIC BRANCH
         ELSE IF (ABS(U0(2) + DU(2)) .GE. TAUULT / DSTIF(2) .AND.
     $      STIFF(2,2) .EQ. DSTIF(2)) THEN
            IF (U0(2) + DU(2) .GE. TAUULT / DSTIF(2)) THEN
               TRA(2) = TAUULT
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               CALL LOGPRI('USRIFC: FIRST PLASTIC BRANCH',0)
            ELSE
               TRA(2) = -TAUULT
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               CALL LOGPRI('USRIFC: FIRST PLASTIC BRANCH',0)
            END IF
C...    BLOCK 3 PLASTIC BRANCH LOADING
         ELSE IF (STIFF(2,2) .EQ. 0 .AND. DU(2) / U0(2) .GT. 0) THEN
            IF (U0(2) + DU(2) .GE. TAUULT / DSTIF(2)) THEN
               TRA(2) = TAUULT
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               CALL LOGPRI('USRIFC: PLASTIC BRANCH LOADING',0)
            ELSE
               TRA(2) = -TAUULT
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               CALL LOGPRI('USRIFC: PLASTIC BRANCH LOADING',0)
            END IF
C...    BLOCK 4 PLASTIC BRANCH FIRST UNLOADING
         ELSE IF (STIFF(2,2) .EQ. 0 .AND. DU(2) / U0(2) .LE. 0) THEN
            IF (ABS(DU(2)) .LE. ABS(U0(2))) THEN
               D22 = TAUULT / ABS(U0(2))
               TRA(2) = D22 * (U0(2) + DU(2))
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               STIFF(2,2) = D22
               CALL LOGPRI('USRIFC: PLASTIC BRANCH FIRST UNLOADING',0)
            ELSE IF (ABS(DU(2)) .GT. ABS(U0(2))) THEN
               DU(2) = U0(2) + DU(2)
               U0(2) = 0.D0
               CALL RSET(0.D0, STIFF, NT*NT)
```

File: /home1/sensink/usrifc.f

```
               STIFF(1,1) = DSTIF(1)
               STIFF(2,2) = DSTIF(2)
               GO TO 100
            END IF
C...    BLOCK 5 SECANT BRANCH
         ELSE IF (STIFF(2,2) .LT. DSTIF(2) .AND.
     $            STIFF(2,2) .GT. 0) THEN
            IF (DU(2) / U0(2) .GT. 0 .AND. ABS(DU(2) + U0(2)) .LE.
     $         TAUULT / STIFF(2,2)) THEN
               TRA(2) = STIFF(2,2) * (U0(2) + DU(2))
               D22 = STIFF(2,2)
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               STIFF(2,2) = D22
               CALL LOGPRI('USRIFC: SECANT BRANCH',0)
            ELSE IF (DU(2) / U0(2) .GT. 0 .AND. ABS(DU(2) + U0(2)) .GT.
     $         TAUULT / STIFF(2,2)) THEN
               STIFF(2,2) = 0
               GO TO 100
            ELSE IF (DU(2) / U0(2) .LE. 0 .AND. ABS(DU(2)) .LE.
     $         ABS(U0(2))) THEN
               TRA(2) = STIFF(2,2) * (U0(2) + DU(2))
               D22 = STIFF(2,2)
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               STIFF(2,2) = D22
               CALL LOGPRI('USRIFC: SECANT BRANCH',0)
            ELSE IF (DU(2) / U0(2) .LE. 0 .AND. ABS(DU(2)) .GT.
     $         ABS(U0(2))) THEN
               DU(2) = U0(2) + DU(2)
               U0(2) = 0.D0
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               STIFF(2,2) = DSTIF(2)
               GO TO 100
            END IF
         END IF
C
C...    MODEL BOND3E: ELASTO-PLASTIC (BI-LINEAR) BONDSLIP TANGENTIAL
C...                  LINEAR NORMAL
C...                  ELASTIC UNLOADING
C...    INPUT FROM DATFILE: USRVAL (SHEAR STRESS LIMIT VALUE)
C...                        USRSTA 0 (ELASTIC BRANCH ORIGIN VALUE)
C...    CALCULATE CURRENT TOTAL TRACTIONS AND TANGENT STIFFNESS
      ELSE IF(USRMOD .EQ. 'BOND3E') THEN
         TRA(1) = DSTIF(1) * (U0(1) + DU(1))
C...    BLOCK 1 INITIAL ELASTIC BRANCH
         IF (ABS(U0(2) + DU(2)) .LT. TAUULT / DSTIF(2) .AND.
     $      USRSTA(1) .EQ. 0) THEN
            TRA(2) = DSTIF(2) * (U0(2) + DU(2))
            CALL RSET(0.D0, STIFF, NT*NT)
            STIFF(1,1) = DSTIF(1)
            STIFF(2,2) = DSTIF(2)
            CALL LOGPRI('USRIFC: INITIAL ELASTIC BRANCH',0)
C...    BLOCK 2 FIRST PLASTIC BRANCH
         ELSE IF (ABS(U0(2) + DU(2)) .GE. TAUULT / DSTIF(2) .AND.
     $      STIFF(2,2) .EQ. DSTIF(2) .AND. USRSTA(1) .EQ. 0) THEN
            IF (U0(2) + DU(2) .GE. TAUULT / DSTIF(2)) THEN
               TRA(2) = TAUULT
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               CALL LOGPRI('USRIFC: FIRST PLASTIC BRANCH',0)
            ELSE
               TRA(2) = -TAUULT
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               CALL LOGPRI('USRIFC: FIRST PLASTIC BRANCH',0)
            END IF
C...    BLOCK 3 PLASTIC BRANCH LOADING
         ELSE IF (STIFF(2,2) .EQ. 0 .AND. DU(2) / U0(2) .GT. 0  .AND.
     $      USRSTA(1) .EQ. 0) THEN
            IF (U0(2) + DU(2) .GE. TAUULT / DSTIF(2)) THEN
               TRA(2) = TAUULT
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               CALL LOGPRI('USRIFC: PLASTIC BRANCH LOADING',0)
            ELSE
               TRA(2) = -TAUULT
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               CALL LOGPRI('USRIFC: PLASTIC BRANCH LOADING',0)
            END IF
C...    BLOCK 4 PLASTIC BRANCH FIRST UNLOADING
         ELSE IF (STIFF(2,2) .EQ. 0 .AND. DU(2) / U0(2) .LE. 0 .AND.
     $      USRSTA(1) .EQ. 0) THEN
            IF (U0(2) .GT. 0 .AND. ABS(DU(2)) .LE. 2*TAUULT / DSTIF(2))
     $      THEN
               USRSTA(1) = U0(2) - TAUULT / DSTIF(2)
               TRA(2) = DSTIF(2)*(U0(2)+DU(2)-USRSTA(1))
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               STIFF(2,2) = DSTIF(2)
               CALL LOGPRI('USRIFC: PLASTIC BRANCH FIRST UNLOADING',0)
            ELSE IF (U0(2) .LE. 0 .AND. ABS(DU(2)) .LE.
     $         2*TAUULT / DSTIF(2)) THEN
               USRSTA(1) = U0(2) + TAUULT / DSTIF(2)
               TRA(2) = DSTIF(2)*(U0(2)+DU(2)-USRSTA(1))
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               STIFF(2,2) = DSTIF(2)
               CALL LOGPRI('USRIFC: PLASTIC BRANCH FIRST UNLOADING',0)
            ELSE IF (U0(2) .GT. 0) THEN
               USRSTA(1) = U0(2) - TAUULT / DSTIF(2)
               TRA(2) = -TAUULT
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               STIFF(2,2) = 0
               CALL LOGPRI('USRIFC: PLASTIC BRANCH FIRST UNLOADING',0)
            ELSE
               USRSTA(1) = U0(2) + TAUULT / DSTIF(2)
               TRA(2) = TAUULT
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               STIFF(2,2) = 0
               CALL LOGPRI('USRIFC: PLASTIC BRANCH FIRST UNLOADING',0)
            END IF
C...    BLOCK 5 SHIFTED ELASTIC BRANCH
         ELSE IF (USRSTA(1) .NE. 0 .AND. ABS(U0(2) + DU(2)) .GT.
     $      ABS(USRSTA(1)) - TAUULT / DSTIF(2) .AND. ABS(U0(2) + DU(2))
     $      .LT. ABS(USRSTA(1)) + TAUULT / DSTIF(2) .AND. STIFF(2,2)
     $      .EQ. DSTIF(2)) THEN
            TRA(2) = DSTIF(2)*(U0(2)+DU(2)-USRSTA(1))
```

File: /home1/sensink/usrifc.f

```
            CALL RSET(0.D0, STIFF, NT*NT)
            STIFF(1,1) = DSTIF(1)
            STIFF(2,2) = DSTIF(2)
            CALL LOGPRI('USRIFC: SHIFTED ELASTIC BRANCH',0)
C...    BLOCK 6 SHIFTED FIRST PLASTIC BRANCH
         ELSE IF (USRSTA(1) .NE. 0 .AND. ABS(U0(2) + DU(2)) .GE.
     $      ABS(USRSTA(1)) + TAUULT / DSTIF(2) .AND. STIFF(2,2) .EQ.
     $      DSTIF(2) .NEQV. (USRSTA(1) .NE. 0 .AND. ABS(U0(2) + DU(2))
     $      .LE. ABS(USRSTA(1)) - TAUULT / DSTIF(2) .AND. STIFF(2,2)
     $      .EQ. DSTIF(2))) THEN
            IF (U0(2) + DU(2) .GT. USRSTA(1)) THEN
               TRA(2) = TAUULT
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               CALL LOGPRI('USRIFC: SHIFTED FIRST PLASTIC BRANCH',0)
            ELSE
               TRA(2) = -TAUULT
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               CALL LOGPRI('USRIFC: SHIFTED FIRST PLASTIC BRANCH',0)
            END IF
C...    BLOCK 7 SHIFTED PLASTIC BRANCH LOADING
         ELSE IF (USRSTA(1) .NE. 0 .AND. STIFF(2,2) .EQ. 0 .AND.
     $      DU(2) / (U0(2) - USRSTA(1)) .GT. 0) THEN
            IF (U0(2) .GT. USRSTA(1)) THEN
               TRA(2) = TAUULT
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               STIFF(2,2) = 0
               CALL LOGPRI('USRIFC: SHIFTED PLASTIC BRANCH LOADING',0)
            ELSE
               TRA(2) = -TAUULT
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               STIFF(2,2) = 0
               CALL LOGPRI('USRIFC: SHIFTED PLASTIC BRANCH LOADING',0)
            END IF
C...    BLOCK 8 SHIFTED PLASTIC BRANCH FIRST UNLOADING
         ELSE IF (USRSTA(1) .NE. 0 .AND. STIFF(2,2) .EQ. 0 .AND.
     $      DU(2) / (U0(2) - USRSTA(1)) .LT. 0) THEN
            IF (U0(2) .GT. 0 .AND. ABS(DU(2)) .LE. 2*TAUULT / DSTIF(2))
     $      THEN
               USRSTA(1) = U0(2) - TAUULT / DSTIF(2)
               TRA(2) = DSTIF(2)*(U0(2)+DU(2)-USRSTA(1))
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               STIFF(2,2) = DSTIF(2)
        CALL LOGPRI('USRIFC: SHIFTED PLASTIC BRANCH FIRST UNLOADING',0)
            ELSE IF (U0(2) .LE. 0 .AND. ABS(DU(2)) .LE.
     $         2*TAUULT / DSTIF(2)) THEN
               USRSTA(1) = U0(2) + TAUULT / DSTIF(2)
               TRA(2) = DSTIF(2)*(U0(2)+DU(2)-USRSTA(1))
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
               STIFF(2,2) = DSTIF(2)
        CALL LOGPRI('USRIFC: SHIFTED PLASTIC BRANCH FIRST UNLOADING',0)
            ELSE IF (U0(2) .GT. 0) THEN
               USRSTA(1) = U0(2) - TAUULT / DSTIF(2)
               TRA(2) = -TAUULT
               CALL RSET(0.D0, STIFF, NT*NT)
               STIFF(1,1) = DSTIF(1)
```

File: /home1/sensink/usrifc.f

```
              STIFF(2,2) = 0
          CALL LOGPRI('USRIFC: SHIFTED PLASTIC BRANCH FIRST UNLOADING',0)
              ELSE
                 USRSTA(1) = U0(2) + TAUULT / DSTIF(2)
                 TRA(2) = TAUULT
                 CALL RSET(0.D0, STIFF, NT*NT)
                 STIFF(1,1) = DSTIF(1)
                 STIFF(2,2) = 0
          CALL LOGPRI('USRIFC: SHIFTED PLASTIC BRANCH FIRST UNLOADING',0)
              END IF
           END IF
        ELSE
        CALL PRGERR('USRIFC',001)
        END IF
        END
```

Two small scale tests using different loading have been performed to check the correctness of the user supplied interface models. The model consists of three elements connected to create a small scale tension-pull model: a CL6TR truss element, an CL12I interface element and an CQ16A axi-symmetric element. Using displacement control the load is given to the truss element. The bi-linear elasto-plastic bond-slip relation is used, see figure 5.5.
In the first test positive load steps are taken until a displacement of 0.075 mm. Then the same loadsteps are given in the opposite direction, see figure J.1 for the results. In the second test load steps are taken that continuously change sign and also jump from a positive plastic region to a negative plastic region, see figure J.2 for the results.

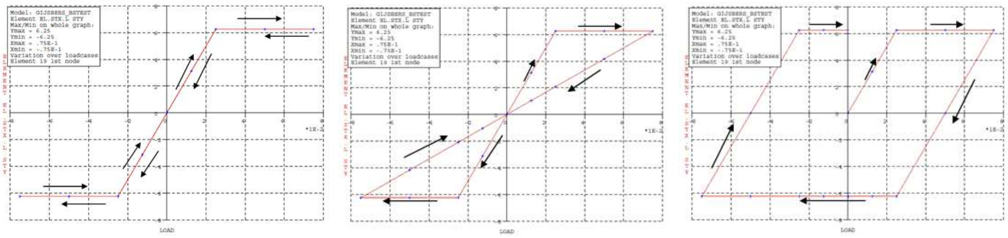


Figure J.1: Results first loading, from left to right: BOND3N, BOND3S and BOND3E

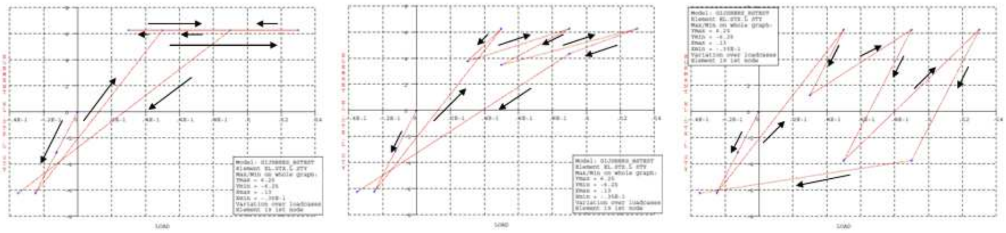


Figure J.2: Results second loading, from left to right: BOND3N, BOND3S and BOND3E

Appendix K

Nonlinear 3D calculation: *DIANA* program settings

Used element types and integration scheme:

CL9TR Truss element (default integration)
CL18I Line-solid interface element (5-point lumped integration)
CTP45 3D wedge element (4x3 integration)
CHX60 3D brick element (default integration)

Other *DIANA* program settings used:

Interface element with ZAXIS 0 0 1 and DIAMET 8

Non-linear settings:
Energy based convergence criterion (default tolerance of 0.0001)
Maximum number of iterations: 100

Appendix L

Sawtooth diagrams used in section 6.5

Table L.1: Accuracy of sawtooth diagrams (with concrete 50 teeth)

	concrete (50 teeth)	bond-slip (23 teeth)	steel 40 teeth)
average of $p+$ en $p-$ [N/mm^2]	(0.087+0.094)/2	(0.22+0.22)/2	(0+28)/2
maximum elastic value [N/mm^2]	2.5	6.25	400
ratio	3.6%	3.5%	3.5%

Table L.2: Accuracy of sawtooth diagrams (with concrete 25 teeth)

	concrete (25 teeth)	bond-slip (12 teeth)	steel (19 teeth)
average of $p+$ en $p-$ [N/mm^2]	(0.17+0.20)/2	(0.46+0.46)/2	(0+59)/2
maximum elastic value [N/mm^2]	2.5	6.25	400
ratio	7.4%	7.4%	7.4%

Table L.3: Accuracy of sawtooth diagrams (with concrete 12 teeth)

	concrete (12 teeth)	bond-slip (6 teeth)	steel (9 teeth)
average of $p+$ en $p-$ [N/mm^2]	(0.34+0.43)/2	(0.96+0.96)/2	(0+123)/2
maximum elastic value [N/mm^2]	2.5	6.25	400
ratio	15.4%	15.4%	15.4%

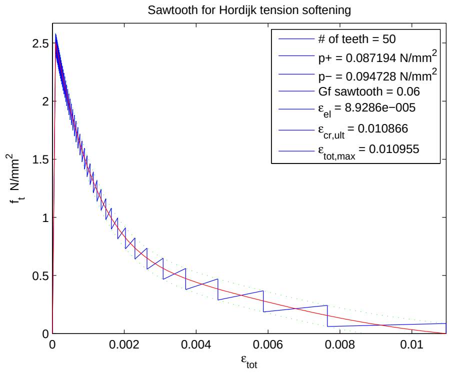


Figure L.1: Sawtooth approximation for concrete using 50 teeth

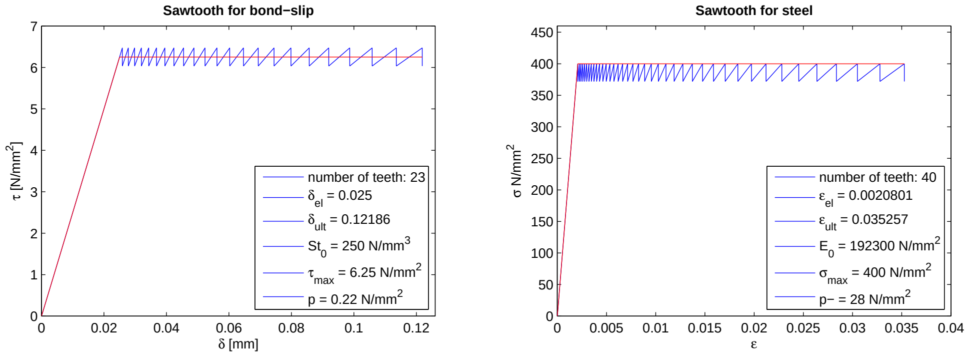


Figure L.2: Sawtooth approximation for bond-slip using 23 teeth (left) and steel using 40 teeth (right)

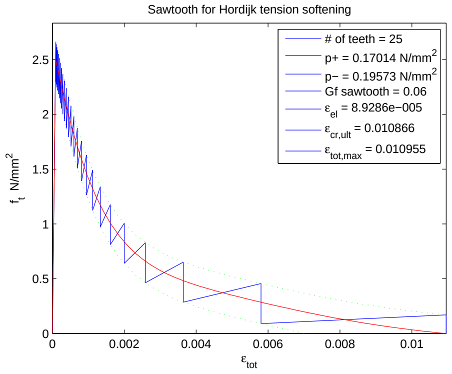


Figure L.3: Sawtooth approximation for concrete using 25 teeth

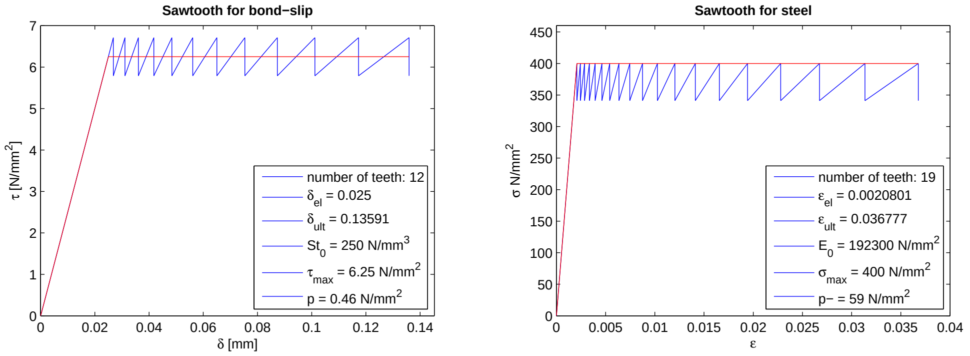


Figure L.4: Sawtooth approximation for bond-slip using 12 teeth (left) and steel using 19 teeth (right)

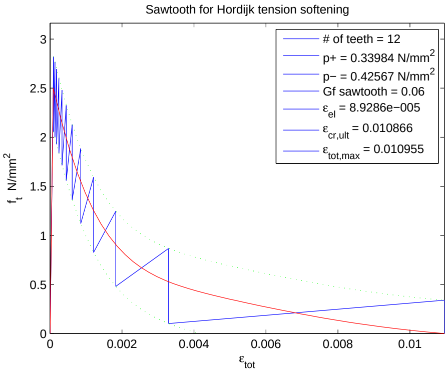


Figure L.5: Sawtooth approximation for concrete using 12 teeth

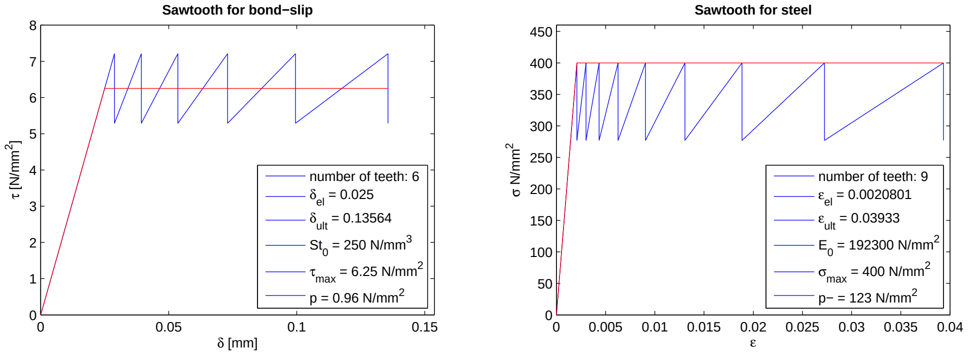


Figure L.6: Sawtooth approximation for bond-slip using 6 teeth (left) and steel using 9 teeth (right)

Appendix M

Sawtooth diagrams used in section 6.7

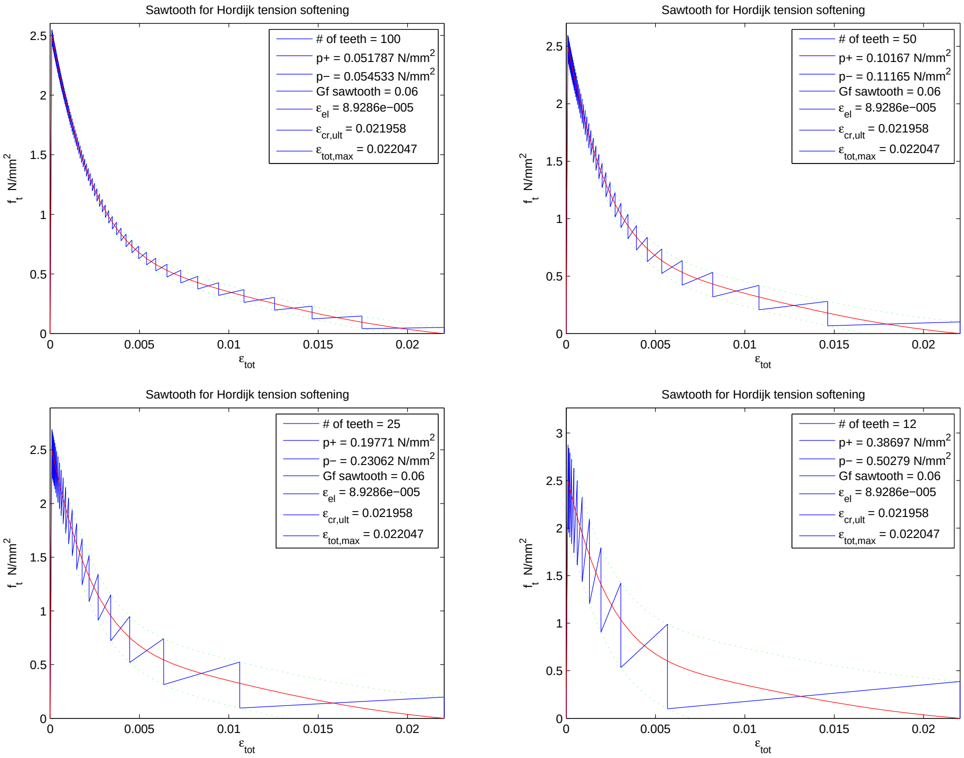


Figure M.1: Sawtooth approximation for concrete used for first mesh refinement, 100 teeth (top left), 50 teeth (top right), 25 teeth (bottom left) and 12 teeth (bottom right)

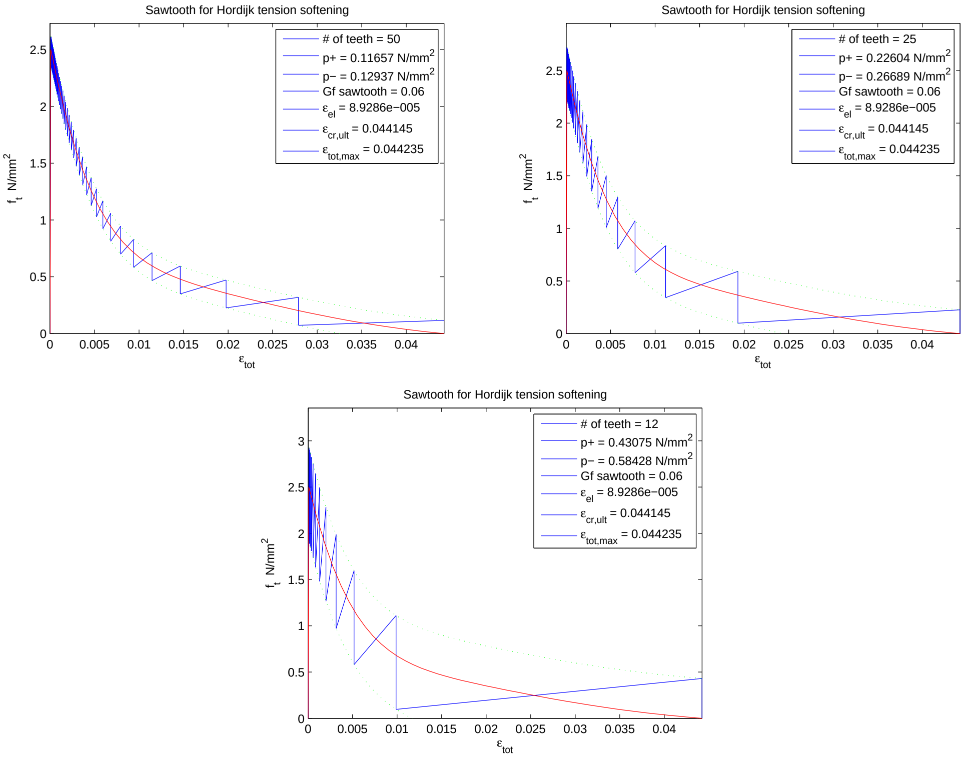


Figure M.2: Sawtooth approximation for concrete used for second mesh refinement, 50 teeth (top left), 25 teeth (top right) and 12 teeth (bottom)

Appendix N

Sawtooth diagrams used in chapter 7

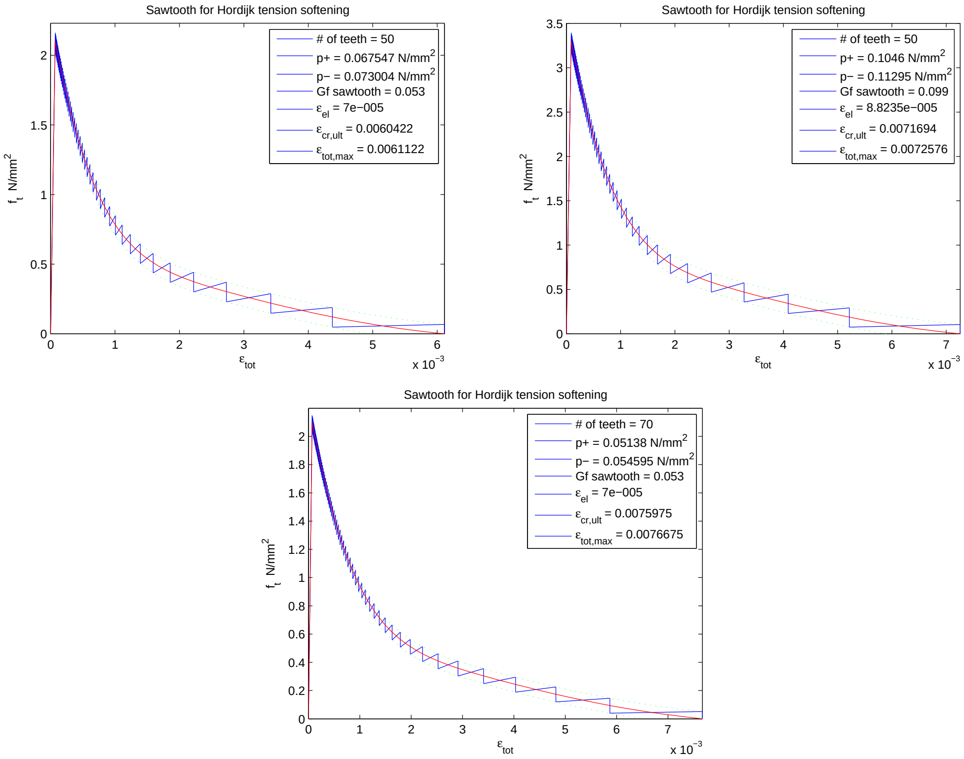


Figure N.1: Sawtooth approximation for concrete, Hordijk 50 teeth B25 used for simulations S4D12AB25 and S8D12AB25 (top left), Hordijk 50 teeth B45 used for simulation S8D12AB45 (top right) and Hordijk 70 teeth used for simulations S4D16AB25, S6D16AB25, S2D25AB25 and S4D25AB25 (bottom)

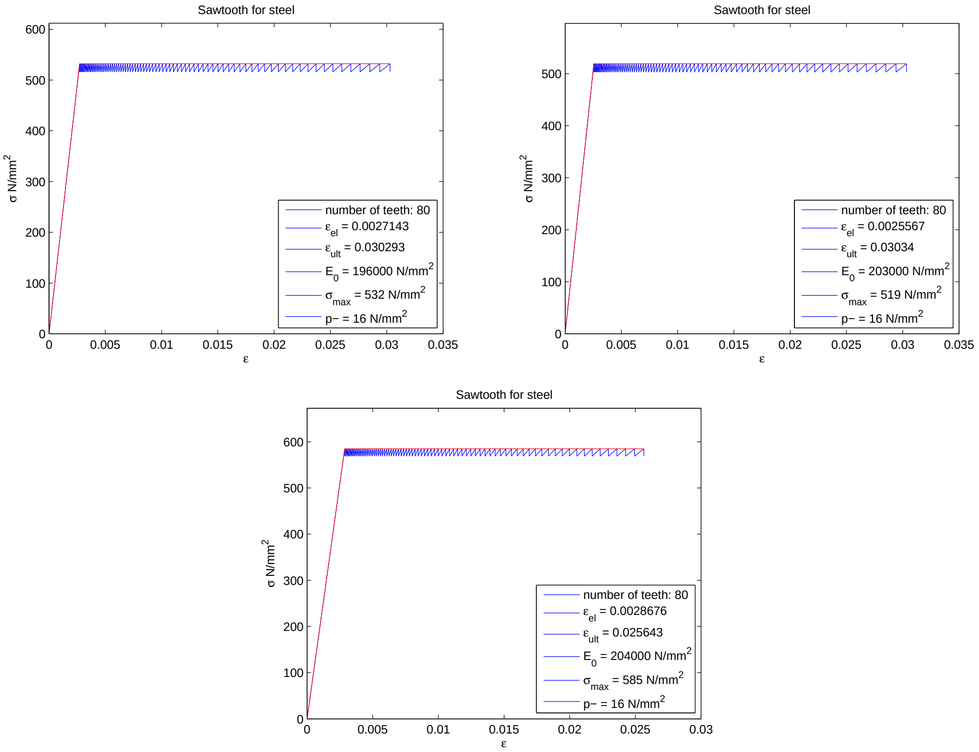


Figure N.2: Sawtooth approximation for steel, Ø12 (top left), Ø16 (top right) and Ø25 (bottom)

Appendix O

Mayer S4D12-A (B25)

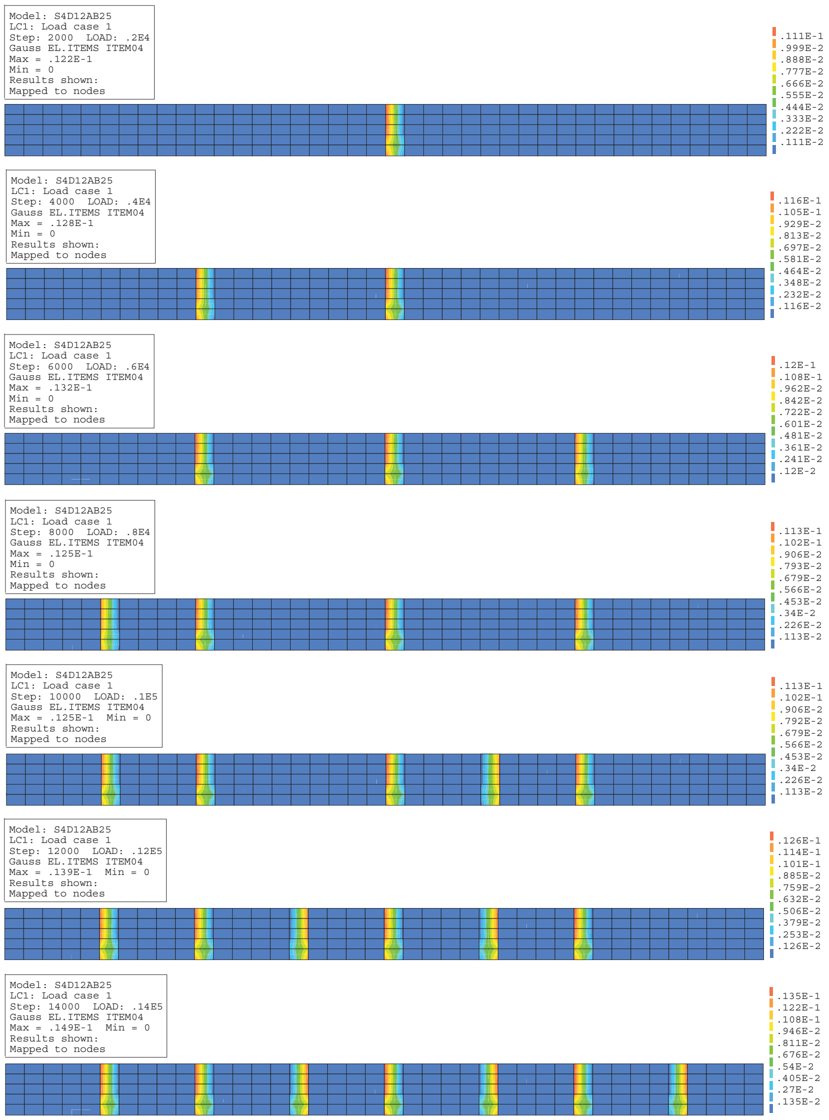


Figure O.1: Crack strain ϵ_{cr}^{nn}, from top to bottom (approximate displacement): u_y=0.85 mm, u_y=1.25 mm, u_y=1.65 mm, u_y=2.20 mm, u_y=2.55 mm, u_y=2.95 mm and u_y=3.65 mm

Appendix P

Mayer S8D12-A (B25)

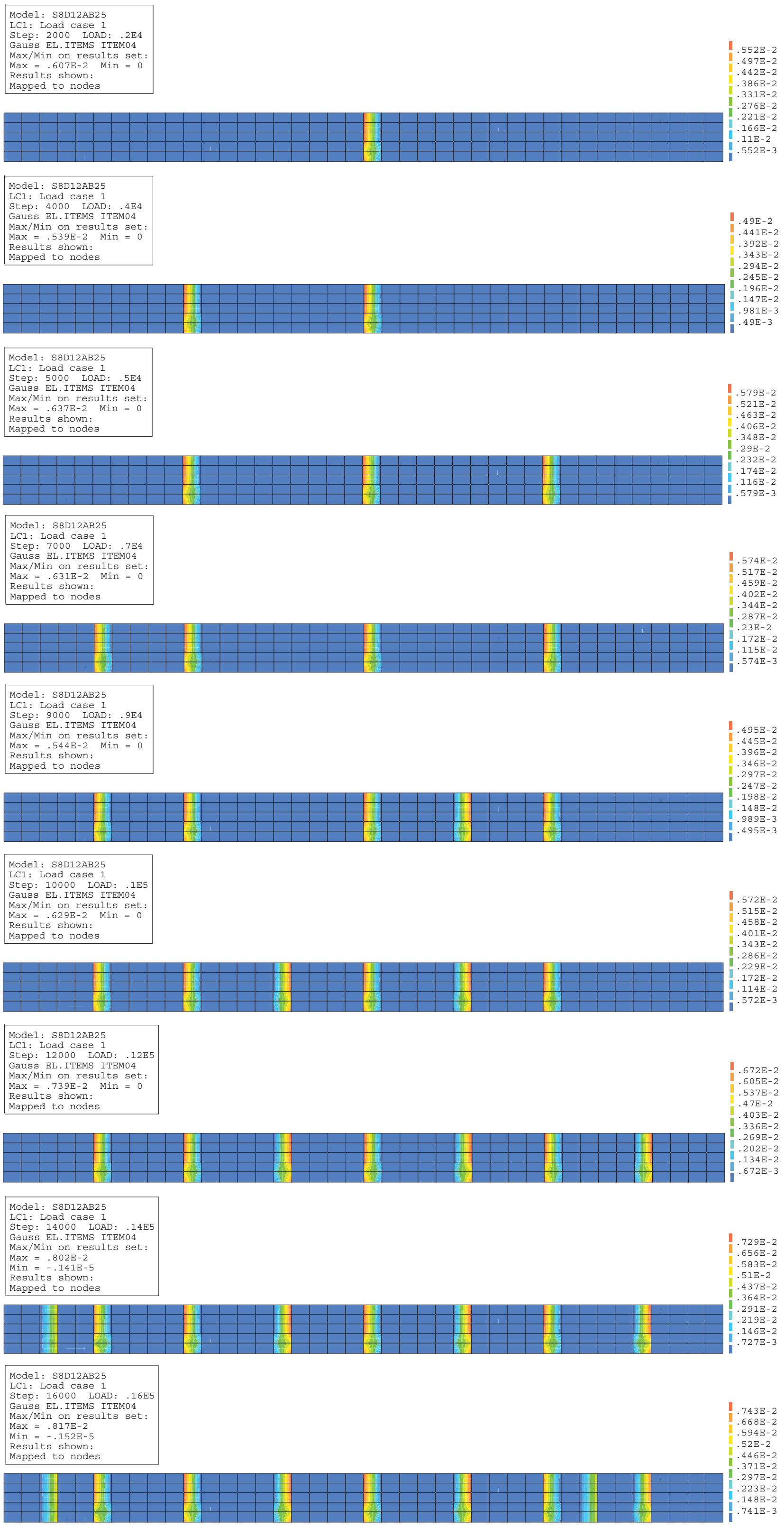


Figure P.1: Crack strain ϵ_{cr}^{nn}, from top to bottom (approximate displacement): u_y=0.50 mm, u_y=0.70 mm, u_y=0.85 mm, u_y=1.05 mm, u_y=1.25 mm, u_y=1.35 mm, u_y=1.80 mm, u_y=2.10 mm and u_y=2.30 mm

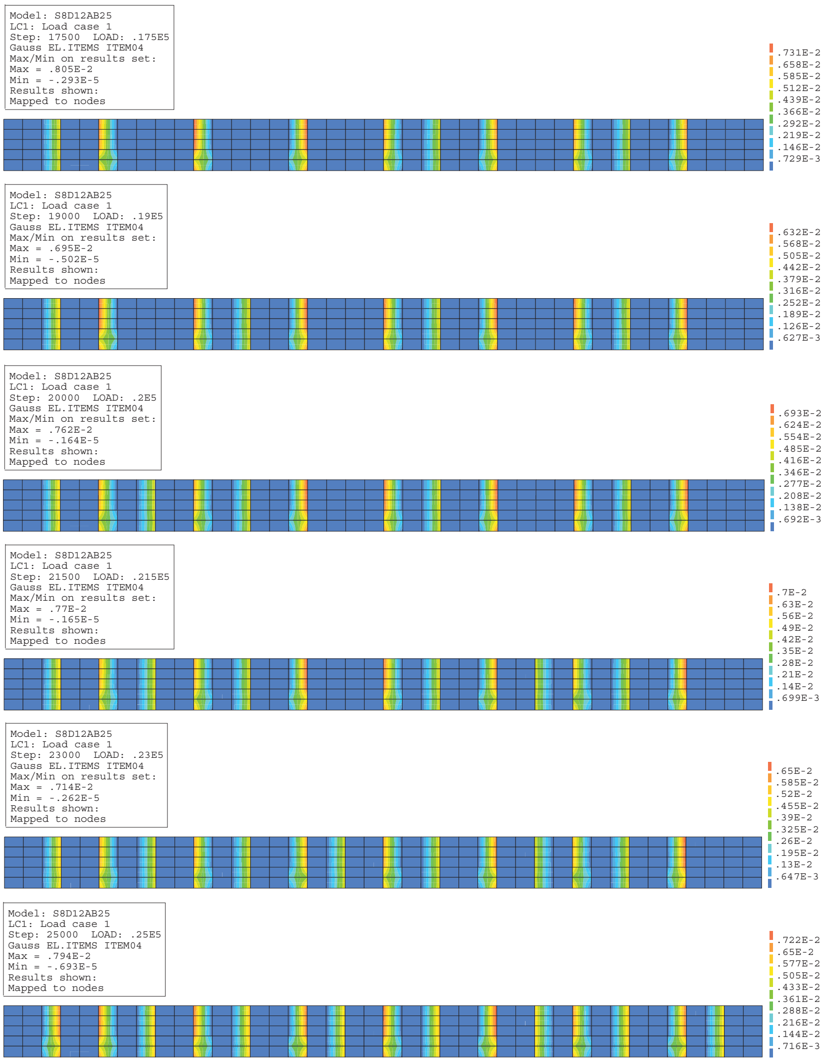


Figure P.2: Crack strain ϵ_{cr}^{nn}, from top to bottom (approximate displacement): u_y=2.40 mm, u_y=2.50 mm, u_y=2.60 mm, u_y=2.70 mm, u_y=2.85 mm and u_y=3.40 mm

Appendix Q

Mayer S8D12-A (B45)

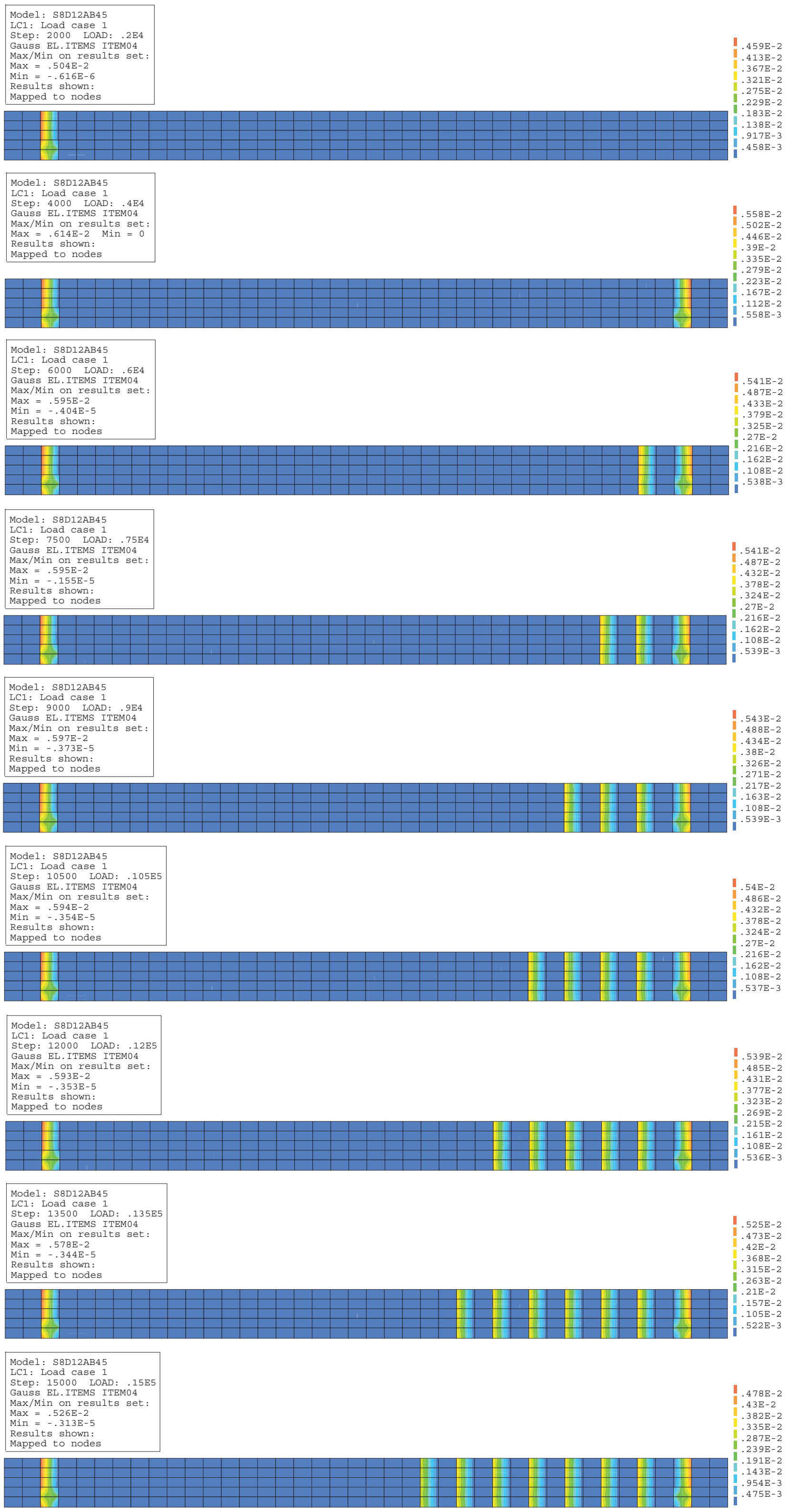


Figure Q.1: Crack strain ϵ_{cr}^{nn}, from top to bottom (approximate displacement): u_y=0.50 mm, u_y=0.70 mm, u_y=0.85 mm, u_y=0.95 mm, u_y=1.10 mm, u_y=1.20 mm, u_y=1.30 mm, u_y=1.40 mm and u_y=1.40 mm

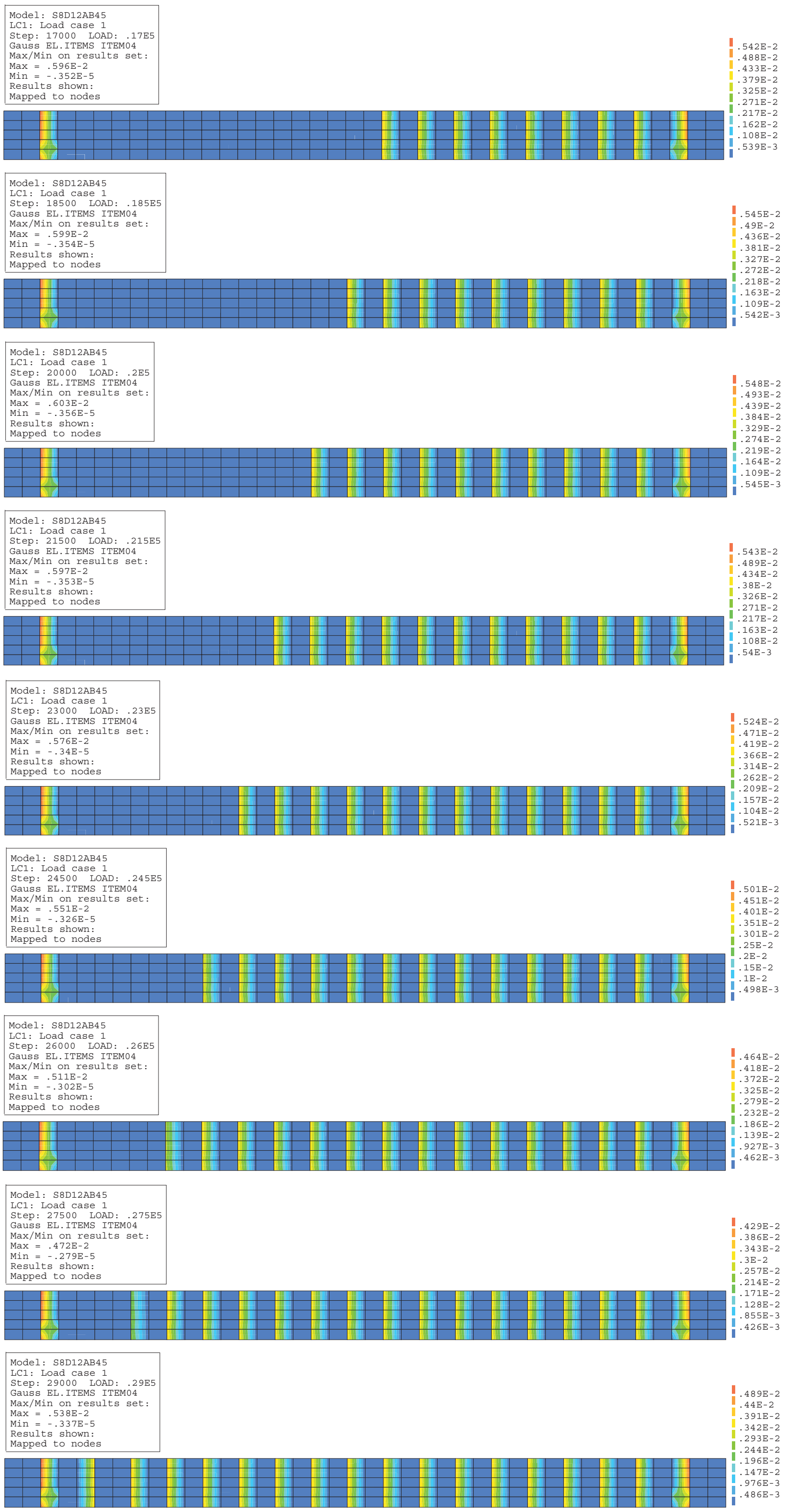


Figure Q.2: Crack strain ϵ_{cr}^{nn}, from top to bottom (approximate displacement): u_y=1.70 mm, u_y=1.85 mm, u_y=2.00 mm, u_y=2.10 mm, u_y=2.15 mm, u_y=2.15 mm, u_y=2.10 mm, u_y=2.05 mm and u_y=2.60 mm

Appendix R

Mayer S4D16-A (B25)

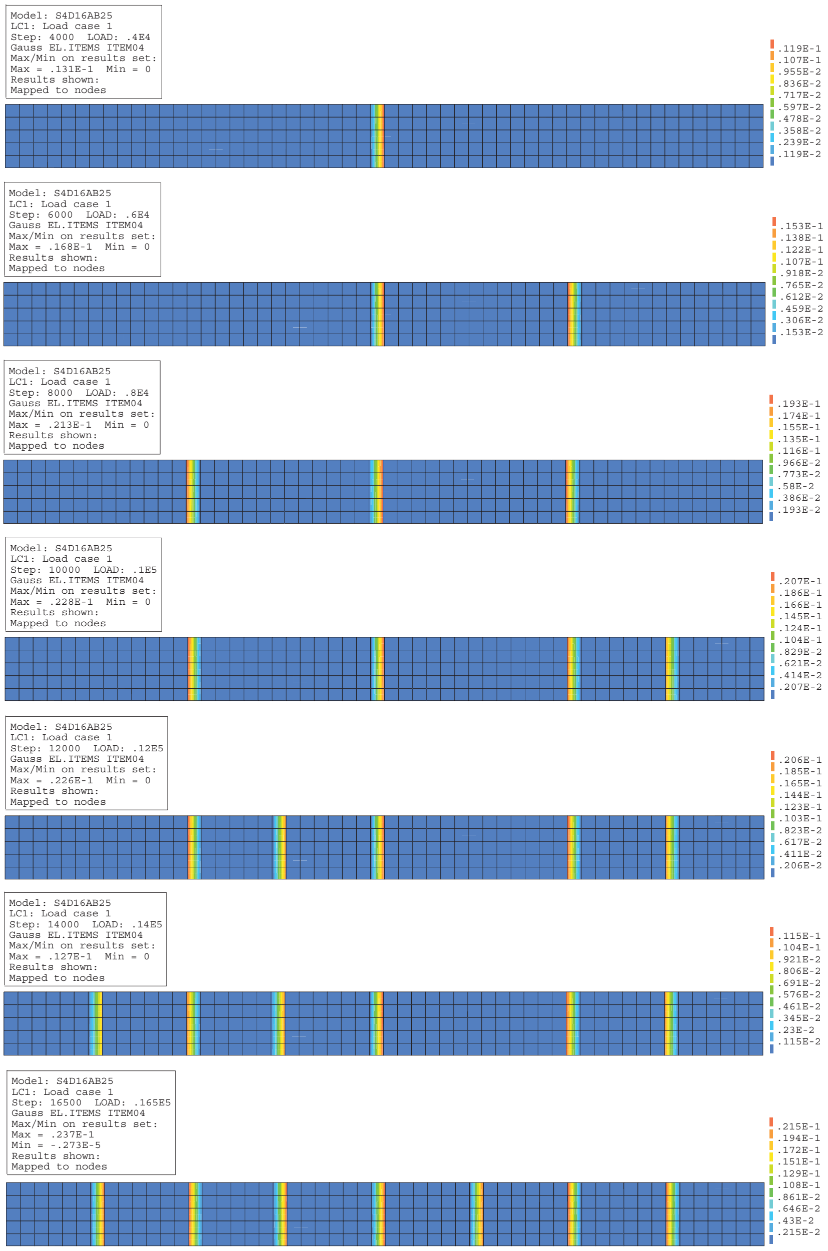


Figure R.1: Crack strain ϵ_{cr}^{nn}, from top to bottom (approximate displacement): u_y=0.75 mm, u_y=1.30 mm, u_y=2.15 mm, u_y=2.75 mm, u_y=3.30 mm, u_y=2.10 mm and u_y=4.60 mm

Appendix S

Mayer S6D16-A (B25)

This appendix shows some detailed results of simulation S6D16AB25 not given in chapter 7. Load-displacement curve (41000 load steps):

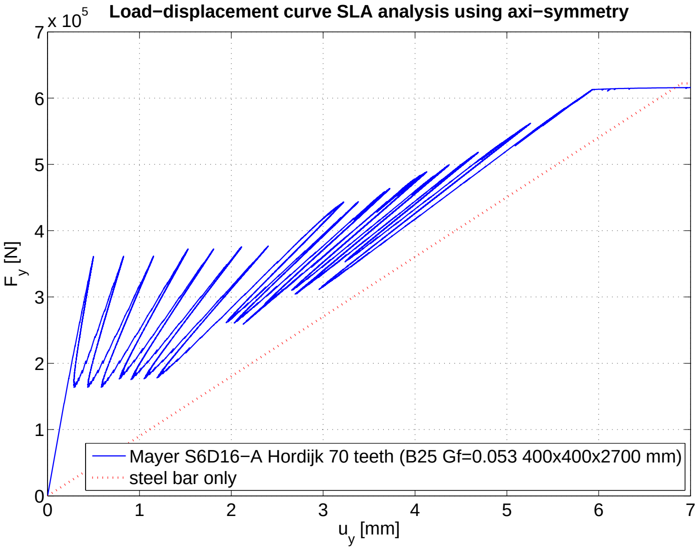


Figure S.1: Load-displacement curve simulation S6D16-A (B25)

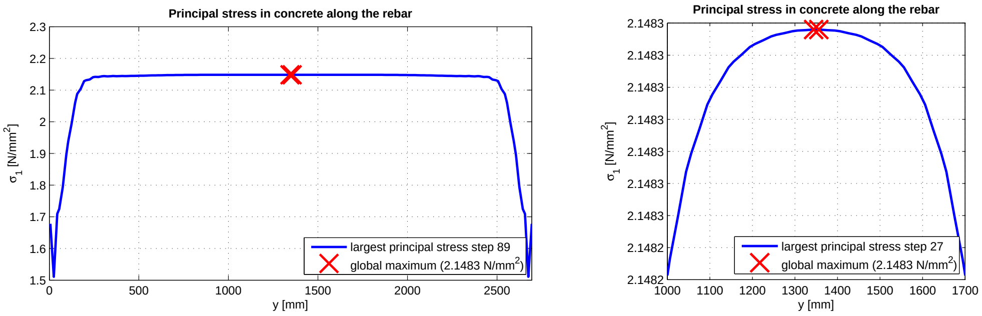


Figure S.2: Principal stress along inner side of rebar at crack initiation

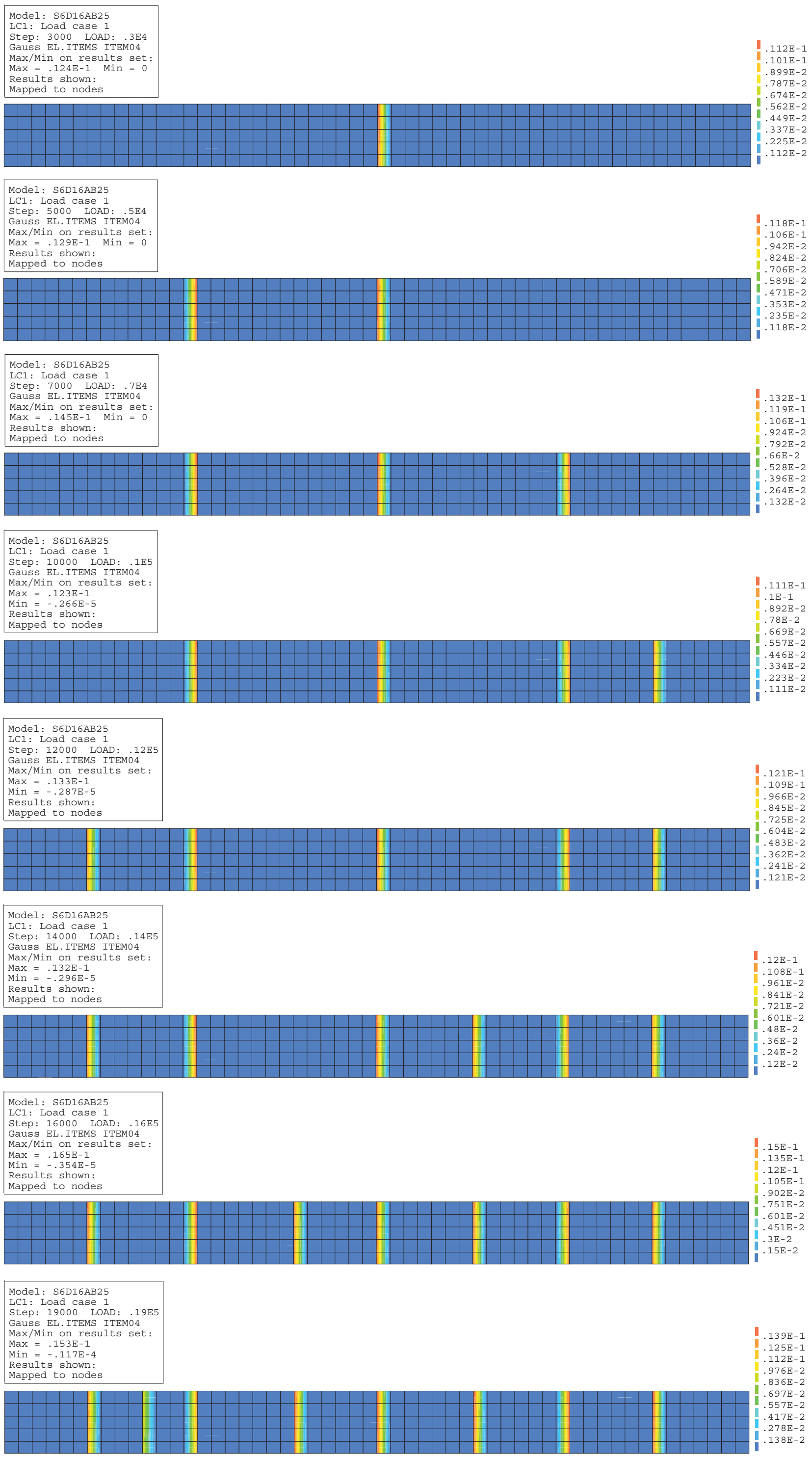


Figure S.3: Crack strain ϵ_{cr}^{nn}, from top to bottom (approximate displacement): u_y=0.75 mm, u_y=1.05 mm, u_y=1.50 mm, u_y=1.55 mm, u_y=1.90 mm, u_y=2.15 mm, u_y=3.15 mm and u_y=3.20 mm

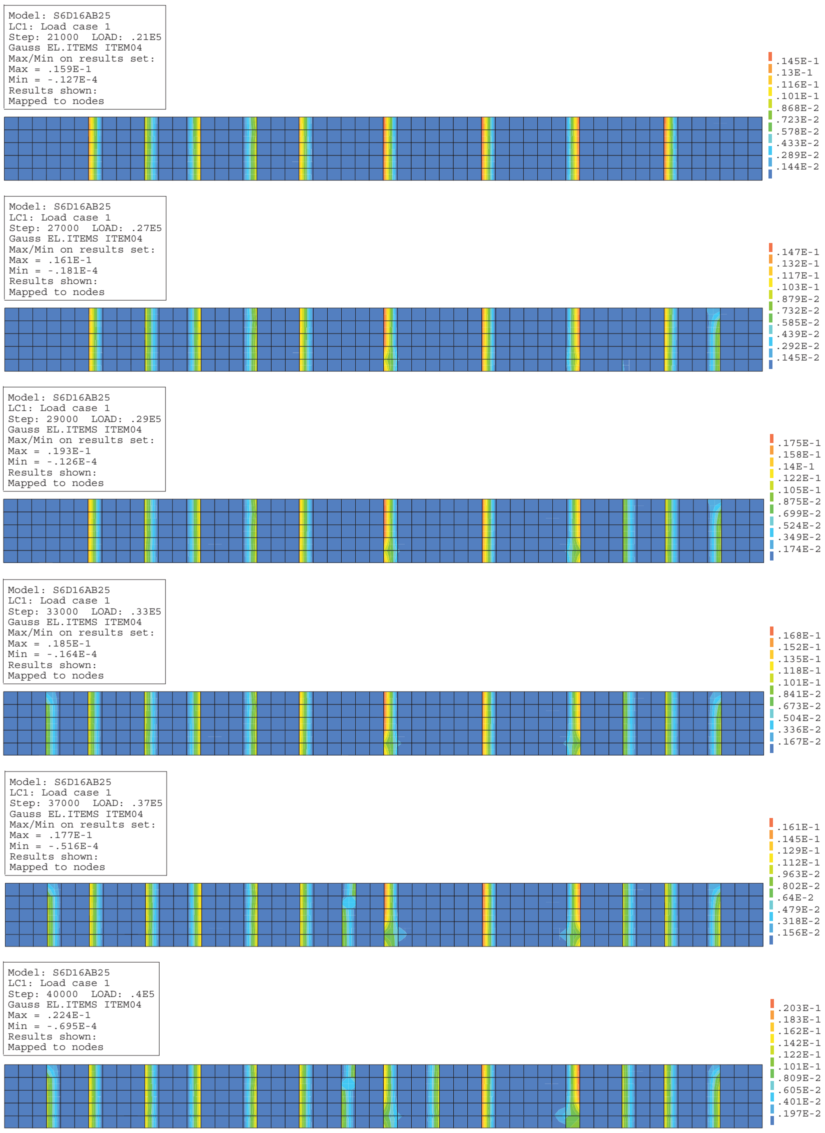


Figure S.4: Crack strain ϵ_{cr}^{nn}, from top to bottom (approximate displacement): u_y=3.45 mm, u_y=3.50 mm, u_y=4.00 mm, u_y=4.20 mm, u_y=4.60 mm and u_y=5.95 mm

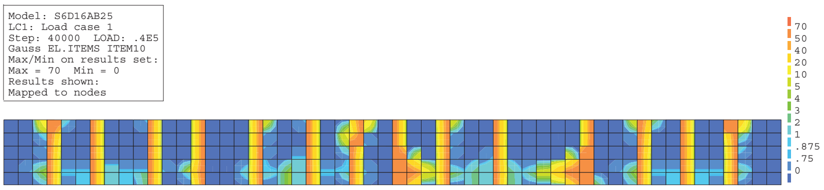


Figure S.5: Damage at u_y=5.95 mm (damage indicator N-direction, 70 equals complete damage)

Appendix T

Mayer S2D25-A (B25)

This appendix shows some detailed results of simulation S2D25AB25 not given in chapter 7. Load-displacement curve (18000 load steps):

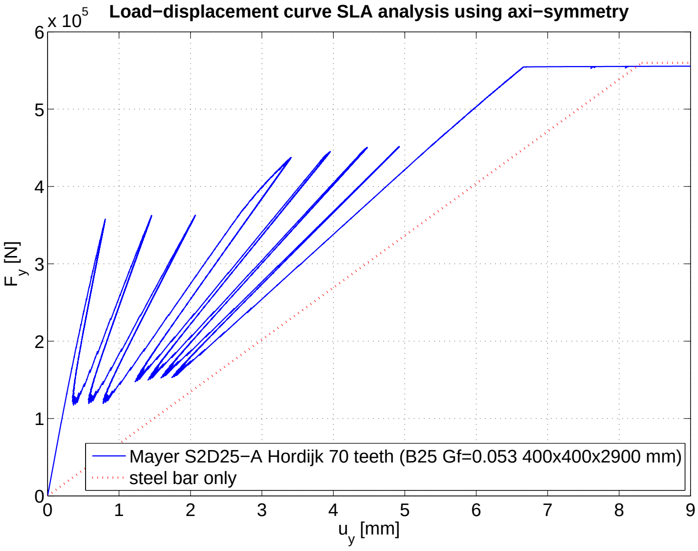


Figure T.1: Load-displacement curve simulation S2D25-A (B25)

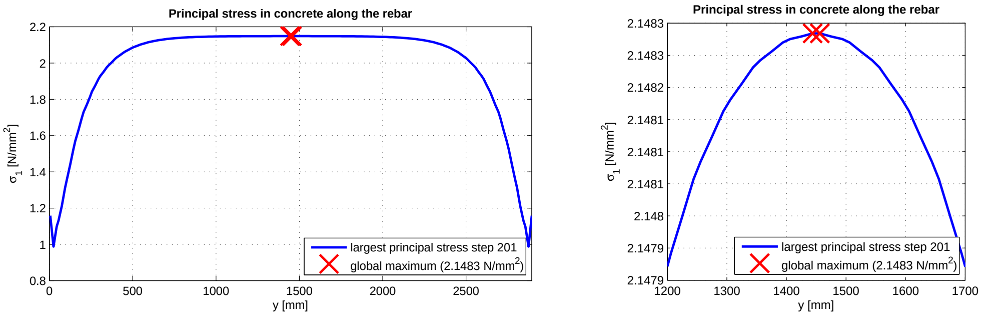


Figure T.2: Principal stress along inner side of rebar at crack initiation

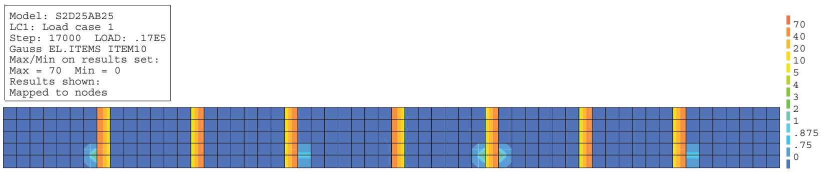


Figure T.3: Damage at u_y=6.40 mm (damage indicator N-direction, 70 equals complete damage)

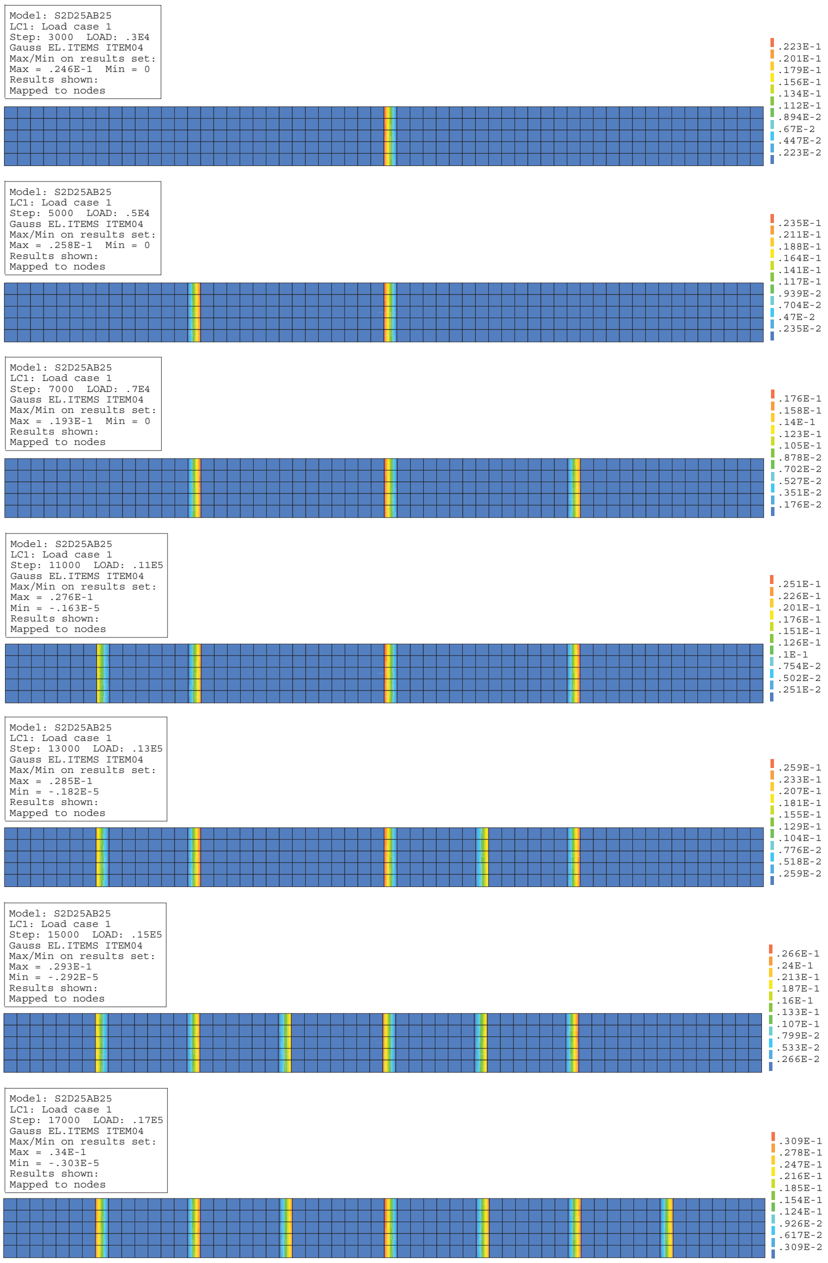


Figure T.4: Crack strain ϵ_{cr}^{nn}, from top to bottom (approximate displacement): u_y=1.40 mm, u_y=2.00 mm, u_y=1.95 mm, u_y=3.20 mm, u_y=3.95 mm, u_y=4.90 mm and u_y=6.40 mm

Appendix U

Mayer S4D25-A (B25)

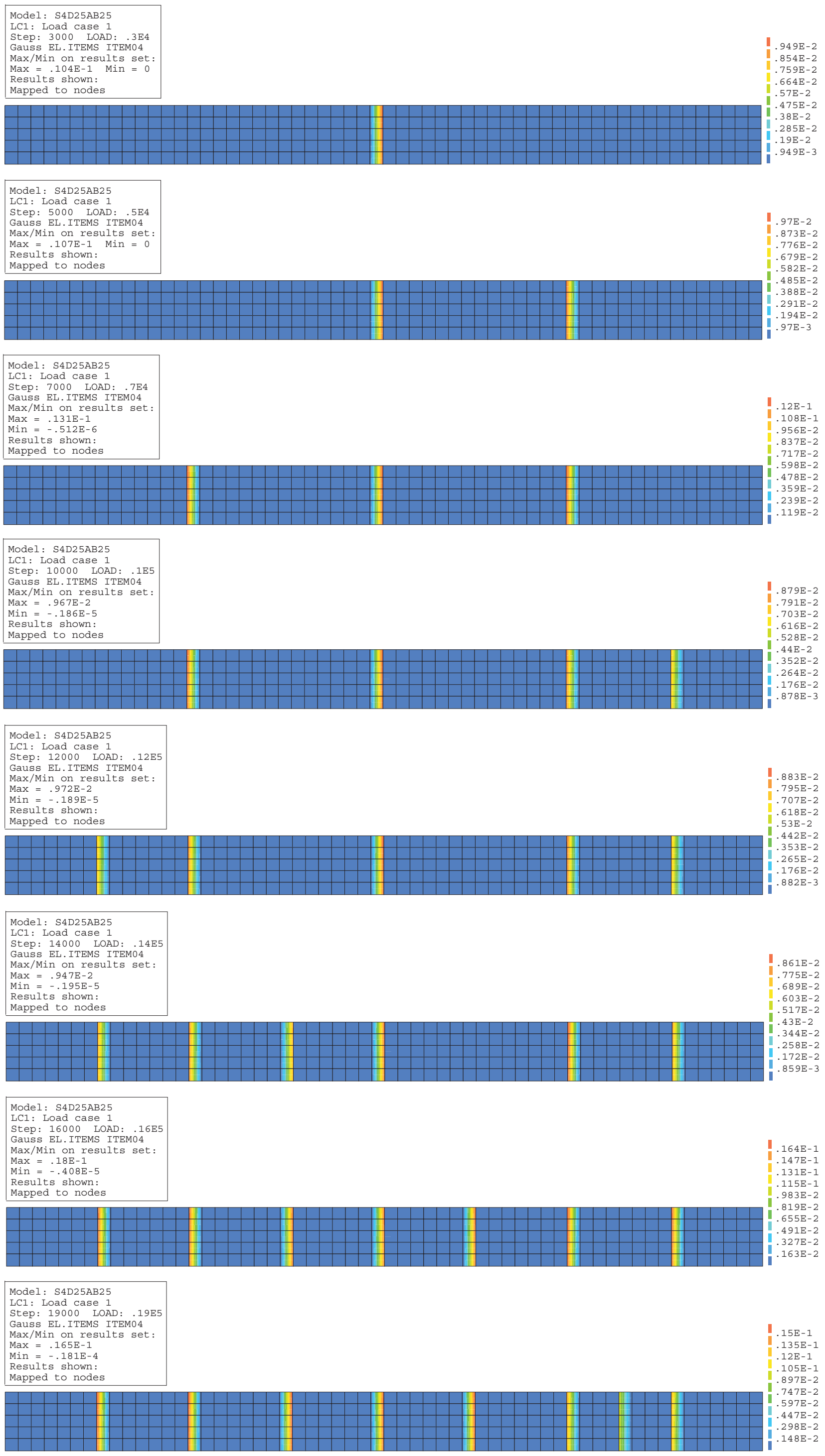


Figure U.1: Crack strain ϵ_{cr}^{nn}, from top to bottom (approximate displacement): u_y=0.65 mm, u_y=0.90 mm, u_y=1.40 mm, u_y=1.20 mm, u_y=1.35 mm, u_y=1.55 mm, u_y=3.40 mm and u_y=3.35 mm

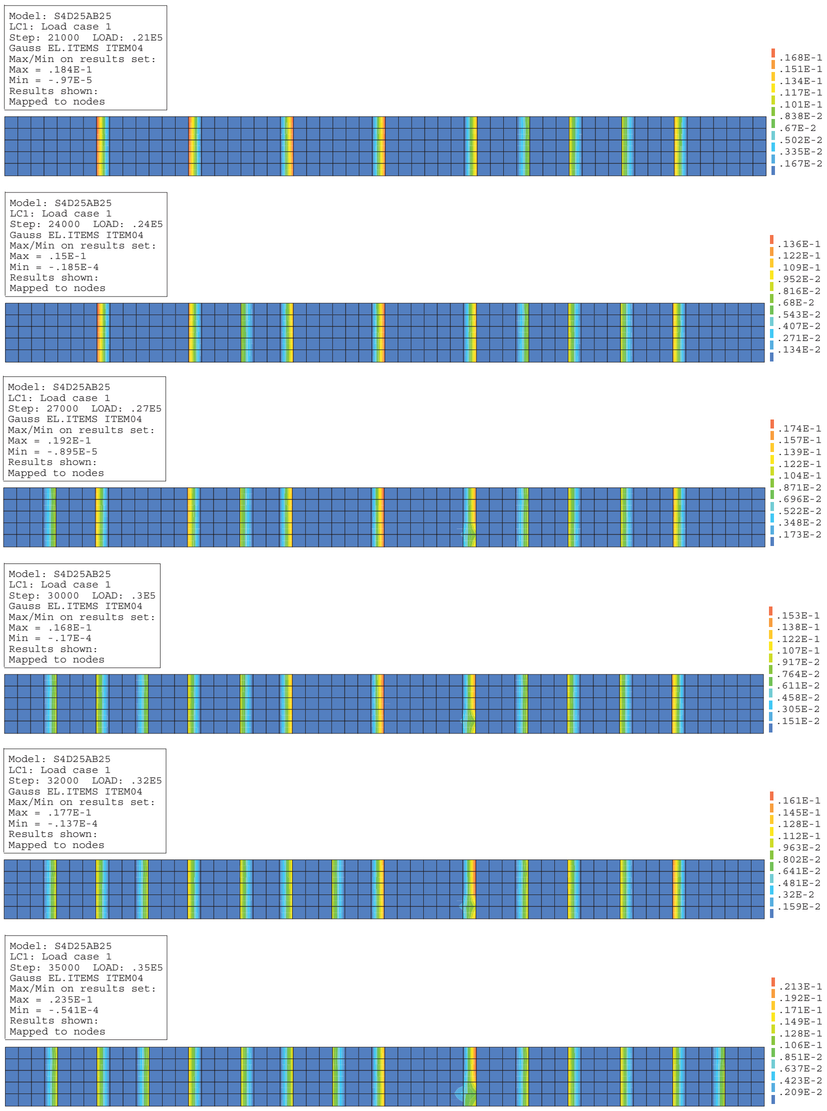


Figure U.2: Crack strain ϵ_{cr}^{nn}, from top to bottom (approximate displacement): u_y=4.25 mm, u_y=4.40 mm, u_y=4.40 mm, u_y=4.95 mm, u_y=4.70 mm and u_y=6.15 mm