



Universitetet
i Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

MASTEROPPGAVE

Studieprogram/spesialisering: Konstruksjoner og Materialer /Byggkonstruksjoner	Vårsemesteret, 2018 Åpen / Konfidensiell
Student/studenter: Otto-Arne Lade signatur (er)
Faglig ansvarlig: Mudiyansele Samindi Manjula K Samarakoon Veileder(e): Samdar Kakay , Daniel Bårdsen	
Tittel på oppgaven: EXPERIMENTAL AND NUMERICAL CRACK FORM SIMULATIONS OF REINFORCED CONCRETE BEAM Engelsk tittel: EXPERIMENTAL AND NUMERICAL CRACK FORM SIMULATIONS OF REINFORCED CONCRETE BEAM	
Studiepoeng: 30	
Emneord: Abaqus, CDP, concrete damage plasticity, crack width	Sidetall: + vedlegg/annet: Stavanger, 15.06.18 dato/år

Preface

This thesis concludes my Master's degree in Constructions and Materials, with specialisation in Civil engineering structures at the University of Stavanger. I want to thank my thesis advisors Mudiyansele Samindi Manjula K Samarakoon and Samdar Kakay for all the good advices during the writing process, and Daniel Bårdsen for the 2d shell model and good help.

Abstract

In this thesis, the Concrete Damage Plasticity module in Abaqus is used to determine crack propagation and crack width with a 3d solid model, and a 2d shell model.

Data is given by previous experiments where 6 concrete beams, casted with 3 different mixtures are tested after either 14 days or 28 days. These were loaded into a 4 point bend test, and loaded until failure. Crack widths were measured visually and by camera during the loading, remains the target for this thesis.

From this dataset, beam 6 was selected, a B35 beam which was tested after 28days.

Along with the test data, material data had been collected by samples casted from the same concrete mixture. These data were used as material input for the Abaqus model.

In addition, material input were extracted from literature.

Results are evaluated for the 3d solid model, and a the 2d shell model.

Appendix

PREFACE	1
ABSTRACT	1
NUMERICAL MODELLING OF REINFORCED CONCRETE BEAMS	6
DESIGN GUIDELINES FOR CONCRETE STRUCTURES	6
THEORY MODELS FOR PLASTICITY OF CONCRETE :	8
ABAQUS CONCRETE DAMAGE PLASTICITY DEFINITIONS AND PARAMETERS.....	13
CONCRETE DAMAGED PLASTICITY MODELS.....	16
GENERAL – FEM:.....	20
ABAQUS :.....	21
<i>Newton-Raphson method</i>	21
<i>Modified Newton-Raphson method</i>	22
<i>Initial stiffness method</i>	22
<i>Quasi-Newton</i>	23
<i>Arch length method</i>	23
<i>Convergence of Abaqus Solvers</i>	24
<i>Time increments for solvers</i>	24
<i>Limitations of the Abaqus Solvers</i>	24
FEM ELEMENTS	25
FLOWCHART FOR 3D SOLID RC BEAM MODEL IN ABAQUS.	29
FLOWCHART FOR 2D SHELL RC BEAM MODEL IN ABAQUS.....	31
TESTING OF REINFORCED CONCRETE BEAMS	32
OVERVIEW :	32
<i>Analysis of test beam, geometry:</i>	32
<i>Strain Gauges:</i>	36
REINFORCEMENT:	37
COMPRESSIVE TESTING.....	38
TENSILE TESTING	40
MODELLING OF REINFORCED CONCRETE BEAMS USING ABAQUS	41
<i>Reinforcement</i>	41
<i>2d shell element model</i>	42
<i>3d models, solid model</i>	43
CALIBRATION OF ABAQUS MODELS.	46
<i>Mesh size</i>	50
OVERVIEW OF SETUP FOR TESTING	54
PARAMETERS IN CDP	55
<i>Overview</i>	55
<i>Dilation angle:</i>	56
<i>Example Shear failure</i>	59
<i>Eccentricity:</i>	62
<i>K parameter:</i>	63
<i>Biaxial parameter:</i>	65
<i>Viscosity parameter :</i>	65
MATERIALS:	69
MATERIAL ADJUSTMENT:	72
CRACK WIDTH:	77
DISCUSSION OF RESULTS	85
RESULTS :	85
CONCLUSION AND FUTURE RESEARCH	87
SUGGESTIONS FOR FUTURE WORK	87
REFERENCES	88

APPENDIX :	90
MATERIALS :	90
<i>Concrete CDP recipes:</i>	<i>90</i>
<i>Material reinforcement input:</i>	<i>95</i>

Figures

FIGURE 1.....	7
FIGURE 2.....	7
FIGURE 3.....	8
FIGURE 4.....	8
FIGURE 5.....	8
FIGURE 6.....	8
FIGURE 7.....	9
FIGURE 8.....	9
FIGURE 9.....	9
FIGURE 10.....	9
FIGURE 11.....	10
FIGURE 12.....	10
FIGURE 13.....	11
FIGURE 14.....	11
FIGURE 15.....	12
FIGURE 16.....	12
FIGURE 17.....	14
FIGURE 18.....	15
FIGURE 19.....	16
FIGURE 20.....	17
FIGURE 21.....	19
FIGURE 24.....	21
FIGURE 25.....	22
FIGURE 26.....	22
FIGURE 27.....	23
FIGURE 28.....	24
FIGURE 29.....	25
FIGURE 30.....	26
FIGURE 31.....	26
FIGURE 32.....	26
FIGURE 33.....	27
FIGURE 34.....	28
FIGURE 35.....	32
FIGURE 36.....	33
FIGURE 37.....	33
FIGURE 38.....	35
FIGURE 39.....	36
FIGURE 40.....	36
FIGURE 41.....	37
FIGURE 42.....	37
FIGURE 43.....	38
FIGURE 44.....	39
FIGURE 45.....	39
FIGURE 46.....	39
FIGURE 47.....	41
FIGURE 48.....	42
FIGURE 49.....	42
FIGURE 50.....	43
FIGURE 51.....	43
FIGURE 52.....	44
FIGURE 53.....	44
FIGURE 54.....	45

FIGURE 55.....	45
FIGURE 56.....	48
FIGURE 57.....	48
FIGURE 58.....	49
FIGURE 59.....	49
FIGURE 60.....	50
FIGURE 61.....	51
FIGURE 62.....	51
FIGURE 63.....	52
FIGURE 64.....	53
FIGURE 65.....	54
FIGURE 66.....	55
FIGURE 67.....	56
FIGURE 68.....	56
FIGURE 69.....	56
FIGURE 70.....	57
FIGURE 71.....	57
FIGURE 72.....	58
FIGURE 73.....	58
FIGURE 74.....	59
FIGURE 75.....	59
FIGURE 76.....	60
FIGURE 77.....	61
FIGURE 78.....	62
FIGURE 79.....	62
FIGURE 80.....	63
FIGURE 81.....	64
FIGURE 82.....	64
FIGURE 83.....	65
FIGURE 84.....	66
FIGURE 85.....	67
FIGURE 86.....	67
FIGURE 87.....	69
FIGURE 88.....	70
FIGURE 89.....	70
FIGURE 90.....	71
FIGURE 91.....	71
FIGURE 92.....	71
FIGURE 93.....	71
FIGURE 94.....	72
FIGURE 95.....	73
FIGURE 96.....	74
FIGURE 97.....	74
FIGURE 98.....	75
FIGURE 99.....	75
FIGURE 100.....	76
FIGURE 101.....	77
FIGURE 102.....	78
FIGURE 103.....	78
FIGURE 104.....	79
FIGURE 105.....	80
FIGURE 106.....	81
FIGURE 107.....	81
FIGURE 108.....	81
FIGURE 109.....	82
FIGURE 107.....	86
FIGURE 108.....	86
FIGURE 106.....	86

Introduction.

In this thesis, the Concrete Damage Plasticity module in the Abaqus software was used.

A full scale reinforced concrete beam had previously been tested until failure under a 4 point bend setup, and data were recorded. Special about this test, was that it had casted concrete sample for testing which were tested, and in addition to the load deflection data, crack widths had been carefully monitored and recorded both visually and with a special camera.

That made this data-set a perfect opportunity to see if an Abaqus CDP modelled beam, with the material input form the dataset, could be able through extraction of stress and strain to be used in the Eurocode 2 crack width formula, and get the results from the actual testing. That is the main purpose of this thesis.

The first chapter of this thesis will explain the necessary background to better understand the plasticity concept for concrete, and understanding the CDP module.

A testing section for the CDP parameters is conducted, to better understand its limitations.

As the model will rely heavily on the reinforcement, the reinforcement bars used for the original concrete beam where tested in this thesis, to have good input material steel data.

In addition, two different models where constructed in Abaqus. On 2d shell model, and one 3d solid continuum element model, both with the CDP module. These modules would be used in parallel, to directly compare them to each other and assess results.

A material section is also included to better understand how they affect the models, and concrete damage plasticity material inputs form literature where taken in as comparison.

Results will be presented, and compared towards the original test data.

Finally we conclude and give recommendation for future work.

Numerical modelling of reinforced concrete beams

The reference for the concrete plasticity theories presented in this chapter is the book “Plasticity in Reinforced Concrete” by Wai-Fah Chen (2007). For practical purposes, I will not refer to the book every time a new theory is presented, but the reader should be aware that this is the source of information.

The Concrete Damage Plasticity (CDP) is the most suitable material modelling technique for reinforced concrete structural members. This thesis uses CDP material model in Abaqus software for modelling reinforced concrete beams, with the aim of determining crack propagation and crack widths. The focus of the thesis, will be on the analyses and results, and comparison to actual lab test results.

This chapter will provide a brief overview of relevant governing regulations, introduction to FEM theory, and present theory relevant for the Abaqus utilization. Additional explanations to the CDP module will be presented. As the aim of this thesis is to use FEM modelling to examine the non-linear plasticity region, the theory section will reflect this, and general derivations will therefore be kept to a minimum.

Design guidelines for concrete structures

The Eurocode Part 2, NS-EN 1992-1-1:2004 (EC2), is the governing standard for concrete design work. EC2 table 3.1 defines the standard strength classes and overview of their class parameter values and limitations. Eurocode Part 2 (EC2) defines all use of reinforced concrete, as from material, to design ULS/SLS and ULS, to execution.

Non-linear behaviour of concrete in compression

EC2 chapter 3.1.5, is the governing chapter for non-linear concrete. It states how the EC2 standard defines the Stress-Strain relation for compressive concrete, under uniaxial loading, as defined in 3.1.5 part 1. This formula uses material input from table 3.1 in the EC2 standard.

$$\frac{\sigma_c}{f_{cm}} = \frac{k \cdot \eta - (\eta^2)}{1 + (k - 2) \cdot \eta} \quad k = 1.05 E_{cm} \cdot \left(\frac{|\varepsilon_{c1}|}{f_{cm}} \right) \quad \eta = \frac{\varepsilon_0}{\varepsilon_{c1}} \quad 0 < |\varepsilon_c| < |\varepsilon_{c1}|$$

This stress strain relation is stated to be valid for short term, one axial bending loading, and is used to create CDP material properties for type [B35-EC2], up to point ε_{c1} .

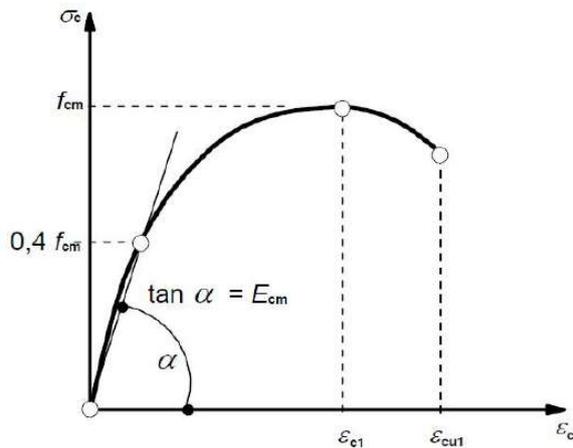


Figure 1

Stress strain relation in Eurocode 2 - compression

3.1.5 part 2, states that EC2 is open to usage of other stress strain relations if they give a similar behaviour in the concrete. This point is important, because it basically opens up for usage of other stress strain relations, as the CDP (Concrete Damage Plasticity) module in Abaqus for EC2 design work.

Stress vs strain behaviour of steel reinforcement

EC2 chapter 3.2 is the governing chapter for reinforcement. This includes rebars and prestressed reinforced concrete. As for this thesis, only B500NC material type rebars are used, which suits all criterias in 3.2.2.

In chapter 3.2.4, the general connection between stress and strain is given, where the definition of the yield point can be seen in Figure 2, and defined as 0.2 % strain.

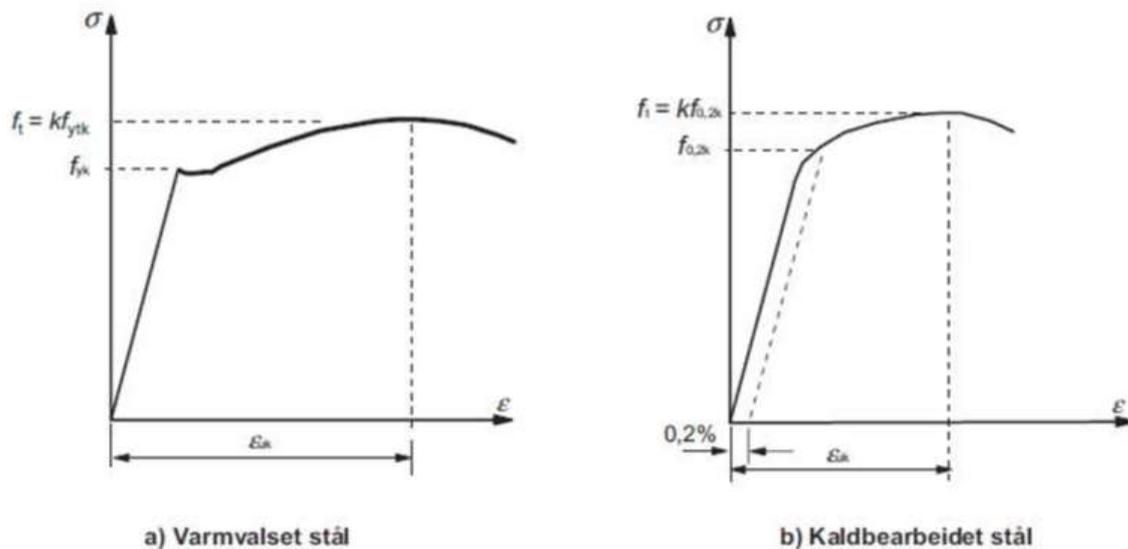


Figure 2

Stress strain diagram of steel

In this thesis, all reinforcement rebars where tensile tested. All passed the B500NC material standard. $\varnothing 16\text{mm}$ reinforcement bars where hot rolled, as the rest had the characteristics of a cold rolled steel.

Theory models for Plasticity of concrete :

As many of these formulas was originally based on rocks and solid, since concrete under large hydrostatical pressure behaves plastically, and non-linearly.

In order to describe plasticity in general, there are 4 points that need to be defined:

- 1) Elastic strain through Hooke's law
- 2) Criterion for yield
- 3) Work hardening rule
- 4) Flow rule, to describe the evolution of plastic strain

Regarding point 1, all yield theories are in accordance with Hooke's law. As for the yield criteria (point 2), many different theories have been described. The background for all these models are Mohr's definition of maxi- and minimum stresses, principle stresses and principle planes, and von Mises's and Tresca's two parameter plane stress models, describing the failure yield criteria. The problem at hand, is that one wants to address the yield criteria for concrete in ductile state under high hydrostatic pressure, and define a fracture criteria in concrete for brittle fractures under small stresses. First yield criteria model was by normal stress theory, that simply made a fail criteria, as the principle stress in each direction, making a cube in the plane view. This was suitable for brittle materials only.

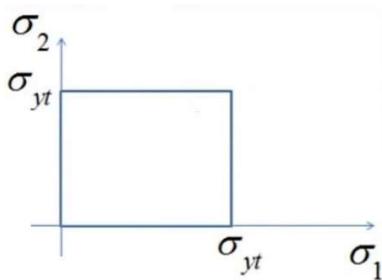


Figure 4

Mohr-Coulomb

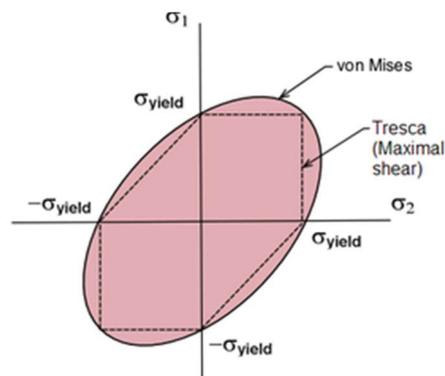


Figure 3

von Mises and Tresca plane stress

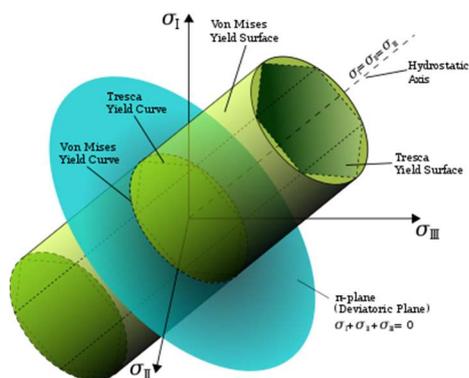


Figure 5

von Mises and Tresca plane stress

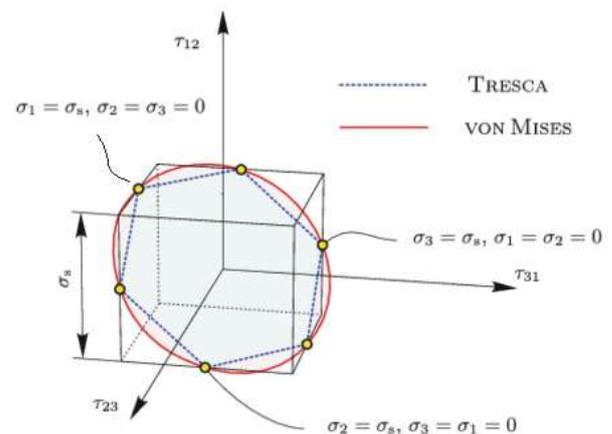


Figure 6

Represented in principal-stress space

Tresca took another approach, and introduced max shear stress theory, that if any of the principle stresses, or the difference between them reached the yield criteria, the material would fail. This gives it the characteristic hexagon shape of the yield limit, utilizing the corners from the mohr-columb model, which worked for ductile materials too.

von Mises introduced the distortion energy theory, which is also a shear stress theory, where he through the E-module and the Poisson-ratio converted through formula the principle stresses in to strain energy of distortion per unit volume. The same process is done with the material defined yield stress, so that the yield criteria could be drawn as a tilted ellipse, following about the same shape as Trescas hexagon for the plane stress state. This was a big improvement as it now was represented as a smooth failure surface, in contrast to Tresca. Tresca's solution suffered from the problem that there is mathematically no normal vector in the middle point of a sharp corner for numerical modelling.

For the use of Mohr-Columb model, one must know in advance what all the 6 sides represent, and it still has the numerical problems with the corners. The Drucker-Prager model as an extension of von Mises is an improvement, but it still gives a linear representation, while experiments have shown a curved tendency for the meridians. The Drucker-Prager- model is very dependent on the β -envelope angle.

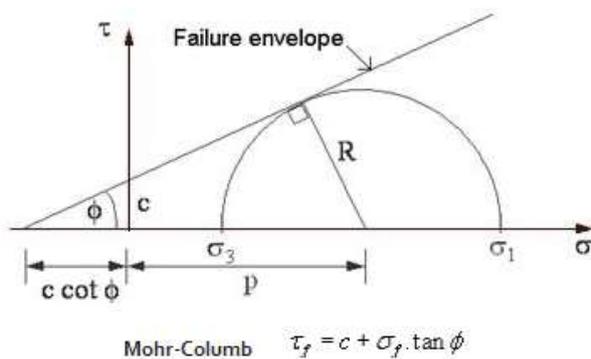


Figure 8

Mohr-Columb meridian

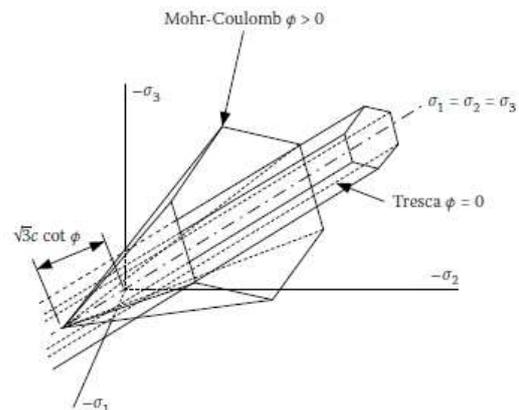
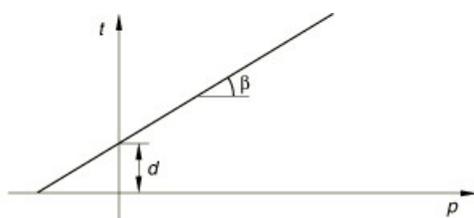


Figure 7

Principal-stress space



a) Linear Drucker-Prager: $F = t - p \tan \beta - d = 0$

Figure 10

Drucker-Prager meridian

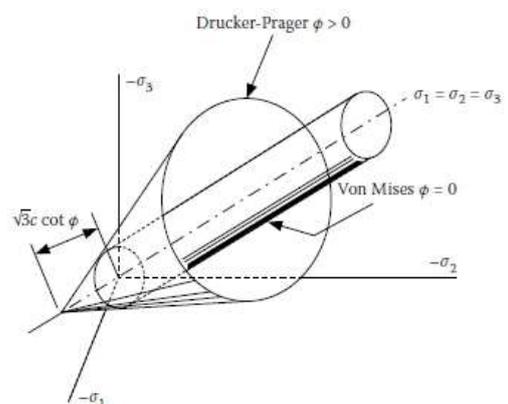


Figure 9

Principal-stress space

Several of the original models here presented have been modified later by others, to better trail fit the same old data sets. Of those modified versions, the Ottosen model and the Willem-Warnke are the most recognized, and have later been even further modified. Their practical usefulness often differs, depending on their complexity and how many parameters that must be known before they can be applied.

Figure 12 shows an overview of how the three different models reflect test data.

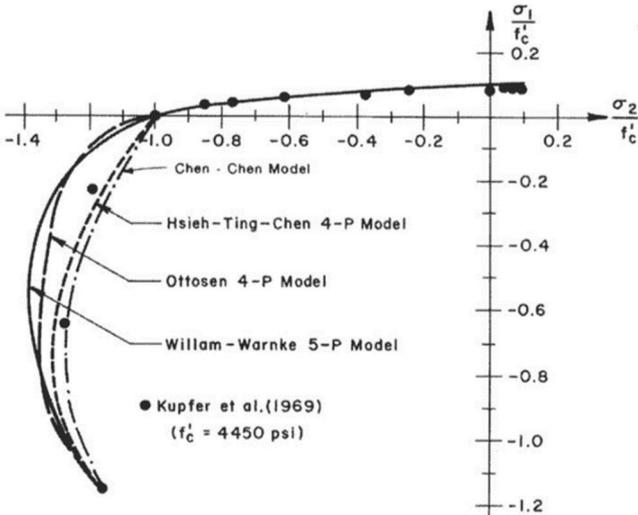


Figure 11
Comparing models, plane principle stress

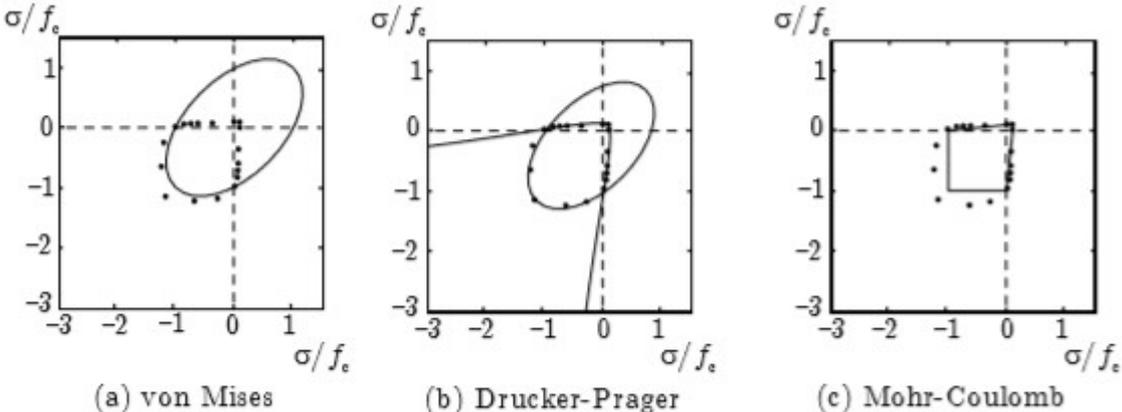


Figure 12
Comparing models, plane principle stress

The third point describes that concrete plasticity is hardening. Both the Mohr-Columb and Drucker-Prager model, later updated by William and Warnke who added elliptical sides to the model and gave the failure surface an almost triangular shape at low stress, as well as a complete circle at high hydrostatic pressure if seen in the deviatoric plane, fails to account for strain hardening.

Hardening, or strain hardening, can either be added as isotropic hardening, or kinematic hardening. With isotropic hardening, this is simply viewed as added rings in all directions, with the outer surface as the failure surface.

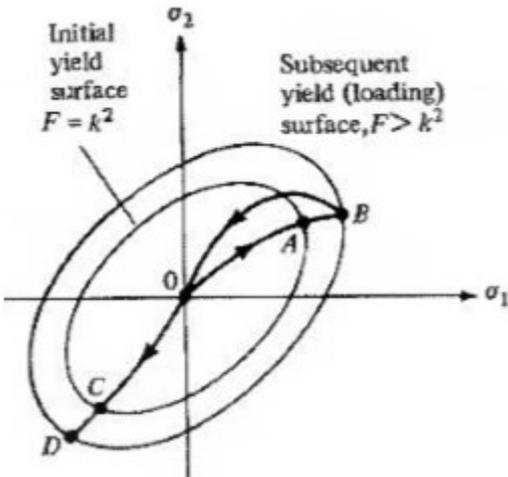


Figure 13
Isotropic Hardening in plane stress

With kinematic hardening, the material is loaded through a cyclic load back to its original state where the start point has now shifted. If repeated, this will look like lots of rings, representing the different hardening curves dependent on numbers of cycles and plastic deformation.

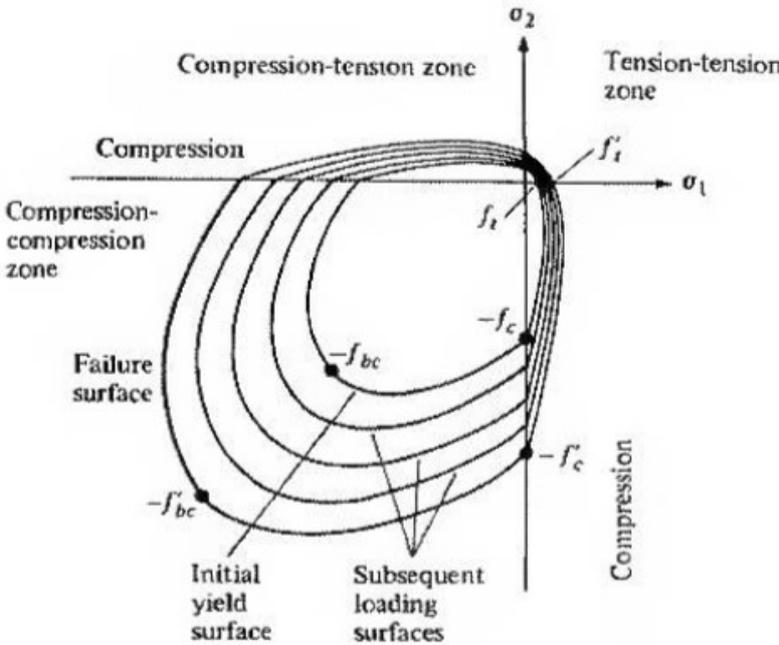


Figure 14
Hardening in plane stress

Figure 14 shows kinematic hardening where the yield surface is expanded into failure surface.

Basic concept of plane stress with hardening

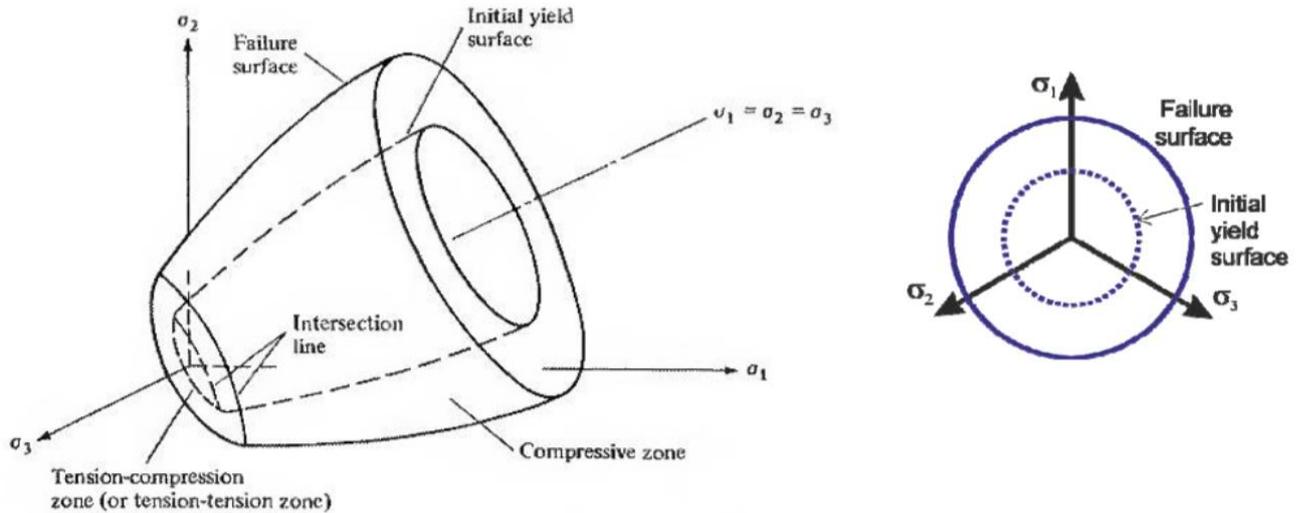


Figure 15

Yield surface and Failure surface defined in principal-stress space

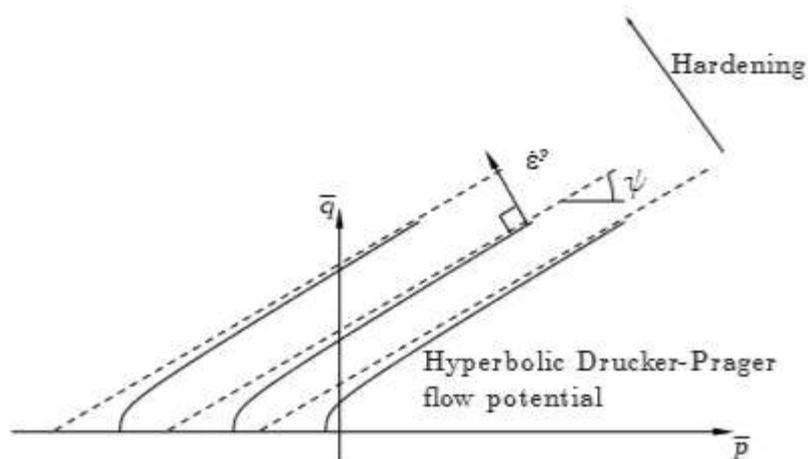


Figure 16

Drucker-Prager hardening

The hardening effects differ between FEM programs, due to the choice of concept and as they rely on different modified versions on theories. This is true for the flow rule as well, as the FEM software developers have made their own adjustments to fit their particular problem type or material range. These new model types define stresses into effective stresses, so simple uniaxial stresses can be extrapolated into the multidimensional state. The models are under constant modifications, and are still not yet perfected.

When the fib 2010 code was made, an effort was made to summarize models and make an easier concrete plasticity for FEM modelling, but it proved to be too complex. Instead, the code gave a recommendation of three generally accepted FEM approaches, where one of three listed was the strain based isotropic damage approach, which is the approach the CDP module in Abaqus builds on.

Abaqus concrete damage plasticity definitions and parameters

In the Abaqus manual, it is stated that the Abaqus CDP-module uses a variant of the Drucker-Prager, which has been modified by Lubliner et. al. (1989), and with additional modifications proposed by Lee and Fenves (1998).

For the CDP module, hardening is defined through stress inelastic strain input, and damage parameters. Abaqus first defines effective stress invariants, hydrostatic pressure, and von Mises equivalent effective stress, to define a flow potential in a Q-P plane. These are also used for the yield criteria.

The plastic flow potential function;

$$G = \sqrt{(\epsilon\sigma_{t0} \tan \psi)^2 + \bar{q}^2} - \bar{p} \tan \psi,$$

Where :

$$\bar{p} = -\frac{1}{3}\text{trace}(\bar{\sigma}), \quad \text{Hydrostatic pressure, with}$$

$$\bar{\sigma} = \mathbf{D}_0^{el} : (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^{pl}). \quad \text{Effective stress as}$$

$$\bar{q} = \sqrt{\frac{3}{2}(\bar{\mathbf{S}} : \bar{\mathbf{S}})}, \quad \text{Von Mises equivalent effective stress, with}$$

$$\bar{\mathbf{S}} = \bar{\boldsymbol{\sigma}} + \bar{p}\mathbf{I}. \quad \text{Effective stress deviator, where I is the first stress invariance.}$$

Definitions for the *flow potential function*, which is set by user :

here ψ is defined as the dilation angle measured in the p-q plane at high confining pressure

and σ_{t0} as the user defined uniaxial tensile stress at failure.

And ϵ is the eccentricity parameter, defining the rate at which the function approaches the defined dilatation angle asymptote.

The yield criteria function;

$$F = \frac{1}{1 - \alpha} \left(\bar{q} - 3\alpha\bar{p} + \beta(\tilde{\epsilon}^{pl}) \langle \hat{\sigma}_{\max} \rangle - \gamma \langle -\hat{\sigma}_{\max} \rangle \right) - \bar{\sigma}_c(\tilde{\epsilon}_c^{pl}) = 0,$$

Where;

$$\alpha = \frac{(\sigma_{b0}/\sigma_{c0}) - 1}{2(\sigma_{b0}/\sigma_{c0}) - 1}; \quad 0 \leq \alpha \leq 0.5,$$

$$\beta = \frac{\bar{\sigma}_c(\tilde{\epsilon}_c^{pl})}{\bar{\sigma}_t(\tilde{\epsilon}_t^{pl})} (1 - \alpha) - (1 + \alpha),$$

$$\gamma = \frac{3(1 - K_c)}{2K_c - 1}.$$

$\hat{\sigma}_{\max}$ The maximum effective principal stress

$\bar{\sigma}_t(\tilde{\epsilon}_t^{pl})$ The effective cohesion tensile stress.

$\bar{\sigma}_c(\tilde{\epsilon}_c^{pl})$ The effective cohesion compressive stress.

Definitions for the *yield criteria function*, which is set by user :

K_c The ratio of; Second stress invariant on the tensile meridian - to the one on the compressive meridian - at initial yield for any given value of the pressure invariant \bar{p} , such that the maximum principal stress is negative. $0.5 < K_c \leq 1.0$

σ_{b0}/σ_{c0} The ratio of; Initial equi-biaxial - to uniaxial , - compressive yield stress.

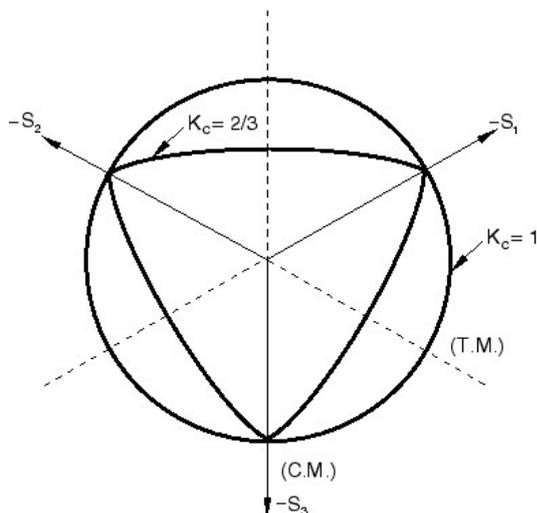


Figure 17

Yield surface deviatoric plane

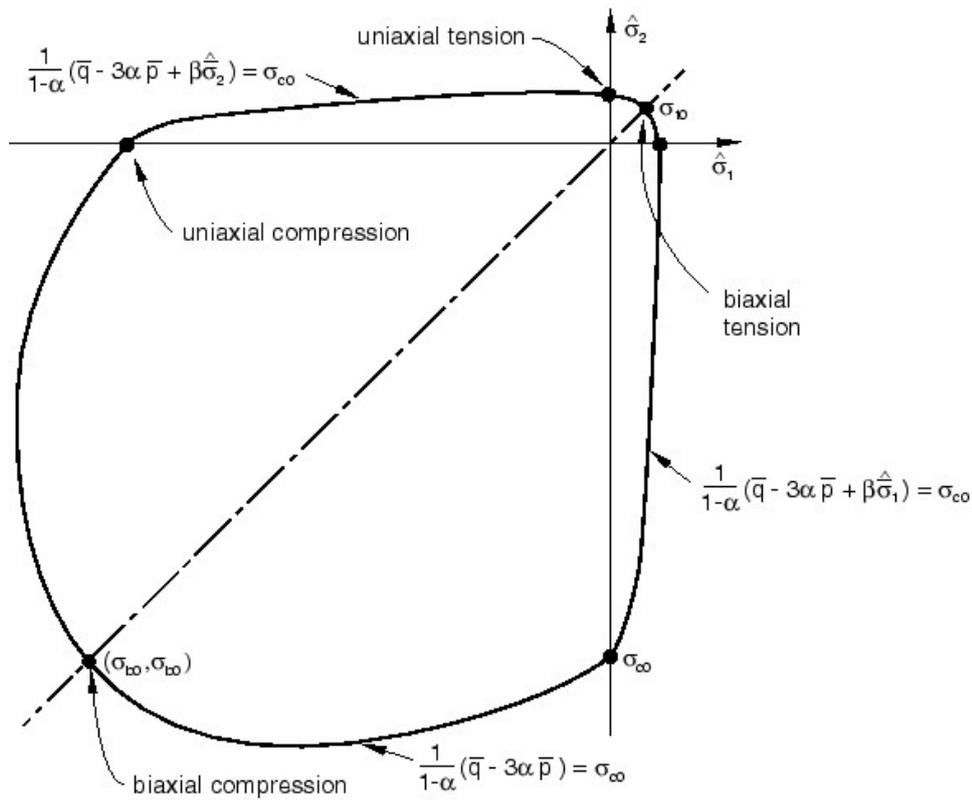


Figure 18
Yield surface plane stress

User defined CDP input parameters.

As seen by the definitions, the dilation angle, eccentricity and K value are all interlinked.

Dilatation angle ψ - Default value is 31.

Eccentricity parameter ϵ - Default value is 0.1

Biaxial to uniaxial ratio σ_{bo}/σ_{co} - Default value 1.16

K_c Parameter - Default value 2/3

Viscosity parameter - Set by user, default is 0.

Concrete Damaged Plasticity models

In Abaqus, materials are defined through modules, where the elastic module defines the Poisson’s ratio and the Emodulus, and a plasticity module is needed for the non-linear behaviour.

When defining a plasticity for a concrete material, CDP is one of Abaqus options available to define how the concrete will behave through the plastic region.

In short, CDP is designed and based on that compression and tensile stress-inelastic strain, are defined, and in addition, one gives information of when the concrete has taken damage, in a damage-strain relation, where 0 is non-damage taken, and 1 is complete loss of capacity.

It is worth noting that if the CDP module is used without reinforcement, it is highly mesh dependent, as it with reinforcement is quite resilient to mesh dependency.

If damage parameters are not set, the model will behave as a plasticity model;

$$\tilde{\epsilon}_c^{pl} = \tilde{\epsilon}_c^{in} \quad \text{and} \quad \tilde{\epsilon}_t^{pl} = \tilde{\epsilon}_t^{ck}$$

Defining concrete in Compression:

The hardening data for concrete in compression, is given by a stress – inelastic strain relation.

The elastic strain, is given by the relation from undamaged concrete, as $\epsilon_{0c}^{el} = \sigma_c / E_0$.

The inelastic crushing strain, is defined as Total strain – Elastic strain, $\tilde{\epsilon}_c^{in} = \epsilon_c - \epsilon_{0c}^{el}$.

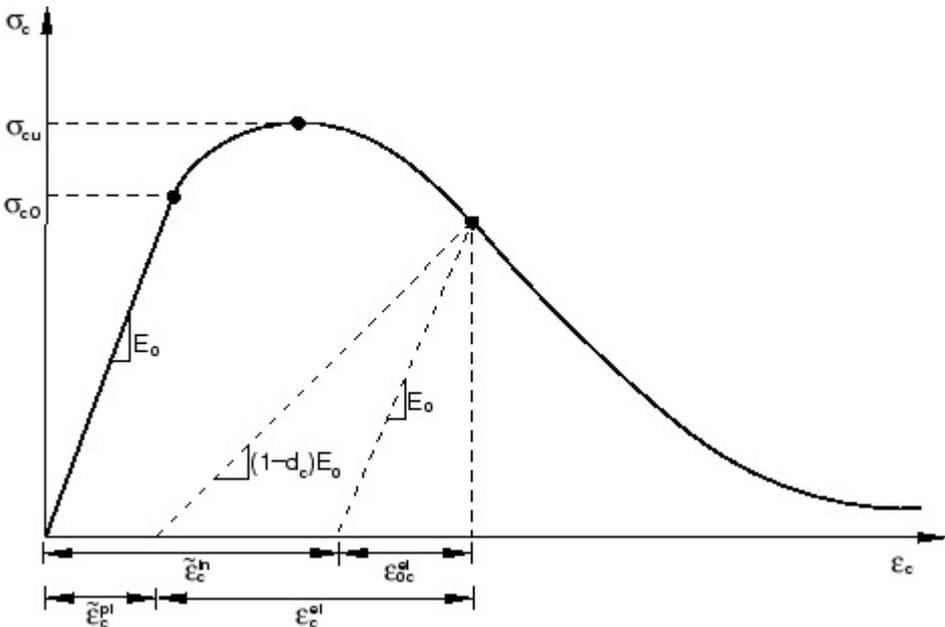


Figure 19
Abaqus manual 2016

Defining concrete in Tension:

The post failure behavior in CDP is modelled with tension stiffening, where the user has to define the strain softening given as a stress – inelastic or cracking strain relation.

The elastic strain, is given by the relation from undamaged concrete, as $\varepsilon_{0t}^{el} = \sigma_t / E_0$.

The inelastic cracking strain, is defined as Total strain – Elastic strain, $\tilde{\varepsilon}_t^{ck} = \varepsilon_t - \varepsilon_{0t}^{el}$.

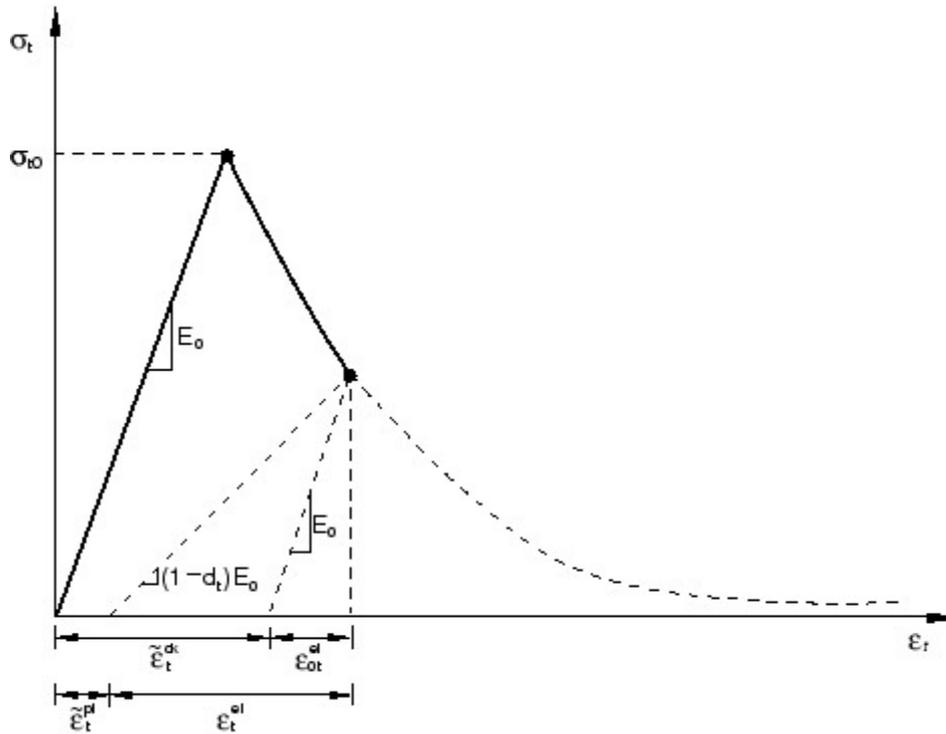


Figure 20

Abaqus manual 6.14

To avoid potential numerical problems, Abaqus enforces a lower limit on the post-failure stress equal to one-hundredth of the initial failure stress: $\sigma_t \geq \sigma_{t0} / 100$.

Further the manual states that for heavily reinforced concrete in a fairly detailed mesh, a reasonable starting point is to assume a linear strain softening form the failure stress down to 0, at a strain rate of 1:10 of the strain at failure. This suggest a typical strain failure in the region of 10^{-4} , and the 0 stress-value in the strain region of 10^{-3} . It also states that this parameter should be calibrated to any particular case.

The choice of tension stiffening has a great deal of effect on the model.

The manual states, that a stiffer model often makes it easier to converge.

Determine the damage parameters

In addition to defining the compressive stress-inelastic strain, and tensile stress-inelastic strain curves, additional input must be set for both compression and tensile;

Compressive damage parameter – inelastic strain

Tensile damage parameter – inelastic strain

The compressive and tensile damage parameter, d_t and d_c , they are given values from 0, undamaged, to 1, which the material has lost all capacity.

$$0 \leq d_t \leq 1,$$

$$0 \leq d_c \leq 1.$$

When E_0 is the undamaged elastic stiffness, the damage parameters, d_t and d_c , stress strain relation is defined as follows :

$$\sigma_t = (1 - d_t)E_0(\varepsilon_t - \tilde{\varepsilon}_t^{pl}),$$

$$\sigma_c = (1 - d_c)E_0(\varepsilon_c - \tilde{\varepsilon}_c^{pl}).$$

Where the defined effective tensile and compressive stresses are defined as

$$\bar{\sigma}_t = \frac{\sigma_t}{(1 - d_t)} = E_0(\varepsilon_t - \tilde{\varepsilon}_t^{pl}),$$

$$\bar{\sigma}_c = \frac{\sigma_c}{(1 - d_c)} = E_0(\varepsilon_c - \tilde{\varepsilon}_c^{pl}).$$

For the compressive damage, input are d_c to inelastic strain, where Abaqus will automatically update the inelastic strains to plastic strains using this formula

$$\tilde{\varepsilon}_c^{pl} = \tilde{\varepsilon}_c^{in} - \frac{d_c}{(1 - d_c)} \frac{\sigma_c}{E_0}.$$

For the tensile damage, input are d_t to inelastic strain, where Abaqus will automatically update the inelastic strains to plastic strains using this formula

$$\tilde{\varepsilon}_t^{pl} = \tilde{\varepsilon}_t^{ck} - \frac{d_t}{(1 - d_t)} \frac{\sigma_t}{E_0}.$$

For each increment, the damage parameter is interpolated between the point values given.

$$d_t|_{t+\Delta t} = \max \left\{ d_t|_t, d_t(\tilde{\varepsilon}_t^{pl}|_{t+\Delta t}, \theta|_{t+\Delta t}, f_i|_{t+\Delta t}) \right\},$$

$$d_c|_{t+\Delta t} = \max \left\{ d_c|_t, d_c(\tilde{\varepsilon}_c^{pl}|_{t+\Delta t}, \theta|_{t+\Delta t}, f_i|_{t+\Delta t}) \right\}.$$

The CDP module is designed for used with uniaxial loading, and to account for the stiffness recovery during cyclic loading.

The figure below, explains how the stiffness recovery is affected by the damage parameter in CDP module through a stress inelastic strain diagram, showing a Tension-> Compression -> Tension load, into the plastic region on both sides.

The assumption assuming recovery factors $w_t = 0$ and $w_c = 1$

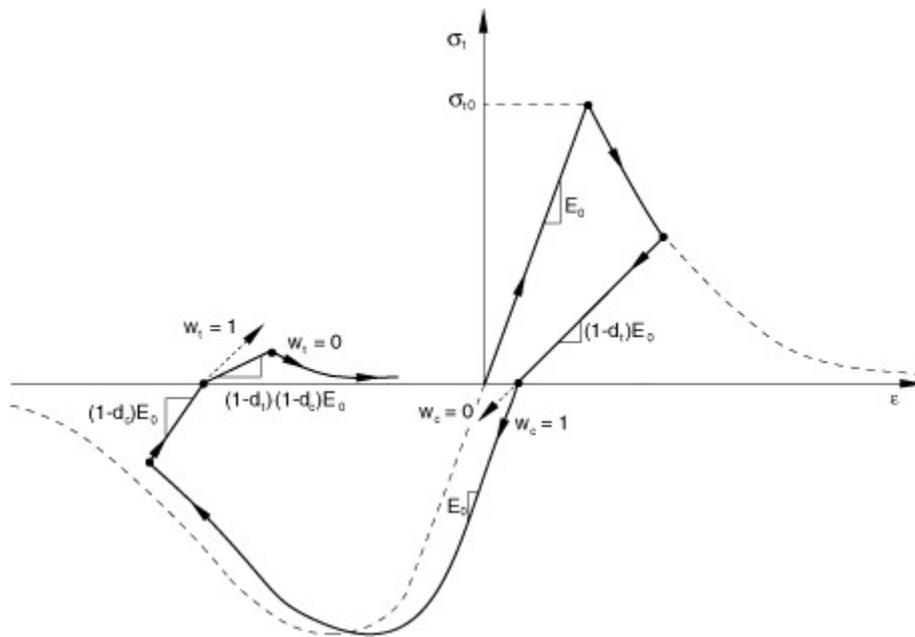


Figure 21

Cyclic load = Tension-> Compression -> Tension

Possible limitation of CDP

One limitation which I have not tested, is that the Abaqus CDP model is a strain-based isotropic Damage model, and according to (Burgers, 2006) is not able to predict a model of a tensile splitting test correctly. Burger suggests that it has to do with the way it models cracks, as it will result in a loss of E-modules in every direction, instead of the crack propagation direction only. It should be mentioned that this was done with an old version (2006) of Abaqus and CDP module, and I would be surprised if this is a case with today's newer versions.

For this thesis, this possible limitation will not play a role, as the beam is subjected to uniaxial loads.

General – FEM:

Finite Element Method (FEM), is a numerical technique that utilize Hooke's law, where the stiffness matrix, multiplied with the displacement matrix, will be equal to the reaction, or load matrix. (Cook, Malkus, Plesha & Witt, 2002)

In general:

$$[K] * \{ D \} = \{ R \}$$

This compatibility-equation must be satisfied for all nodes throughout the system. This can be done analytically and is sometimes referred to as the stiffness method, when the stiffness matrix is kept linear. Each small finite element is then solved with respect to the equation of inner and outer forces (principle of virtual displacement). By calculating through all the finite elements, a complete solution is found for the model. It can be applied to 2d- and 3d-models, where it can be able to represent large arbitrary shaped complex structures.

When utilizing materials that do not behave linearly (steel plasticity), the stiffness matrix too has to become a non-linear matrix. Non-linearity is not limited to material properties, as there can also be contact non-linearity and geometric non-linearity if the deformation is too large.

To satisfy the same equation of virtual displacement, the external load must be implemented in smaller steps, or increments. The stiffness matrix will then be subject to an adjustment process, where the stiffness will be adjusted to compensate for the new incremental load. This is repeated until either a wanted load is reached, or the model can no longer converge a solution through iterations.

Abaqus :

For this thesis, Abaqus Standard is used.

Methods used by solver:

The problem is static, and the model will deform in a non-linear way. For a static, non-linear problem, the Abaqus manual lists to standard solvers, General and RIKS. General solver can use New-Raphson method, modified New-Raphson method or quasi-Newton, as the RIKS solver uses the arch length method. Non-linear geometry behaviour is not taken into account in this thesis.

When solving non-linear behaviour, a direct analytical solution is not possible, as the stiffness matrix and sometime the load itself, can be a function of the displacement and deformation.

To solve these kind of problems, different methods use steps of iteration, to numerically solve the problem. For more detail explanation, see Vasios, N. (2015). *Nonlinear Analysis of Structures. The Arc Length Method: Formulation, Implementation and Applications.*

Newton-Raphson method

In Abaqus General solver, this method is referred to as Newton's method.

A load P is added, and displacement initially is 0. When displacement is 0 to begin with, an addition Δu is needed to keep the balance. A small increment is needed, where an additional displacement is added, which generates an load imbalance, e_{pa} , which is the $P.1$ load minus stiffness times the added displacement. The load $p.1$ is kept constant, as displacement and stiffness is undergoing iterations for each step. As iterations for each step is calculated, and the load imbalance will be reduced, and when the balance is 0, or pre-defined or close enough 0, the solution has converged.

This can be repeated for additional $P.2$ loads, giving convergence for the $P.2$ load on top of the already calculated $P.1$. Summarizing all the P loads with their solutions, to the original load is reached, hence the complete force / deflection curve is calculated.

It is often referred to as the full Newton-Raphson method.

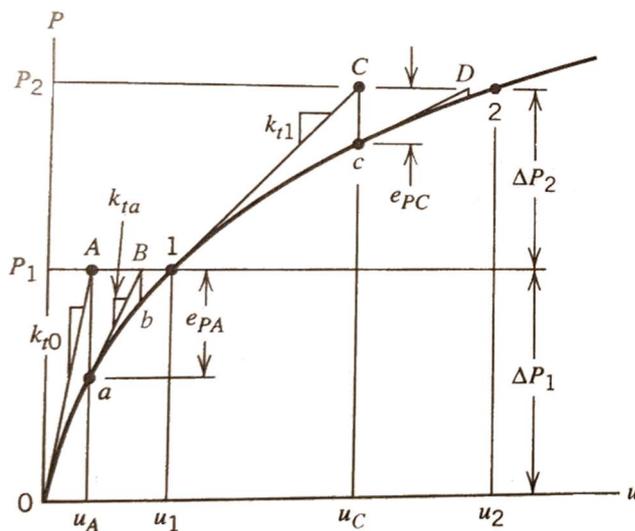


Figure 22

Newton-Raphson method

Force imbalance ; $e_{PA} = P.1 - k.t0 * u.a$

As seen on the figure, through iteration A and B , it has converged for the load P_1 . Additional P_2 load is added, and after new iteration C and D, it has now converged on load P_2 .

Basically, for the Newton-Raphson method, the sequence will continue until convergence is reached. Usually, only the incremental values are calculated and traced through the iterations, and when the final convergence is achieved, either at the final full load or the last incremental load, the last current state at convergence is set as a valid result.

The cost of using this method, is basically determined by the amount of change in the stiffness throughout the iteration process. Therefore, this method it is widely used, and many versions exist.

Modified Newton-Raphson method

As the full Newton-Raphson method updates a full k.t stiffness matrix for each increment, the modified version uses the same k.t0 tangent initial stiffness for all iterations until convergence for load P.1 is reached. At convergence, it uses the next k.t1 stiffness for all increments until convergence for load P.2 is reached, hence the effort of calculating each step is reduced, but the total convergence rate is slower.

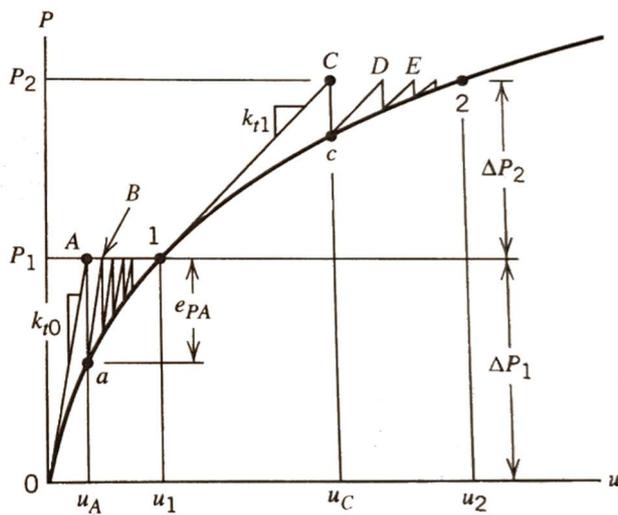


Figure 23

Modified Newton-Raphson method

Initial tangential stiffness is kept until the iterations converge. Then the process is repeated.

The effort of calculating each step is reduced, but the total convergence rate is slower.

Initial stiffness method

This method builds on Newton-Raphson method, but here only the initial stiffness matrix is kept constant all through the iterations, and all non-linear parts are repeatedly updated for deflection and reaction, until convergence is reached with the initial stiffness. This approach converges slowly.

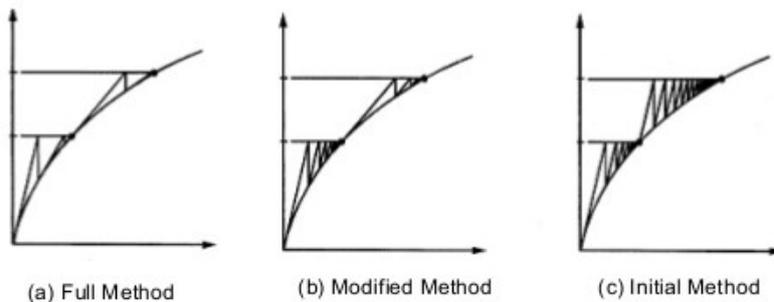


Figure 24

Quick summary of the different Newton methods

Convergence of Abaqus Solvers

Abaqus uses many different parameters for convergence, which is default all set and left at default setting. In general, these settings are regarded as being quite strict.

Time increments for solvers

For the General solver, for each step of increment, time is added. For a dynamic problem, this is the real time clock. For a static problem, time represents a percentage of the achieved final load, called Load Proportional Factor(LPF). Time is from 0s to 1s full load, and at LPF = 1, it has converged on the final target load.

For the RIKS solver, time is also here added for each step of increment, but it differs as loading time is defined from 0s to convergence. This is because the convergence goal of the increment sequences for the RISK analysis, is not the load as in Newton-Raphson methods, but the arch length convergence. Typically, the arch length method converges within a period of 1-5 seconds depending on the complexity, where the end time at convergence can be seen as the equivalent of the LPF =1s for the newton methods.

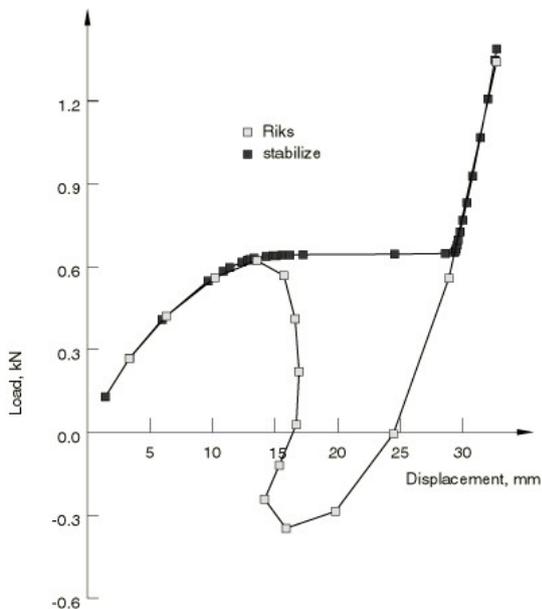


Figure 26

Snap-back curve, with RIKS and General solver

Limitations of the Abaqus Solvers

As seen in the figure above, the Newton methods cannot numerically calculate any load lower than the peak load at 12-13mm displacement, and follow the curve down again. Here the incremental additional stiffness is 0, until the load is equal to previous load, and an incremental load is added, and it can converge on this new deflection point.

In practical terms, this means that the Newton methods for a Beam, is valid up until UTS point, as the RIKS method can be used to go “over” the UTS top.

FEM Elements

Dassault System's Simulia Abaqus software, is a truly multipurpose FEM program, as it can be used to structural, static, dynamic, CFD, heat transferee, electrical problems, connection problems, large scale deformations, which means it has a substantial element library included.

For structural problems, continuum, shell elements, beam elements and truss elements are the usual structural elements

When constructing a model, a choice has been made to which type of elements it should use. The selection of elements inflicts the model's ability to produce accurate results, and greatly inflicts the time spent on calculations.

Abaqus element library

Elements :

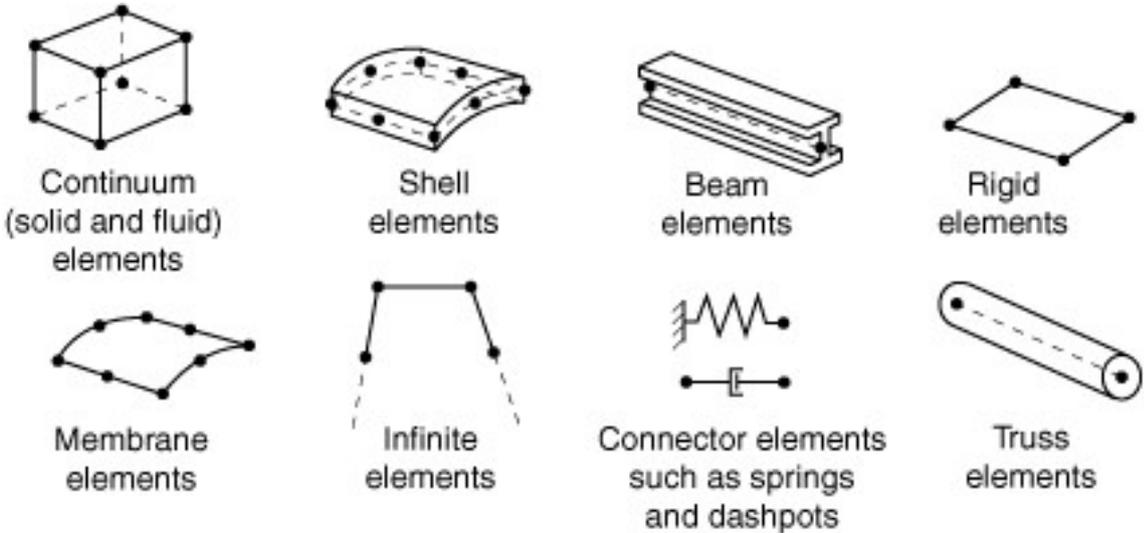


Figure 27

Standard FEM elements in Abaqus

Abaqus starts by all elements either in the Standard/Implicit category, or in the Explicit category. This choice of element types, defines if the model can run either implicit or explicit solvers, and the two categories, can not be mixed.

Abaqus standard elements, include first order elements with linear interpolation, or second order elements with quadratic interpolations, for both 1d, 2d and 3d elements.

The most common 1d element, is the truss element, as used for reinforcement for concrete beams.

For 2d problems, triangular and quadratic elements can be selected, as for 3d problems, hexahedra (bricks), wedge (Prism) or tetrahedron (pyramids) elements can be used.

Elements can also be combined, as for an RC beam, where a 1d truss element are embedded into a full 3d continuum element meshed part. The 1d element are given more dimensions by applying a profile to cover around the line along in it's length direction.

In the same way, a 3d element can be represented by 2d, shell elements. Shell elements for 3d are normally extruded from solid shapes, but can also be used in plain 2d models. Although shell elements are specified with a thickness, it only uses the geometry of the model they represent when calculating the elements nodes.

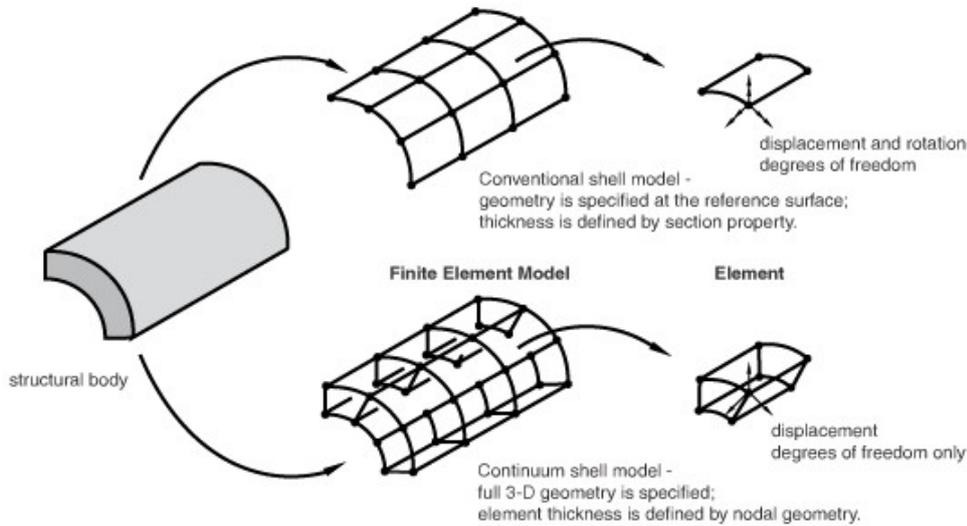


Figure 28
Abaqus manual 6.10

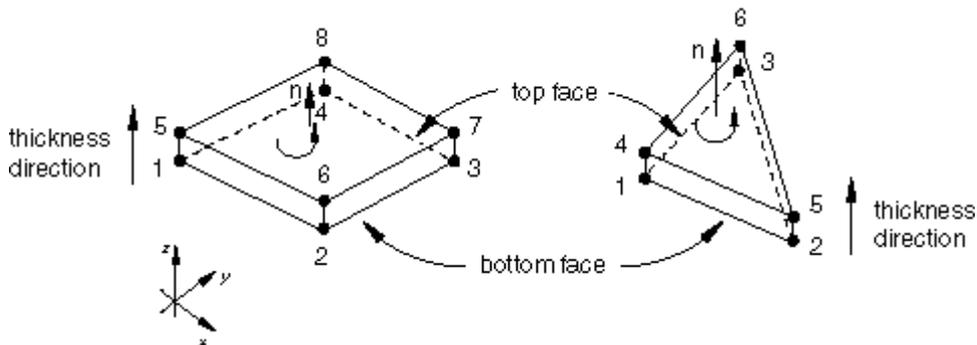


Figure 29

Shell elements

For continuum elements, Abaqus defines HEX- elements as a combined definition of both 8-node and 20-node brick elements, and TET- elements is the equivalent definition for tetrahedron or trapeze .

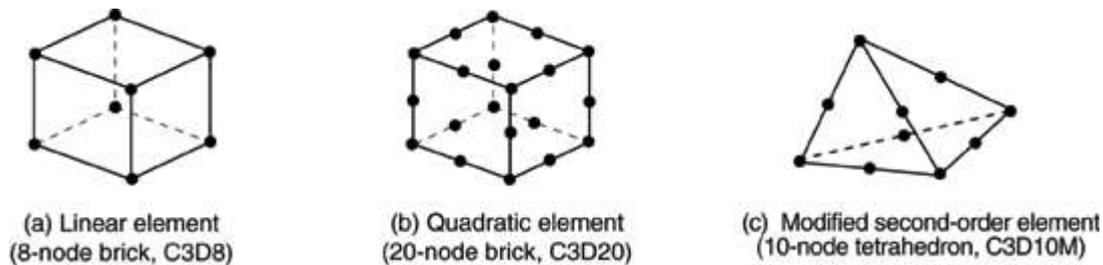


Figure 30

Solid elements

The Abaqus manual states that for “smooth” problems that do not involve large element distortion the second order elements give better accuracy than first order linear elements. It also recommends using second order elements for bending dominated problems, specifically if few elements are used in modelling the structural part’s thickness.

Triangular or tetrahedral elements are practical, as they can be used to mesh complex shapes, but a good mesh of Hex elements will often converge on a solution of equal accuracy faster. Hex elements perform best if they are kept close to a rectangular shape. This means that Hex elements are more sensitive on their initial shape and does not perform well if they get heavily distorted from their original shape, or is initially distorted.

Continuum elements in Abaqus in addition to the option of being linear and quadratic, can be opted to use full or reduced integration (labelled R) and hour glass control, which have a significant effect on the accuracy of the element for a given problem.

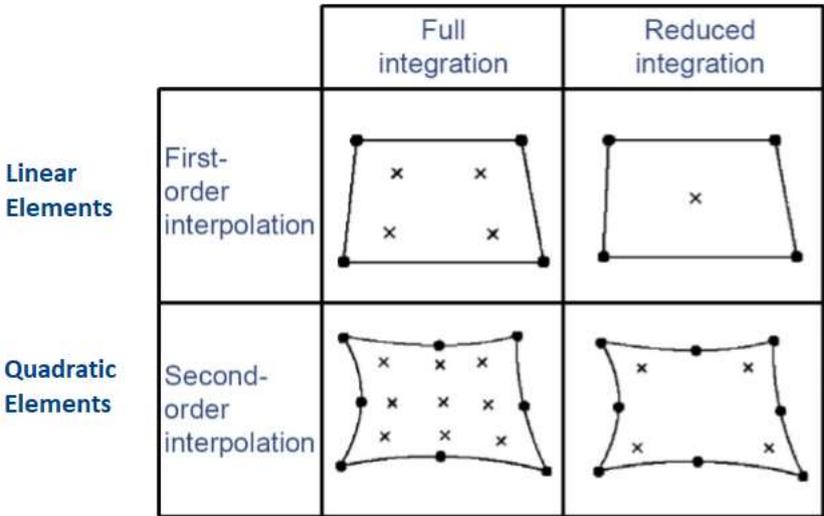


Figure 31
Full and reduced integral

This chart visualizes the difference, representing a 2d element, or one of the sides of a 3d element. The same difference is found in linear and quadratic triangular and tetrahedron elements.

As the stiffness and mass are found through the integration points within the element, an element with reduced integration will require less computational effort. As an example of the impact of number of interpolation points, Abaqus manual states that just from going from a full integrated 27node brick element, to the same element with reduced integration, the calculation cost is reduced by 3,5 times. As the sampling points dictates the shape functions, they also greatly influence how an element behaves in the model. (dof can be used for displacement and or rotations, heat transferee and similar). Reduced integration simply reduces interpolation points for the specific element, and hourglass control helps to prevent shear locking effects for linear elements.

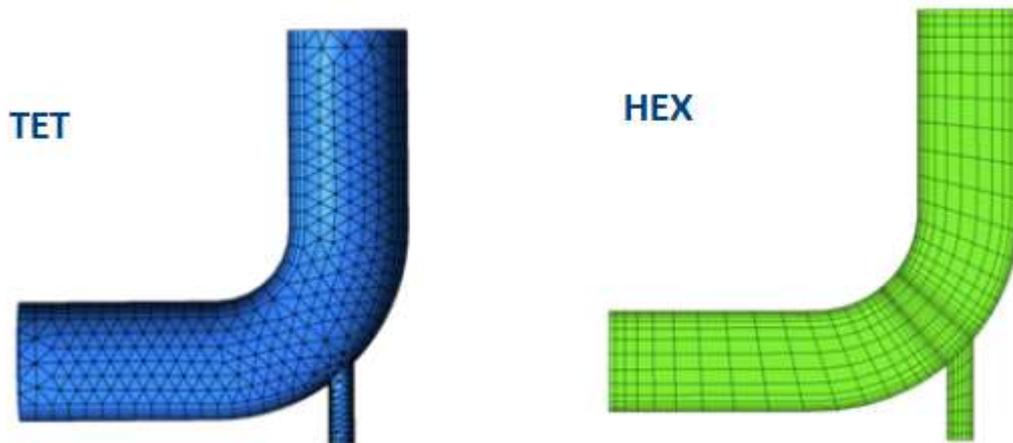


Figure 32

Tet and Hex elements

Figure 34 shows different meshing using Tet elements to the left, where tetrahedron are facing inwards along the curved sides, and to the right, only use of hex elements where both mesh types produced a good mesh.

Limitations

Linear elements does not have curved sides, and are in need of a fine mesh to produce accurate results. For complex shapes, the accuracy of the shape can quickly be not good enough. Number of elements used through the thickness also effects the efficiency. If a C3D8 element is used alone in pure bending, it can introduce shear locking, and hourglass shape, where the element numerically generates a shear strain which is no present in the real problem. But if 4 or more C3D8 are used over the thickness, this effect is reduced to almost be neglectable. A C3D20 element will not have any of these problems.

Note

The most important thing when choosing elements, either 2d or 3d, is that the elements corner nodes are always connected to other nodes. If a node is left "hanging", with no connection to other nodes inside the model, the model will produce incorrect results. This is particularly important when having two types of mesh, for example if going from a tet mesh to a hex mesh, or from a coarse hex mesh to a fine mesh, that all nodes have a connecting node, unless they are the very nodes on the edge of a model.

In Abaqus, partition lines can be defined, as they will guide the mesh generation to use these lines for node contact points for the elements used.

Flowchart for 3d solid RC beam model in Abaqus.

Make a choice about inputs. Here it is used mm and MPa, or N.

→ 10 MPa = 10

→ 1mm->1

1 – Create Parts

Solid 3d; Part as the concrete beam, and part for load/support plates; extrusion.

Wire 3d; Parts for reinforcement $\phi 12$ $\phi 16$, stirrups $\phi 8$

2 – Properties

Create material for concrete CDP, load plate and reinforcement.

Density, elasticity and Plasticity needed (CDP concrete, and plasticity rebars)

Create sections for Solid parts and wire parts truss(define CS area)

Define material directions, and section direction.

3 – Assembly

Add parts Concrete beam, reinforcements and stirrups.

Make mesh individual for all parts.

Place load and support plates, copy and place out reinforcement.

Add partition planes in transitions from load plate to solid beams.

Split load plate in two, length wise. Important for supports conditions.

Optional additional partition planes placed.

4 – Steps

Add Static- then either General solver or RIKS solver. See calibration chapter

User must define one set of supports and loads, for each solver. Same for field and history output, or else other definitions are common. In this thesis, General solver is mainly used.

Add Field output and History output and add default. n=1

Used input in this thesis see chapter 3.

If CDP is used, add [sss

5 – Constraints

Add all reinforcement as Embedded. Create Set for all reinforcement and add to Solid beam.

Add supports along slip line created on support plates. Constrain vertical direction.

Add – TIE – restrain between surfaces of load plates and support plates towards Solid beam.

6 – Loads

Add load, define pressure and mark two load plate top areas.

Load is calculated as [total load / surface of both top plates]

See calibration chapter.

7 – Mesh - Used for solid elements and wire.



Defines how to mesh the elements. Wedge, hex, hex dominated or tet.



Defines the specific element to be used; Tet, hex, linear, quadratc, truss and other options.



Defines the overall mesh level for an assembly or by part, m50 m25 etc..



Select all assembly parts and use this to add mesh to the assembly, or by individual part.

See calibration chapter

8 - Job

Create job- Select full analysis, keep defaults and to speed things up, cpu to 2, and gpu til 1.

9 - Results/ Visualisation

a

Load completed job to the visualizer by right click on job and select results..

All load increments can be selected forward and backwards to see progress towards end result, or animated through playback. Many options, view by part, split sections etc..

U – displacement – See deformed state and end result for

S – Stress

Damaget – CDP damage tensile

Damagec – CDP damage compressive



Used to extract output from history output or field output.

Method used in this thesis; select field output, select “Unique node” as target, for example displacement u2. Use edit function, cut model in half and find a middle node, select and save, which generates a list with output from each increment it has had on the way to full load.



Use this data manager to access the list, use either Edit, to copy to excel, or plot to view.

(hold down shift to mark lines)

List reflects time vs displacement u2. Time is represented by (1s = full load)

Flowchart for 2d shell RC beam model in Abaqus.

Make a choice about inputs. Here used *meter* is defined through input file.
2d shell model is made through XFEM software.

1m → 1 or 1mm → $1 * 10^{-3}$

1 – Create Parts

Select – File – import – model, select imp file. (2d shell element .imp file)

This makes the model as one part, including beam, rebars stirrups, support and load triangular created. Mesh is predefined, to match height of reinforcement placement 40mm.

As 2 Top is 39mm down, As 1 bottom is 41mm up.

2 – Properties

Create material for concrete, load plate and reinforcement. *

Change support and load triangulars, to steel. Avoid added deformation of system.

Change cross section area of bottom reinforcement to 603 mm

3 – Assembly

No assembly, one part.

4 – Steps

Add Static- then either General solver or RIKS solver.

User must define one set of supports and loads, for each solver. Same for field and history output, or else other definitions are common. In this thesis, General solver is mainly used.

Add Field output and History output and add default.

5 – Constraints

No constrains

6 – Loads

Two point loads added on top of load triangles. See calibration

[point 1 + point 2 = total load]

From this point, it is the same as for the 3d solid model

Testing of reinforced concrete beams

Overview :

6 reinforced concrete beams were casted with 3 different concrete mixtures, and then put through a 4-point bend test until failure, after 14 days and 28 days of maturity. Prior to casting, the beams had strain gauges glued at the reinforcement at strategic places. As part of the testing, first 10 cracks were manually noted and visually measured along with the increased loading.

Concrete samples for the 3 different mixtures were casted, then tested at the same day as the failure test, or within one day. All this was done as a part of a Bachelor's thesis at UiS, and that data is the input for my Master's thesis.

Analysis of test beam, geometry:

All 6 beams had equal geometry and reinforcement, measuring 2200x300x250 mm (LxHxW), with 3 $\phi 16$ mm as tensile bottom reinforcement, and 2 $\phi 12$ mm as compression reinforcement, in addition $\phi 8$ mm shear reinforcement was added c/c 130 mm.

All beams were loaded up to and slightly over their ultimate tensile stress (UTS) point, where the load max resistance was noted.

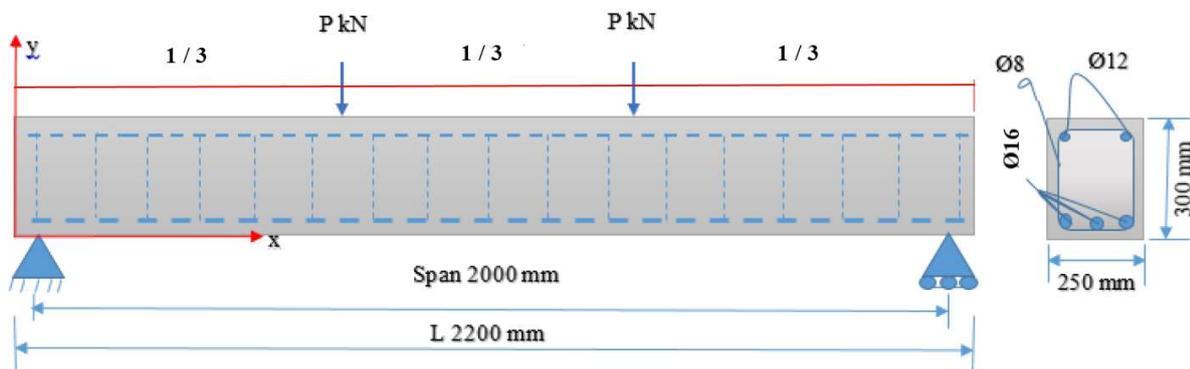


Figure 33

Beam dimensions and reinforcement

Crack widths were also registered, as the first 10 cracks were noted and measured manually.

The last beams were in addition filmed by a camera which after calibration was able to trace stress concentrations through changes of color on the concrete surface.

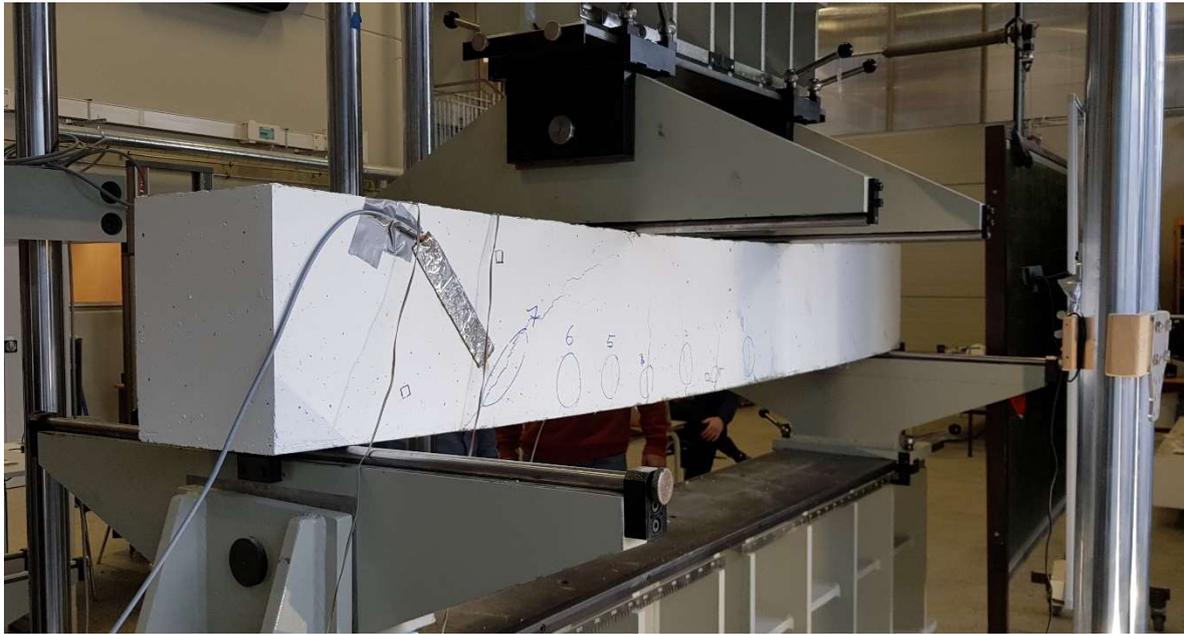


Figure 34

Four-point bending setup, simply supported with two point loads

The beams were placed as simple supported beams, resting on top of a set of rollers, spanned 2m apart. Two equal point loads were placed 1/3 of the length (0.67m) apart from the supports, creating 3 equal distances over the midspan. This setup creates a constant bending moment diagram for the middle part, as seen in the figure below. Beams were then tested to their limits.

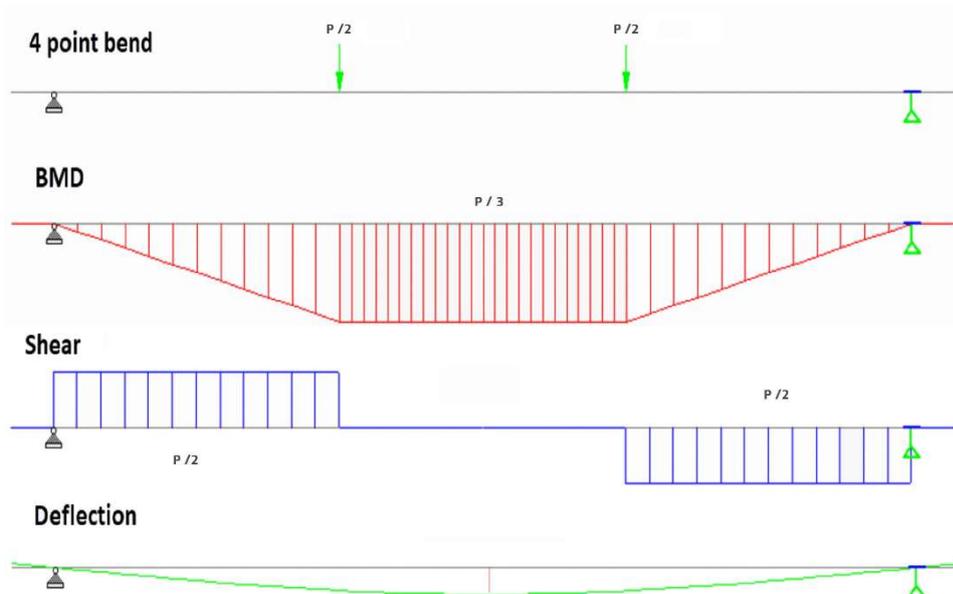


Figure 35

BMD, shear and deflection for the 4-point bending setup

From the Toni-Tech load setup, forces and displacement from the beams failure testing were registered. The strain gauges glued to the reinforcement prior to casting inside the beam, were connected to an amplifier and data was registered. Gauges were also attached on the surface of the concrete in areas where cracks were expected to appear. Material testing were conducted as compressive crushing of cubes and splitting of cylinders and E-module testing.

For this thesis, only one dataset from one beam would be used, both as reference and target.

The chosen beam was beam 6, as this beam had the most consistent data available.

BEAM 6 : designed as B35 concrete - tested after 28 days.

Hand calculations for M.Rd were calculated with no safety factors, where both concrete tensile capacity and steel yield value from tensile testing were added together, see appendix.

If one further looks at the hand calculations, the imbalance between the top compression half and the tensile part becomes clear. This means that the reinforcement values from a tensile test is needed.

* Inserted with $\phi 16\text{mm}$ tensile test results at f_{uk} (700MPa) See appendix for test.

** Inserted with 500 MPa for $\phi 16\text{mm}$ rebars.

Table: 1 (maximum load capacity at UTS)

B35 hand calculation UTS*	298.4 kN
Lab test result	296.9 kN
M.Rd.c **	227.5 kNm
M.Rd.t **	78.8 kNm
AS1 reinforcement "balanced"	1764 mm ²
AS1 3x $\phi 16\text{mm}$	603 mm ²
Stadium 1 F _{.CR}	38.5 kN
Stadium 1 M _{.CR}	12.9 kNm
Stadium 1 Self-Weight displ.	0.2mm

Deflection from self-weight was found to be 0.2mm , with an approximate weight of 4.4 kN including all reinforcement.

This was compensated for in the reference displacement diagram for beam 6.

Test plot was stopped at UTS, max capacity, as this part of the curve is the target value.

This is the reference curve for calibration and during test of parameters.

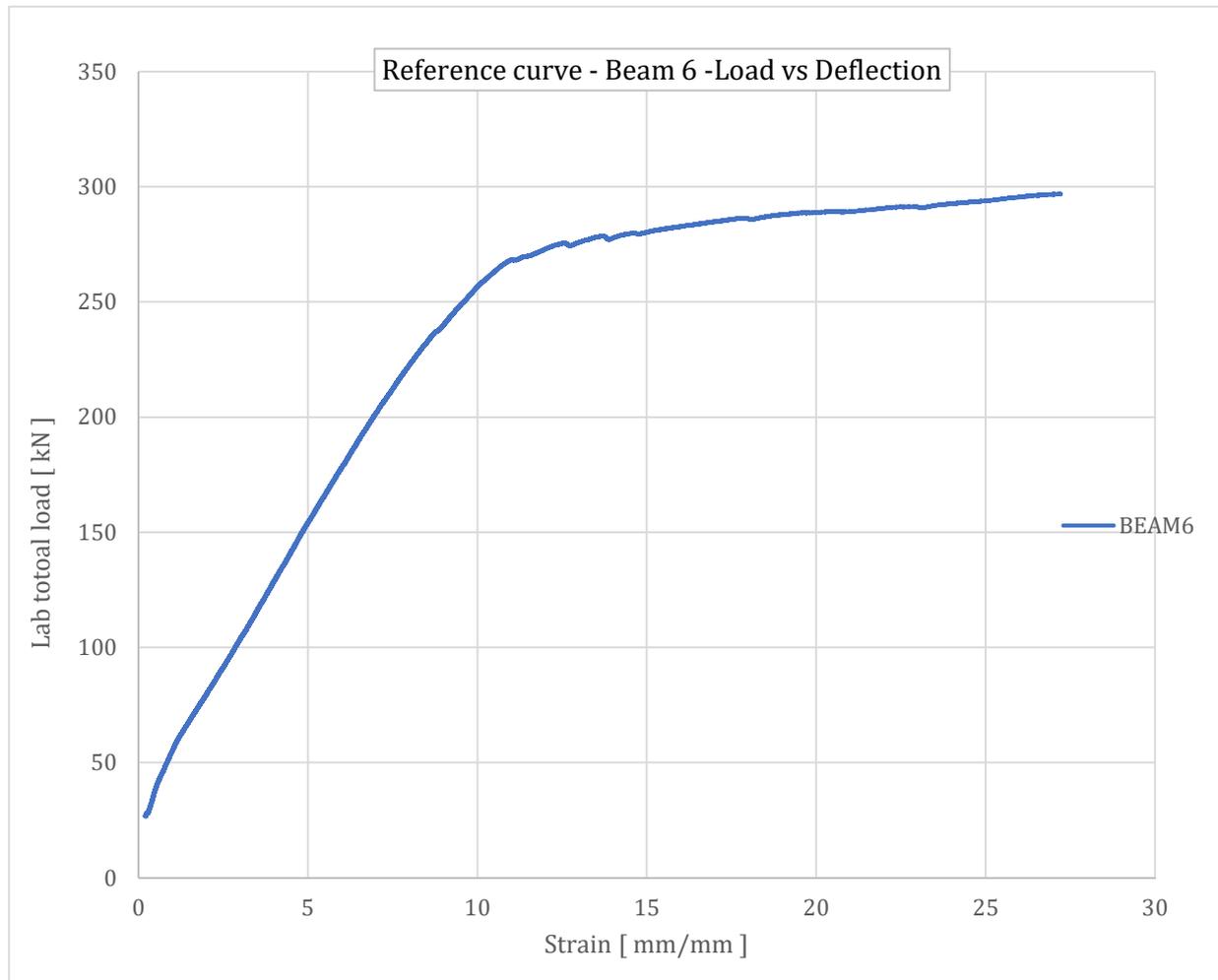


Figure 36

Load deflection curve adjusted for 25kN calibration start and self weight

The procedure for the load test machine, is to let it apply pressure on the beam, until it has registered 20kN of pressure. From this point, load and deflections are set to zero, and measured. This means that for plotting the correct curve, an additional load of 20kN needs to be added, and compensated for self-weight deflection.

This is included in the curve under, as it's shifted 25kN up, and deflection is added 0.2mm

Strain Gauges:

Strain gauges measure small changes in the material it is attached to, as it has constant current going through it. If the voltage drops, the resistance has been increased, because the pathway it took inside the strain gauge has been lengthened. When amplifying this change, we can read off the gauge that the material has undergone elongating, measured in strain as to original values. If the resistance measured is lower, then the material has been compressed. It can only read values perpendicular to the attached long side of the gauge.



Figure 37

Beam prior to testing

Figure 39 and Figure 40 shows strain gauges visible glued to the concrete at the angle 45*, 90 deg towards expected crack elongation direction, ready to be tested.



Figure 38

Strain gauge, and numbering of tensile cracks

Unfortunately, only beam 1 had successfully readings, as the other results were too unreliable. This means that strain for the reinforcement will be taken from the model instead.

Reinforcement:



Figure 39

Stress strain $\phi 16\text{mm}$ and test specimens

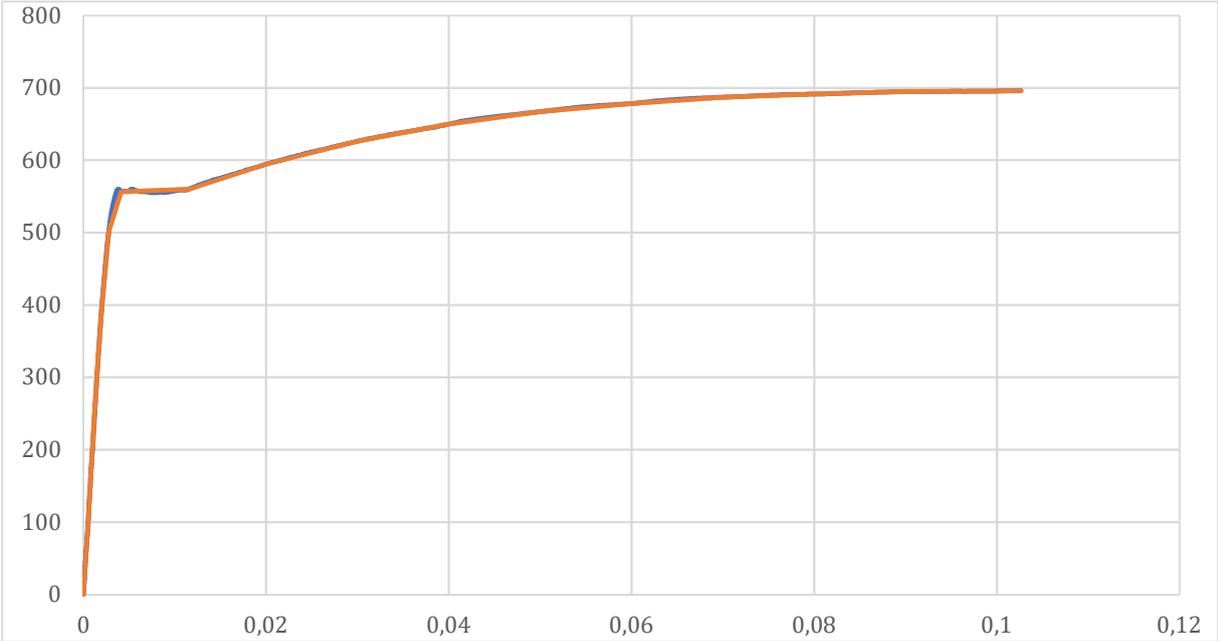


Figure 40

Tensile test $\phi 16\text{mm}$, plotted to UTS

All reinforcement were put through tensile tests, stretched until full failure. This was to ensure all sizes rebars where their yield 500 MPa specifications. Special attention were made to the $\phi 16\text{mm}$ reinforcement, as they are the main component of the tensile strength in the RC beam.

To avoid making the models over-complicated, the tensile data from the $\phi 16\text{mm}$ test were used for all types of rebars.

Concrete

When the concrete for the beams were mixed, additional samples were casted, to be tested with the same maturity age as the test beam.

Within a day from the beam testing, these samples were put to compression and tensile tests, to measure material input data. Compression crushing test, for stress strain compression curves, and split test for concrete cylinders to test tensile strength and calculate Emodules.

Compressive testing

The class for the concrete mixture was B35. Four compressive samples were tested.

The four test samples stress strain curves were plotted, as shown in Figure 43.

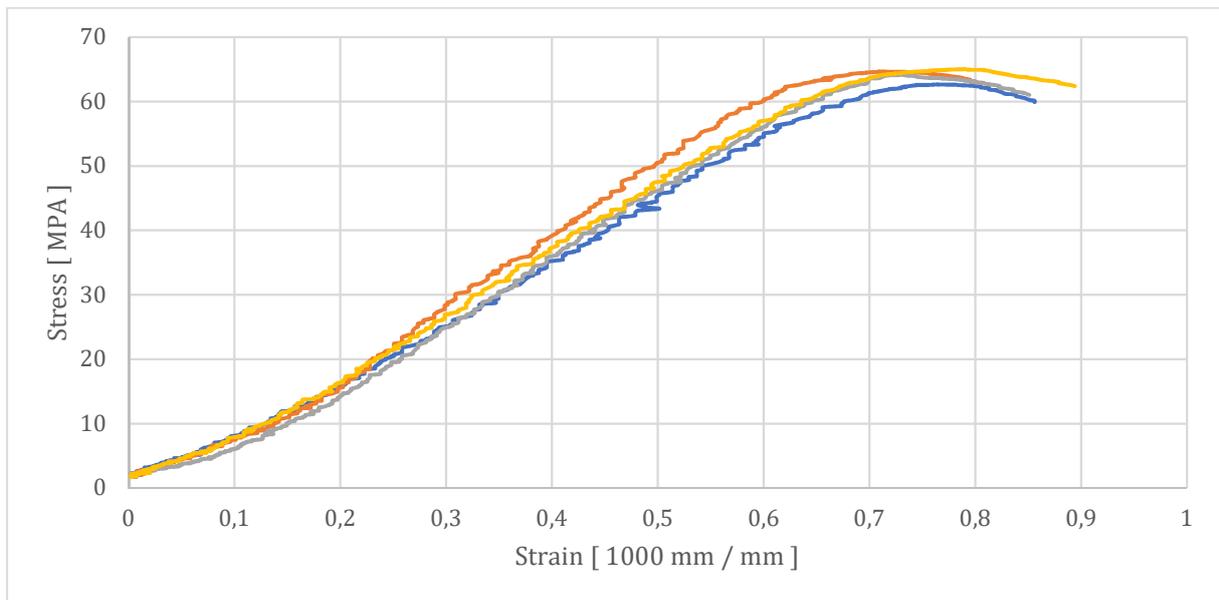


Figure 41

Stress strain graphs for compression samples 1-4

σ_c	MPa
Sample 1	62.65
Sample 2	64.65
Sample 3	62.12
Sample 4	64.95
Mean value	63,46
SD	1.2265
f.cm	61.57

Table: 2 (Concrete compressive)

The 4 compressive concrete samples give a good representation of the registered strength. Strain readings are at f_{cm}, is 0.00074, which is very low as normal values for this peak is expected around strain levels of 0.002.

Both EC2 and ifb2010, indicates level of 0.002

From the shape of the sample curve, it is visible that this is a stress total strain curve, and not a stress inelastic curve, and the calibration must have either been slightly off, or there must have been some other error.

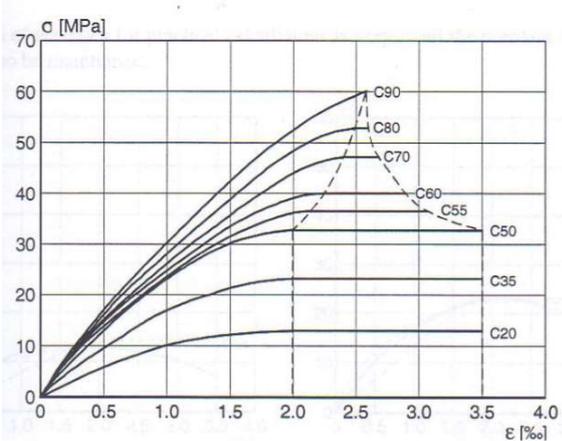


Figure 42
ifb 2010 code

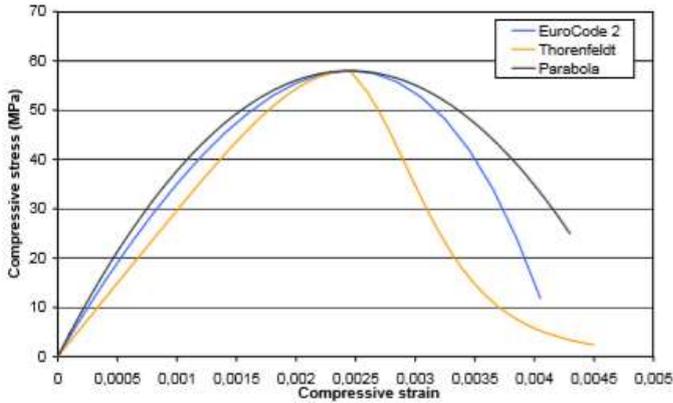


Figure 43
Thorenfeldt and EC2 compressive

f_{ck} (MPa)	12	16	20	25	30	35	40	45	50	55	60	70	80	90
ϵ_{c1} (‰)	1,8	1,9	2,0	2,1	2,2	2,25	2,3	2,4	2,45	2,5	2,6	2,7	2,8	2,8

Figure 44
Cut out from the Table 3.1 in EC2 - Strength class and ϵ_{c1} .strain values

As shortened version of the table 3.1 in EC2 is presented, to give a picture of EC2 expectations of a strain value, at f_{cm} peak. In EC2 this value is given as ϵ_{c1} .

The strain values from the compression test is the fore disregarded, as the strength values can be used further.

Tensile testing

Tensile tests were performed with cylinder samples, in a wedge split test, and a compression test with extensometer to determine tensile strength and e-modulus.

These values would be used to create material data import for Abaqus model.

Table: 3 (Tensile strength)

Tensile split	MPa
Sample 1	2,02
Sample 2	2,20
Mean value	2,11
SD	0,09
f.ct.sp	1,96
f.ctm	2,18

Table: 4 (E-modulus)

E-modulus	MPa
Sample 1	29840
Sample 2	29520
Mean value	29680
SD	160
E.cm	29400

As for determining the stress – inelastic strain curve, linear approximation is to be used as a starting point.

These values would be used to create material data import for the Abaqus model.

Modelling of reinforced concrete beams using Abaqus

There are several ways to model a problem with the Abaqus software.

Shell elements either in 3d or 2d, or solid elements. Beam Dimensions 2200x300x250 with 3 ϕ 16mm tensile reinforcement, 8mm shear c/c 160mm and 2 ϕ 12mm as compression reinforcement.

In this thesis, a solid element model was chosen, with reinforcement as trusses, embedded into the concrete, but as the test setup can be seen as a 2d problem with only bending in one direction, Daniel Bårdsen also built a 2d shell model to replicate the setup.

This gives a great opportunity to compare the two different approaches to the problem.

The 2d model consists of several different shell elements. Rebars are placed into the mesh grid system, so values can be extracted by the location of mesh element nodes. This means that the placement will here differ somewhat from the real beam. The mesh of the 2d model is about equal to a mesh 40 for the 3d solid mode.

Both the 2d shell and the 3d solid models have the same goal, to be able to represent the RC beams behaviour as measured during the lab testing. The best way to do this, is to compare load/deflection curves. CAE interface was used for both types.

Runtime for the two different models are vastly different, which makes the comparison more interesting.

Reinforcement

Both models utilizes the T3D2 elements 2 noded truss element as element for reinforcement.

The 2d and 3d model differ mostly in the way they model the reinforcement.

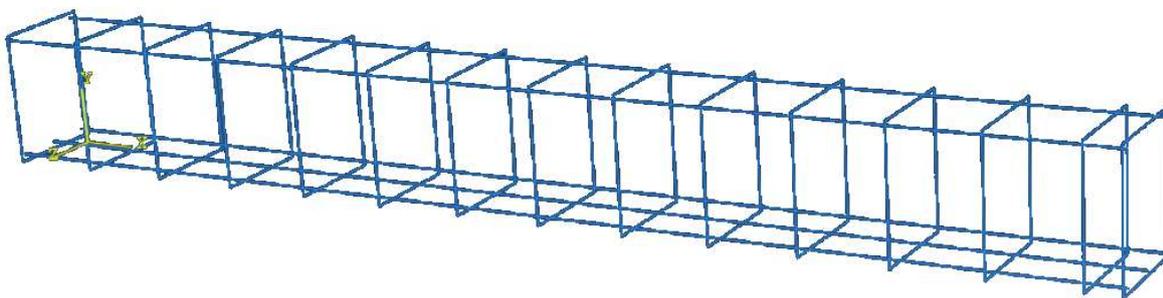


Figure 45

Reinforcement setup for tested beams

As the 3d has the exact setup, with C-min = 25 mm, the 2d model places the rebars directly on the grid lines, so it is easy to extract nodal information from the rebar, just by checking the nodes.

Some parts of the testing were done with only the longitude reinforcement, to speed up convergence.

2d shell element model

Supports as steel, as concrete material proved to be subjected to deformation/crushing and added displacement to the result.

This 2d shell model were only modified trough material selection, support constrains and loading.

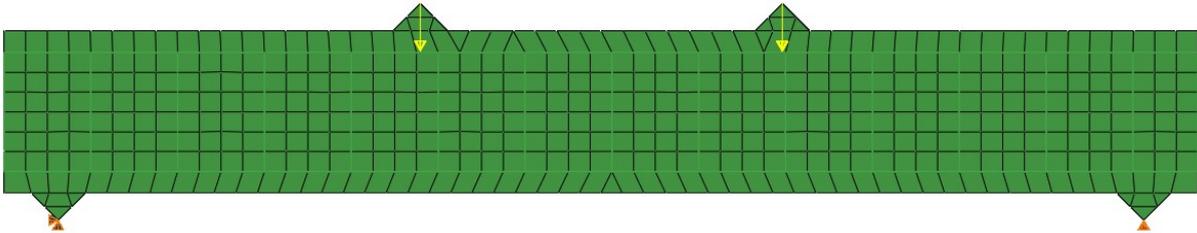


Figure 46

Support and load points

Load: Concentrated point loads. Constant vertical direction of loads

Support and load points are defined as very high strength steel to avoid support points adding deflection to the model during loading.

This model consists of S4 Linear 4 node and S3 Linear 3 node shell elements for the concrete beam and support/load points. The reinforcement is modelled with 3DT2 2 node truss elements.

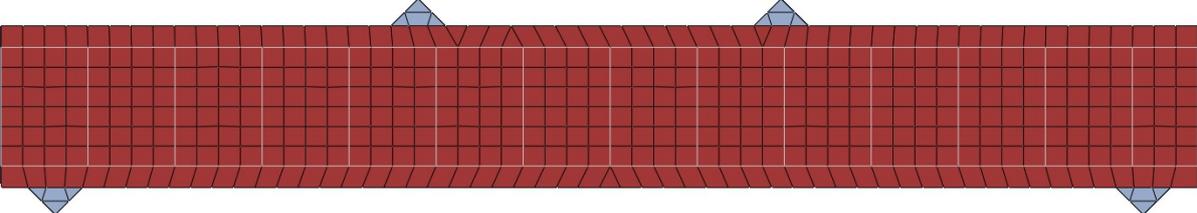


Figure 47

Material selection 2d shell model

This model utilizes the mesh grid with and height, and all rebars are moved slightly to fit directly into the mesh gird. In this way, it is possible to extract nodal points inside the rebar positions.

3d models, solid model

Regarding the procedure for the 4-point bending RC beam lab experiment, rollers were used for both support and load points. This can be done in Abaqus too, but proved to be quite complicated, and introduced a lot of uncertainties of contact surfaces and deformation, and was rejected

With no rollers, load was tried to be applied directly to the concrete. This resulted in local deformation in top half, as it simply deformed downwards in the midpoint, producing incorrect deformation. To solve this, a plate of 20x20x250mm where added, to represent the $\varnothing 20$ mm rollers. Support plates of 30x20x250mm were added to avoid unnecessary local crushing resulting in deformation errors and convergence issues. Plates where constrained to the solid beam with the Abaqus function "tie", and model deformed correctly

Load: Pressure over load plate. Constant vertical direction of loads.

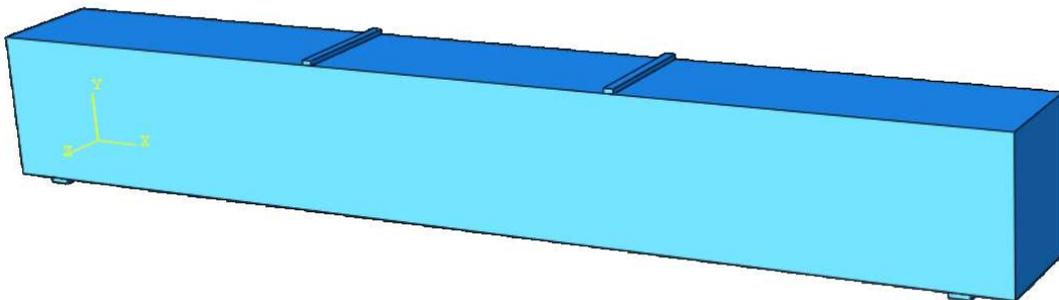


Figure 48

Initial model without partition planes

As seen in the element theory chapter 1, this model violates the rule of having nodes line up directly to each other, which was clear as its deformation behaviour was completely off. Here the mesh level dictates where the node goes, and is not lined up to where the connection plates meet the solid beam, even though they are connected with Tie restriction. As a direct result, partition planes were added to cut through along the edges of load and support plates, and sorted the problem.

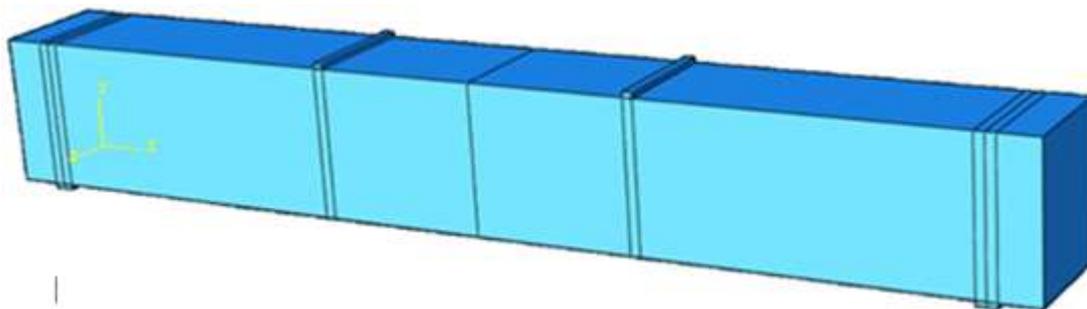


Figure 49

First model - partition planes added at tie-in points

Error in the first model

This model worked fine for a long time, but under certain load conditions and, it would produce incorrect force displacement curve. The error was created by a too large difference between element sizes in general for mesh 50, and the narrow bands 20mm wide where the support and load plates partition planes cut into the solid. This error could be ironed out by applying mesh20, which is hopelessly impractical choice considering computational time.

Error in the first model, shown by increased amount of increments in the general solver.

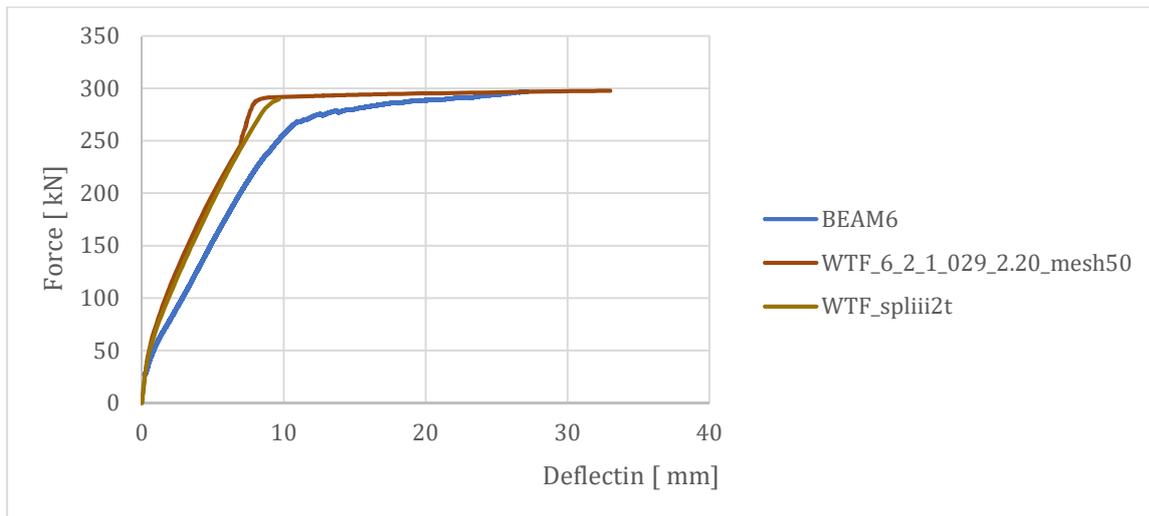


Figure 50

Incorrect force displacement curve, sudden rise before bend point

The solution was again to replace the model, opt1, with wider load plates and supports, optimized to run well with mesh 50, mimicking the element size of mesh50 producing a more consistent element size throughout the model, which sorted out the problem.

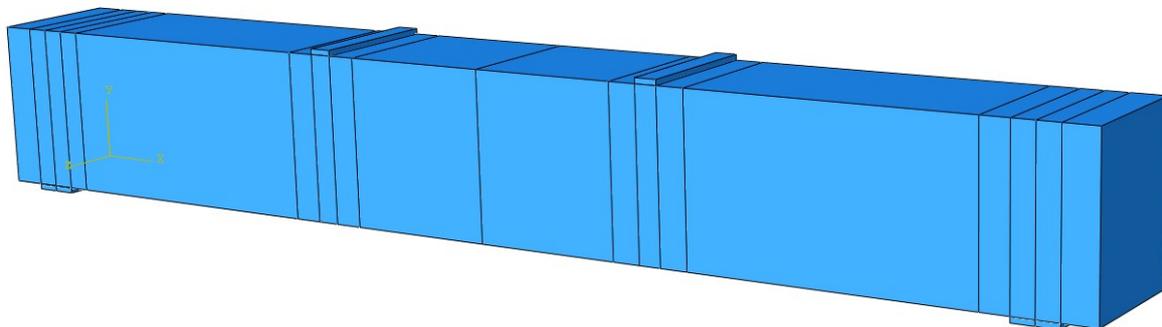


Figure 51

Opt1, Abaqus 3d solid model

Another set of models were tested, with additional partition planes, as to get more even mesh sized elements in the crucial places. This also solved the problem, and mimicked results of finer meshes, but still was unpractically slow. Opt 2 is forcing a mesh level of 20, as Opt 3 is giving a mesh of 10. It still had the transition between narrow bands and higher mesh, but not in critical places. Opt2 and opt1 align perfectly on the error graph on previous page.

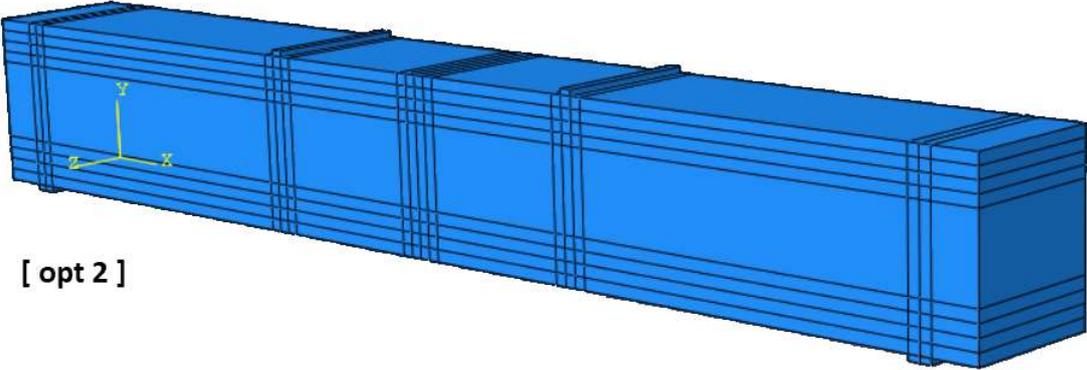


Figure 52
Option 2, mesh 20

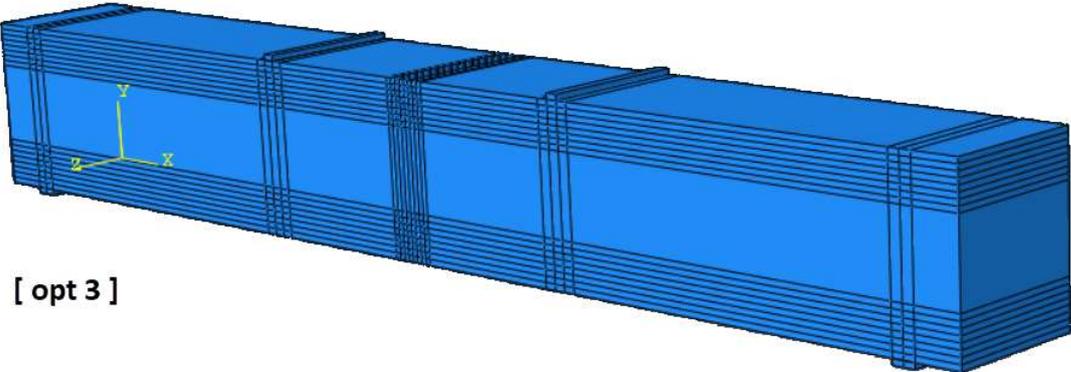


Figure 53
Option 3, mesh 10

Opt 3) ensured 10 mesh size around midpoint and towards top bottom surfaces.

Both models have significantly longer calculation time than opt 1, but still faster than mesh20 and significantly faster than mesh10.

This is only used for final runs, some Part 4 and strain extractions.

Calibration of Abaqus models.

The Abaqus software is by default, unitless, meaning the user can determine the input units, being length as meter, millimetre, inches, stress as kips or N/mm, newton or kN.

Verification of unit input, and unit matching of the 2d shell model and 3d model. The choice of inputs where made to use [MPa] and [mm], as main inputs.

This is mostly done as being more easily able to adapt and compare other CDP concrete material setups, which were mostly found MPa and mm. As the 2d shell model was modelled in meter, a trail test had to be set up to see when the unites would be correct. Frist step here is to add both models as solid beams of steel, with E200GPa.

As the 2d shell model is in meter, and the 3d solid model is in mm, and loads and inputs from CDP recepies are in MPa, loads were adjusted to correct values with MPa as input.

Input used : Beam [E200GPa] , 500MPa] - Selfweight [7860 kg / m³]

Total load Deflection	1800 kN [mm]	3600 kN [mm]	Self-weight Steel 0.981 [mm]
Mathcad	2.297	4.811	0.011
Staad.Pro	2.098	5.054	0.011
2d shell model	2.257	4.764	0.0106
3d solid mode [opt1]- Mesh 50	2.439	4.879	0.0112

Table: 5 (material steel input

Just to make sure a quick test to see if the point load added, or functioned as a summation:

	1x1800 kN [mm]	Max σ .s Beam [N/mm ²]	2x1800 kN [mm]	Max σ .s beam
Staad.Pro	2,38 mm	182 MPa	4,79 mm	213 MPa
2d shell model	2,25 mm	307 MPa	4,54 mm	320 MPa

Table: 6 (material steel input

To define total applied load 290kN, and output deflection in mm, stress from models:

	3d solid opt 1	3d solid opt 2	2d shell
290 kN defined as :	14.5	29	0.145
Deflection [mm]	10 ⁰	10 ⁰	10 ⁻³
Stress [MPa]	10 ⁰	10 ⁰	10 ⁰

Table 7 (Units input)

Note

During this load calibration, it was found that the 2d shell model had at one point double load, and 3xØ20 rebars in tension. On a mishap, this version was copied to the workstation and used to produce the CDP parameter comparison. For the comparisons, this has mishap has no influence, but it does explain why the numbers on the 2d shell model diagram seems off.

For all other uses, the correct 2d shell model was used.

This step involved adding Concrete Damage Plasticity module for Abaqus, to determine the concrete material behaviour both 2d and 3d models. As Abaqus manual states that both -General solver, and -RIKS solver are suitable for static linear and non-linear problems. As this thesis only investigates the behaviour up to the UTS of the beam, it also indirectly means that both solvers will fit the task. If we were to investigate beyond the UTS point, only the RIKS solver would be usable.

From the material input, a concrete with minimum input from CDP was used with values from EC2, B35 class concrete. See attachment for material parameters, [B35_Default], and will be used for most cases of testing in this thesis.

First, simple pointloads

Model used, opt 1. , [B35_DEFAULT] , and rebar material input [PL_true].

Simple test: Deflection

Solver	Model	Load 180 [kN]	Load 280 [kN]
Output :		Displacement	Displacement
General	2d shell	2.9 mm	5.9 mm
RIKS	2d shell	2.9 mm	5.9 mm
General opt1.	3d solid	3.3 mm	6.6 mm
RIKS opt1.	3d solid	3.3 mm	6.6 mm
General opt2.	3d solid	3.3 mm	6.5 mm
RIKS opt2.	3d solid	3.3 mm	6.5 mm

Table: 8 (displacement)

As seen here, the ability to produce end results, which is the main function of a solver, works for both solvers. Two identical runs where done with General and RIKS solver.

But as it would be seen, the two different solvers have one big difference in how they can present results. For all solvers, the path way is just something that needs to be done, to get to the end result, where the aim is to get there as efficient as possible. This is true for both solvers, but the general solver takes the same path each time, as the RIKS takes a new arch length. This makes the General solver much more efficient to use where you want to plot graphs, based on increments.

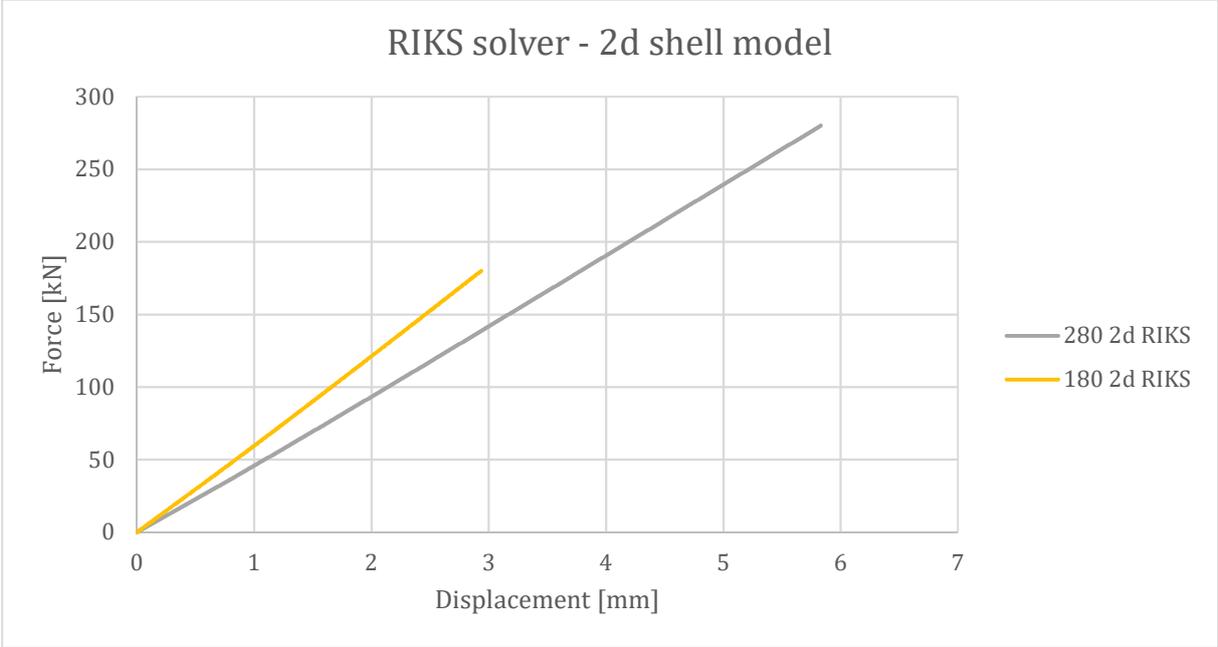


Figure 54

RIKS, plotted by increment steps

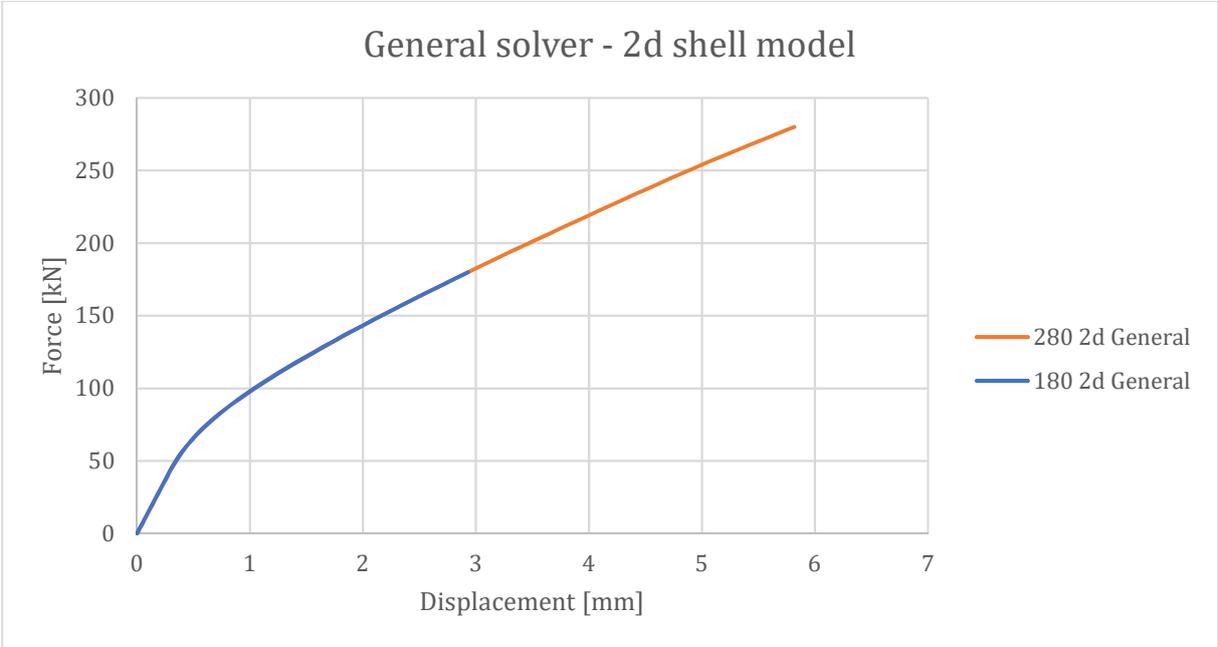


Figure 55

General solver, plotted by increment steps

As seen on the previous page, to plot a from the RIKS solver, the solver has to converge on each load creating an accurate end point, and then use these endpoints to form a graph. In contrast, the general solver gives the opportunity to extract usable data from each increment step. But as each step is only an assumption along the path, again here only the end can be truly trusted to be accurate on a convergence.

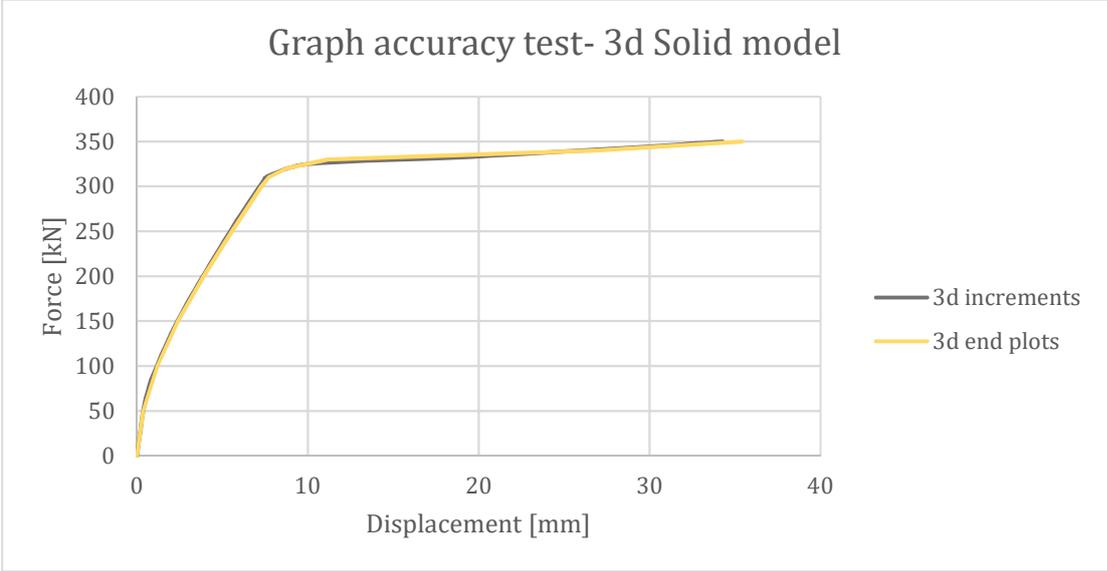


Figure 56

Test 3d solid model

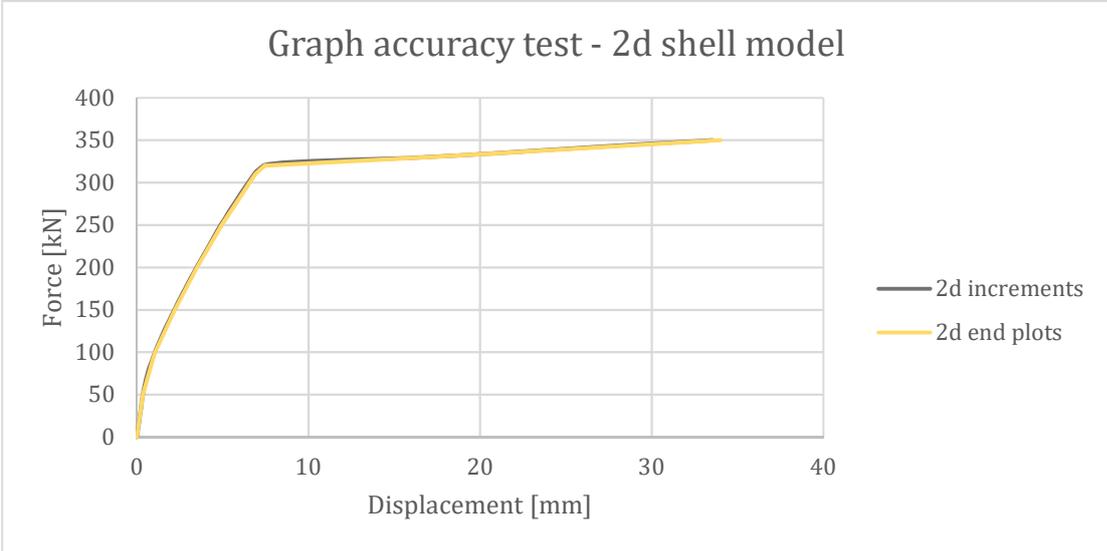


Figure 57

Test 2d shell model

Here several runs were made to test the extraction of increments vs plot points, and as there were minor differences, the correlations were very good, and the method of extracting data from each increment step was accurate. Therefore, this method is the preferred choice for graph plotting using the General solver.

Mesh size

Mesh size in this thesis, is only valid for the 3d solid model. Adaptive meshing was not used, as it only works with Tet elements, and which for this model is no gain over using HEX elements and less elements.

The reason to look into this, is to verify if the model is mesh dependent. This is tested by letting all input to be kept as constant, only varying the mesh level. Otp1 was chosen.

Model opt 1. , [B35_DEFAULT] , and rebar material input [PL_true]:

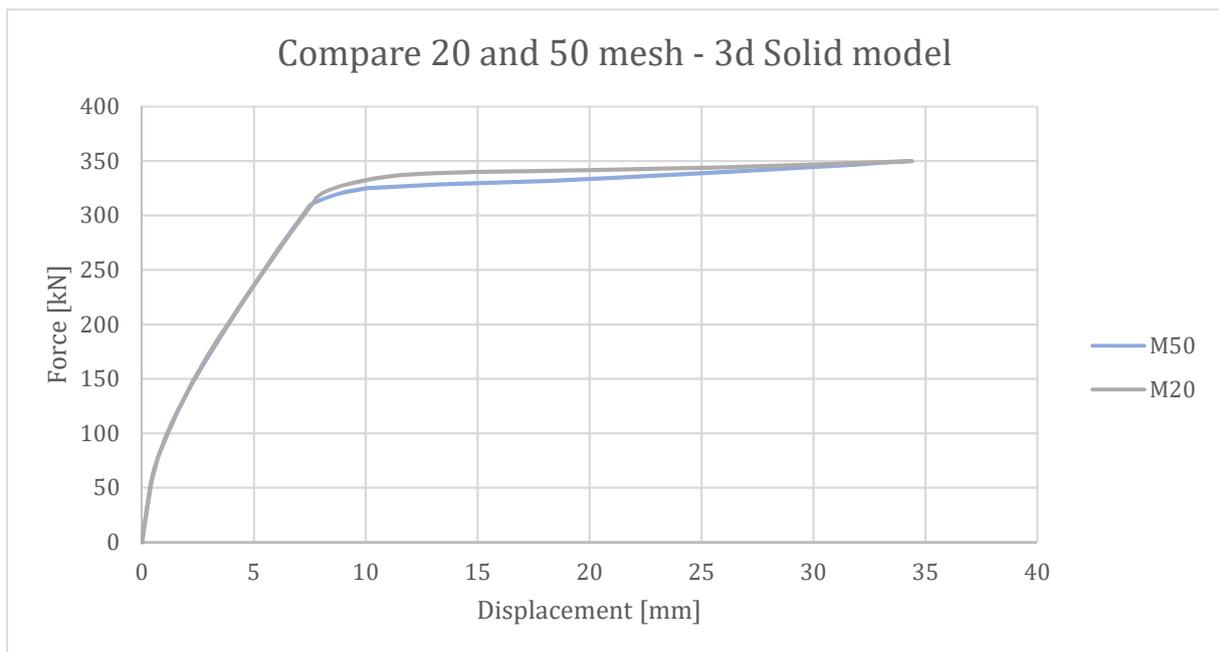


Figure 58

3d solid model mesh dependence

As we can see, the opt1 model is mesh dependent. The first model made, was much more mesh dependent, and therefore rejected.

There are three words that go together; mesh size, viscosity parameter and convergence. If the model does not converge, we can either solve it by refining the mesh, or adding viscosity. The effect of the viscosity parameter will be mentioned in the CDP section.

It was also tested if the General solution, with the stabilizer option on, had any big effect on the results. This function greatly helped in producing consistent result, as convergence was possible over a wide range of loads. In most cases, CDP specific concrete setups, will not run into the non-linear area without stabilizing and a small amount of viscosity.

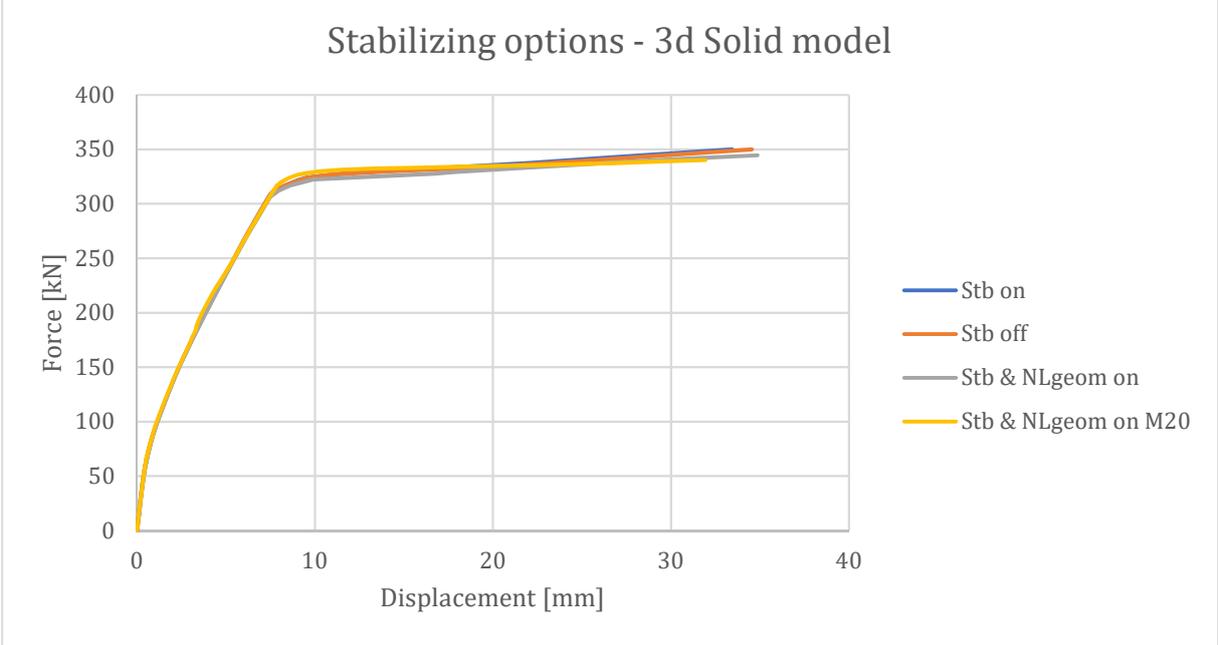


Figure 59

Stabilizing Parameter in General solver – 3d solid model

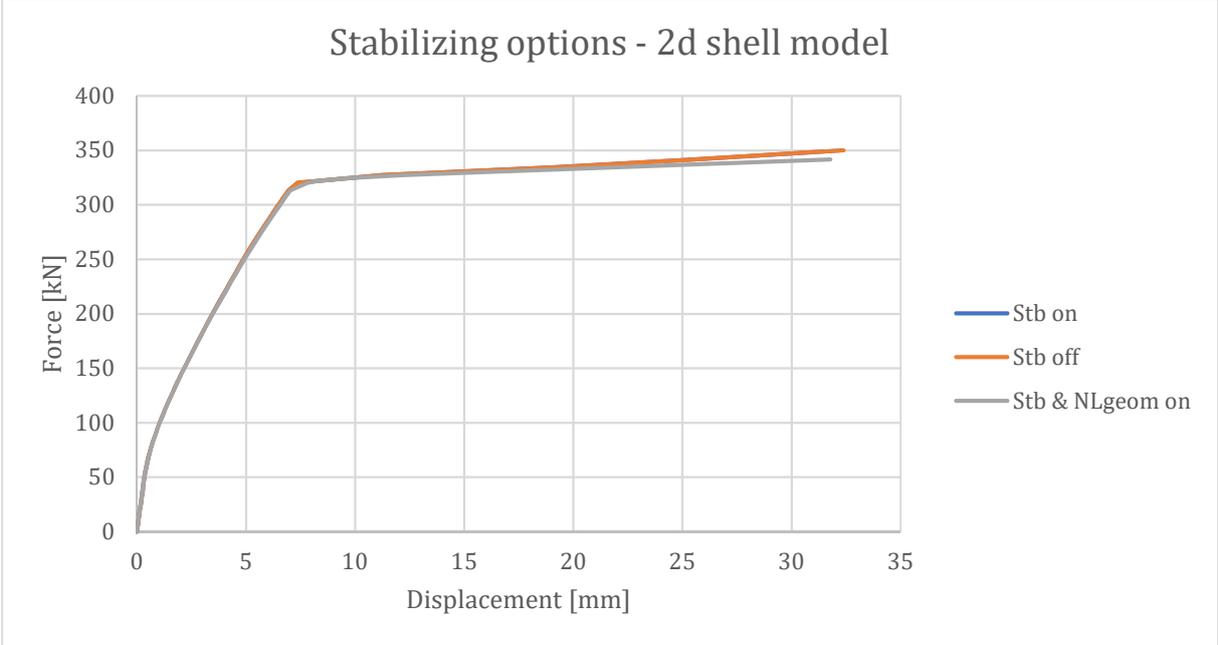


Figure 60

Stabilizing Parameter in General solver – 2d shell model

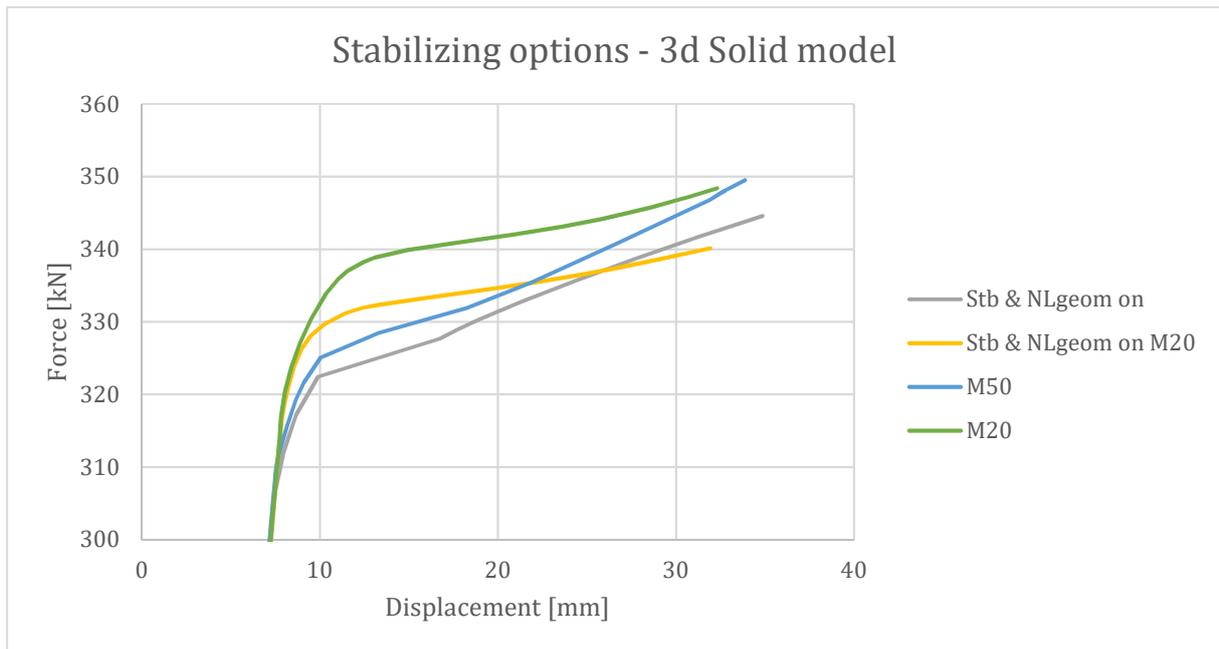


Figure 61

Enlarged differences between stabilizers

The most surprising about the graph above, is that the rise in stiffness seen for m20 on top figure, is reduced once the non-linear modulus has been initiated. The conclusion from these two graphs, is that a mesh 50 with NLgeom off, is as accurate as a m20 with NLgeom enabled, for the displacement range up to sub 30mm as is where the test beam UTS point is.

Mesh 50 will be used further for testing and material plots.

As for element testing for the 3d solid modes, both C3D20R, quadratic 20 node element, inc R hourglass control, and 10 node Tet elements where tested.

Except for the x5 increase in time, the difference where within 0.2mm-0.3mm , which there is no use in plotting. A test between truss and bar elements where done too, with even less difference in result.

For the entire thesis, the model will consist of C3D8R, linear quadratic element with reduced integration, with truss T3D2 elements.

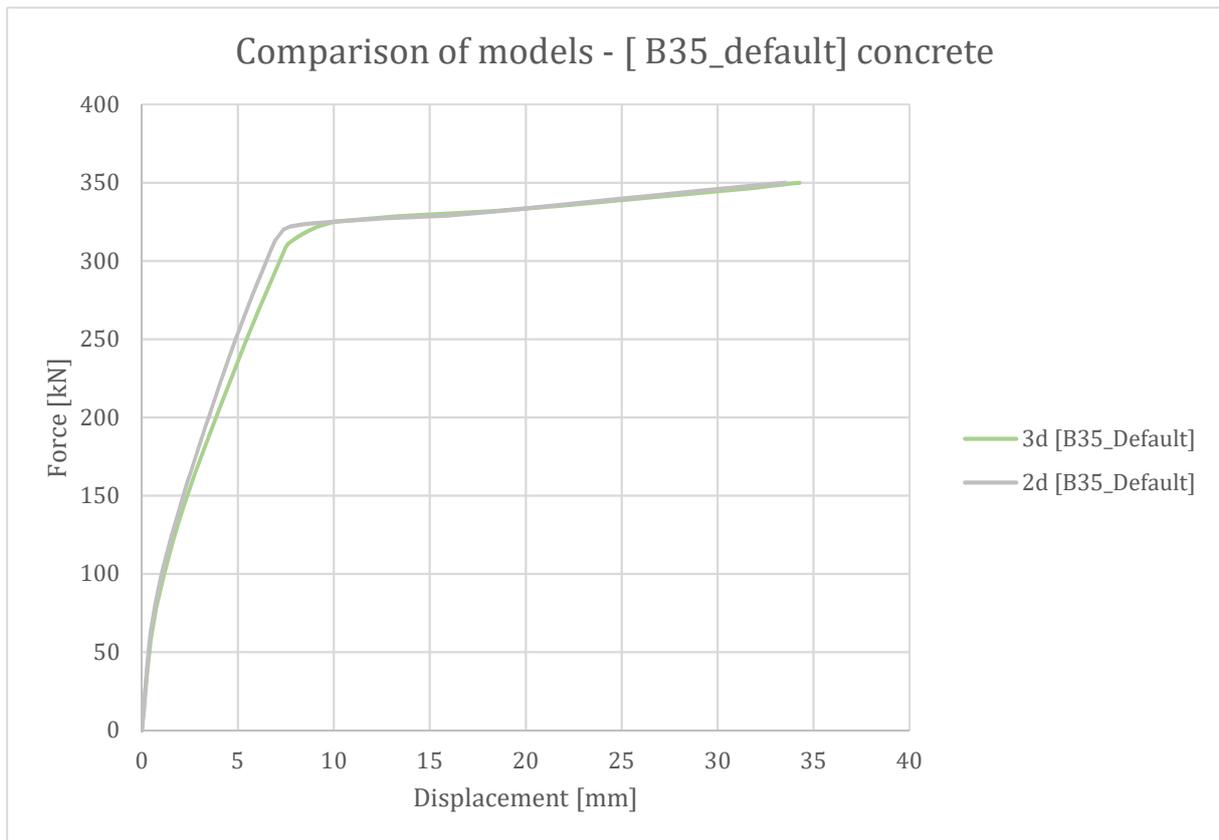


Figure 62

Direct comparison, viscosity parameter = 0.

The 2d shell model is a tad more stiff, while both have identical hardening curves. The models are so close that they can easily replace each other for general purposes.

What the 2d loses, is flexibility of repositioning the reinforcement and add changes, without having to rebuild the model and adjust mesh accordingly.

In general, the 2d shell model could roughly be said to be 3-5 times quicker to converge that the 3d solid model with a m50 mesh, which makes it very capable.

In addition, the 2d model is less affected by convergence errors, needs less viscosity and seems generally more stable.

It would have been interesting to see a 2d shell model being compared to a 3d shell model, as from a lot of reading during this thesis, the 3d shell model might be a better option than solid type.

For the model calibration, the concrete [B35_default] is used, as it has few inputs and minimizes it's affect on the model calibration.

Adjustment of the material input parameters will be done in section Adjustment.

Overview of setup for testing

Deflection:

Deflection is used for both target and as validation. Unless otherwise is stated, reference to deflection in this thesis means deflection measured at the bottom side at the mid-point length wise, as near the centre width midpoint as possible, given by the mesh size.

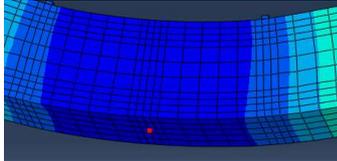


Figure 63

Midpoint - deflection ref point

Solvers :

Increments. All runs are done with these input for increments, both General and RIKS solvers. This needs to be done to ensure that enough increment steps are produced through out all the loads range, and not just the last part. See appendix.

	Initial	Minimum	Maximum
Increment size:	0.001	1E-035	0.02

General solver would have the stability option on, as default, with these setting:

The solver automatically determines the model as table, and only enables the settings if it is found to be unstable.

Automatic stabilization: Specify dissipated energy fraction :
 Use adaptive stabilization with max. ratio of stabilization to strain energy:

Concrete :

If nothing else is specified, viscosity 0.00001 is used.

Concrete [B35_default] is used for testing, see appendix.

This is a concrete setup with minimum requirements inputs for the CDP module to run. This is done to make sure the input material is a limiting or contributing factor to errors or convergence issues.

All testing is done with true stress true strain reinforcement test results input [PL_true], see appendix.

Parameters in CDP

Overview

The Concrete Damage Plasticity module, is an option in Abaqus software to model concrete. Materials are first defined through density, then an elasticity module, where Poisson ratio and e-modulus is defined. For the plasticity region, the CDP module is used.

It defines the concrete as anisotropic, and input is needed for compressive and tensile behaviour. There are many options inside this module, where the Compression and Tensile input with the damage setup is discussed in the material selection.

In this section, only these CDP specific parameters will be discussed:
The theory behind these are showed in chapter one.

This chapter will be more of a summary of their effects, and how they affect the two models at hand given all other inputs are constant.

Dilation Angle	Eccentricity	fb0/fc0	K	Viscosity Parameter
31	0.1	1.16	0.67	0

Figure 64

CDP specific parameters with default values

The first 4 parameters are related to the material in use, as the last viscosity parameter is an aid to help the model to converge, or speed up the process.

Here the test setup is to see the effect these parameters have on the two models in a force displacement diagram, and compare this to observed findings from references.

For all testing in this chapter [B35_Default] is used with [PL_true] reinforcement setup.

Mesh level 50 is used. Opt 1.

Dilation angle:

This angle is often commented as the concrete's internal friction angle.

But, as from soil mechanics, there are several parallel ways to convert the internal friction ϕ angle towards the β dilation angle, since they are on two different planes. But they are close in range, in a very approximately manner of $\sin(\phi) = \tan(\beta)$, where $\phi=37$ is about $\beta=31$.

Dilation angle basically defines the shape of the envelope shape as the concrete enters the yield, as the flow rule follows the angle and defines the shear failure surface.

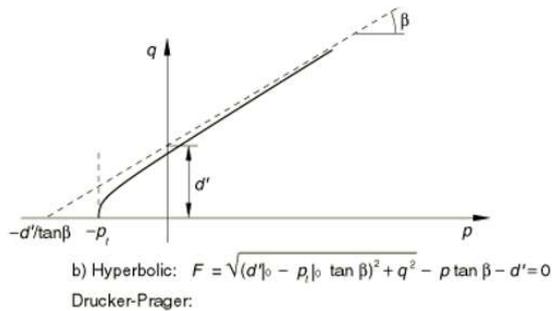


Figure 66

Actual Q-P plan for CDP

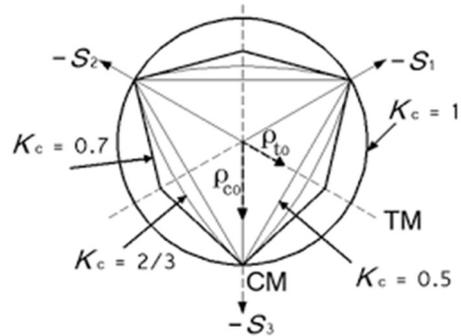


Figure 65

K parameter

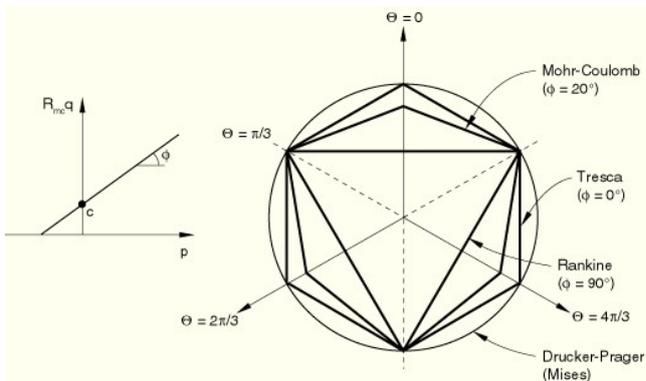


Figure 67

Connection linear dilation angle and curve type

Here figure 68 can be used to explain the eccentric parameter, as it defines how rapidly the flow rule will converge toward the asymptote, which follows the dilation angle.

Figure 67, shows the connection between K parameter values, and type of curve.

Figure 69 shows the connection between K parameter values, and dilation angle, here given by ϕ in the q-p plane.

If we combine these figures, it means that the K values and dilation angles are set up for each other. That means, if we change the dilation angle, we should change the K parameter accordingly too. As seen from later in this chapter, the K value does not have a large effect on the problem at hand here, and is probably the reason it is often left as default value, even with dilation angles adjusted.

Dilation angle effect on models

3d solid model – To make it more readable, these angles have identical shape; ($20 = 25$, $31 = 36 = 40$).



Figure 68

Dilation angle – Force / displacement diagram 3d

2d shell model

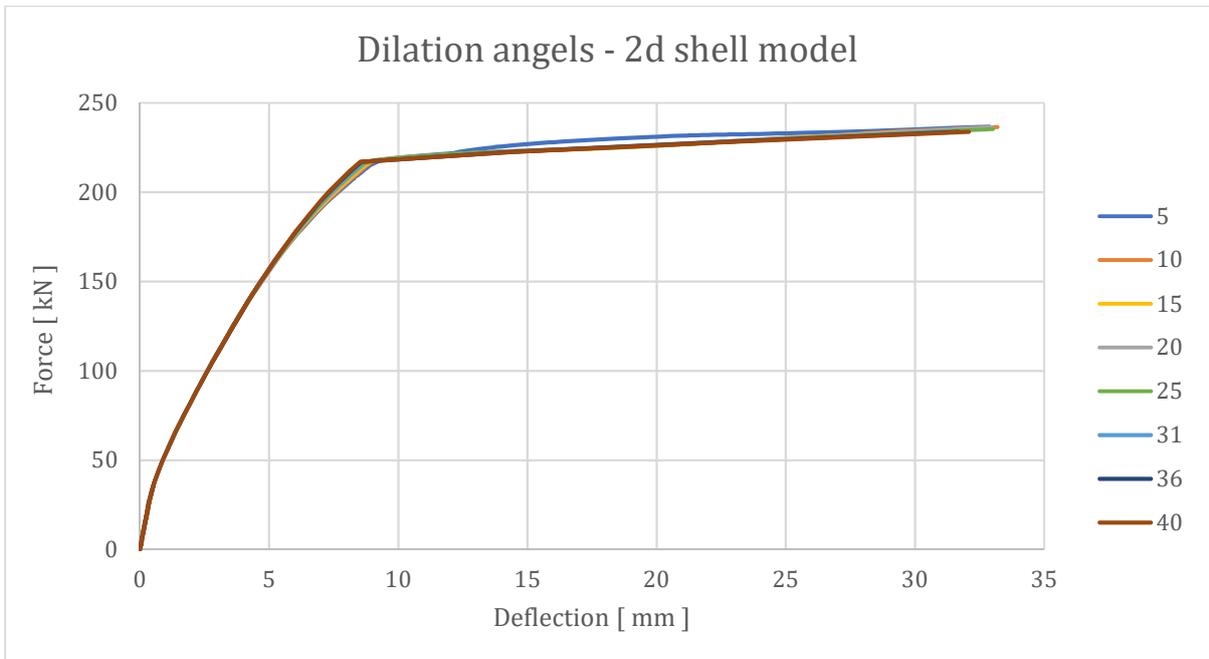


Figure 69

Dilation angle – Force / displacement diagram 2d

Two groups have gathered, as the first angles 5-10-15-20-25 deg, provides a softer stiffness, and the 31-36-40 deg which has a steeper elastic curve and a more flat hardening curve. All angles except 10 degrees converge at the same point, but the main difference is the stiffness during the elastic region for the reinforcement.

3d solid model – To make it more readable, these angles have equal shape; (20 =25 , 31 =36=40).

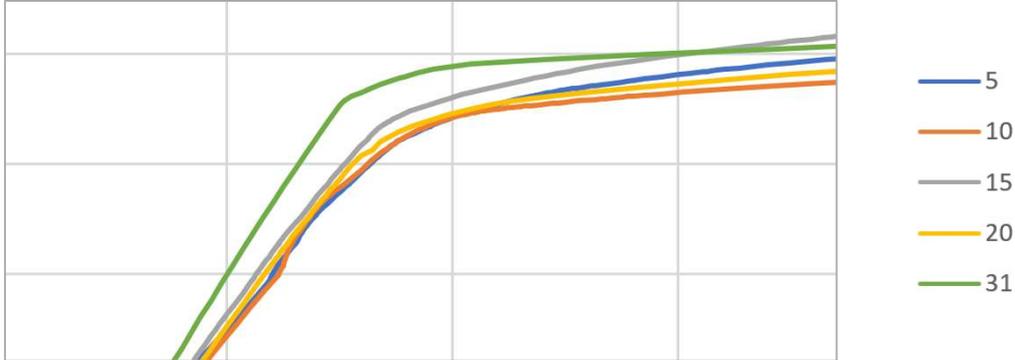


Figure 70
View only to enhance shape

Here the effect is quite varying, and the test had to be double checked to see if the results were produced by some errors.

For the 2d shell model, the variation is much more confined, and all angles give small variation, except dilation angle 5 deg, which gives and increased value for the midspan hardening, before it flats out and converge on the same point as for the other angles.

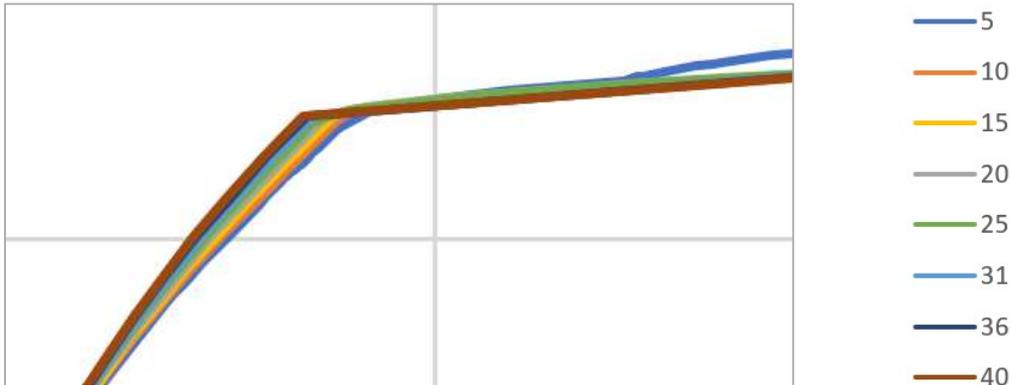


Figure 71
View only to enhance shape

As we zoom in on the corner, it follows the same path as for the 3d solid model, with angle 5 being the softest curve, and the 40 angle gives the characteristic hard nudge around the beams yield point. As seen here, the effect of the dilation angle is less on the 2d shell mode. Note the 5 degree angle gives a softer elastic curve, and a more angled hardening curve, until they all converge around the same point around 30mm deflection.

As default value, Abaqus has provided 31 degrees.

Example Shear failure

During testing with different strain rates, I noticed that a modified [6_2_1] setup, with tension set to 2,6 MPa and end strain cracking rate to 0.0038, was actually producing a shear failure in the model. This was possible due to a 160c/c shear reinforcement placement, instead of the correct 130 c/c shear reinforcement. – Dilation angle was set to 31 degrees, and exposed to shear failure.

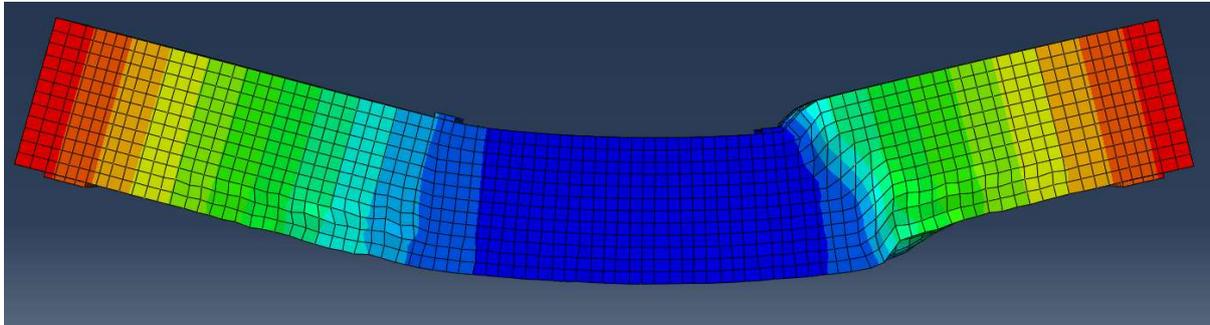


Figure 72

31deg - shear cracking visible

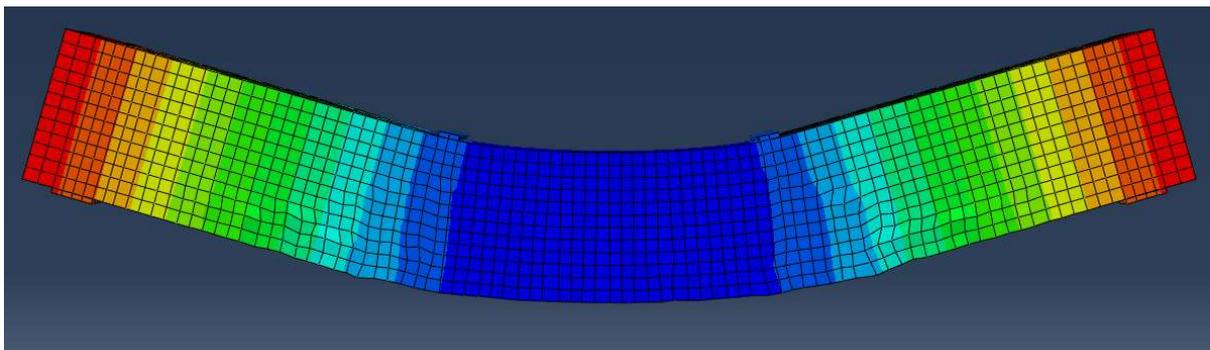


Figure 73

36deg - no shear cracking

By only changing the dilation angle to 36, which in the test runs for force displacement curve gave no difference from 31, - here has made the shear failure to not occur, as the increased dilation angle has expanded shear failure envelope.

More interestingly, was that going back to 31 degrees, then increasing the K factor from 0.667 to 0.8, it gave the same effect.

This means that for constructions where shear failure would be the dominant failure type, dilation angle would be critical. After correcting the shear reinforcement placement, it would no longer fail. As seen later on the eccentricity diagram, a high level of 0.15 or above would probably have had the same effect as 36 deg or $K=0.8$, and increased the shear failure enough to prevent the shear failure at 31 degrees.

Examples from other references about the dilation angle, includes this paper [(Masti, Maghsoudi & Rahgozar, 2008)] , which experienced the same as in this thesis, that the different models reacts differently towards dilation anglers.

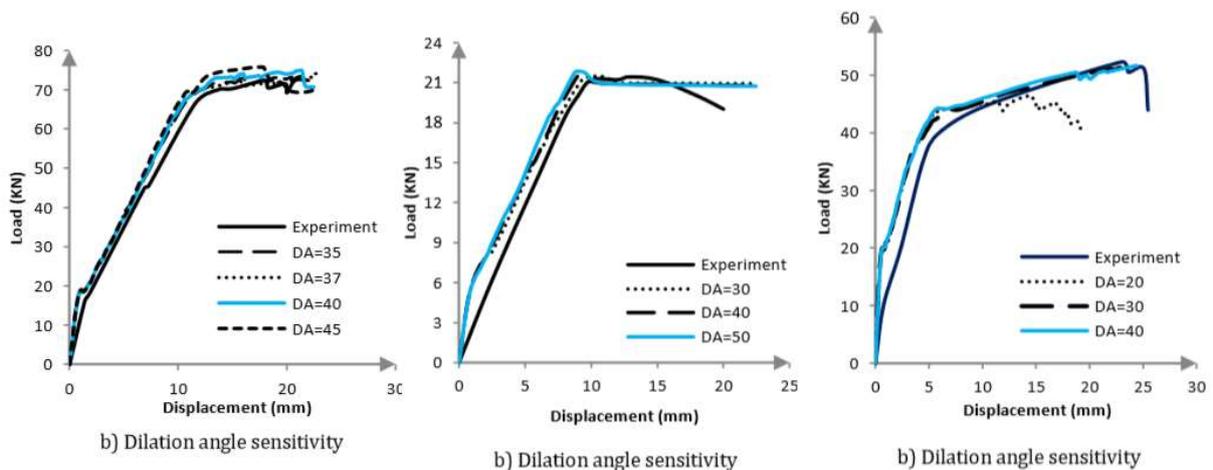


Figure 74

Mesh dependency

As one can argue about the reference above that both concrete and models were unequal between the three models, in my case the material was the same for both models, which indicates that dilation angle sensitivity is model dependent. If this is shell element or 2d / 3d dependent needs separate testing.

As for the default value of 31 degrees, I have found any conclusive answer, it is reason to believe the default dilation angle 31 deg is based on an experimental value, rather than a converted value from internal friction of concrete.

At the testing it was performed a triaxial test on concrete, where K value and dilation angle were derived. Here it was found a $K = 0.75$, correspondent with 30deg dilation angle.

At least in my models, the K value have little to none effect the case problem for this thesis, and the values above would be very equal to $K = 0.667$ and 31 dilation angle.

From paper that uses dilation angle different from default 31 (Esfahani, Hejazi & Vaghei, 2017), the main values are 36 and 38. Other papers uses 36 and 38 (Santuk & Pul, 2017), (Jankowiak & Lodygowski, 2013)

Many of these base their findings on how they fit a displacement curve, but I do not know if that would be the correct way of using the dilation angle, knowing the connection to the shear failure plane. Most do not state the reason for the use of dilation angle, which one could only guess if it's from the concrete internal friction, measured value or simply a value found in one of Abaqus's concrete examples or other references.

One reference that uses dilation angle to calibrate against old data is this (Michal & Andrzej, 2015), where the conclusion is to recommend dilation angle 5, or atleast below 15. As seen on the graphs on the next page, where the dilation angle is compared against old experiment with a confirmed dataset.

From the (Michal & Andrzej, 2015), the 5 degree comes form a series of replication of old tested data, here given by two examples.

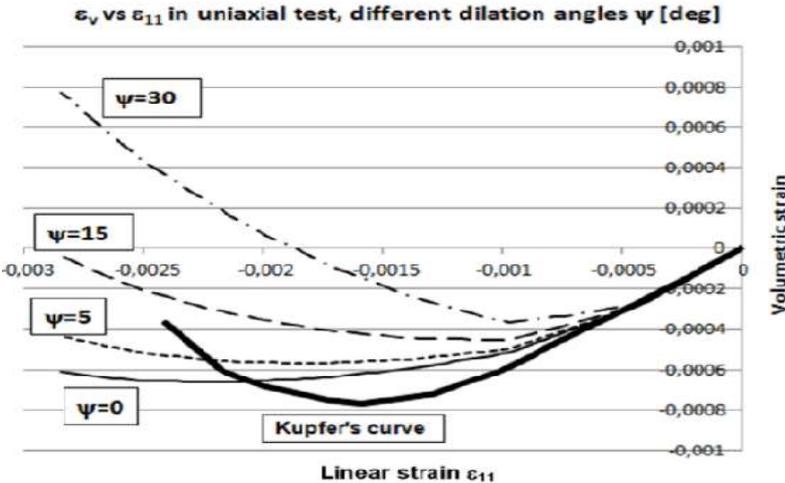


Fig. 14. Volumetric strain in uniaxial compression test

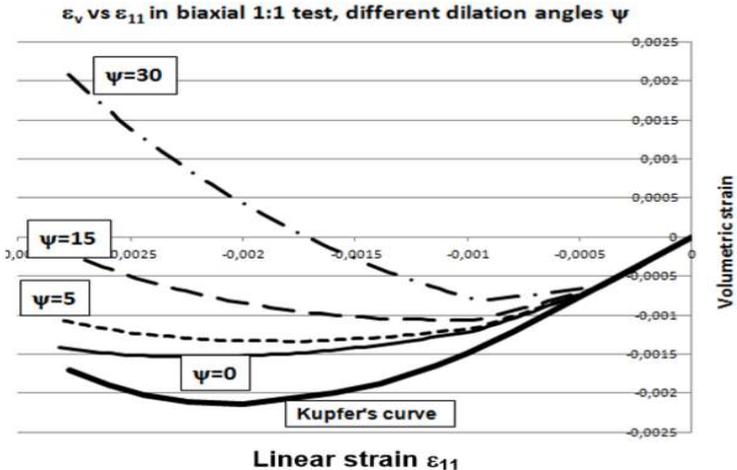


Fig. 15. Volumetric strain in biaxial 1:1 compression test

Figure 75

(Michal & Andrzej, 2015)

In this paper, the K value and Eccetricity was kept at default. As seen on the figures, it is recommended to keep the angle below 15*, or else it would produce an positive volumetric strain, and an overly stiff concrete.

Eccentricity:

See Figure 67 (Actual Q-P plan for CDP), in dilation angle. With increasing level of eccentricity, the increases the curvature of the flow rule to more quickly gain the shape of the dilation angle. This means that a high eccentricity has a higher level of dilation angle for the entire yield span. Default 0.1

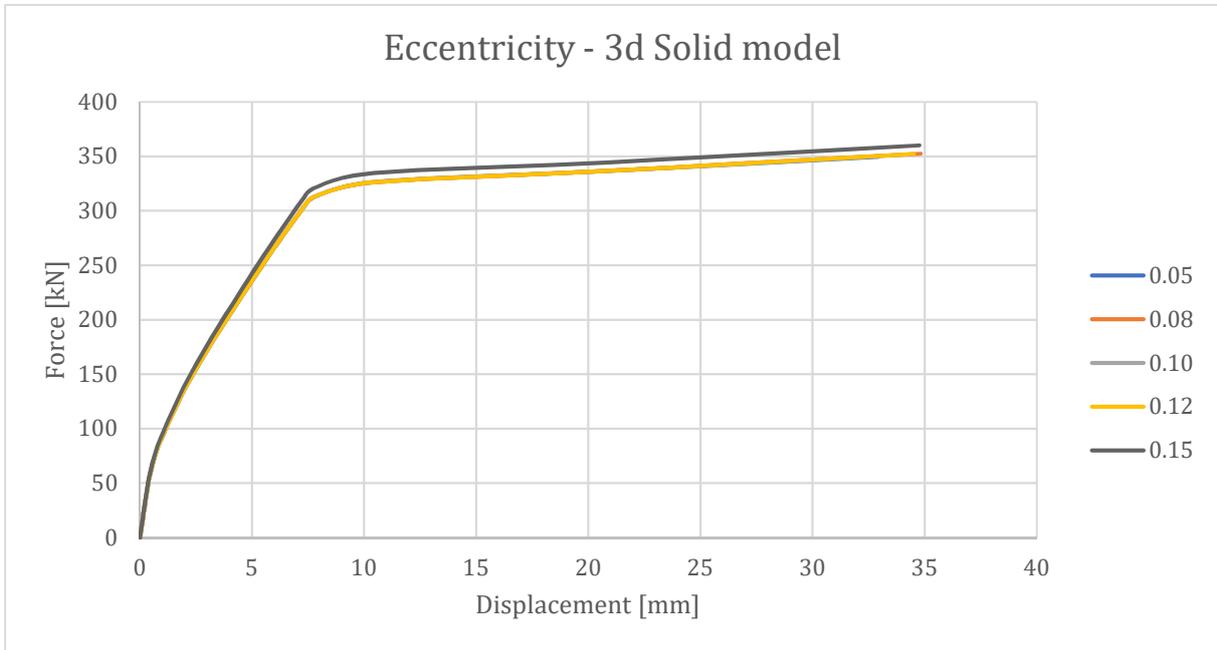


Figure 76

3d solid

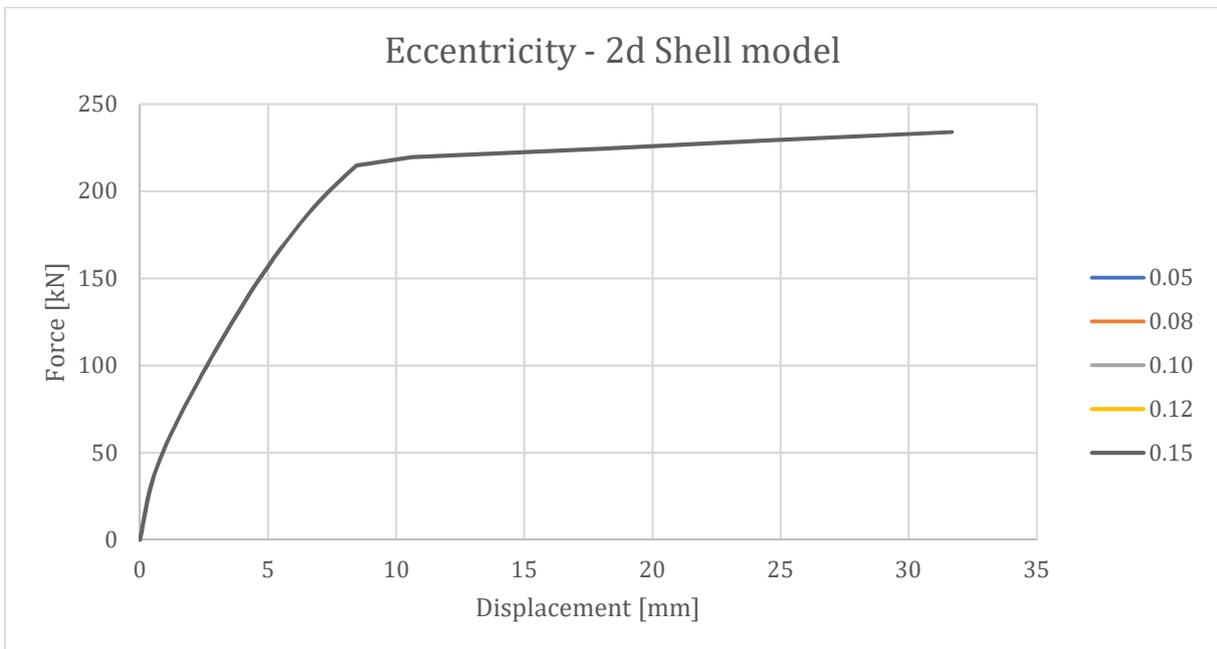


Figure 77

2d shell

All curves follow the same shape, except the 0.15 value on the 3d solid model, which increases the hardness. The differences between the eccentricity parameter values tested are neglectable considering the problem at hand.

K parameter:

Default 0.667

In literature studying cdp parameters, this one is just set at $2/3$ as it's a common ratio.

A K factor of 1 is regarded as a full von mises curve, as a value less than 1, is following the decker-flocker diagram

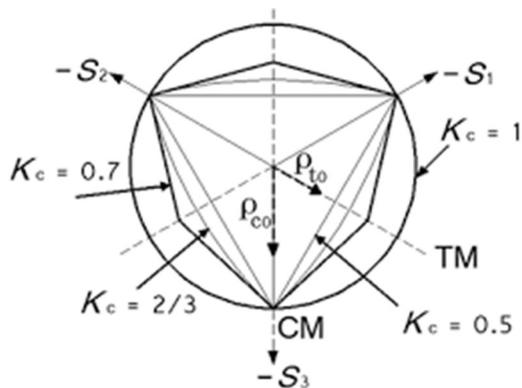


Figure 78

Relation K value and θ

K_c is the ratio of the second stress invariant on the tensile meridian to compressive meridian at initial yield with default value of $2/3$ (Abaqus User Manual, 2008). The parameter K_c should be defined based on the full triaxial tests of concrete.

The link between the dilation angle and K value, could loosely be defined as if the K value goes up, the dilation angle needs to decline, to have the same stiffness in the model.

The parameter K_c should be defined based on the full triaxial tests of concrete, moreover biaxial laboratory test is necessary to define the value of σ_b0/σ_c0 . It does not discuss the identification procedure for parameters ϵ , σ_b0/σ_c0 , and K_c because tests that are going to be verified in this study do not have such information

K-Parameter effect on models

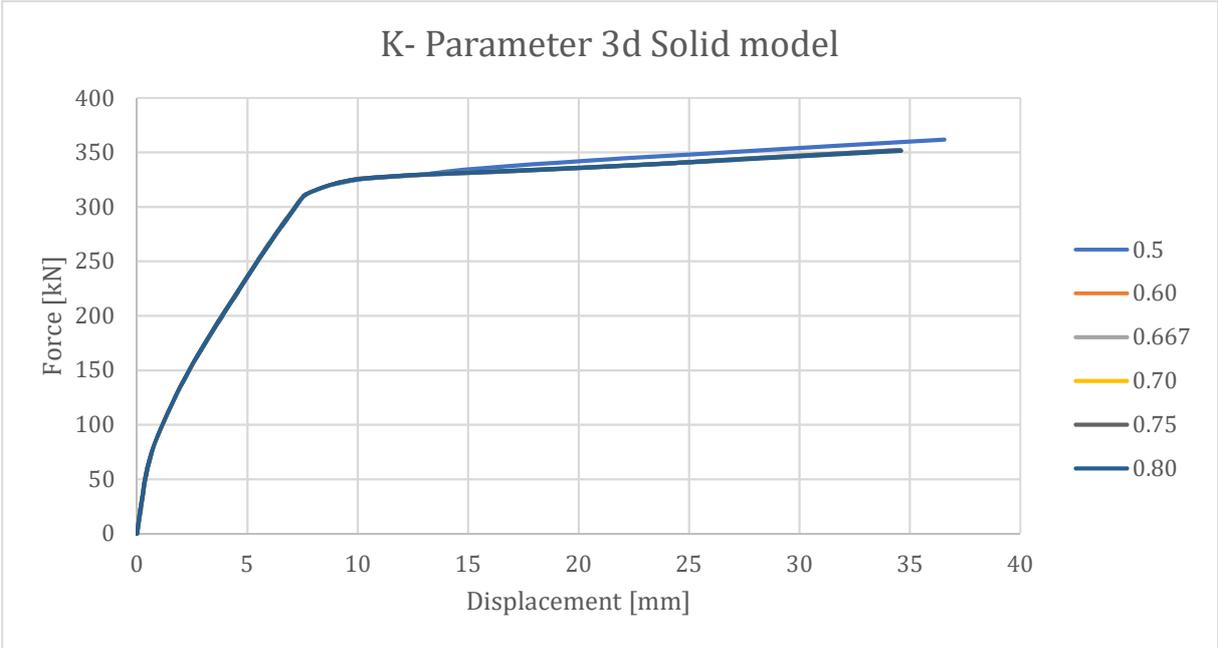


Figure 79

3d model – only the K=0.5 value stands out

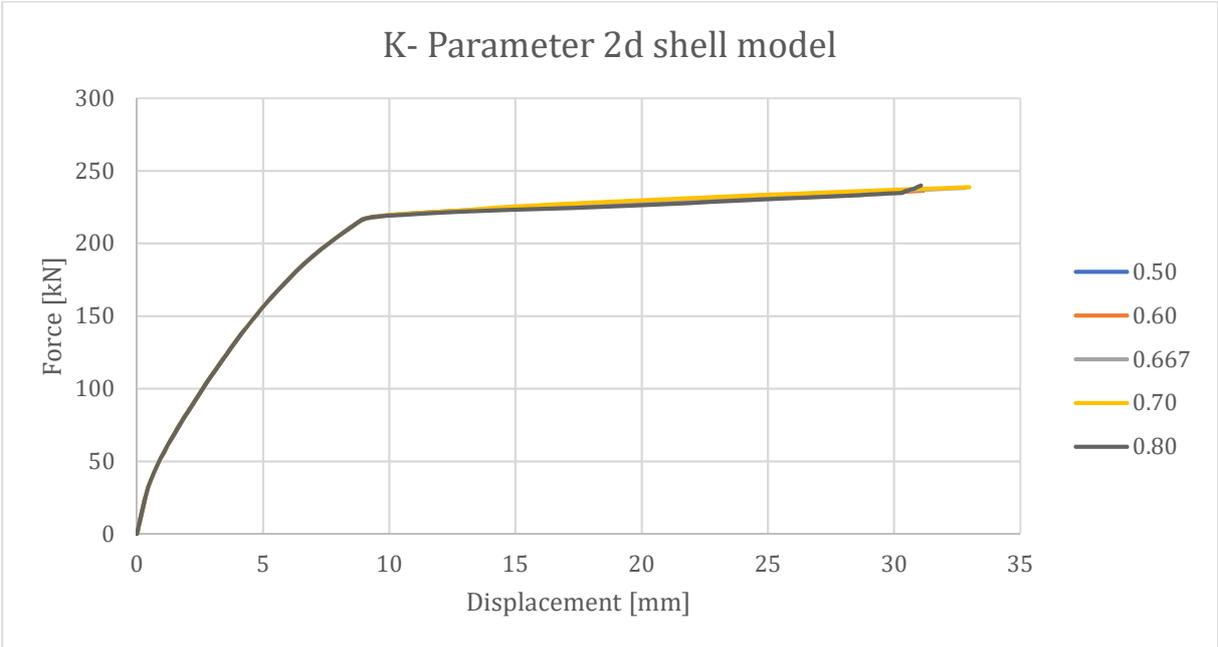


Figure 80

2d mode - all remains very similar

For this kind of problem, without shear failure, the K parameters effect is neglectable for both model. As for the 3d solid model, an increase in stiffness is only registered for the K = 0.5 . As an increase in K gives larger shear failure value, it at the same time decreases the overall stiffness of the beam.

Biaxial parameter:

Fb0/fc0: default 1.16

The fb0/fc0 and K parameter is defined as the maximum stress found in bixaxial testing compared to uniaxial testing, where the ratio is the parameter. This is not done in this thesis, and the only way to set this parameter is through testing. The only other value I have seen used is 1.12.

Viscosity parameter :

Viscosity parameter is often referred to as relaxation parameter, and could potentially have a major impact on results from the models when in use. Default is 0.

This parameter is used to allow for mass scaling, to suppress numerical instabilities, overcome convergence problems, and to speed up the time spent on calculations according to the Abaqus manual. At default 0, Abaqus does not enable any viscosity, or relaxation.

The reason for adding the viscosity, could be convergence issues when doing non-linear problems, seen below form an early testing , with a different concrete input, with a severe drawback is selected too high.

Another thing with this parameter, is that it is directly linked to the time frame of the applied load. In all studies in this thesis, load is applied within a frame of 1 second. But if you do some dynamic and the applied load frame setup is different form 1 second, these values are not representable again, and need to be set accordingly to the load time frame Szczecina and Winnicki (2017) for more detail. Proposed Viscosity Parameter Limit.

General Solver	0.0	0.00001 max
RIKS Solver	0.0	0.0001 max

These limits are set to the highest usage of viscosity parameter while keeping accuracy.

For the general solver, I highly recommend turning on the stabilizer function rather than using the viscosity parameter if encountering convergence issues.

As during testing, and specially with the RIKS solver, issues could occur with convergence if the parameter was left at 0, while loading into the non-linear area. The same issue could be found to a lesser degree for the General solver.

The best way to show the effect of viscosity on a micro structural plan, is perhaps by showing pictures form where the effect had been studied on a fracture case.

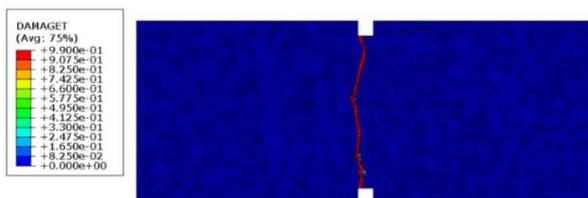


Fig. 5. DAMAGET for viscosity parameter = 0

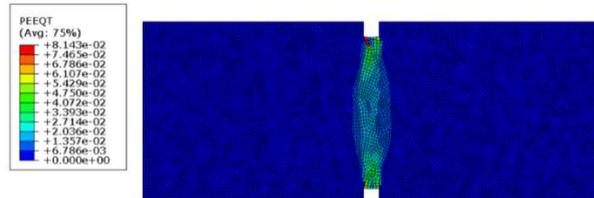


Fig. 8. PEEQT for viscosity parameter = 0.0001

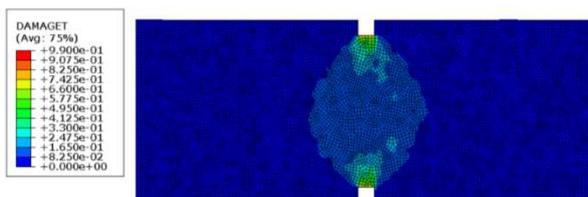


Fig. 9. DAMAGET for viscosity parameter = 0.001

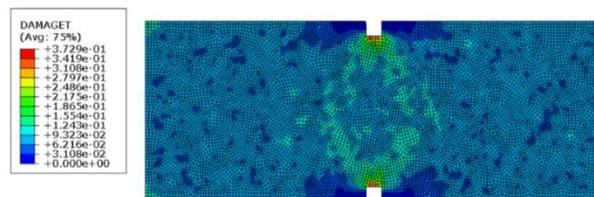


Fig. 11. DAMAGET for viscosity parameter = 0.01

Viscosity parameter effect []

Crack propagation is clearly defined with viscosity = 0, and is smudged into something unrecognizable by leaving the parameter at 0.01. The effect is severe.

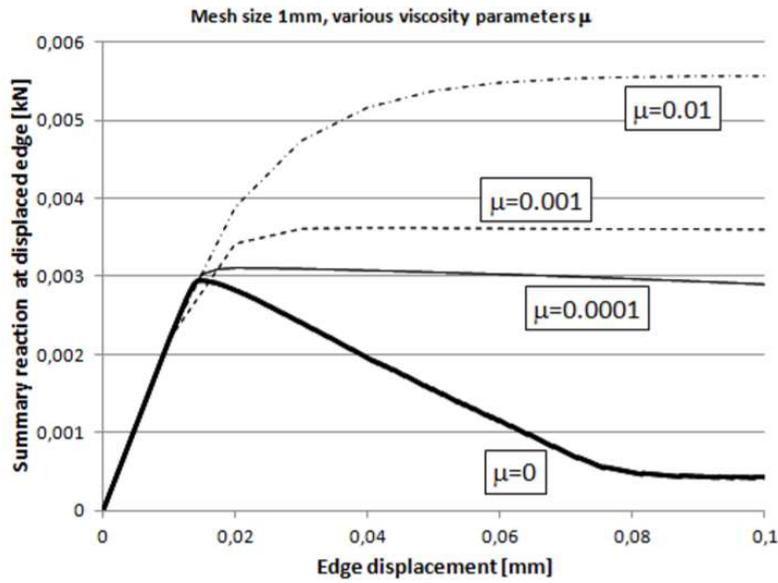


Figure 82

Load at the tip of a crack []

This is a stress displacement diagram from a crack tip, where the stress rises on the edge due to the crack. As we move inward in the material, it should decline as shown with viscosity = 0. The errors presented with increasing Viscosity effect is massive, and is backed up by the test from the models.

Viscosity Parameter effect on models

General solver :

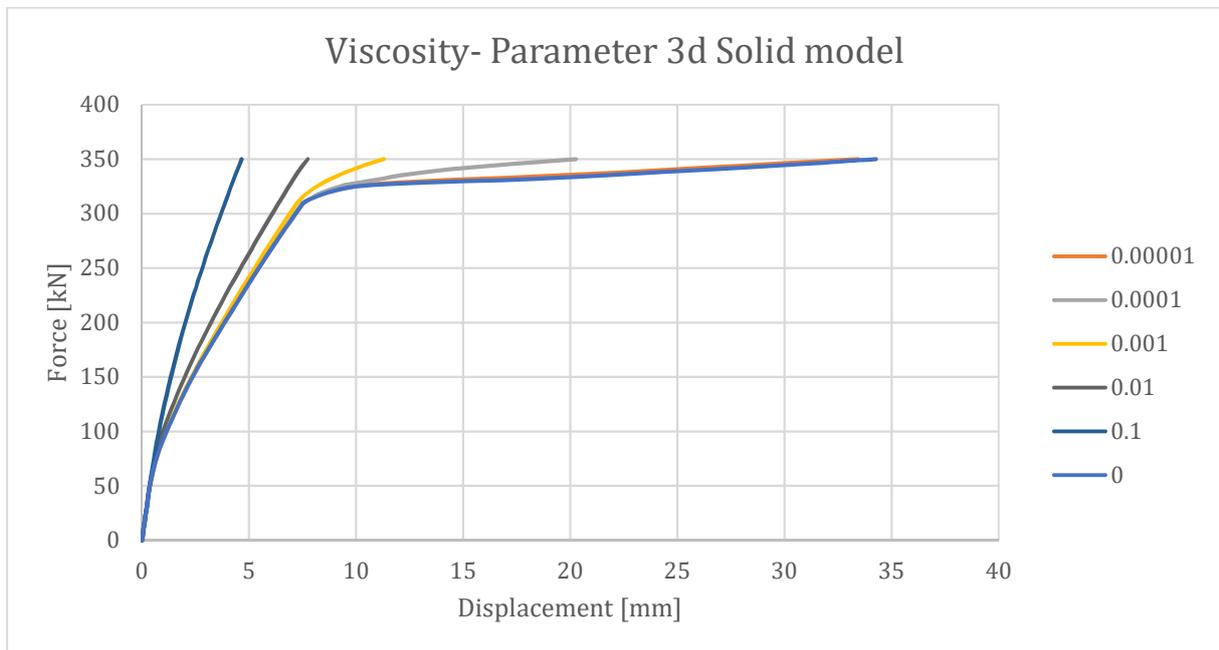


Figure 83

Viscosity parameter tested 2d shell model

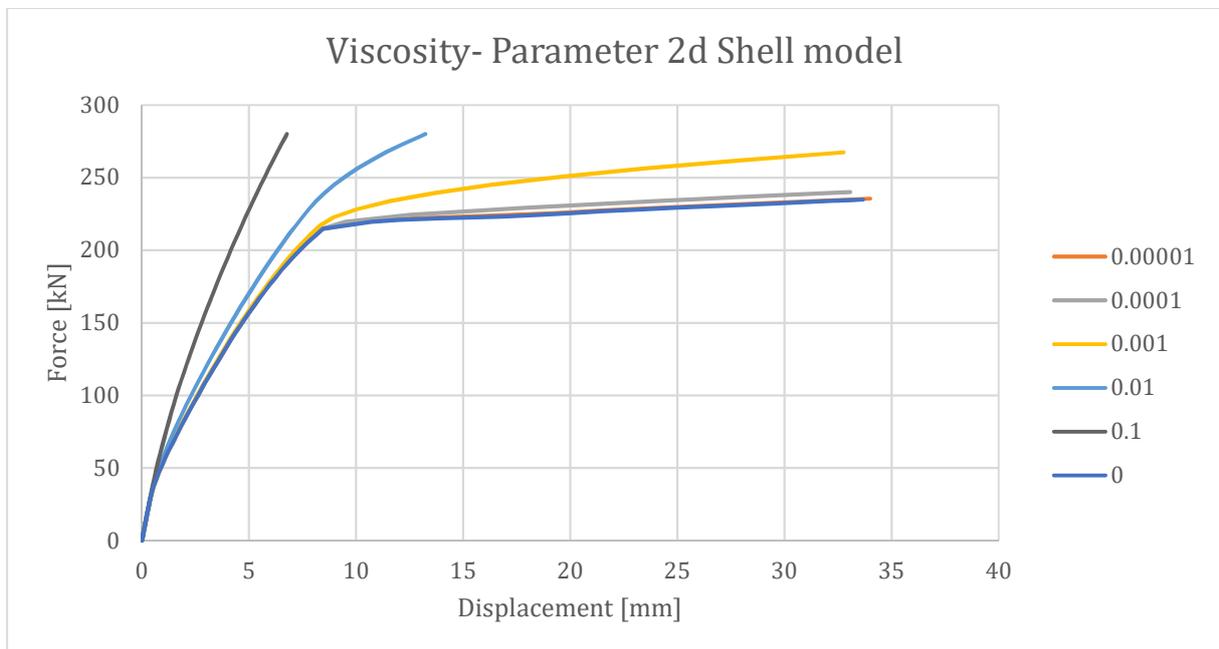


Figure 84

Viscosity parameter tested 2d shell model

It is very noticeable that going above 0.00001 , quickly creates huge errors in representation, for the General solver.

3D Solid model: E.c = 25GPa, fck=20MPa, f.ctk=1.8MPa, Es=200GPa and input form testing.

Here a more softer concrete than the standard B35_Default is used to more quickly push the model towards the limits.

Total Load applied					200 kN	250 kN	270 kN	280 kN
G	0	E25	B20	PL_true	5,742 mm	Not tested	17,79 mm	Not tested
G	0.00001	E25	B20	PL_true	5,738 mm	Not tested	17,79 mm	Not tested
G	0.0001	E25	B20	PL_true	5,770 mm	Not tested	12,01 mm	Not tested
G	0.001	E25	B20	PL_true	5,492 mm	Not tested	11,24 mm	Not tested
G	0.01	E25	B20	PL_true	5,035 mm	Not tested	7,83 mm	Not tested
G	0.1	E25	B20	PL_true	3,438 mm	Not tested	4,94 mm	Not tested
R	0	E25	B20	PL_true	5,797 mm	failed	failed	failed
R	0.00001	E25	B20	PL_true	5,792 mm	9,782 mm	25,81 mm	failed
R	0.0001	E25	B20	PL_true	5,792 mm	9,410 mm	29,69 mm	47,71 mm
R	0.001	E25	B20	PL_true	5,798 mm	9,271 mm	28,01 mm	28,35 mm*
R	0.01	E25	B20	PL_true	5,486 mm	8,116 mm	10,99 mm	16,17 mm
R	0.1	E25	B20	PL_true	4,408 mm	5,973 mm	6,71 mm	6,95 mm

Table: 9 (3d solid model, viscosity limits) * Deforms incorrectly

From table 9 , we can see that the RIKS solver is less effected, as it still is severely effected if The effect where tested on a 2d and 3d beam, with both the General and the RIKS solver.

As for the RIKS and General solver in the 2d shell model, they follow the same path follows the same path as for the 3d model, struggling with convergence if not used.

From these findings, my proposed limits of 0.00001 for General, and 0.0001 for RIKS, is set.

The reason why RIKS is less effected, has to do with time. As general solver uses 0-1s as time frame, RIKS can stretch over 1-6 seconds on a solution. This means that the parameters value is stretched over a wider timeframe, and the effect is smaller. The effect is then not immediately clear, as the runtime varies with the load, but the number from the tests shows that a us of 0.0001 or less, grants good results.

The results also show that sane usage of the viscosity parameter is has good effect on convergence while keeping the precision spot on.

To get the majority of CDP material input to run in the non-elastic area, viscosity is needed.

Materials:

Concrete is viewed upon as a homogenous material, but it will have differences within, as it is influenced by a vast number of factors until in service, and even then continue to change. This makes it interesting to see how the material input, effects the models, and if a change in material input has more effect than the CDP parameters.

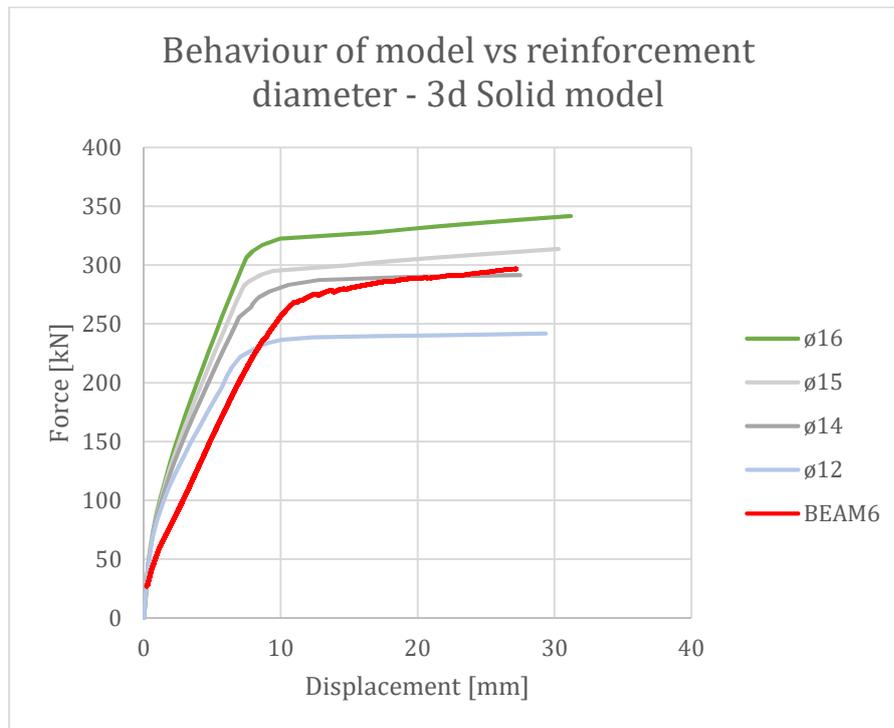


Figure 85

Effect of changing diameter of reinforcement

The first figure is to ensure that the reinforcement works. The imaginary size ø14 and ø15 only show the steps upwards in stiffness to ø16.

First of all, I noticed that almost all models of non-linear concrete, produced the classic straight line in the hardening area, instead of a hardening curve up maximum load capacity.

As we have tensile data from the rebars used, it would be interesting to see if the rebar data could contribute to a hardening line, or if this is a concrete dependent behaviour. Form the calculations, the tensile both behaviour and strength comes from the rebars.

Abaqus states in the manual, that material to input as true stree and true strain. Trail run to see if there is any difference, between s500 , S556 and PL_eng and PL_true, see appendix for detail. As seen below, all three (500 = 556 with a shift in yield), gives a hardening curve, and do not differ very much. But if seen for the [Lab_3r] concrete, it is clear that the more detailed reinforcement gives a more correct hardening curve for the RC beam.

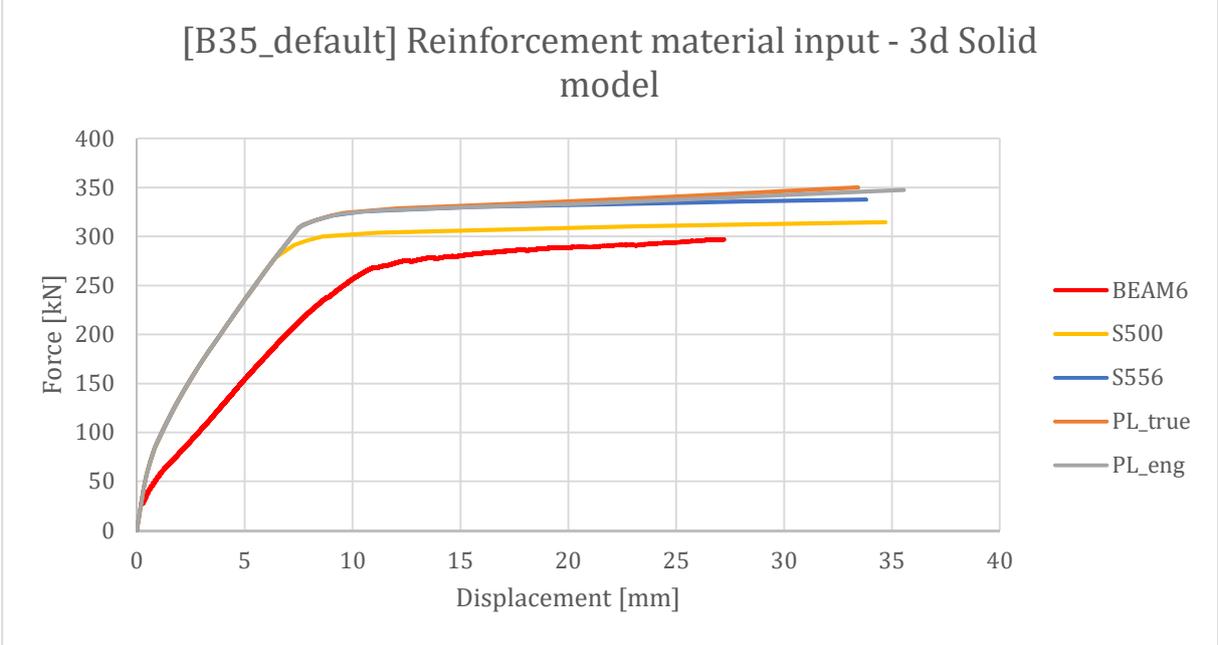


Figure 86

Input for [B35_default]

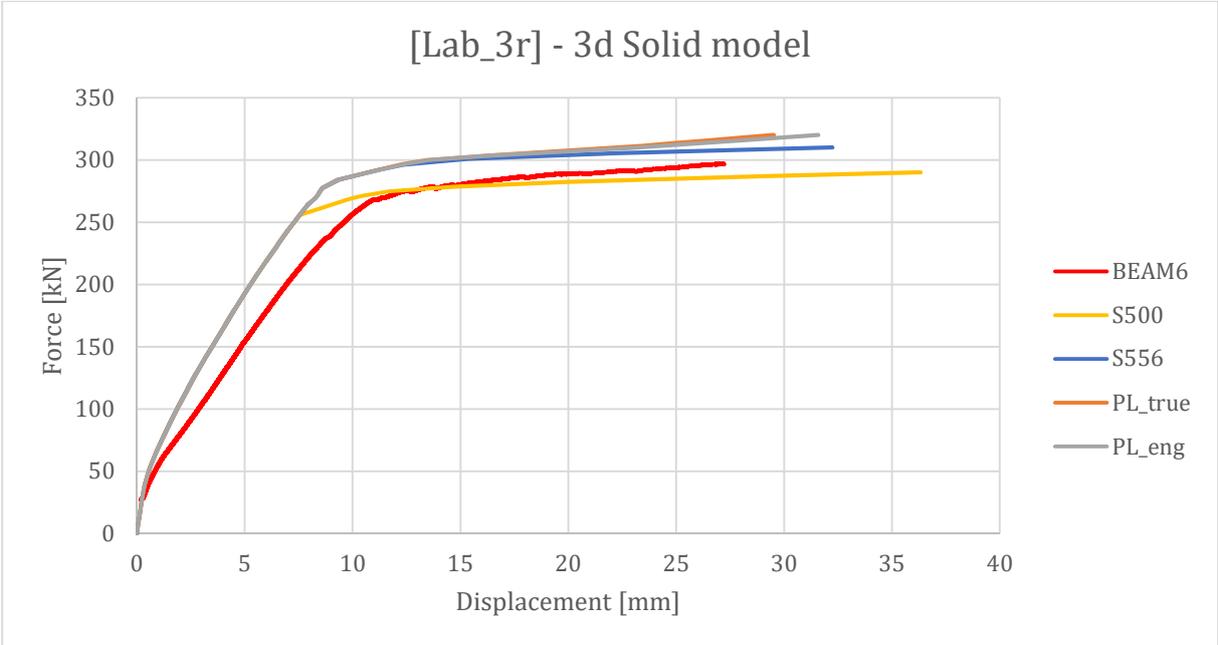


Figure 87

Input for [Lab_3r]

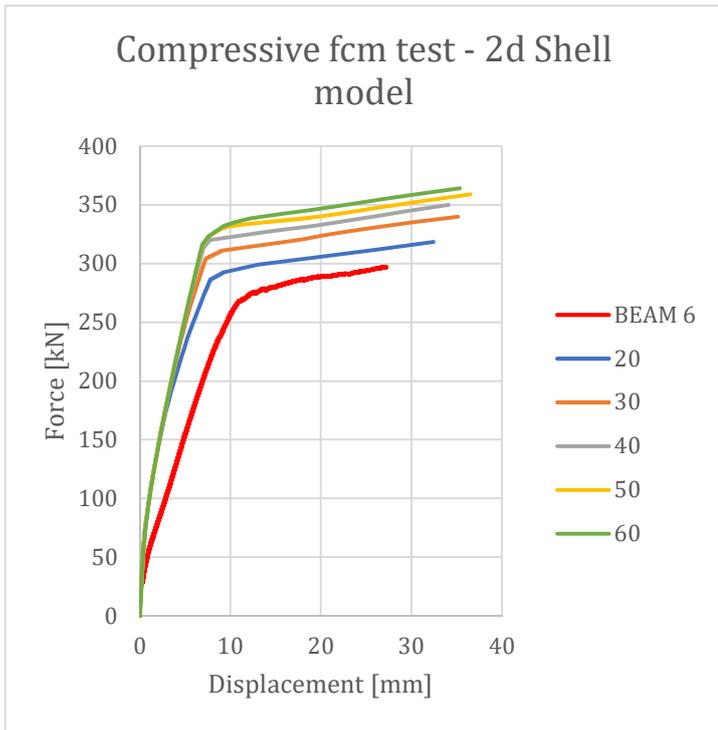


Figure 89

Compressive

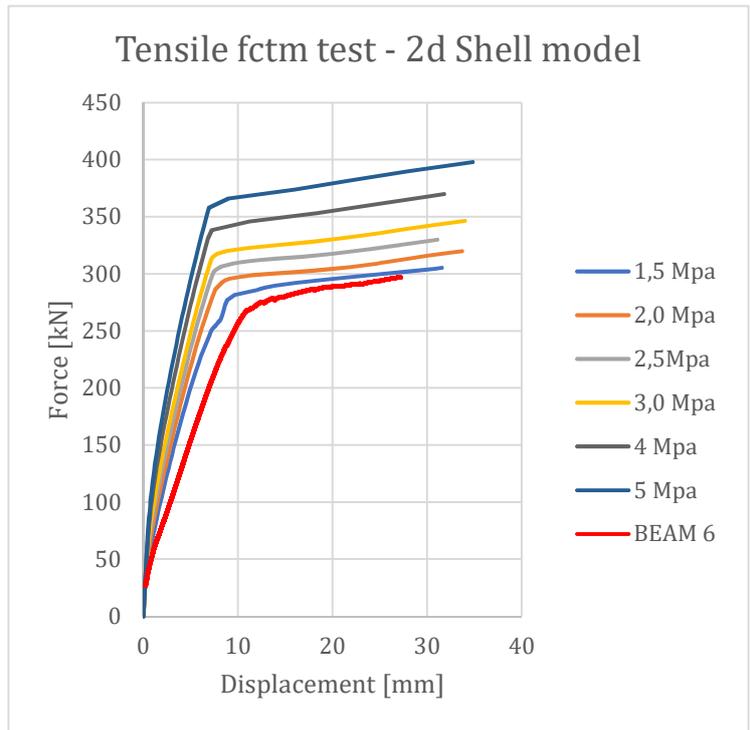


Figure 90

Tensile

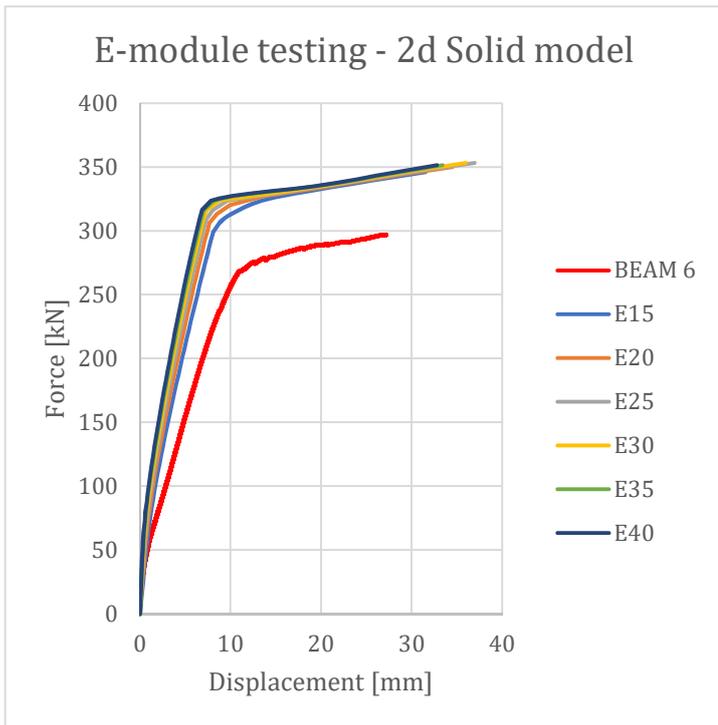


Figure 91

E-module

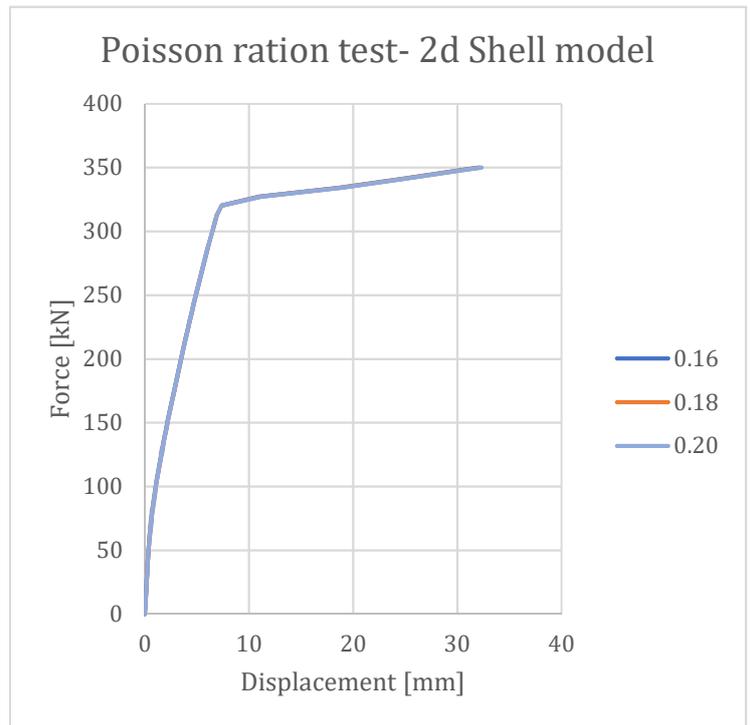


Figure 88

Poisson ratio

Default for these numbers are $f_{ctm} = 3.2$, and $f_{cm} = 43$ and $E_c = 34\text{GPa}$ from EC2.

As seen on these figures, the biggest impact is from the tensile strength as it shifts not only the strength, but also the overall stiffness reaction of the model. Poisson's ratio is neglectable.

Material adjustment:

As the concrete [B35_default] is a perfect candidate to check model functionality and compare them to each other, it is a very poor representation for the reinforced beam tested. It vastly overestimates the load bearing capacity, and needs to be replaced.

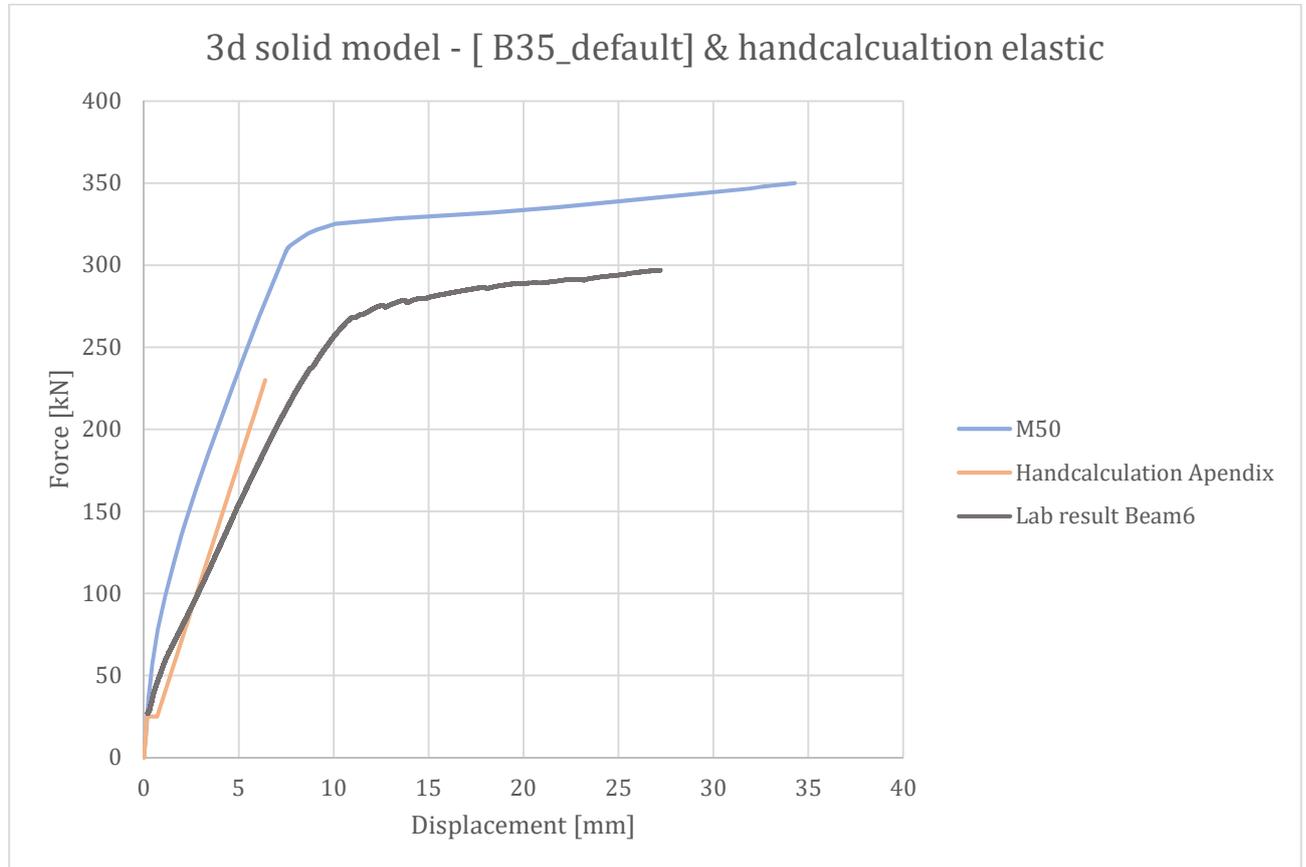


Figure 92

Test [B35_default]

In addition, hand calculations with B35 EC2 inputs are used to estimate the BEAM 6 behaviour. It's a bit softer than expected, as one can see the curve of the hand calculation and [B35_default] follows each other very well over the elastic region.

From the material testing chapter, the strain from the concrete compression test was put in question, and chosen to be disregarded.

In addition, it was always meant to find a way to define the tensile concrete strain softening behaviour. One choice is linear decline, as in the Abaqus manual and used in Hognestad stress strain curves. Another choice is to use Thorenfeldt stress train curves, as the compression result just barely puts in into the "high strength = 60 MPa " concrete definition for the use of Thorenfeldt curves.

Two preliminary concrete types were constructed using the test results.

The E-module of 29680 MPa was kept, so was the 61 MPa compressive strength, and the 2.11 MPa tensile. One tested the far-fetched possibility of the strain curve being an inelastic curve, as this would fit quite nicely to the numbers, combined with a linear 2.11 MPa tensile no end input.

The other used a "f_{ck}" version, with 53 MPa instead of 61 MPa, and combined with the [B50-paper 1] compressive strain values, and 2.2 MPa tensile with end values 0.02 MPa and 0.0038 strain.

Another concrete was built up from the B35 EC2 chapter 3.1.5 formula, with tensile 3.2 MPa and end points 0.032 MPa with 0.00120 strain.

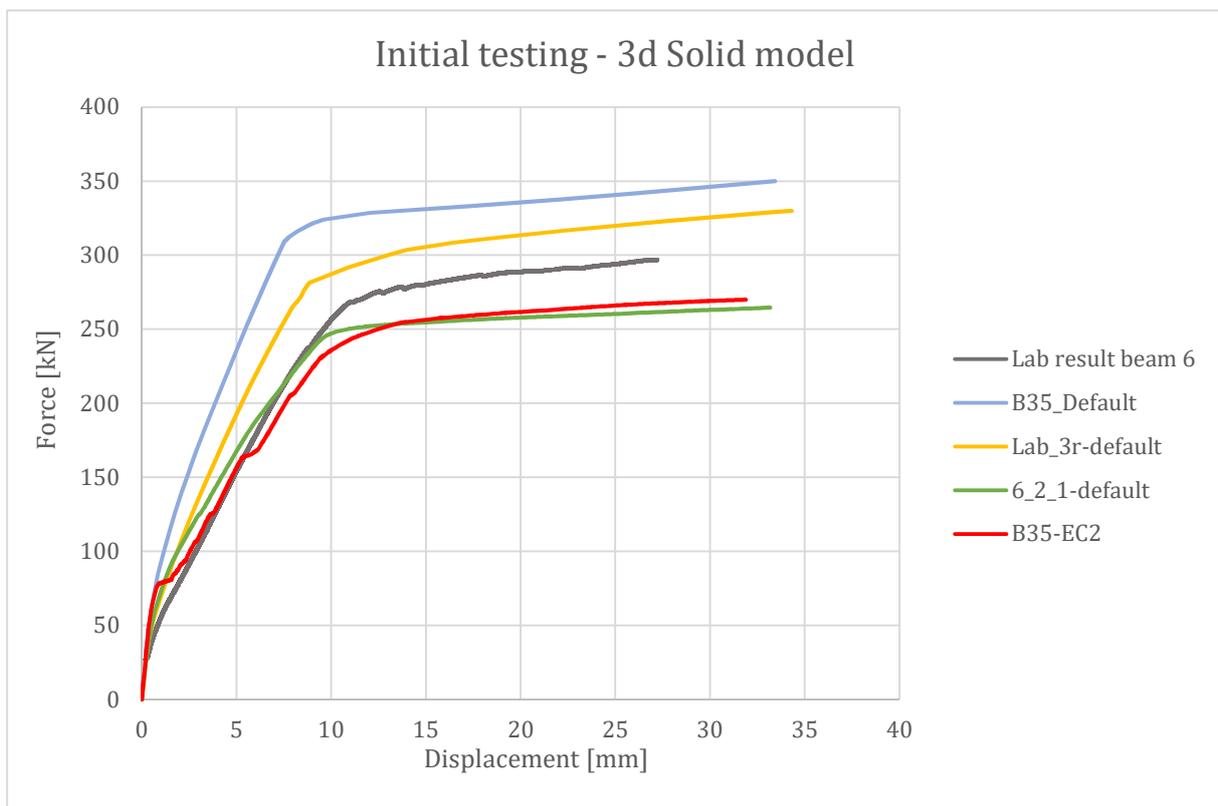


Figure 93

Initial testing

From the graphs above, the EC2 formula created concrete was the best fit, but it was very difficult to work with, as it had convergence difficulties and produced a lot of errors during calculation. This is also reflected in the curve, and due to difficulties, is not an option.

Next is the green curve, which has a semi fit for the elastic region, while missing the non-linear part completely. And if one looks closely, the starting point is just not right, as the concrete material input is stiffer at the beginning.

To better be able to compare the different and find “normal” variety of compressive and tensile stress inelastic strain, different CDP concrete’s were plotted.

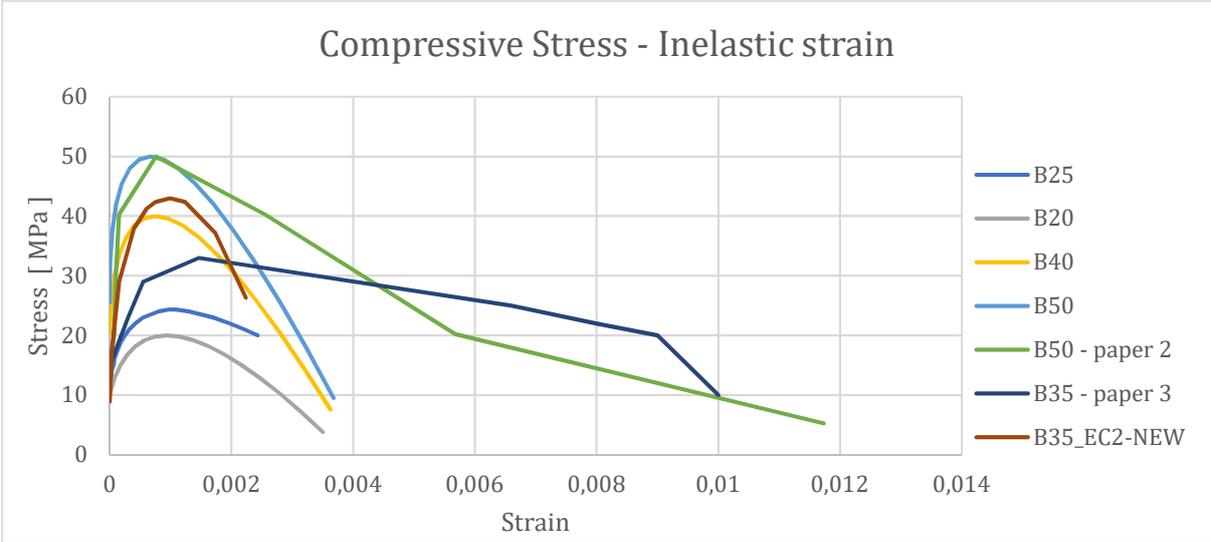


Figure 94

Compressive, stress inelastic strain

Concrete type [6_2_1] is built on the inelastic strain curve of B50 above.

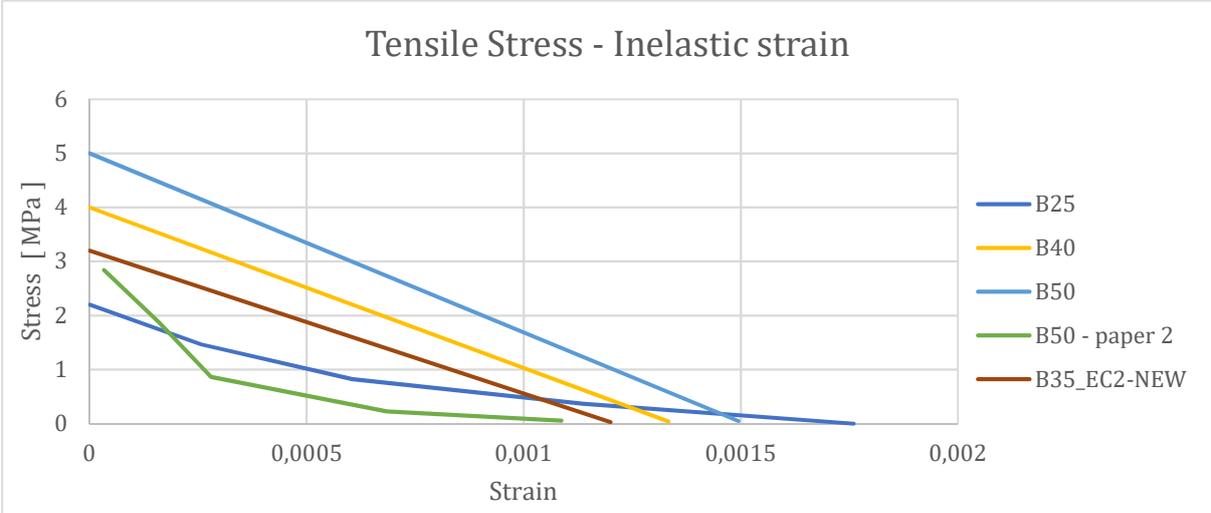


Figure 95

Tensile, stress inelastic strain

By looking at the compressive inelastic strain, they differ quite a bit as from the range of 0.004 up to 0.01, which is considered as large step, especially as the EC2 strain stops at 0.00223.

Tensile strain is a bit more consistent, as it lands in the region of 0.0012 – 0.0018. Here the difference is more if they are represented by one linear line, or multi linear stepwise representation. For the B35-EC2, I gave it a linear tensile stress inelastic curve.

B25 is an unconfirmed versatile concrete setup as it is very soft and easy to from, see appendix example for example.

Next thought was that it might have something with the bonding between the reinforcement and concrete.

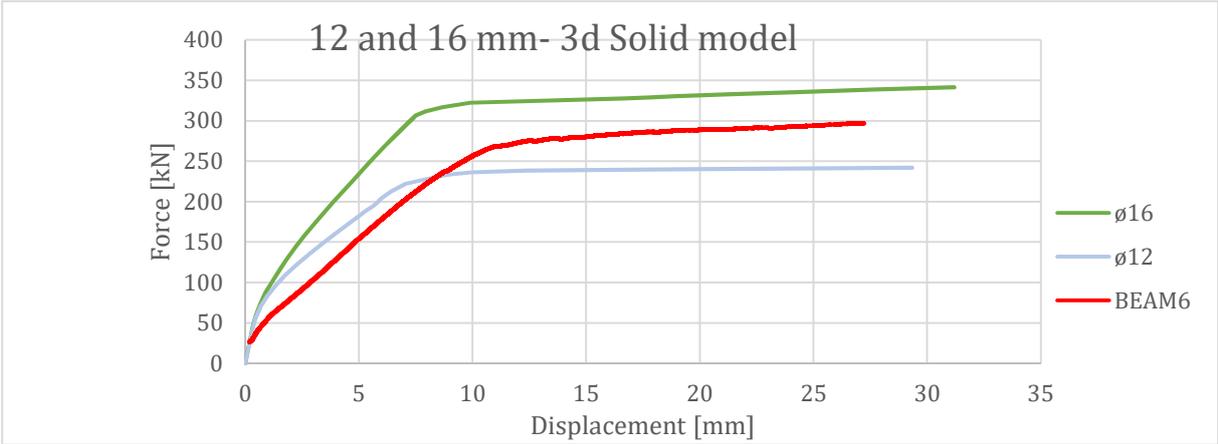


Figure 96

Full bondage assumption

The idea behind this figure, is that if there was a bond problem, it would behave as if there was less reinforcement present, which there is if one switch to ø12 instead of ø16. But as seen above as tested previously, the BEAM 6 curve is much more alike the ø16 curve than the ø12 curve, both in the linear and non-linear area, which means the full bondage assumption of Abaqus is correct. To determine which part dictates which behaviour, the next setup where done line this, as the two swapped their tensile parameters.

In addition, both were tested with Thorenfeldt generated strain curves for 2.11 MPa fctm.

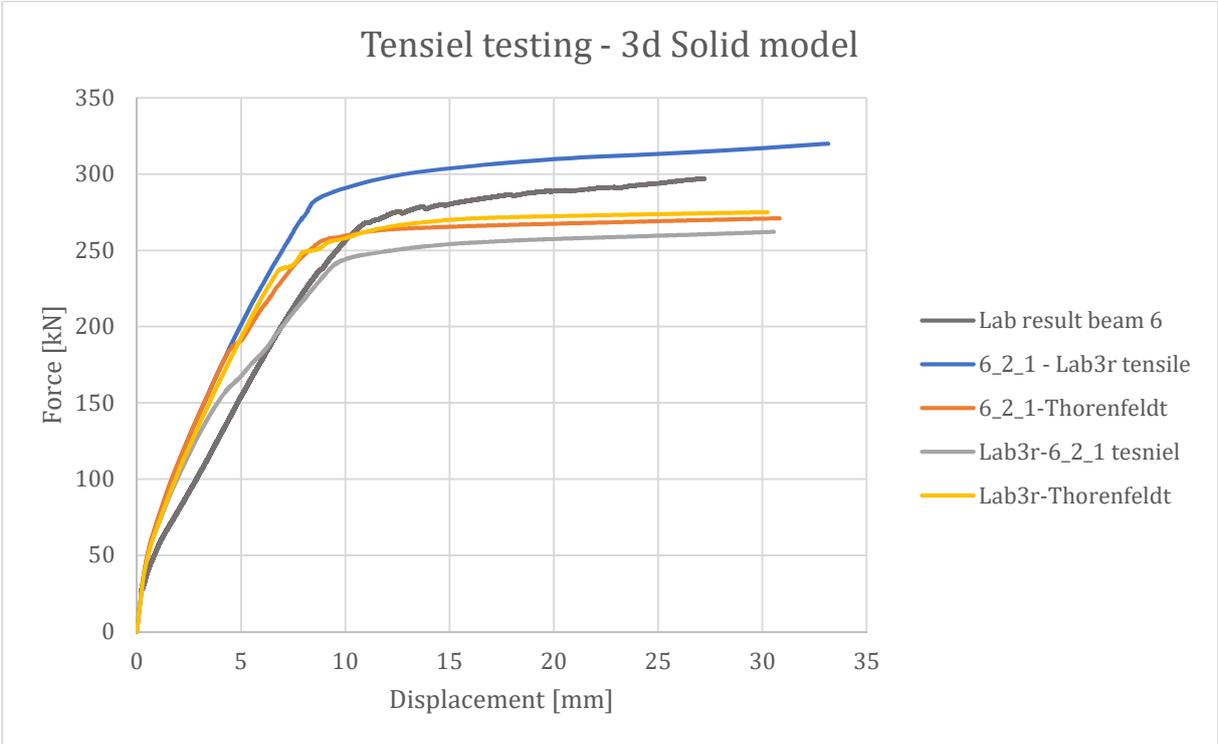


Figure 97

Tensile testing

Many other combinations within the parameters measured, but these were the closest once.

As none of these is a good fit, and they do not solve the issue about the initial stiffening being too steep, next is to assess and test CDP concrete inputs from literature.

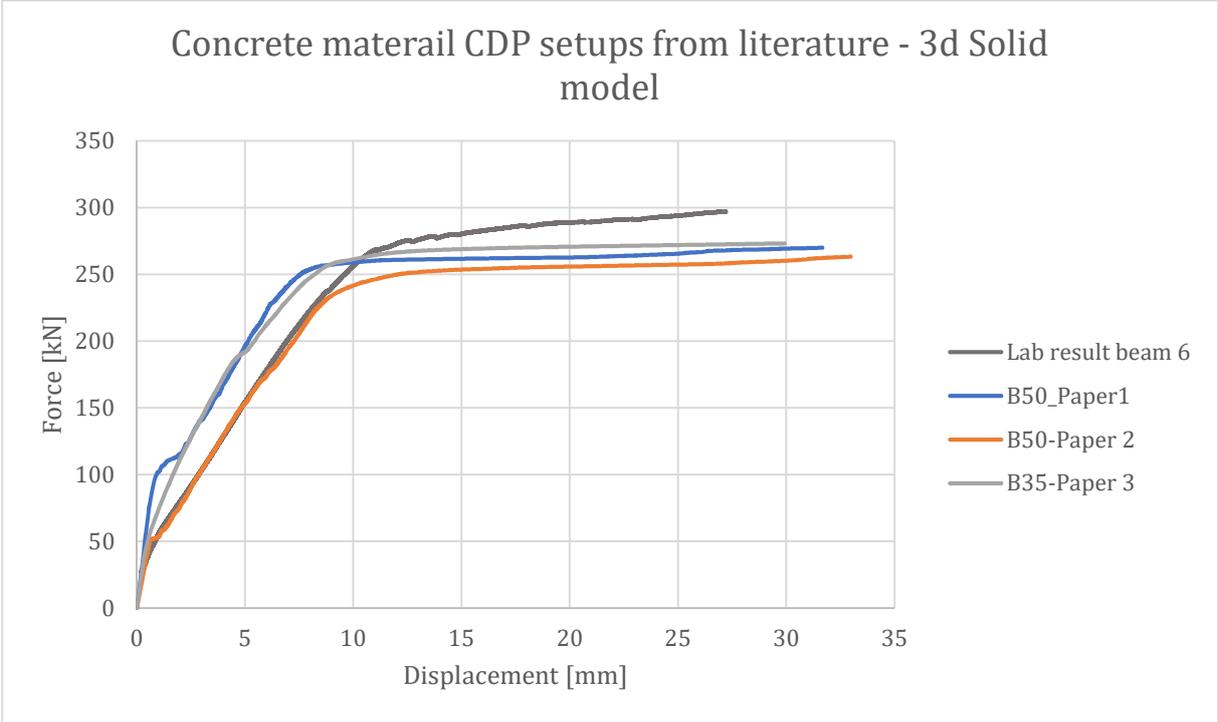


Figure 98

CDP concrete inputs from literature

From this plot, the B50-paper 2 actually fits really good, and will be selected for the crack width testing. As is it far from perfect with an off-set of 25kN in the non-linear area, the elastic region is more or less identical, and it follows with the same initial stiffness.

The reason why it fits, is much due to the lower E-module of 19.7 GPa, and probably the more ductile compressive strain, see Figure 96 and 97.

Ref , see appendix for details.

B50-paper 1 = (Esfahani, Hejazi & Vaghei, 2017)

B50-paper 2 = (Jankowiak & Lodygowski, 2013)

B35-paper 3 = (Santuk & Pul, 2017)

Crack width:

Through stadium 2 and 3, cracking of the will occur. During load testing in the lab, visual and camera-based inspection (DIC system) was done to estimate crack size.

The idea behind the use of the CDP module, was to build a model with material input from the lab, to accurately represent the tested RC beam, where data from the reinforcement could be extracted in the non-linear area to give input to the EC2 formulas for crack width and compare with the table above.

As in the material adjustment, I did not manage to create a sufficient material input from the lab testing to match behaviour of the test beam, which kind of defeats the purpose of this chapter and thesis, as there is no sense in comparing data which by visual inspection will not fit. Back up solution was to see if one could find a material input which could fit the data.

The B50 (Jankowiak & Lodygowski, 2013), is a good candidate for the elastic part, but is does not cover the non-linear part, as it was not possible to run it further due to convergence issues.

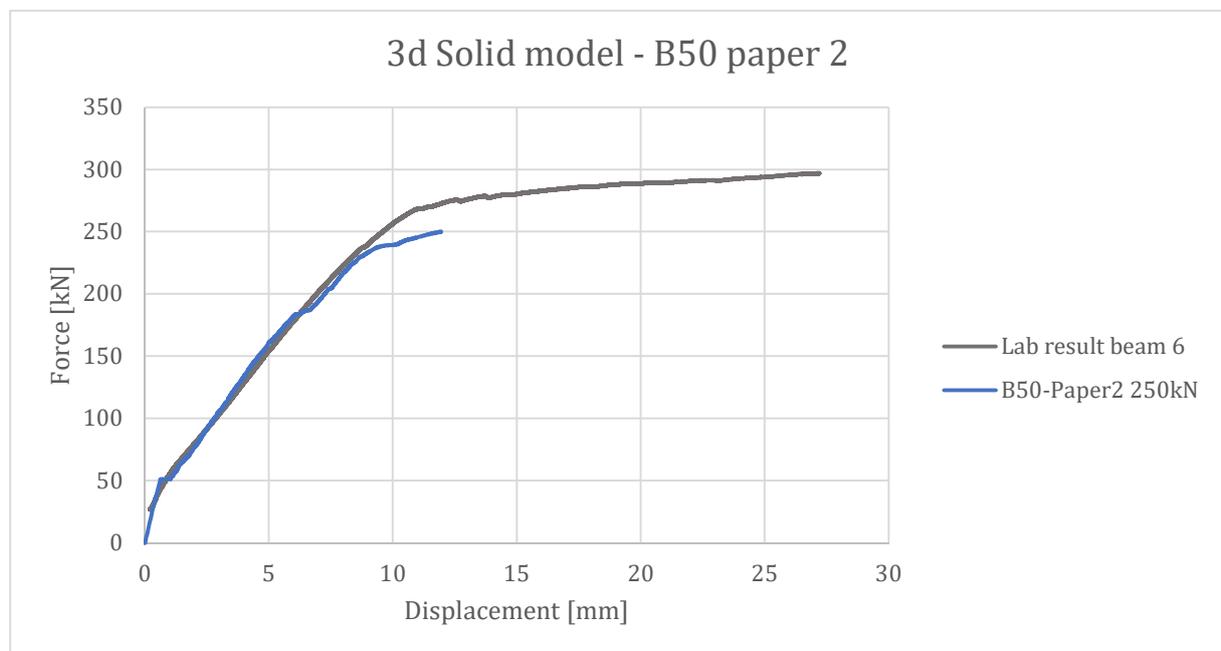


Figure 99

Target curve

Further the damage function in CDP will be used and evaluated, but the built-in Abaqus function "XFEM crack" will not be used. This is a very useful function, as it can predict and calculate crack propagation for a specific crack, but its drawback is that it requires a start point, or line. And as this thesis seeks to find these start points, the function is not applicable and will not be used.

Overview reinforcement for 3d solid beam.

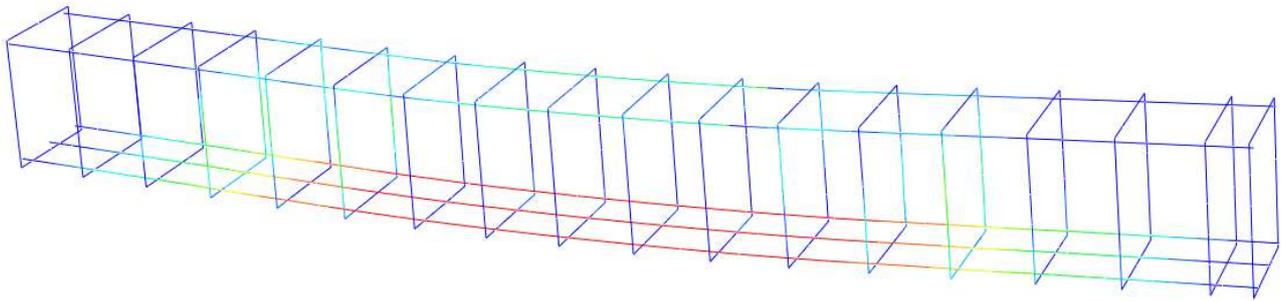
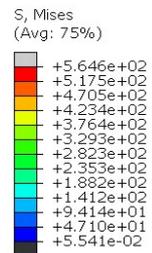


Figure 100
Reinforcement 3d solid model

Here plotted with von Mises, it is possible to see how the load is transferred through the beam from the load points and down in an angle towards the supports, through higher stresses on affected shear stirrups.

As noted earlier, this beam's compression capacity is much greater than its tensile capacity, and it is easily shown here, as the top reinforcement barely passing 180 MPa, the tensile reinforcement is pushing close to its yield strength.



This may be easier to show through max principal stresses, here at different loading stages to show how the crack propagation close to the supports first will go upwards, then inwards, as the loading continues.

Load transfer

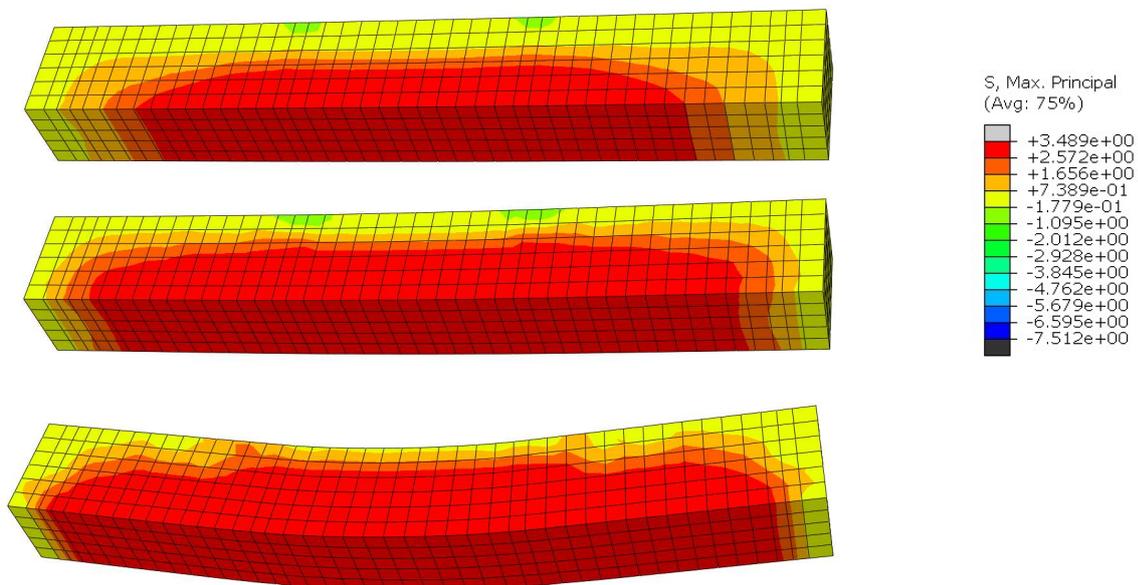


Figure 101

Principal stress concrete only

Crack propagation seen through the tensile damage parameter, for the 3d solid model

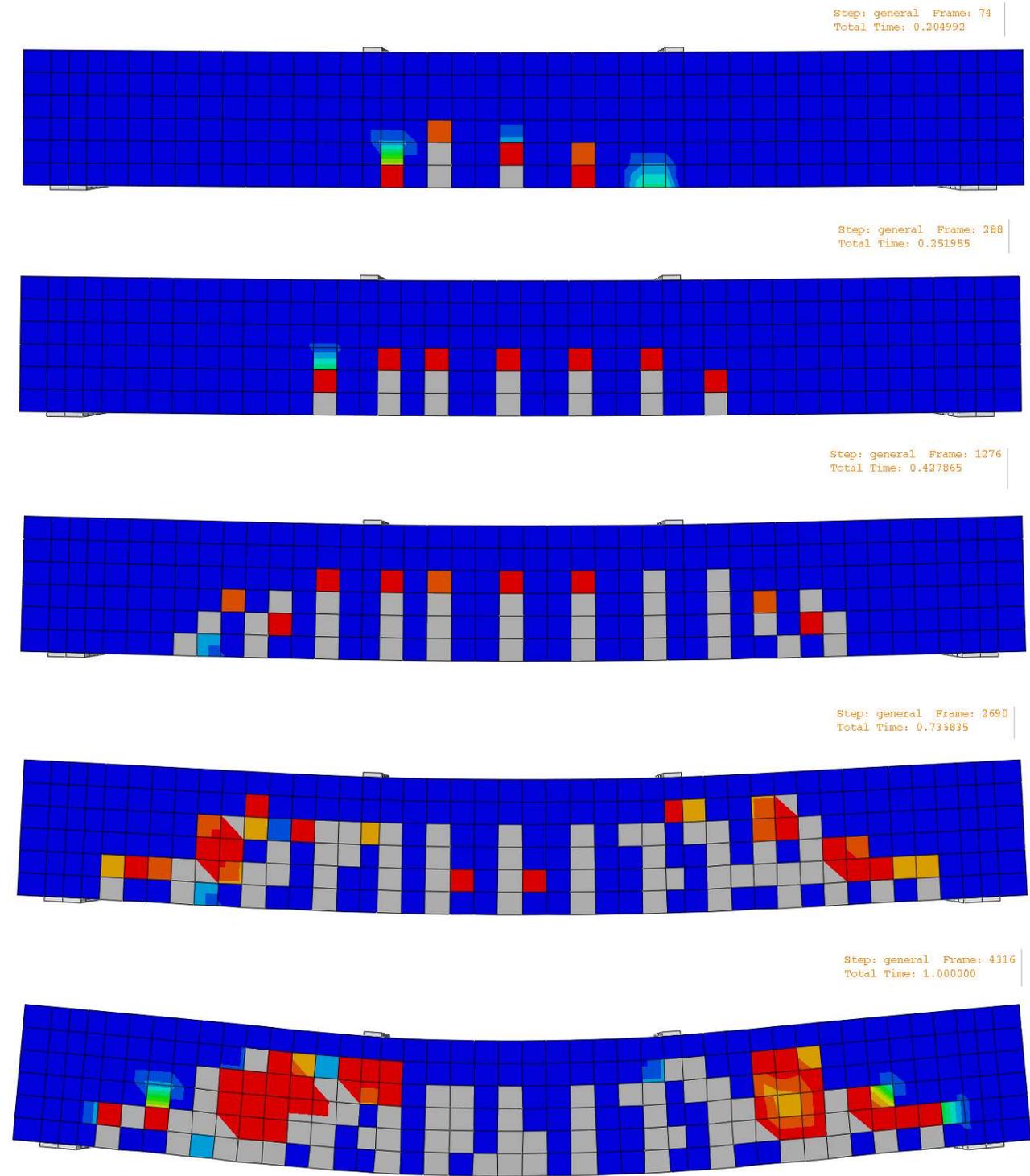
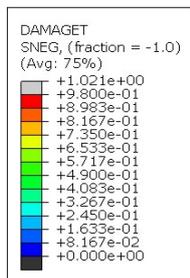


Figure 102



Crack propagation 3d solid model – end deflection 34mm

Crack propagation seen through the tensile damage parameter, for the 2d shell model

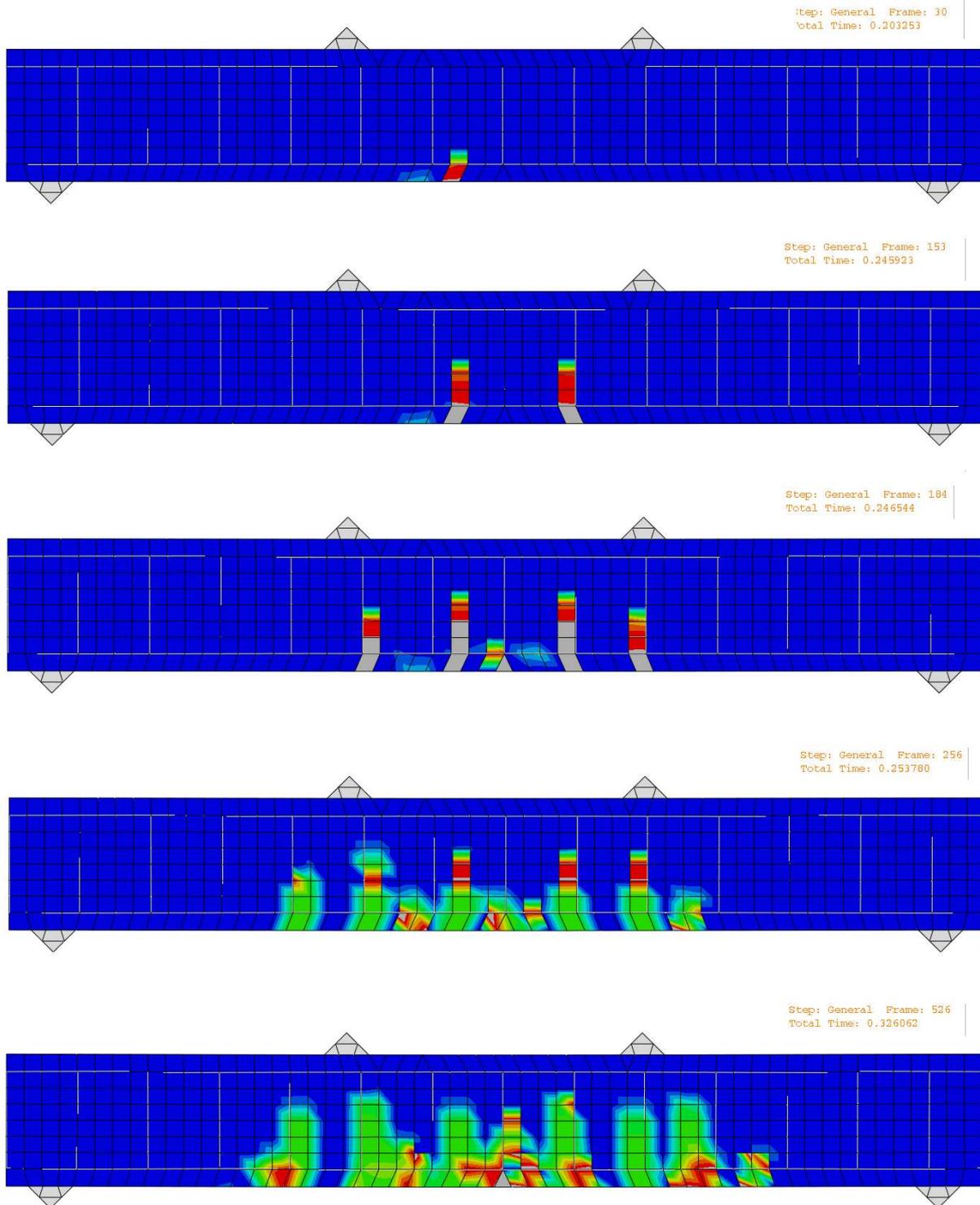
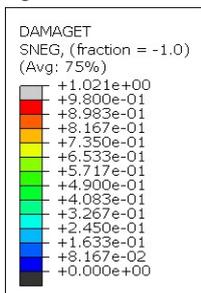


Figure 103



Crack propagation 2d shell, only up to 78 kN

Comparing crack patterns

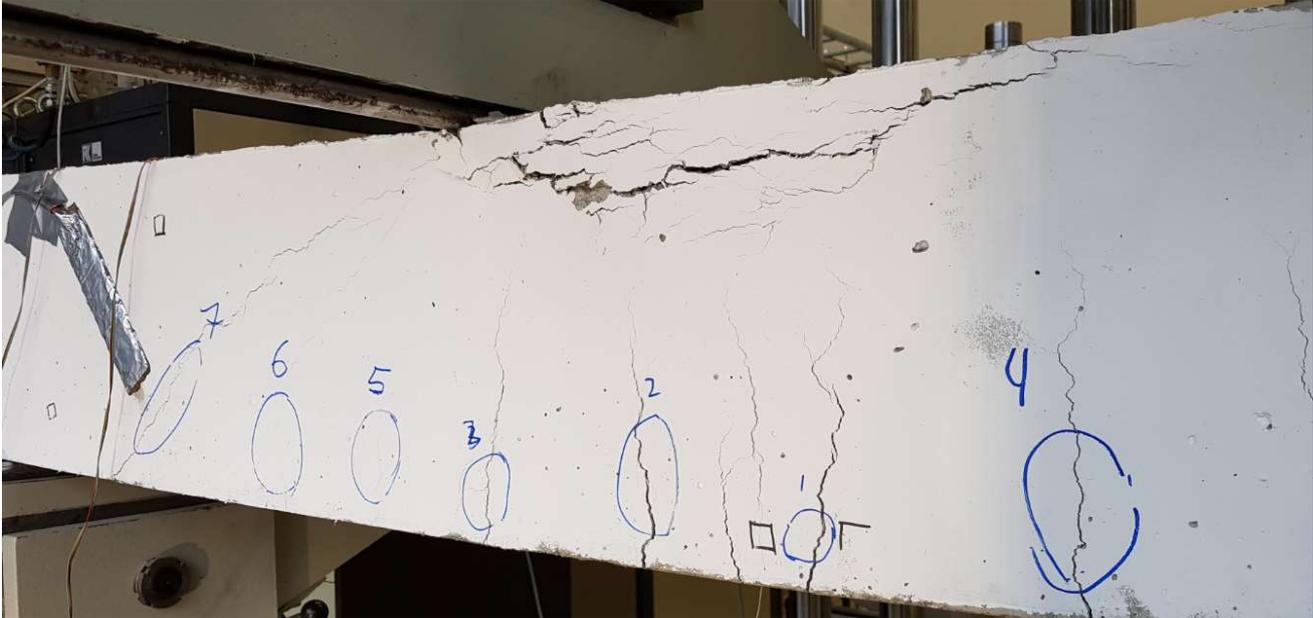


Figure 104
Test of beam at failure

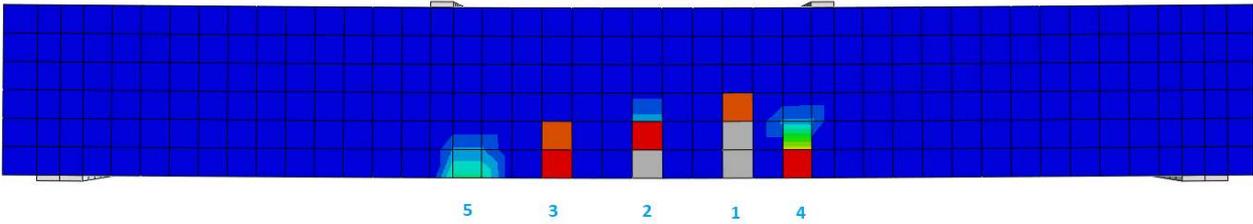


Figure 105
3d solid model, stopped after first 5 cracks

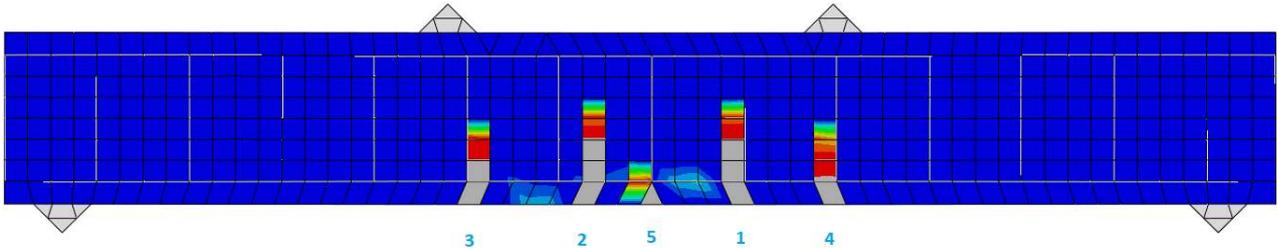


Figure 106
2d shell model, stopped after first 5 cracks

If you look at the Figure 106, the first crack marked 1 is a little bit to the right of the midpoint.

That means the 3d solid model is a bit off with the location of the first crack, while at the same time, the placement of the first 5 cracks are correct. For the 2d beam, it can't reproduce the same degree of precision, but in general follows the same path as in Figure 106.

When trying to run the [B50-paper 2] material input on the 2d shell model, it fails to converge, as it hits a dead stop at 0.326 of total load, which is just only 78kN. It's look like the gained stability it had previously is lost once the damage parameter is entered into the materials.

The crack propagation is not as good as for 2d shell model as for the 3d solid, as it estimates two center cracks, instead of one center crack first, but in general it's an overall good representation.

This is small difference is most likely due to mesh level geometry, and I believe if the mesh refinement had been exactly doubled to keep the neat arrangement for the rebars, this model would actually produce correct crack propagation.

The same extraction method used for the 3d solid, can also be used on the 2d shell element model.

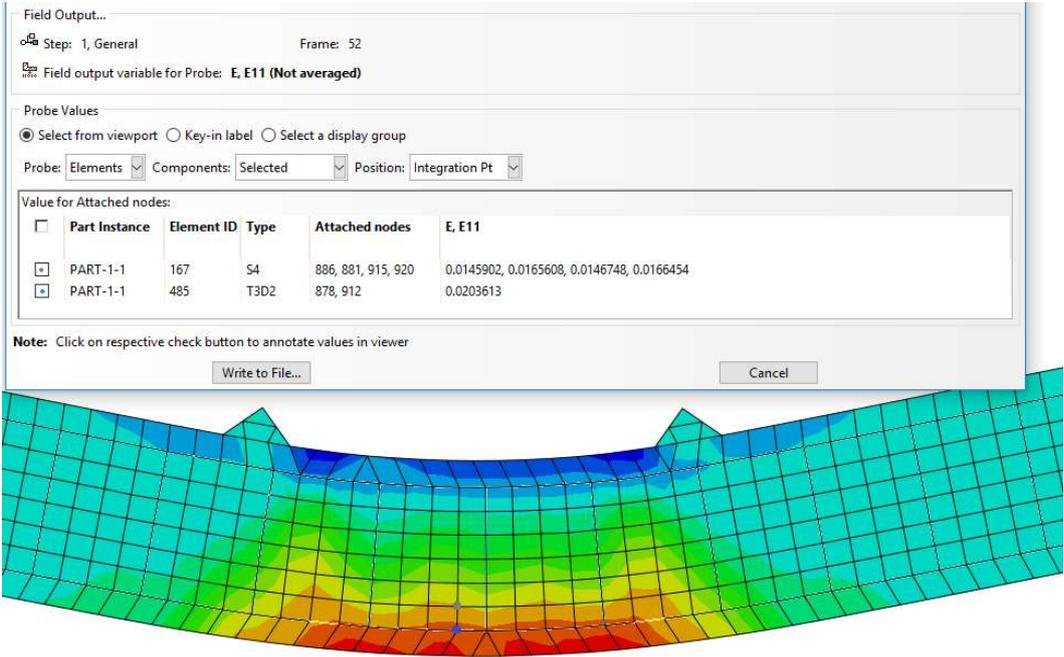


Figure 107

Probe function, extraction of data

As the 2d shell element had significant problems with the [B50-paper2] concrete, the illustration picture above is taken from the [B35_default] concrete.

As the models have shown a good compatibility towards the actual tested beam, the 3d solid model is selected to be used to extract data .

Strain from the mid-section will be extracted from the concrete and the stress and strain from the $\phi 16$ reinforcement.

Using the strain to calculate crack width.

$$w_k = S_{r,max} \cdot \varepsilon_{CR}$$

Where $\varepsilon_{CR} = \varepsilon_{sm} - \varepsilon_{cm}$ which must be larger or equal to $0.6 \frac{\sigma_s}{E_s} = 1.5 \times 10^{-3}$

$$\varepsilon_{CR} := \frac{\sigma_s - k_t \cdot \frac{f_{ct,eff}}{\rho_{P,eff}} \cdot (1 + \alpha_e \cdot \rho_{P,eff})}{E_s} = 2.017 \times 10^{-3}$$

And defined by geometry $S_{r,max} := k_3 \cdot C + \frac{k_1 \cdot k_2 \cdot k_4 \cdot 16 \text{ mm}}{\rho_{P,eff}} = 197.9 \text{ mm}$

Method used is probe, element integration point for concrete strain, and node point for rebar extraction. Using time frame to extract at exact load.

Example 75 kN, is found from $0.3 \cdot 250 = 75$ kN, where 250 kN is max load at time 1s.

Table extracted values from 3d model.

Extracted values from midpoint	Strain concrete	Strain $\phi 16$	Stress $\phi 16$
75 kN	7.8*E-5	0.000682	137 MPa
125 kN	9.4*E-5	0.000132	265 MPa
175 kN	0.0001265	0.000194	388 MPa
235 kN	0.00015132	0.002365	473 MPa

Table: 10 (extraction through strain)

Crack width calculated from: (input from table XX above)

Extracted values from midpoint	Strain	stress
75 kN	0.120 mm	0.067 mm
125 kN	0.243 mm	0.193 mm
175 kN	0.359 mm	0.315 mm
235 kN	0.438 mm	0.40 mm

Table 11 (extraction through stress)

Comparing crack width:

Load measured	Real total load on Beam	Crack width observations mm	Crack width DIC system mm	Strain extraction mm	Stress extraction mm
50 kN	75 kN	<0.100	0.041	0.120	0.067
100 kN	125 kN	0.200	0.125	0.243	0.193
150 kN	175 kN	0.300	0.225	0.359	0.315
210 kN	235 kN	---	0.415	0.438	0.400

Table: 12 (crack width comparison)

This result is in line with the observed crack widths, and can absolutely be said to be a good r representation.

Discussion of results

Results :

In the concrete lab, 6 geometrically identical reinforced concrete beams were tested created with 3 different concrete mixtures. These were either tested in a 4-point load to failure after 14 days, or 28days, along with concrete samples for the same mixture. From these beams, one was picked for this thesis. This beam was set to be the benchmark for which an Abaqus model using the Concrete Damage Plasticity module would be used.

As crack widths were measured visually and with a camera system during the testing, the thesis seeks to be able to replicate these crack widths through the Abaqus model, with input from the material testing. That is the main purpose of this thesis.

As the material input proved to be difficult, many of the standard solutions for stress strain were used and tested, but not found accurate enough to be a good representation of Beam 6.

Through testing of different published CDP concrete material input, one matched the test beam, and was used.

Two different methods of calculating crack widths were done; The first method was to measure the strain values from the reinforcement and from the nearby concrete, and the other was extracting stress values from the rebars, before the values were used in the EC2 formulas for crack width.

These values were compared to the visual and camera captured values at given load levels. Table 13 is adjusted with +25kN due to self-calibration of the test machine done at 20kN and self-weight, labelled "total load".

As the 2d shell model could not run the selected material input due to convergence issues, only results from the 3d solid model were obtained.

Comparing crack width, 3d solid model results in green:

		Lab result	Lab result	Abaqus	Abaqus
Load measured	Total load	Crack width DIC system mm	Strain extraction mm	Stress extraction mm	Stress extraction mm
50 kN	75 kN	<0.100	0.041	0.120	0.067
100 kN	125 kN	0.200	0.125	0.243	0.193
150 kN	175 kN	0.300	0.225	0.359	0.315
210 kN	235 kN	---	0.415	0.438	0.400

Table: 13 (crack width comparison)

What makes the Concrete Damage plasticity module special, is the damage parameter. This is an additional set of parameters, which is defined from 0 = no damage, to 1 = full damage, and are linked to inelastic strain values for both compression and tensile.

This function has been tested with good result for both models, where the 3d solid model could predict correct placement for the first 5 cracks, limited by mesh size.

Comparing crack patterns.

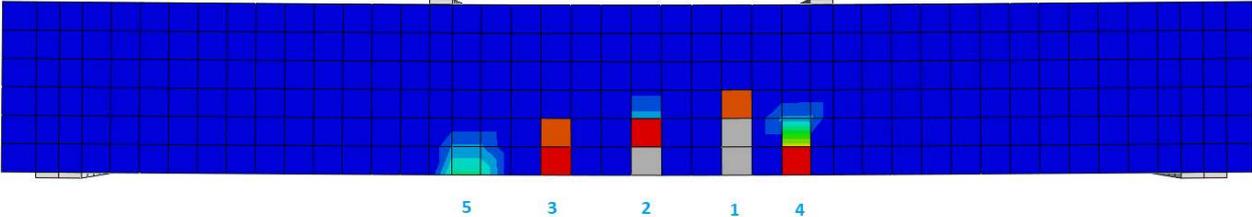


Figure 108
3d solid model, stopped after first 5 cracks

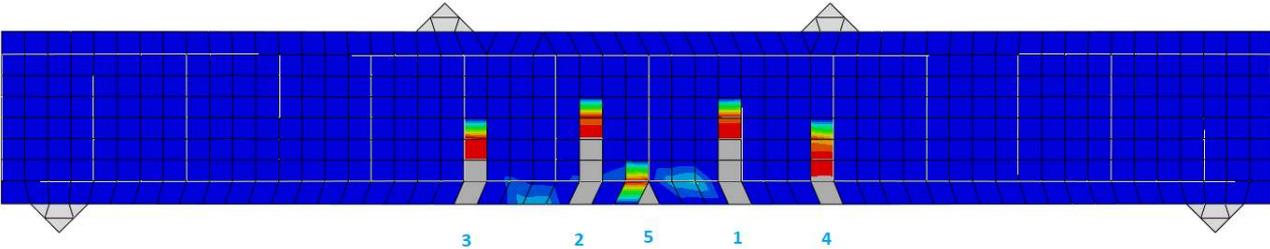


Figure 109
2d shell model, stopped after first 5 cracks

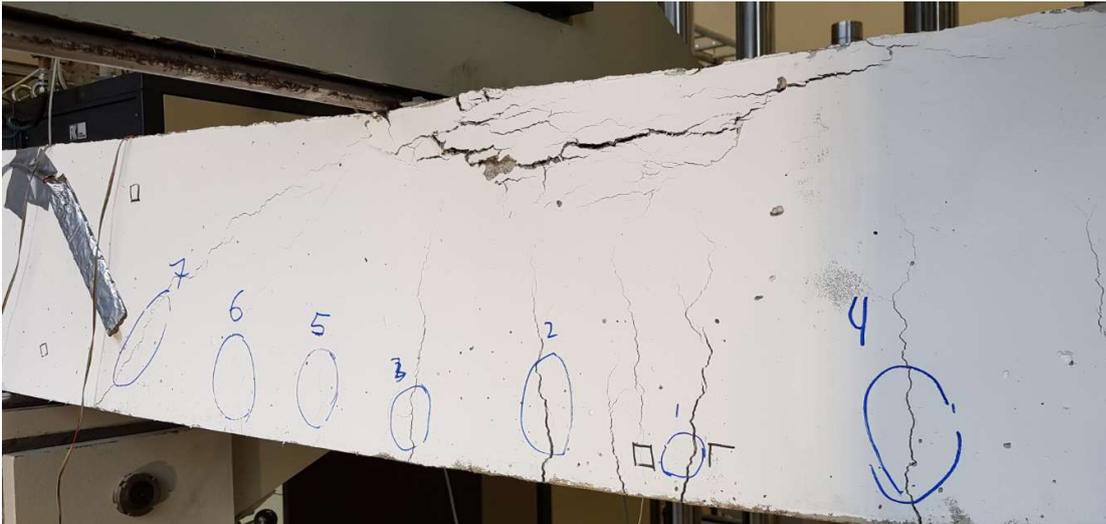


Figure 110
Test of beam at failure

Conclusion and Future research

The Abaqus Concrete Damage Plasticity module is suitable to model concrete RC beams.

Creating a material input from lab testing proved to be unexpectedly difficult. As for user-friendliness, the CDP-module is a bit challenging with its inelastic strain inputs, but the behaviour and the damage parameter functionality works good. And being modelled inside the Abaqus system, one of the big pluses is that it can be combined further into a variety of analyses.

The major drawback for the CDP module, would have to be time consumption, because once you include all the parameters, the time consumption rises dramatically. The [B35_default] had 76 increments to convergence with a load giving about 30mm deflection, the [B50-paper2] needed 2919 increments to do the same, and [B50-paper1] needed 4209 increments. That is a 55x increase in time, all on the same mesh level, going from "min" to "max" level of inputs.

But as it is a newer software, it utilizes GPU assistance in calculation, and as the future holds more cores for all ordinary workstations, this is an issue today, but not necessarily an big issue in 2-3 years of time.

As in comparison, both models did well, and for most cases, and for most cases the 2d shell model is a valid representation of the heavier 3d solid mode, but with full functionality of the damage parameter, the 2d shell model suffered from convergence issues.

The overall conclusion, is that it is possible to extract data for both strain and stress from the models in Abaqus CDP-module. The extracted data provides good input to the EC2 standard to calculate crack widths, with good accuracy. Regarding the CDP, one must once again stress the point of using the viscosity parameters with great caution. This thesis recommend the following limits:

Proposed Viscosity Parameter Limits

General Solver	0.0	0.00001 max
RIKS Solver	0.0	0.0001 max

Suggestions for future work

As a closing remark, there is still work to be done within this field of research. One should test if 3D shell elements suffer the same mesh sensitivity as the 3D continuum-element. Further, one should test strain effects using the Abaqus power law modus, especially when testing beams under 28 days. Regarding material, there should be constructed a material catalogue for the most typical concrete mixtures within its strength class, and build a catalogue with concrete parameters, which both Abaqus and other FEM-software would benefit from. This could be used in combination with the power law to retest all 6 beams to learn more about the material modelling. Finally, more work needs to be done regarding testing the implisit/dynamic Quasi-static solver, to be able to define the non-linear behaviour past the maximum load point.

References

- Burgers, R. (2006). *Non-linear FEM modelling of steel fibre reinforced concrete for the analysis of tunnel segments in the thrust jack phase*. (Master thesis). TU Delft/Delft University of Technology, Netherlands.
- Chen, W-F. (2007). *Plasticity in reinforced concrete*. New York: McGrawHill.
- Cook, R.D., Malkus, D.S., Plesha, M.E. & Witt, R.J. (2002). *Concepts and applications of finite element analysis* (4th ed.). New York: Wiley.
- Esfahani, M. H., Hejazi, F. and Vaghei, R. (2017). Simplified Damage Plasticity Model for Concrete. *Structural engineering International* 27 (1), 68-78.
- Jankowiak, T. & Lodygowski, T. (2013). *Identification of parameters of concrete damage plasticity constitutive model*. Retrieved from url: https://www.researchgate.net/publication/228525599_Identification_of_parameters_of_concrete_damage_plasticity_constitutive_model on May 14, 2018.
- Kaklauskas, G., Gribniak, V., Salys, D., Sokolov, A., & Meskenas, A. (2011). Tension-Stiffening Model Attributed to Tensile Reinforcement for Concrete Flexural Members. *Procedia Engineering*, 14, 1433-1438.
- Krok, A., Peciar, M., & Fekete, R. (2014). Numerical investigation into the influence of the punch shape on the mechanical behavior of pharmaceutical powders during compaction. *Particuology*, 16(C), 116-131.
- Malm, R. (2009). *Predicting shear type crack initiation and growth in concrete with non-linear finite element method*. Retrieved from url: <http://www.diva-portal.org/smash/get/diva2:209582/fulltext01> on May 11, 2018.
- Masti, K., Maghsoudi, A.A. & Rahgozar, R. (2008). Nonlinear Models and Experimental Investigation of Lifetime History of HSC Flexural Beams. *American Journal of Applied Sciences*, 5(3), 248-262.
- Michal and Andzej. (2015). *Calibration of the CDP model parameters in Abaqus*. Retrieved from url: http://www.i-asem.org/publication_conf/ase15/3.CTCS15/1w/W1D.1.AWinnicki.CAC.pdf on May 28th, 2018.
- Nordic Concrete Federation. (2017). *Crack width calculation methods for large concrete structures. Workshop proceedings from a Nordic miniseminar, Oslo, Norway, August 29.-30., 2017*. Retrieved from url: <https://nordicconcrete.net/wp-content/uploads/2017/11/Workshop-Proceedings-No-12-aug17-Oslo.pdf> on April 25, 2018.
- Santuk, M. & Pul, S. (2017). *Finite Element Analysis for Obtaining Structural Performance of Bridge Pier Interacting with Soil*. Retrieved from url: https://www.empa.ch/smar/SMAR_2017_Proceedings/papers/284.pdf on June 1, 2018.
- Sevim and Sönmez. (2013). *Investigation of Concrete Quality in Earthquake Regions after Devastating Earthquakes Occurred in Turkey*. Retrieved from url: https://www.researchgate.net/publication/320421530_Investigation_of_Concrete_Quality_in_Earthquake_Regions_after_Devastating_Earthquakes_Occurred_in_Turkey on May 2, 2018.

Sümer, Y. & Aktas, M. (2015). Defining parameters for concrete damage plasticity model. *Challenge Journal of Structural Mechanics* 1(3), 149-155.

Szczecina, M., & Winnicki, A. (2017). Relaxation Time in CDP Model Used for Analyses of RC Structures. *Procedia Engineering*, 193, 369-376.

U.S. Department of Transportation Federal Highway Administration (2006). *Optimized Sections for High-Strength Concrete Bridge Girders--Effect of Deck Concrete Strength*. Retrieved from url: <https://www.fhwa.dot.gov/publications/research/infrastructure/structures/05058/03.cfm> on May 20, 2018.

Vasios, N. (2015). *Nonlinear Analysis of Structures. The Arc Length Method: Formulation, Implementation and Applications*. Retrieved from url: <https://scholar.harvard.edu/files/vasios/files/ArcLength.pdf> on June 3, 2018.

Walraven. (2008). *Eurocode 2: Design of concrete structures EN1992-1-1*. Retrieved from url: http://eurocodes.jrc.ec.europa.eu/doc/WS2008/EN1992_1_Walraven.pdf on April 13, 2018.

Appendix :

Materials :

List :

Concrete CDP recipes:

[B35_Default] - Default CDP, settings used for testing.

[B25-EC2] - Values calculated from calculated from EC2.

B50 [paper 1] * (Esfahani, Hejazi & Vaghei, 2017)

B50 [paper 2] * (Jankowiak & Lodygowski, 2013)

[PL_eng] - Reinforcement material input stress inelastic strain

[PL_true] - Reinforcement material input true stress, inelastic true strain

[PL_log] - Reinforcement material input true stress, inelastic log strain

For all CDP recipes,

RIKS solver : Viscosity parameter 0.0001 or lower.

General Solver : Viscosity parameter 0.00001 or lower.

[B35_Default]

B35_Default					
Concrete	E modulus		Poisson Ratio		
parameters :	34000 MPa		0.2		
CDP parameters:	<i>Dilation Angle</i>	<i>Eccentricity</i>	<i>fb0/fc0</i>	<i>K</i>	<i>Viscosity Parameter</i>
	31	0.1	1.16	0.667	*
[MPa]	<i>Yield Stress</i>	<i>Inelastic strain</i>		<i>Yield Stress</i>	<i>Cracking strain</i>
Compression	43 MPa	0	Tensile	3.2	0
Compression dmg	No-input	No-input	Tension dmg	No-input	No-input

[B35_EC2]

B35_EC2					
Concrete	E-modulus		Poisson Ratio		
parameters :	34 GPa		0.2		
CDP parameters:	<i>Dilation Angle</i>	<i>Eccentricity</i>	<i>fb0/fc0</i>	<i>K</i>	<i>Viscosity Parameter</i>
	31	0.1	1.16	0.667	*
[MPa]	<i>Yield Stress</i>	<i>Inelastic strain</i>	[MPa]	<i>Yield Stress</i>	<i>Cracking strain</i>
Compression	28.5	0	Tensile	3.2	0
	32.25	0.000051		0.0000	0.0038
	40.313	0.000314		-	-
	43	0.0007353		-	-
	43	0.0022353		-	-
	0	0.01		-	-
Compression damage	0	0	Tension damage	0	0
	0	0.000051		0.99	0.0038
	0	0.000314		-	-
	0	0.0007353		-	-
	0.05	0.0022353		-	-
	0.99	0.01		-	-

B50 [paper 1] *

B50 Paper 1		dilation angle	
			31
			0.1
33.4			1.16
0.2			0.67
			0
Mpa	inelastic strain	C	inelastic strain
25.5	0	0	0
32	0.0000057382	0	0.0000057382
37.5	0.000041363	0	0.000041363
42	0.000106874	0	0.000106874
45.5	0.000202271	0	0.000202271
48	0.000327555	0	0.000327555
49.5	0.000482726	0	0.000482726
50	0.000667782	0	0.000667782
49.5	0.000882726	0.01	0.000882726
48	0.001127555	0.04	0.001127555
45.5	0.001402271	0.09	0.001402271
42	0.001706874	0.16	0.001706874
37.5	0.002041363	0.25	0.002041363
32	0.002405738	0.36	0.002405738
25.5	0.0028	0.49	0.0028
18	0.003224148	0.64	0.003224148
9.5	0.003678183	0.81	0.003678183
Tension	cracking strain	T	cracking strain
5	0	0	0
0.05	0.001494322	0.99	0.001494322

[B30-2] [paper 2] *

B50 original paper 2		dilation angle	
19.7			38
0.19			0.1
			1.12
			0.666
			0
Mpa	inelastic strain	C	inelastic strain
15	0	0	0
20.197804	0.0000747307	0	0.0000747307
30.000609	0.0000988479	0	0.0000988479
40.303781	0.000154123	0	0.000154123
50.007692	0.000761538	0	0.000761538
40.23609	0.002557559	0.195402	0.002557559
20.23609	0.005675431	0.596382	0.005675431
5.257557	0.01173319	0.894865	0.01173319
Tension	cracking strain	T	cracking strain
1.99893	0	0	0
2.842	0.00003333	0	0.00003333
1.86981	0.000160427	0.406411	0.000160427
0.862723	0.000279763	0.69638	0.000279763
0.226254	0.000684593	0.920389	0.000684593
0.056576	0.00108673	0.980093	0.00108673

B35 -EC2 – BUILD / B50-EC2

Input B50 from EC2.

$$E_{cm} := 37000 \text{ MPa} \quad \varepsilon_{cu2} := 0.0035 \quad n := 2.0 \quad E_c := E_{cm}$$

$$\text{Stress at fcm peak } f_{cm} := 58 \text{ MPa}$$

$$\text{Strain at fcm peak } \varepsilon_{c2} := 0.002$$

Elastic Stress - Strain

Max value elastic strain

$$\varepsilon_{\text{elastis.c}}(f_c) := \frac{f_c}{E_c}$$

Valid up to

$$\frac{f_{cm}}{E_c} = 0.001568$$

Stress - Strain curve fig 3.3

Formula [3.17]

$$\sigma_c = f_{cm} \quad \varepsilon_{c2} < \varepsilon_c < \varepsilon_{cu2}$$

$$\sigma_c(\varepsilon_c) := f_{cm} \cdot \left[1 - \left(1 - \frac{\varepsilon_c}{\varepsilon_{c2}} \right)^n \right]$$

Finding elastic strain, plastic strain, total strain, through STRESS

$$\varepsilon_{\text{Total}} = \varepsilon_{\text{elastis.c}}(\sigma_c(\varepsilon_{\text{Total}})) + \varepsilon_{\text{pl.c}}$$

Chosen value, as $\varepsilon_{\text{Total}}$ as variable R: $R = \varepsilon_{\text{Total}} \quad R := 0.003$

Finding stress at given total strain value

$$\sigma_c(R) = 25.375 \text{ MPa}$$

$$\varepsilon_{\text{elastis.c}}(\sigma_c(R)) = 0.0006858$$

Elastic strain part from level of comp. stress

$$\varepsilon_{\text{pl.c}} := R - \varepsilon_{\text{elastis.c}}(\sigma_c(R)) = 0.00281419$$

Plastic strain = (Total - Elastic strain)

$$\varepsilon_{\text{elastis.c}} := \frac{f_{cm}}{E_c} = 0.0015676$$

If above total strain = 0.0020 --> Plastic strain : $\varepsilon_{\text{pl.c}} := R - \varepsilon_{\text{elastis.c}} = 0.001932$

Material reinforcement input:

Emodule = 200 GPA.
 Sample size , ø16mm B500NC rebars.
 Cut to ~300mm length, tested with extensometer along the rigde on the reinforcement bar.
 Testing found E moduls, yield points and UTS. Inelastic strain, is calculated by measuring strainpoints from yield up to UTS, and then remove elastic strain at yield point.
 True strain is converted with abaqus, and is the one used in this thesis.
 Log strain had no positive contribution, and was only used for testing.

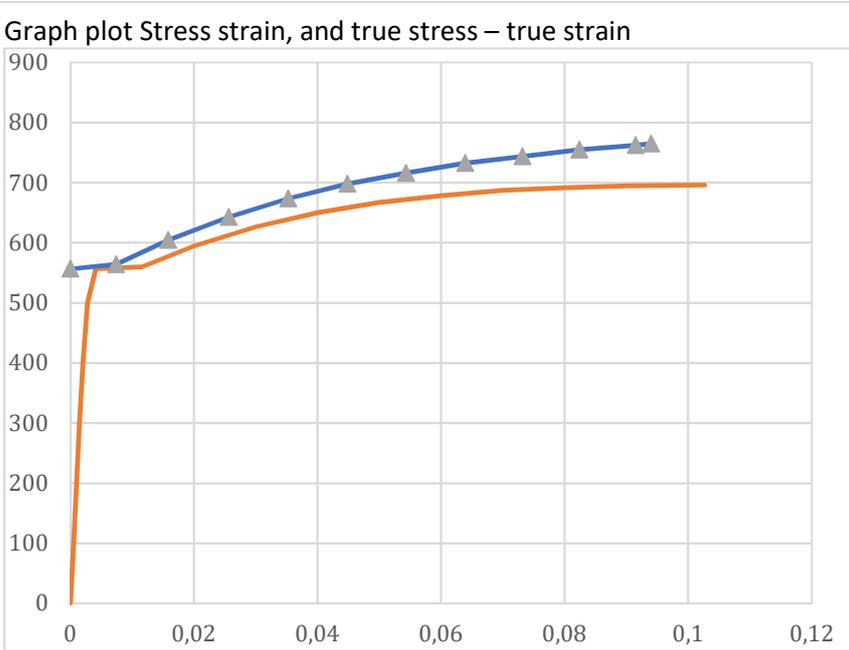


Figure 108

<i>[PL_eng]</i>	
Stress	Inelastic strain
(MPa)	(mm/mm)
556,46	0
560,12	0,00743
595,12	0,01597
626,57	0,02597
650,12	0,0359
667,24	0,04587
678,23	0,05582
687,24	0,06596
691,38	0,07593
694,87	0,08585
695,73	0,09582
696,18	0,09852

<i>[PL_true]</i>	
True stress	True inelastic strain
(MPa)	(mm/mm)
556,460	0
564,282	0,00740
604,624	0,01584
642,842	0,02564
673,459	0,03527
697,846	0,04485
716,089	0,05432
732,570	0,06388
743,876	0,07319
754,525	0,08236
762,395	0,09150
764,768	0,09396

<i>[PL_log]</i>	
True stress	Logaritmic inelastic strain
(MPa)	(mm/mm)
556.46	0
564.282	0.004570662
604.624	0.012739212
642.842	0.022176569
673.459	0.03140498
697.846	0.040527402
716.089	0.049492081
732.57	0.058469786
743.876	0.067159481
754.525	0.075655064
762.395	0.08405707
767,643	0,094235474

Table: 14 (Material input for Abaqus : Reinforcement)

Example of mesh dependency.

An extensive test where done on the first 3d solid model, to find the mesh dependency. difference as test have the same potential error, in addition to be easy to spot when plotted at full graph.

Material input [LAB3r] , [PL_true]

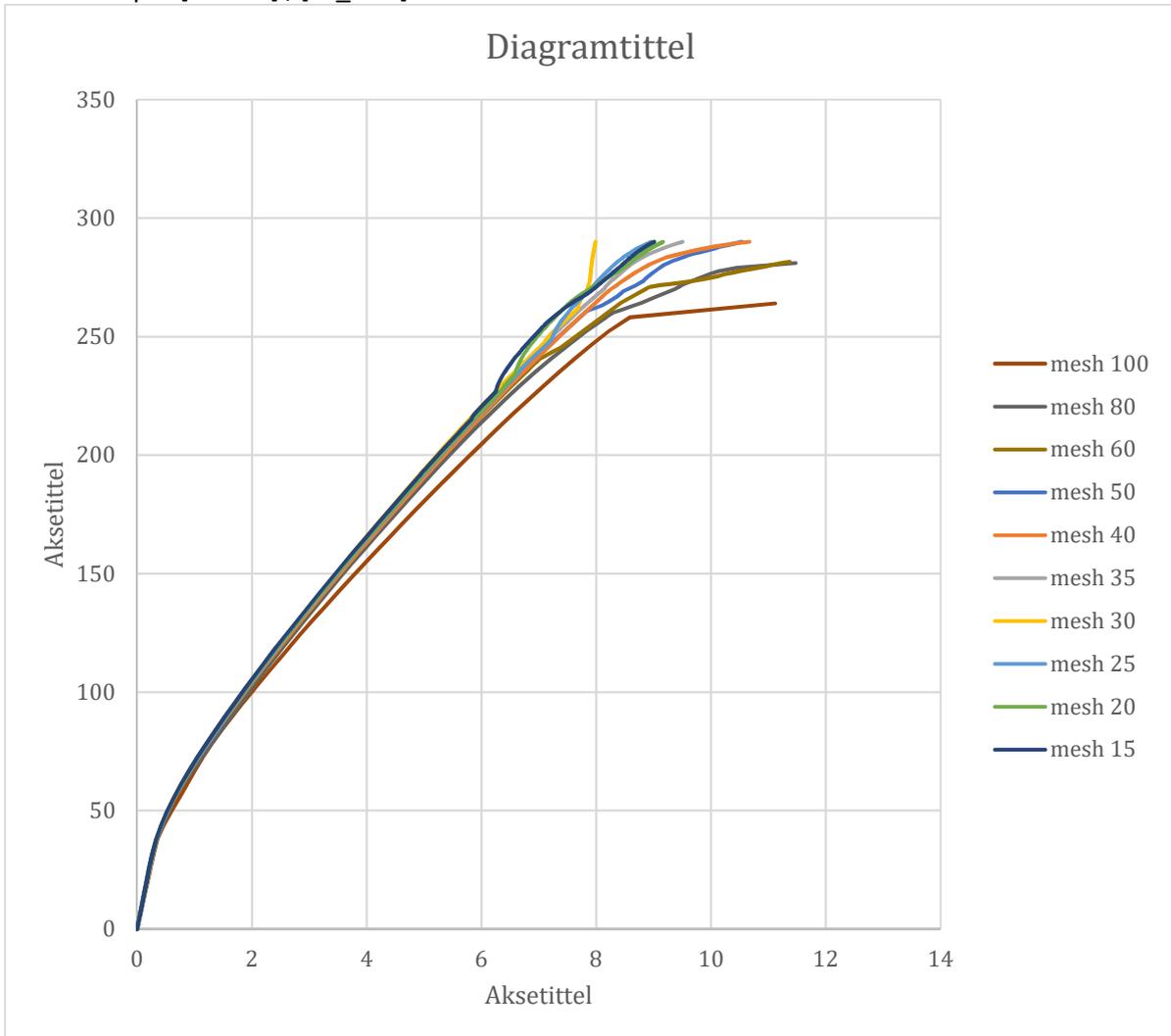


Figure: 109 (First model was found to be too mesh dependent, and rejected.)

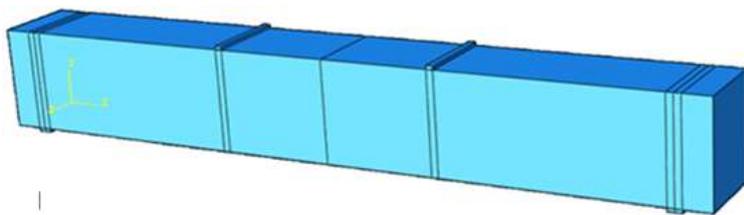


Figure: 110 (too narrow bands, and replaced with opt 1.)

The graph also shows that mesh sizes that does not "add up" according to the beams dimensions, give strange effects, as the mesh 15.

General solver
 Input : [B35_DEFAULT]
 Steel : [PL_true]

General solver: Overview test of increment sizes: Load 200kN

[B35_DEFAULT] [PL_true]	Start increment	Max increment	2d Shell model
<i>Stabilize: on</i>	1	1	16,95
<i>Stabilize: on</i>	0.001	0.01	-
<i>Stabilize: on</i>	0.001	0.02	16,71
<i>Stabilize: on</i>	0.01	0.02	16,72
<i>Stabilize: on</i>	0.1	0.1	16,81
<i>Stabilize: on</i>	0.3	0.3	16,71
<i>Stabilize: off</i>	1	1	16,99
<i>Stabilize: off</i>	0.001	0.01	-
<i>Stabilize: off</i>	0.001	0.02	16,72
<i>Stabilize: off</i>	0.01	0.02	16,72
<i>Stabilize: off</i>	0.1	0.1	16,82
<i>Stabilize: off</i>	0.3	0.3	16,87

Table: 14 (Increment step size test)

Very little variation was found