

Metodik för kalibrering av en beräkningsmodell

Develop a method to determine material properties

Examensarbete för högskoleingenjörsexamen inom Maskiningenjörsprogrammet

ELIN ARVIDSSON
SARA JOHANSSON

Institutionen för Material- och tillverkningsteknik
Avdelningen för Avancerad oförstörande provning
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige, 2015
Examinator: Gert Persson Examensarbete nr 127/2015

Metodik för kalibrering av en beräkningsmodell

© Elin Arvidsson, Sara Johansson 2015
Examensarbete nr 127/2015
Avdelning material- och tillverkningsteknik
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

FÖRORD

Den här rapporten är resultatet av examensarbetet som handlat om att skapa en metodik för att kalibrera en beräkningsmodell i Matlab med hjälp av CalFem. Vi vill tacka Sveriges tekniska forskningsinstitut (SP) avd. Mekanik för att vi fått genomföra vårt examensarbete hos dem och för all hjälp under projektets gång.

Ett extra stort tack till vår handledare Erik Dartfeldt som varit drivande och delaktig och gjort projektet möjligt att genomföra. Samt till Torsten Sjögren på SP som varit en hjälpande hand.

Ett stort tack till vår handledare och examinator på Chalmers, Gert Persson, som vart till stor hjälp under projektets gång för att få oss att nå vårt resultat.

SAMMANFATTNING

Projektet har utförts på Sveriges tekniska forskningsinstitut, SP. De är ledande inom forskning, utveckling och analys. De har också ett stort intresse i att undersöka nya områden. Syftet med projektet var att skapa en metodik för att fastställa materialparametrar. De materialegenskaper som skulle fastställas var E-modulen och Poissons tal för ett dragprov. Syftet skulle uppnås genom att skapa en beräkningsmodell i Matlab med hjälp av CalFem.

Beräkningsmodellen togs fram för ett statiskt belastningsfall med linjärelastisk materialmodell. Experimentdatan togs från ett dragprov analyserat med DIC (Digital Image Correlation).

Provstavens förskjutningar och tillförda kraft togs in i beräkningsmodellen. För att hitta materialparametrarna ställdes lämpliga intervall upp för E-modulen och Poissons tal. Genom att variera dessa värden i beräkningsmodellen kunde energi som krävdes för att uppfylla experimentets förskjutningar beräknas. När experimentets energi och beräkningsmodellens energi var samma kunde materialparametrarna fastställas.

Experimentdata från en provstav av titan togs in i beräkningsmodellen. E-modulen fick ett intervall mellan 98,5-101 GPa och Poissons tal mellan 0,315-0,365. Dessa intervall ligger inom rimliga värden för titans egenskaper.

SUMMARY

This project has been completed at the technical research institute of Sweden, this is a leading company in research, development and analysis. The company has a strong interest in exploring new areas. The purpose for the project was to develop a method to determine material properties. The properties to be determined E-modulus and poisson's ratio and this would be achieved by creating calculation model.

The calculation model was created in Matlab using CalFem, a toolbox with functions of the finite element method. The calculation model is developed for a static load case with a linear elastic material. The boundary conditions should simulate a tensile test. The material properties poisson's ratio and elastic modulus was determined after data from a real tension test analyzed by digital image correlation (DIC) and was used in the calculation model.

The properties were found by using two different types of objective functions. The objective functions calculate the energy that it takes to reach the boundary conditions. A lot of different candidates with diverse values on the elastic modulus and poisson's ratio were calculated. The energy that was supplied to the physical experiment was known. When the energy and the displacement are the same between the physical experiment and the candidate the material properties have been found.

The tensile test was performed with a titanium sample and the result from the calculation model gave that the elastic modulus had a range between 98.5-101 GPa and poisson's ratio had a range between 0.315-0.365. The properties for titanium should be between these ranges. The method that was created to find material properties was a success.

INNEHÅLLSFÖRTECKNING

1 INLEDNING.....	1
1.1 Bakgrund	1
1.2 Syfte.....	1
1.3 Avgränsningar.....	1
1.4 Precisering av frågeställning	2
2. TEORETISK REFERENSRAM	3
2.2 Hållfasthetslära.....	3
2.3 Finita elementmetoden (FEM)	5
2.4 Fysikaliskt arbete	6
2.5 Interpolation.....	6
3. METOD.....	7
3.1 Första statiska belastningsfallet	9
3.2 Kalibreringsmetodik	10
3.3 Digital image correlation (DIC)	11
3.4 Målfunktioner vid de verkliga experimentet	13
4. GENOMFÖRANDE.....	14
4.1 Första statiska belastningsfallet.....	14
4.2 Matlabkod där en referensmodell skapas	15
4.3 Konvergens.....	19
4.4 Tre olika fall med olika materialegenskaper	19
4.5 Jämförelse av energi.....	20
4.6 Inverkan av elementstorlek.....	21
4.7 Experimentdata från två olika dragprov	21
5 ANALYS OCH RESULTAT	23
5.1 Konvergens.....	23
5.2 Tre olika fall med olika materialegenskaper	26
5.2.1 Fall 1 – Referensen	26
5.2.2 Fall 2	28
5.2.3 Fall 3	30
5.3 Jämförelse av energi.....	32
5.3.1 Jämförelse av energin då E-modulen varierar	33

5.3.2 Jämförelse av energin då poissons tal varierar	35
5.4 Inverkan av elementstorlek.....	37
5.5 Experimentdata från två olika dragprov	38
5.5.1 Dragprov 1	39
5.5.2 Dragprov 2	42
6. SLUTSATS	47
6.1 Fortsatt arbete	48
REFERENSLISTA.....	49

BETECKNINGAR

I beteckningslistan beskrivs de olika funktioner som används från CalFem och Matlab samt funktioner som är uppbyggda för projektet.

hooke - en funktion från CalFem som skapar en materialmatris med linjär elastiska egenskaper. En materialmodell som beskriver hur materialet beter sig. Funktionen kräver tre indata: **p_{type}**, **E** och **v**. **p_{type}** = typ av analys, fyra olika fall kan väljas (plan spänning, plan töjning, axelsymmetri eller 3D-analys). **E** = elasticitetsmodulen och **v** = poissons tal.

quad_mesh – en funktion som meshar/delar upp en 2D rektangel i element. Skapad av Jim Brouzoulis, februari 2011. De indata som krävs för funktionen är: **p1**, **p2**, **nelx**, **nely**, **ndofs**. **p1** = x -och y-koordinat på nedre vänstra hörnet av rektangel. **p2** = x -och y-koordinat på övre högra hörnet av rektangeln. **nelx** = antal element i x-riktning. **nely** = antal element i y-riktning. **ndofs** = antal frihetsgrader per nod. Utdata är **Edof**, **Ex**, **Ey** och **B_i** där $i=1,2,3,4$. **Edof** = matris som beskriver varje elements kringliggande frihetsgrader. Första kolonnen anger vilket element och tillhörande rad är frihetsgraderna som tillhör elementet. **Ex** = anger varje elementets x-koordinater. **Ey** = anger varje elementets y-koordinater. **B_i** = frihetsgrader på utsidan av rektangel där **B1** = frihetsgraderna längst ner på rektangeln, **B2** = frihetsgraderna längst till höger på rektangeln, **B3** = frihetsgraderna längst upp på rektangel, **B4** = frihetsgraderna längst till vänster på rektangeln.

randvillkor – en funktion som bestämmer randvillkor för kroppen. Funktionen är skapad för projektet. Indata: **B1** och **B3** som fås av funktionen **quad_mesh** och anger de översta och nedersta noderna. Utdata är matrisen **bc** = en (nx2) matris som anger randvillkoren för noderna. Består av n rader beroende på antal noder och 2 kolonner, i den första kolonnen anges vilka noder som berörs och i den andra kolonnen anges randvillkoren. Fast inspänning resulterar i en nolla och förskjuts noden anges hur mycket den förskjuts i enheten meter. Är randvillkoret positivt förskjuts figuren i positiv riktning och utsätts för drag. Är randvillkoret negativt förskjuts figuren i negativ riktning och utsätts för tryck.

plane – en funktion från CalFem som beräknar styvheten hos varje rektangulärt element. Funktionen kräver fyra indata: **Ex**, **Ey**, **ep** och **D**. **Ex** = anger varje elementets x-koordinater. **Ey** = anger varje elementets y-koordinater. **ep** = en matris som innehåller **ptype** (se beskrivningen av **hooke**) och **t** = tjockleken hos figuren. **D** = vilken materialmatris man har valt exempelvis **hooke**. Utdata är matrisen **Ke** som är den lokala element styvhetsmatrisen.

assem - en funktion från CalFem som kopplar ihop lokala elementstyvhetsmatriserna till en global matris. Indata är **Edof**, **K** och **Ke**. **Edof** får man från funktionen **quad_mesh**. **K** = globala styvhetsmatrisen, denna skickas med för att inte skrivas över när varje element läggs in för sig. **Ke** = den lokala styvhetsmatrisen för elementet. Utdata är globala styvhetsmatrisen **K**.

solveq – en funktion från CalFem som beräknar nodförskjutningar och ekvivalenta nodkrafter. Indata är matriserna **K** och **bc**. **K** = globala styvhetsmatrisen. **bc** = randvillkor. Utdata är matriserna **u** och **f**. **u** = nodförskjutningarna. **f** = ekvivalenta nodkrafterna, vilket är kraften som krävs för att flytta varje nod till sin nya position.

extract – en funktion från CalFem som för samman nodförskjutningarna med tillhörande element. Indata är **Edof** och **u**. **Edof** fås av **quad_mesh** och är en matris som beskriver varje elements tillhörande frihetsgrader. **u** fås av **solveq** och är deformationen för varje nod. Utdata som fås är matrisen **ed** = element deformationsmatris, hur de tillhörande frihetsgrader deformeras i varje element.

eldraw2- en funktion från CalFem som plottar en odeformerad 2D figur och dess elementuppdelning som exempelvis en rektangel eller balkelement. Indata som krävs för funktionen är **Ex**, **Ey**, **plotpar** och **elnum**. **Ex** = anger varje elementets x-koordinater. **Ey** = anger varje elementets y-koordinater, **plotpar** = en matris där man anger linjetyp, linjefärg och typ av nodmarkering. **elnum** = en matris med alla element i stegrande ordning. Utdata är att funktionen plottar upp den odeformerade figuren och dess elementuppdelning.

eldisp2 - en funktion från CalFem som plottar en deformerad 2D figur och dess elementuppdelning som exempelvis en rektangel eller balkelement. Indata som krävs för funktionen är ***Ex***, ***Ey***, ***ed***, ***plotpar*** och ***sfac***. ***Ex*** = anger varje elementets x-koordinater. ***Ey*** = anger varje elementets y-koordinater. ***ed*** = globala deformationsmatrisen. ***plotpar*** = en matris där man anger linjetyp, linjefärg och typ av nodmarkering. ***sfac*** = skalfaktorn på deformationen vilket kan användas om deformationen är väldigt liten. Utdata är att funktionen plottar upp den deformerade figuren och dess elementuppdelning.

fill - en funktion från Matlab som fyller en 2D-figur med angivna x-och y-värden med motsvarande c-värde. Indata är x-koordinater, y-koordinater och c-värden som exempelvis kan motsvara spänning eller töjningsvärden för de givna koordinaterna. Utdata är att funktionen fyller figuren med olika färger. Färgkarta eller enskild färg kan tilläggas för önskat utseende.

griddata - en funktion från Matlab som interpolerar fram sökta värden. Indata är ***x*** = x-koordinater för de givna värdena i figuren, ***y*** = y-koordinater för de givna värdena i figuren, ***v*** = värden som tillhör x-och y-koordinaterna, ***xq*** = x-koordinater för de koordinater som ska interpoleras fram och ***yq*** = y-koordinater för de koordinater som ska interpoleras fram. Utdata är ***vq*** = tillhörande värden till de xq-och yq-koordinater som ska interpoleras fram.

quiver - en funktion från Matlab som skapar ett riktningsfält. Indata är ***x,y,u*** och ***v***. ***x*** och ***y*** anger startvärde för pilarna och ***u*** och ***v*** anger riktningen som pilarna ska peka mot. En skalfaktor på storleken på pilarna kan också anges. Utdata är att funktionen ritar upp riktningspilarna eller riktningsfältet.

1 INLEDNING

1.1 Bakgrund

Forskning, utveckling och analys av teknik, material, produkter och processer är Sveriges tekniska forskningsinstitut (SP) kännetecken. De är ledande inom denna bransch därför att de har ett stort intresse av att undersöka nya områden. Projektet trädde fram då nyfikenhet fanns inom att bindas samman verkligdata och beräkningar. Dessa beräkningar kan ske i stödprogram som exempelvis Matlab. Genom att förbättra beräkningsmodeller kan antalet fysiska experiment minska och därmed kostnaderna. För att få fram verklig data kan tex en metod som heter DIC (Digital Image Correlation) användas där förändringar av ett mönster registreras genom bilder. Med hjälp av dessa bilder kan deformationer och töjningar identifieras Andra metoder som används för att få fram experimentdata är t.ex. krocktest, dragprov och slagprov. En fördel med att ha bra beräkningsmodeller är att tester som är svåra att genomföra i verkligheten kan analyseras teoretiskt. Det kan t.ex. vara komplicerade konstruktioner eller objekt med svår geometri. I miljöaspekten är beräkningsmodeller bra för att spara på resurser, detta då experiment kräver mycket material som därefter går till avfall.

1.2 Syfte

Syftet med arbetet är att skapa en metod för att fastställa materialparametrar. De egenskaper som ska fastställas är E-modulen och Poissons tal. Syftet skall uppnås genom att skapa en beräkningsmodell i Matlab med hjälp av CalFem. Projektets mål är att bindasamman beräkningsmodellens resultat och experimentdata.

1.3 Avgränsningar

Arbetet kommer att fokusera på beräkningar av statiska belastningsfall med enkla geometrier. En linjärelastisk materialmodell (Hookes lag) kommer att användas för att simulera materialets mekaniska egenskaper. Den experimentdata som kommer att användas fås från ett dragprov.

1.4 Precisering av frågeställning

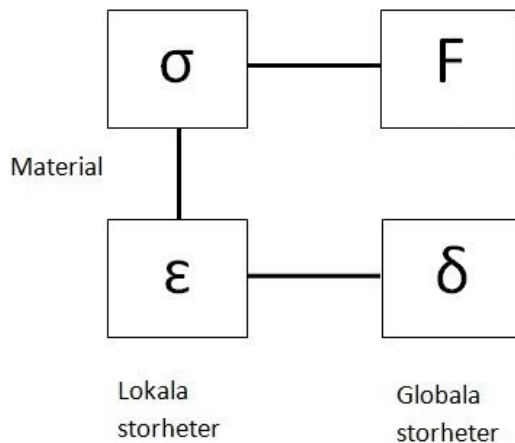
- Kan beräkningsmodellen bete sig som ett verkligt dragprov?
- Kan beräkningsmodellen få ut trovärdiga värden på E-modulen och poissons tal?

2. TEORETISK REFERENS RAM

I teoretiska referensramen beskrivs viktiga begrepp som behövs för att genomföra projektet. Som exempelvis begrepp inom hållfasthetslära och hur finita elementmetoden är uppbyggd.

2.2 Hållfasthetslära

Hållfasthetslärans teorier bygger på olika typer av samband. De tre viktigaste är jämviktssamband, deformationssamband och konstitutiva samband. Jämviktssamband kopplar ihop krafter (F) och spänningar (σ). Deformationssamband och kompatibilitetssamband kopplar ihop deformationer (δ) och töjning (ϵ). Kompatibilitetssamband beskriver hur strukturen hålls samman. Spänningar (σ) och töjningar (ϵ) kopplas samman genom materialsamband eller konstitutiva samband. Mellan kraft och deformation finns inget samband utan materialegenskaper krävs för att binda samman dessa parametrar. Figur 2.1 visar hur sambanden och de olika parametrarna hänger ihop. Kallas även inom hållfasthetslära för den gyllene kvadraten.



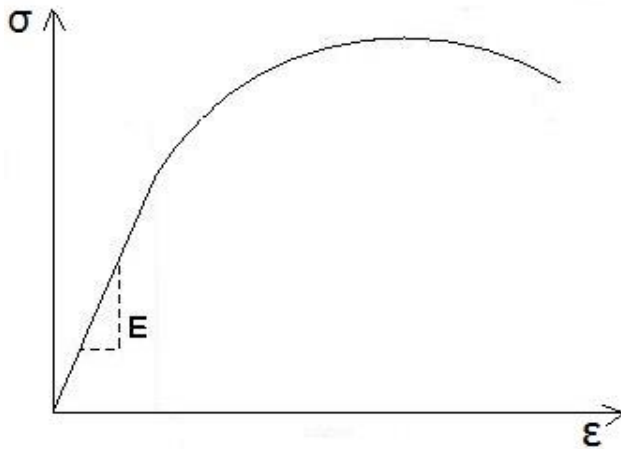
Figur 2.1 Gyllene kvadraten

Hookes lag är ett exempel på ett konstitutivt samband mellan spänning och töjning enligt ekvation 1 nedan:

$$\epsilon = \frac{\sigma}{E} \quad (1)$$

Hookes lag utgör en linjärelastisk materialmodell. Viktiga materialparametrar är elasticitetsmodulen (E-modulen) och Poissons tal (ν). E-modulen erhålls som den räta linjens

lutning i spänning-töjningsdiagrammet, se figur 2.2. E-modulen är en konstant som karakteriserar styvheten hos ett elastiskt material. Material som återgår till sin ursprungliga form efter avlastning kallas elastiska. Om sambandet mellan töjning och spänning är linjärt kallas detta att materialet är linjärt elastiskt.



Figur 2.2 Spänning-töjnings diagram

Poissons tal även kallat tvärkontraktionstalet är sambandet mellan tvärgående töjning och längsgående töjning när ett material belastas med dragning. Det innebär att när materialet deformeras längst med dragriktningen men även i tvärgående riktning. Poissons tal kan anta värden mellan 0 till 0,5. Ju högre tal ju mer deformeras tvärriktningen och bildar en tydligare midja. De finns två olika typer av töjning, teknisk töjning och sann töjning. Poissons tal bygger på den sanna töjningen som även kallas logaritmisk töjning. Den sanna töjningen betecknas enligt ekvation 2 nedan:

$$\epsilon = \ln\left(\frac{AL}{L_0}\right) \quad (2)$$

Den tekniska är en linjär approximation av töjningen som betecknas enligt ekvation 3:

$$\epsilon = \frac{\text{förlängningen}}{\text{ursprungliga längden}} \quad (3)$$

Den tekniska töjningen kan endast användas om $\mathcal{E} \ll 1$ vid elastisk deformation då den annars bidrar till grova beräkningsfel. ¹

2.3 Finita elementmetoden (FEM)

Finita elementmetoden växte fram för att kunna lösa komplicerade partiella differentialekvationer som inte går att lösa med klassiska metoder. Det kan exempelvis vara att kroppen har svår geometri eller komplicerade randvillkor. I finita elementmetoden delas en verklig struktur eller solidkropp upp i ett antal element. Kallas även att göra en mesh (ett nät) över en 1D, 2D eller 3D-kropp. Elementen har enkel geometri som små rektanglar eller trianglar. I varje hörn i elementen finns så kallade noder. Varje nod har frihetsgrader som beskriver i vilka riktningar noden kan förflytta sig. Genom att ansätta randvillkor över figuren och ange hur frihetsgraderna ska förskjutas kan töjningar och spänningar för varje element i figuren beräknas. De linjära ekvationerna definierar de konstitutiva sambanden samt lokala och globala storheter som exempelvis styvhet och förskjutning. Ekvationerna byggs upp i matriser för att kunna hantera den stora mängden information från alla element. De konstitutiva sambanden beskriver materialegenskaperna. De lokala storheterna syftar på varje enskilt element medan de globala är för hela kroppen. För varje element beräknas en lokal styvhetsmatris \mathbf{K}_e som sedan måste transformeras till en global styvhetsmatris \mathbf{K} . Den globala styvhetsmatrisen behövs för att kunna lösa de okända nodförskjutningarna. Då sambandet mellan den globala styvhetsmatrisen \mathbf{K} , ekvivalenta nodkrafter \mathbf{f} och nodförskjutningar \mathbf{d} ges av ekvation 4 nedan: ²

$$\mathbf{K} \times \mathbf{d} = \mathbf{f} \quad (4)$$

För att transformera lokala styvhetsmatrisen till den globala multipliceras den lokala styvhetsmatrisen med en transformationsmatris. Ekvationen för att transformera styvhetsmatriserna skrivs enligt ekvation 5. Transformationsmatrisen omplaceras den lokala matrisen till ett globalt koordinatsystem. När matriserna transformerats till det globala koordinatsystemet kan töjningar och spänningar i hela figuren beräknas. ³

$$\mathbf{K} = \mathbf{T} \times \mathbf{K}_e \quad (5)$$

För att se om beräkningarna närmar sig ett särskilt värde undersöks om resultaten man fått konvergerar mot antalet element i figuren. Genom att ändra antalet element och successivt öka

dessa kan man avläsa när resultaten närmar sig ett visst värde. När detta inträffar har man en tillräckligt bra modell. Beroende på vilka beräkningar som skall tas fram kan modellen konvergera vid olika resultat. Exempel på beräkningar som undersöks för konvergens är spänningar och töjningar. Dessa kan då konvergera vid olika antal element.²

I praktiken används finita elementmetoden för hållfasthetsanalys i program som exempelvis ABAQUS⁴ och ANSYS⁵ men kan även appliceras i Matlab.

2.4 Fysikaliskt arbete

För att beräkna tillförd energi för en kropp i FEM används ekvation 6. \mathbf{K} är den globala styvhetsmatrisen och \mathbf{u} är nodförskjutningsmatrisen.⁶

$$W = \frac{(\mathbf{u}^T \times \mathbf{K} \times \mathbf{u})}{2} \quad (6)$$

Energiekvationen kan härledas ur jämviktsekvationen enligt ekvation 7.⁷

$$W = \text{kraft} \times \text{väg} \quad (7)$$

2.5 Interpolation

Interpolation är en metod för att utifrån kända omkringliggande värden approximativt bestämma mellanliggande värden. Den vanligaste varianten av interpolation är linjär interpolation.⁸

Ekvationen för linjär interpolation visas i ekvation 8 nedan:

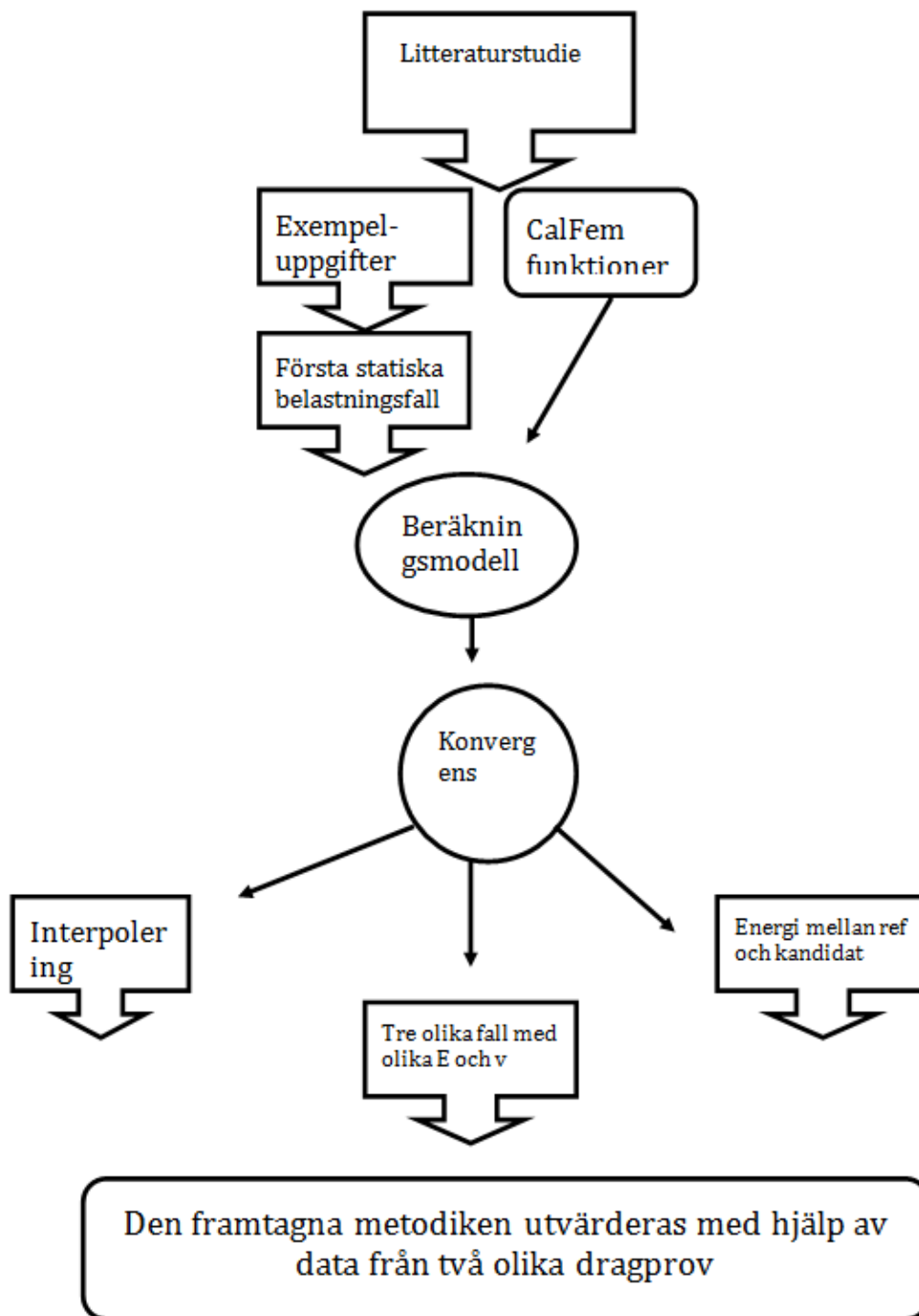
$$y = y_0 + \frac{y_1 - y_0}{x_1 - x_0} (x - x_0) \quad (8)$$

Där (x_0, y_0) och (x_1, y_1) är två kända punkter. y är det okända värdet man vill interpolera fram ifrån ett givet värde x som ligger emellan x_0 och x_1 . Det finns flera typer av interpolation. Flera omkringliggande punkter kan tas hänsyn till för att få fram ett närmevärde emellan. Metoden blir noggrannare om det finns många kända omkringliggande värden. Ju färre kända värden som finns ju sämre blir approximationen. Om avståndet mellan de kända värdena är långt kan approximationen också bli försämrad. Motsatsen till att interpolera kallas extrapolera. Det innebär att istället för att interpolera fram ett mellanliggande värde, extrapoleras ett värde fram som ligger utanför det kända mätområdet.⁹

3. METOD

För att kunna uppnå projektets slutmål har arbetet delats in i nio olika delar. Detta görs för att det skall bli enklare att kunna genomföra projektet och för att hålla ett jämnt tempo igenom hela projektet.

1. Litteraturstudie om CalFem, finita elementmetoden och hållfasthetslära.
2. Utföra exempeluppgifter från CalFem i Matlab.
3. Välja ett första statiskt belastningsfall och utföra detta utan större hjälp av funktionsfiler.
4. Skapa en kod i Matlab där en referensmodell skapas med $E=210$ GPa och $\nu=0.3$.
5. Undersök ifall resultaten förskjutningar, töjningar och spänningar konvergerar vid något specifikt elementantal.
6. Genomföra beräkningar på tre olika fall där varje fall har olika materialegenskaper.
7. Jämförelse av energi mellan referensfallet och olika kandidater.
8. Interpolering.
9. Metodiken utvärderas med hjälp av data från två olika experiment.



Figur 3.1 Flödesschema över hur de olika metoddelarna hänger ihop och i vilken ordning de genomförs.

Det första som genomförs är en litteraturstudie över de olika områden som arbetet berör, hållfasthetslära, finita elementmetoden och CalFem. Inom hållfasthetsläran är Poissons tal och

elasticitetsmodulen två väldigt viktiga parametrar. Att förstå hur dessa parametrar påverkar ett material och dess egenskaper är därför ytterst viktiga i detta projekt, då olika fall med olika materialegenskaper skall prövas.

De beräkningar som utförs i projektet bygger på finita elementmetoden (FEM). En inledande del av projektet handlar därför om att sätta sig in i FEM. I detta projekt skall analyserna genomföras i Matlab där programmet heter CalFem och information om hur just CalFem fungerar och används behövs därför.

Matlab är ett programmeringsprogram för tekniska och naturvetenskapliga beräkningar. Programmet har inbyggda funktioner för att lösa numeriska beräkningar inom linjär algebra och matematisk analys. Matlab har även kommandon för att grafiskt rita upp och presentera data i 2D-och 3D.¹⁰

CalFem är en förkortning av ”Computer Aided Learning of the Finite Element Method”. Programmet är uppbyggt av Matlab-funktioner som bygger på finita elementmetoden. CalFem har skapats och utvecklats vid avdelningen för Byggnadsmekanik på Lunds universitet. Tillsammans med CalFem programmet finns även en manual där uppbyggnaden och funktionerna är beskrivna samt övningsexempel. Funktionerna i programmet är uppdelade i olika grupper som material-, element- eller systemfunktioner. Materialfunktionerna innehåller funktioner för exempelvis linjär elastisk material (Hookes lag). De bilda materialmatriser med E-modulen och Poissons tal samt antaganden som exempelvis plan töjning eller plan spänningstillstånd. Elementfunktionerna beskriver olika elementtyper, som exempelvis balkelement, rektangulära eller 3D solider. Systemfunktionerna omfattar funktioner som exempelvis kan beräkna nodförskjutningarna, töjningar och spänningar.¹¹

Efter att en grundlig förståelse skapats för CalFem genomförs en del av de exempeluppgifter som finns i manualen. Detta för att få en djupare förståelse och för att se hur funktionerna fungerar.

3.1 Första statiska belastningsfallet

Därefter väljs ett första belastningsfall där en rektangulär kropp med bestämda randvillkor skall analyseras. Resultat som kan tas fram är: en plott på den odeformerade och deformerade kroppen

samt förskjutningar. För att få en bättre förståelse över hur CalFemfunktionerna beräknar resultaten så används väldigt få CalFem funktioner. Istället skrivs koden upp på egenhand, se bilaga 1. Detta betyder att t.ex. kroppens form skapas med egen kod och även dess randvillkor. Genom att utföra det första belastningsfallet med egen kod skapas en förståelse som annars hade varit svår att få.

En referensmodell skapas med E-modulen 210GPa och poissons tal 0.3. Detta är typiska värden för ett konstruktionsstål. Denna modell är den beräkningmodell som skall kalibreras och skapas med en generell Matlabkod med de alla olika CalFem funktioner som behövs, se bilaga 2. De resultat som tas fram är en plott över hur kroppen ser ut både innan och efter deformation, spännings- och töjningsplottar för att se var max och min värden uppstår i kroppen. I koden skall endast antalet element, materialets egenskaper och dimensioner behöva anges. Därefter när programet körs skall Matlabkoden räkna ut alla resultat.

Antalet element som kroppen skall delas in i behövs bestämmas. En kropp kan delas in i oändligt många element till bara ett element. För att analyserna inte skall ta alldeles för lång tid och för att resultaten skall bli trovärdiga undersöks om resultaten efter ett visst antal element konvergerar.

Referensmodellen delas in i det antal element som bestämts. Nästa steg i processen är att genomföra tre olika fall där materialegenskaperna ändras inför varje fall. Detta genomförs för att se hur resultaten ändras när t.ex. E-modulen ökar gentemot referensmodellen. De resultat som jämförs är max spänning, töjning och nodförskjutning.

3.2 Kalibreringsmetodik

För att kunna undersöka ifall beräkningsmodellen som skapats i Matlab fungerar och kan hitta materialegenskaperna så skapas målfunktioner. De materialegenskaper som skall hittas är referensens egenskaper, alltså E-modulen 210 GPa och poissons tal 0.3. Energin som tillförs till provstaven med referensens egenskaper kan beräknas, ekvation 6. För att räkna ut den energi används referensens förskjutningar \mathbf{u} och dess globalastyvhetmatris \mathbf{K} . Referensmodellens kropp har förskjutits på ett visst sätt efter att randvillkoren har satts. Referensens noder är då placerade på specifika platser. Materialegenskaperna skall nu hittas genom att används ekvation 6 och de två målfunktioner som skapas ekvation 9 och 10.

$$\mathbf{R}_u = \mathbf{W} - \mathbf{W}_R \quad (9)$$

$$\mathbf{R}_{bc} = \mathbf{W}_k - \mathbf{W}_R \quad (10)$$

Det som jämförs är hur mycket tillförd energi som skiljs mellan referensen och de kandidater som skapas, bilaga 5 och 6. Kandidaterna som skapas är olika kroppar med varierande materialegenskaper, fast med samma dimension som referensmodellen. Kandidaternas tillförda energi räknas också ut, dock räknas två olika energier ut för kandidaten. Dessa olika energier kallas \mathbf{W} och \mathbf{W}_k . Den energi \mathbf{W} hos kandidaten är den energi som har behövts tillföras till kandidaten. Den energi som har blivit tillagd för att tvinga kandidatens noder till samma placering som hos referensen. Kandidatens andra energi som beräknas är \mathbf{W}_k . Detta är den energi som tillförts för att kroppen skall uppnå samma randvillkor som referensen. Alla noder i kandidatens kropp är alltså fria, förutom de noder där randvillkor existerar.

Målfunktionerna som skapats för att hitta materialegenskaperna \mathbf{R}_u och \mathbf{R}_{bc} står för den energiskillnad som skiljer referensen tillfördaenergi mot kandidatens tillfördaenergi. Referensens energi \mathbf{W}_R är den energi som krävs för att referensen skall ändra form utefter de randvillkor som sats. Om båda målfunktionerna är noll betyder detta att kandidaten har samma materialegenskaper som referensen.

3.3 Digital image correlation (DIC)

För att kunna koppla samman ett verkligt experiment med beräkningsmodellen tas data in från ett dragprov via Digital image correlation (DIC), se figur 3.2. Data som tas in i beräkningsmodellen är provstavens förskjutningar och den kraft som provet belastats med.



Figur 3.2, Dragprovstrustning.

Digital image correlation (DIC) är en bildanalysmetod baserad på digitala bilder. Ur bilderna kan konturer och förskjutningar utläsas av ett föremål under belastning. I detta projekt är det en provstav som belastas och analyseras. För att spåra förändringar i provstaven skapas ett slumpmässigt mönster genom att spraya provstaven med vit och svart färg, se figur 3.3.



Figur 3.3, En provstav som blivit sprayad med vit och svart färg för att skapa ett slumpmässigt mönster som metoden kan registrera.

DIC registrerar ett antal slumpmässiga mönster som existerar på provstaven och analyserar hur dessa mönster förändras under att provstavens belastas. Var dessa mönster är placerade på provstaven registreras i ett koordinatsystem. Varje bild registrerar var mönstren är placerade i koordinatsystemet. Hur mönstren har förskjutits fås fram via att jämföra första och sista placeringarna.¹²

Då DIC's antal registrerade punkter/områden skiljer sig till beräkningsmodellens antal nodpunkter kan en metod som heter interpolering användas. Denna metod används för att beräkningsmodellens nodpunkter skall kunna representera experimentets placeringar.

3.4 Målfunktioner vid de verkliga experimentet

När kunskap om hur interpolering genomförs så analyseras två olika dragprov med hjälp av DIC. Den första provbiten som analyseras har dimensionerna 8mm*4mm*2mm och 678 punkter tas ut på provbiten. Hur dessa punkter förskjuts under dragprovet analyseras. Eftersom antalet noder i beräkningsmodellen är 180 stycken skall dragprovets värden interpoleras. De 678 stycken punkter som kommer från experimentet skall representeras via de 180 stycken noder som existerar i beräkningsmodellen, se bilaga 8. Det andra experimentet som genomförs så har provbiten dimensionerna 8mm*4mm*2,7mm och 608 punkter/områden på provet tas ut. Dessa skall på samma sätt som i första experimentet interpoleras till 180 stycken noder, se bilaga 12. De analyser som skall genomföras hos dessa två experiment är i princip samma analyser som genomfördes för de olika kandidaterna, dock ser målfunktionerna annorlunda ut. Två målfunktioner skapas, ekvation 11 och 12.

$$\mathbf{R}_u = \mathbf{W}_u - \mathbf{W}_{bc} \quad (11)$$

$$\mathbf{R}_{bc} = (\mathbf{F}_{nod} - \mathbf{F}_{exp}) \times (\mathbf{U}_{topp} - \mathbf{U}_{botten}) \quad (12)$$

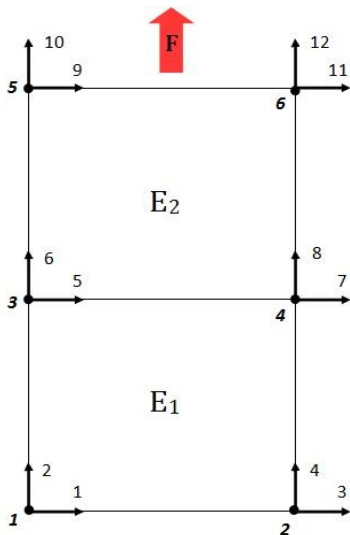
\mathbf{R}_u står för den energiskillnad som uppstår mellan \mathbf{W}_u och \mathbf{W}_{bc} . Där \mathbf{W}_u är den energi som krävs för att förflytta alla kandidatens nodplaceringar samma plats som experimentets. \mathbf{W}_{bc} är den energi som krävs för att kandidatens kroppen bara skall uppfylla randvillkoren. Samma randvillkor som experimentet har. Är \mathbf{R}_u lika med noll betyder det att ingen energi skiljer de olika kropparna åt, \mathbf{W}_{bc} har då samma nodplaceringar som \mathbf{W}_u .

\mathbf{F}_{exp} är den kända kraft som har påverkat provstaven under dragprovet. \mathbf{F}_{nod} varierar beroende på vilka materialegenskaper provstaven har, vilken kandidat som undersöks i beräkningsmodellen. Så länge någon av målfunktionerna inte är noll så skiljer sig kandidaten från de verkliga provet.

4. GENOMFÖRANDE

4.1 Första statiska belastningsfallet

Ett första statiskt belastningsfall genomförs för att få en djupare förståelse över hur element, noder och frihetsgrader fungerar. En rektangel med bredden 5mm, höjden 10mm och tjockleken 1mm. De randvillkor som existerar på kroppen är att den är fast inspänd i ena kortsidan medan den andra kortsidan skall förskjuts uppåt 2mm. Rektangeln delas upp i 2 element (E1 och E2) vilket ger 6 stycken noder och 12 stycken frihetsgrader, se figur 4.1.



Figur 4.1 Beskrivning av element (E1 och E2), Noder (de punkter som existerar i varje element hörn) och frihetsgrader (varje pil som existerar i varje nod)

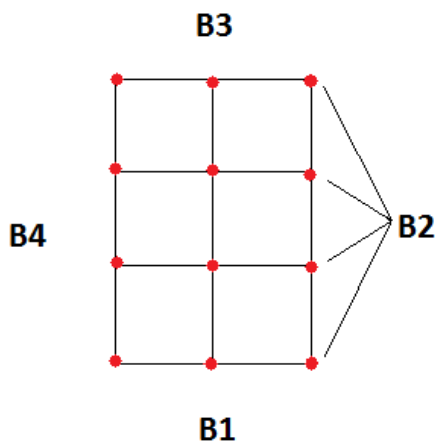
De funktioner från CalFem som används för att lösa problemet är följande: **hooke**, **planqe**, **assem** och **solveq**. Det första steget är att få fram modellens materialmatris **D**. Matrisen beskriver hur materialet beter sig, bestäms utifrån materialets egenskaper E-modulen och poissons tal. Materialstrukturen fås fram genom att använda CalFem funktionen **hooke**. När kroppen delas in i flera element beräknas varje element för sig. Varje element får en lokal styvhetsmatris **K_e**, dessa räknas fram genom CalFem funktionen **planqe**. För att få fram hur hela kroppen påverkas används CalFem funktionen **assem**. Denna funktion kopplar samman de lokala elementstyvhetsmatriserna till en global styvhetsmatris **K**. Beskrivning av varje elements noder och frihetsgrader har skrivits in i Matlab utan användning av CalFemfunktionerna. Detta för att förstå uppbyggnaden av kroppen. Alla okända parametrar är nu framtagna och resultat om hur

modellen deformeras och med vilken kraft varje frihetsgrad har påverkats kan nu fås fram med funktionen *solveq*, se bilaga 1.

En kropp delas oftast in i många element. Att skriva in alla dessa element för hand skulle då vara tidkrävande. Genom att då skapa nya funktioner och använda fler av CalFemfunktionerna så har en generell kod tagits fram, se bilaga 2.

4.2 Matlabkod där en referensmodell skapas

För att undgå att skriva in antalet element, nodernas koordinater och dess frihetsgrader förhand i Matlabkoden har funktionen *quad_mesh* används och ersatt detta arbete. Genom att skriva in rektangelns x – och y -koordinat längst ner till vänster och längst upp till höger så skapas storleken och utseendet på kroppen. Dimensionerna på den nya kropp som analyseras är 5mm*10mm*1mm. Antalet element som kroppen delas in i skapas genom att antalet element i x- och y-led skrivs in. Sista delen som anges för att kunna använda funktionen *quad_mesh* är antalet frihetsgrader som skall förekomma i varje nod. Funktionen *quad_mesh* får nu fram *Edof*, varje elements tillhörande frihetsgrader och matriser med frihetsgraderna för yttersidorna på kroppen. Yttersidornas frihetsgrader på kroppen sparas i vektorerna **B1**=bottensidan, **B2**=högersidan, **B3**=ovansidan, **B4**= vänstersidan, se figur 4.2.



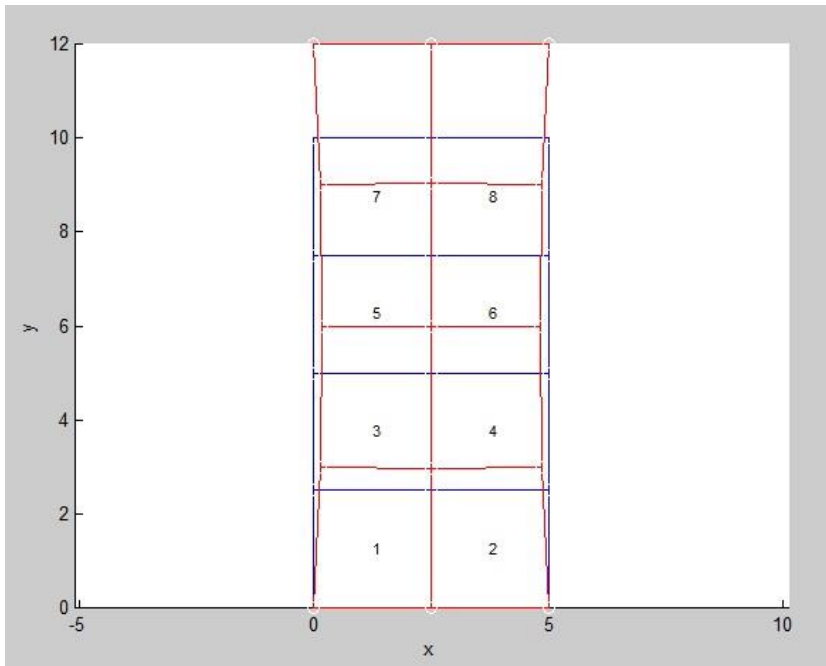
Figur 4.2, Förklaring vilka noder *quad_mesh* får ut som benämns med **B1**, **B2**, **B3** och **B4**. Alla röda punkter i figuren är noder. De noder som tillhör **B2** är yttre noderna till höger, detta visas genom strecket som är draget från **B2** till de tillhörande noderna.

Randvillkoren som skapas inför det första statiska belastningsfallet används återigen på denna nya kropp som belastas. Botten av rektangeln skall vara fast inspänd och den övre sidan skall dras upp 2mm. En funktion *randvillkor* skapas för att ange randvillkorsmatrisen *bc*, se bilaga 3. För att sätta randvillkor på frihetsgraderna i botten och i toppen på kroppen används matriserna *B1* och *B3* som angivits ur funktionen *quad_mesh*. *B1* skall vara fast inspänd i både x -och y-led medan *B3* inte skall röra sig i x-led men dras upp 2mm i y-led.

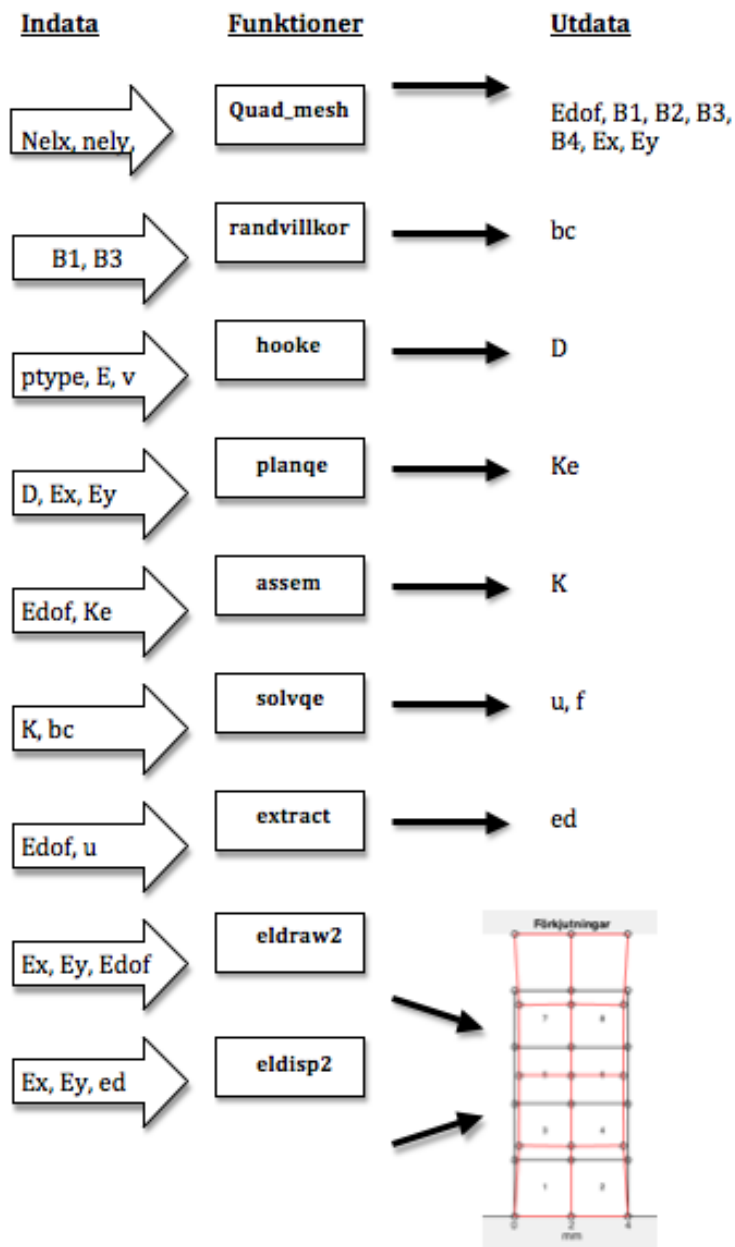
Den lokala styvhetsmatrisen för varje element tas fram genom användning av funktionen *planqe*. Denna funktion användes också i första statiska belastningsfallet, enda skillnad är att en for-lopp skapas då det är många element som skall föras in. När den första lokala styvhetsmatrisen har skapats i for-loppen förs denna in i nästa funktion *assem*, där en global styvhetsmatris skapas efter att varje lokal styvhetsmatris förts in.

Den funktion som används för att få fram u =frihetsgradsförskjutning och f =ekvivalenta nodkraftsvektorn är *solveq*. Indata i *solveq* är den globala styvhetsvektorn *K* och randvillkorsmatrisen *bc*.

För att se hur kroppen ser ut och hur många element den har indelats i används funktionen *eldraw2*. Den indata som krävs för att plotta upp kroppen är *Edof*, koordinaterna i x-led och y-led för varje element och ange vad för utseende figuren skall ha, ränder, punkter, färg osv. När detta förts in i funktionen plottar Matlab upp kroppen, se figur 4.3. För att få fram hur figuren ser ut efter att den deformerats används funktionen *eldisp2*. Här anges koordinaterna i x-och y-led för varje element precis som i funktionen innan, elementförskjutningsvektorn *ed* samt en skalfaktor som kan användas om förskjutningarna är väldigt små. Ett flödesschema över hur funktionerna används visas i figur 4.4.



Figur 4.3, Utseende på kroppen före och efter deformation. Kroppen som är plottad med blå färg är den odeformerade och den röda är den deformerade kroppen.



Figur 4.4, Vilka indata olika funktioner behöver för att kunna få fram utdata.

Andra resultat som tas fram är plott med förskjutningsvektorer, spänning och töjningsplottar. Förskjutningsvektorerna som existerar i varje nod plottas upp med Matlabkommandot *quiver*. Denna funktion plottar pilar i varje nod som visar vilken riktning förskjutningen sker i. För att ta fram spännings- och töjningsplottar i x-och y-led tas värdena fram via funktionen *planqs*. Dessa plottas upp med hjälp av Matlabkommandot *fill*. Då fylls den deformerade kroppen i med olika färger motsvarande töjning och spänningsvärde för varje element. De skapas två olika spänningsplottar, en för spänningarna i x-led och en annan för i y-led. Samma utförande sker för töjningen.

4.3 Konvergens

För att få fram antalet element som referensmodellen skall delas in i, undersöks om resultaten konvergerar. Referenskroppen delas först in i två element och resultat som tas fram är: max spänning i x -och y-led, max töjning i x -och y-led och den näst största nodförskjutningen som existerar i kroppen (Den näst största nodförskjutningen tas pga. den största nodförskjutningen i kroppen existerar vid randvillkoren). Detta arbete utförs och dokumenteras för varje gång antalet element ändras. Antalet element som analyseras är från 2 till 200 stycken element med cirka 10 steg mellan varje. Om den tredje decimalen är lika i resultaten antas värdena vara tillräckligt nära varandra för att säga att de konvergerar. Beroende på vad undersökningen söker kan värdena sägas konvergera vid olika noggranheter.

Det är ytterst viktigt att kontrollera om de värden som tas ut är placerade omkring samma område på kroppen, annars kan de vara en felkälla till att värdena inte konvergerar.

4.4 Tre olika fall med olika materialegenskaper

Tre olika fall genomförs där materialegenskaperna för varje fall ändras. Den Matlabkod som används även här är bilaga 2 och 3. Dimensionerna på kroppen är 4mm bred, 8mm hög och 2mm tjock. De olika fall är följande:

- Fall 1: Referensmodellen, E-modulen $E=210\text{GPa}$ och poissons tal $\nu=0,3$.
- Fall 2: E-modulen $E=400\text{GPa}$ och poissons tal $\nu=0,3$.
- Fall 3: E-modulen $E=210\text{GPa}$ och poissons tal $\nu=0,1$.

De resultat som tas fram för de olika fallen är max spänning, töjning och förskjutning. I fall 2 ökas E-modulen gentemot fall 1, detta för att se hur E-modulen påverkar resultaten. I fall 3 minskas Poissons tal gentemot fall 1, detta för att se hur Poissons tal påverkar resultaten.

4.5 Jämförelse av energi

Det finns många olika metoder för att jämföra olika kandidater med varierande materialegenskaper mot referensmodellen, den metod som valts att användas i detta projekt är att mäta skillnaden i tillförd energi. För att få fram hur mycket energi som tillförs i varje fall används ekvation 6 och genom två skapade målfunktioner ekvation 9 och 10 jämförs kandidatens energi gentemot referensens energi. De olika kandidaterna har olika materialegenskaper men randvillkoren och antalet element på de olika kandidaterna förblir oförändrade.

Två olika Matlabkoder skapas där första koden är när kandidaterna varierar E-modulen mellan 100GPa till 300GPa och Poissons tal är 0,3, se bilaga 6. Den andra koden är när kandidaterna varierar Poissonstal mellan 0,1 till 0,4 och E-modulen är 210GPa, se bilaga 5. Det resultat som tas fram för den första koden är följande:

1. Den energi som krävs för att kandidaten skall uppnå samma nodplaceringar som referensen, ekvation 6.
2. Den energi som krävs för att uppnå den första målfunktionen, ekvation 9.
3. Den energi som krävs för att uppnå den andra målfunktion, ekvation 10.
4. Skillnaden i energi för kandidaterna att bara uppnå randvillkoren och sedan för att uppnå referensens nodplaceringar ekvation 13.

$$R_{nod} = W_k - W \quad (13)$$

De resultat som tas fram för den andra koden är följande:

1. Den energi som krävs för att kandidaten skall uppnå samma nodplaceringar som referensen, se ekvation 6.
2. Den energi som krävs för att uppnå den första målfunktionen, ekvation 9
3. Den energi som krävs för att uppnå den andra målfunktion, ekvation 10

4.6 Inverkan av elementstorlek

Då data skall tas in från ett dragprov där indelningen av kroppen kan variera gentemot beräkningsmodellens indelning så undersöks hur interpolationen påverkar energi mängden. Detta undersöks genom att ändra antalet element på beräkningsmodellen. Antalet element börjar på 50 och ändras sedan men jämna steg upp till 200 stycken element. Vid varje steg räknas den tillförda energin ut, se bilaga 7.

4.7 Experimentdata från två olika dragprov

Två dragprov genomförs och analyseras med DIC. Det första experimentet som genomförs är en rektangulär provbit med dimensionerna 8mm*4mm*2mm. DIC registrerar 678 stycken specifika punkter/områden och hur dessa punkter/områden förflyttas under belastning. För att inte beräkningarna skall ta för långtid har antalet noder/punkter i beräkningsmodellen valts till 180 stycken. Detta betyder att kroppen har delats in i 152 stycken element. Att antalet element valts just till 152 stycken är pga. den undersökningen som skedde tidigare i projektet för att se när resultaten konvergerade. DIC registrerar förändringar i 678 stycken punkter/områden. Beräkningsmodellen registrerar endast 180 stycken nodförändringar. Detta innebär att en interpolering mellan värdena behöver ske, se bilaga 8. I nästa experiment som genomförs har provbiten ny dimensioner 8mm*4mm*2,7mm. DIC registrerade 608 stycken punkter/områden på denna provbit och därefter genomförs samma steg som för det första experimentet, se bilaga 12. En ny funktion *randvillkorEXP* skapas för att ange de randvillkor som förekommer på provstaven, se bilaga 11.

Efter att interpolering skett beräknas energitillförseln då E-modulen och poissons tal varierar, ekvation 6. Kandidater skapas med varierande E-modul mellan 70GPa till 115GPa och konstant poissons tal 0.3, bilaga 9 och 13. Därefter varierar poissons tal mellan 0.25 till 0.35 och E-modulen är konstant 70 GPa se bilaga 10 och 14. För att finna dragprovets materialegenskaper jämförs energiskillnaden mellan de olika kandidater som skapats och den energitillförseln i experimentet. De två målfunktioner som räknas ut är ekvation 11 och 12. När båda dessa målfunktioner är noll betyder det att ingen energiskillnad existerar mellan experimentet och kandidaten.

Istället för att variera materialegenskaperna i två olika omgångar och i två olika Matlabfiler så skapas en ny Matlabkod, se bilaga 15 och 16. I den nya koden ändras båda parametrarna och en plott skapas för att visa inom vilket intervall poissons tal ligger och en plott för inom vilket intervall E-modulen borde ligga.

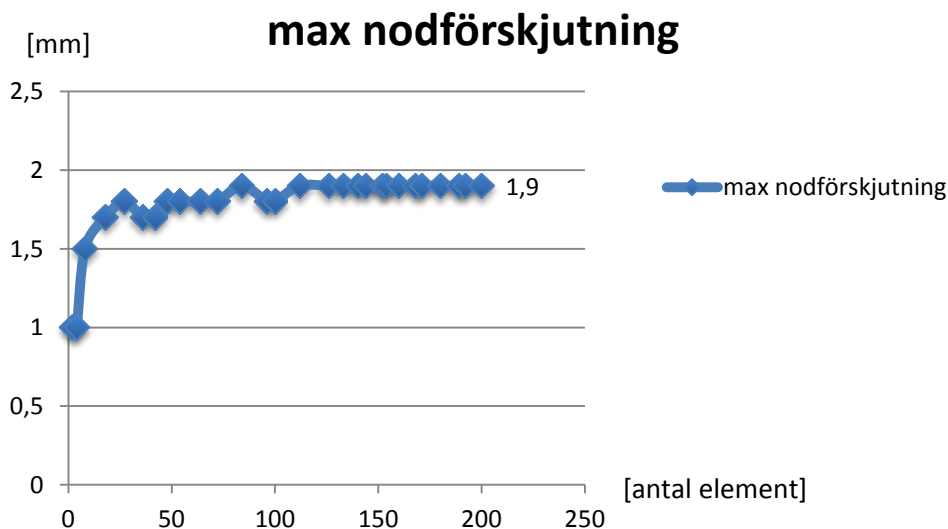
5 ANALYS OCH RESULTAT

I analys och resultat redogörs alla resultat som projektet har fått fram samt analyser utifrån resultaten.

5.1 Konvergens

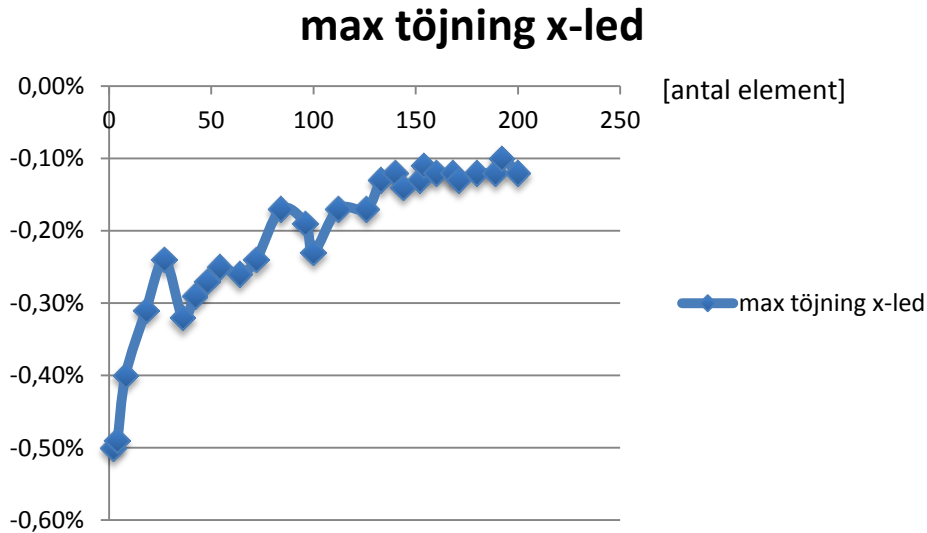
Beräkningsmodellen konvergerade vid 152 element. Konvergens undersöktes i max nodförskjutning samt max töjning och spänning i x-och y-led. De materialegenskaper som användes var E-modulen 210GPa och poissons tal 0,3. Se bilaga 4 för fullständig konvergenstabell.

Maximala nodförskjutningen konvergerade väldigt bra efter 100 element och var stadigt 1,9mm. Se figur 5.1.



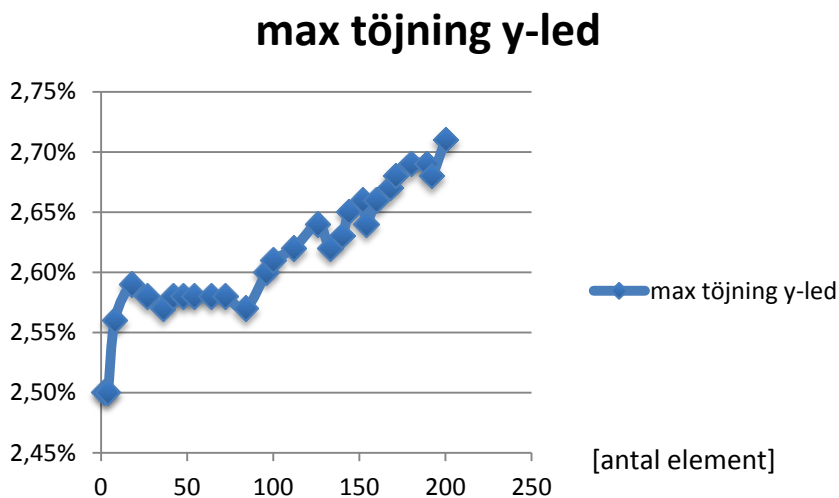
Figur 5.1 Konvergens för maximala nodförskjutningen.

Maximala töjningen i x-led konvergerade väldigt tydligt vid -0,13% vilket skedde efter 100 element, se figur 5.2. Värdena varierande i början mellan 2 till 100 element men efter det stabiliserade sig kurvan.



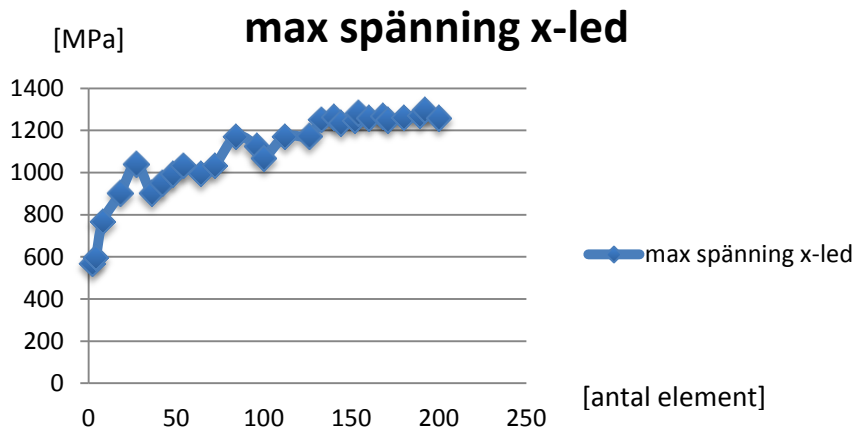
Figur 5.2 Konvergens för maximala töjningen i x-led.

Maximala töjningen i y-led stabiliserade sig inte lika tydligt, se figur 5.3. I början med mindre antal element varierade inte kurvan lika mycket som i slutet. Värdena varierade mellan 2,5% till 2,7% då element antalet varierades från 2 till 200 element. Vilket innebär att variationen var väldigt liten och att töjningsvärdet i y-led redan från början var tillräckligt bra. Vid 152 element var maximala töjningen i y-led 2,66%.



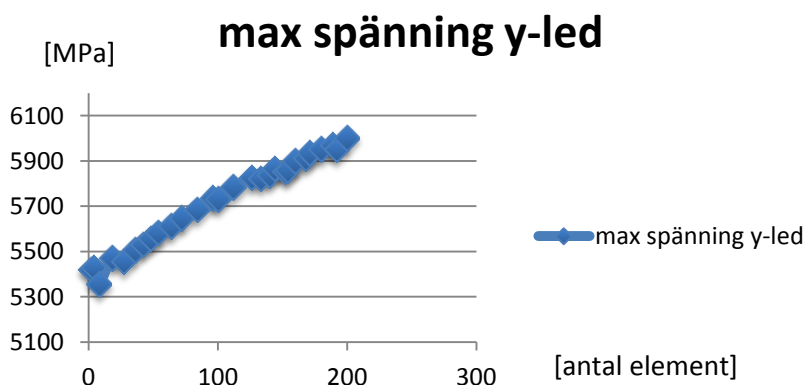
Figur 5.3 Konvergens för maximala töjningen i y-led.

I x-led konvergerade spänningen då antal element närmade sig 150 element. Vid 152 element var spänningen 1248,2MPa, se figur 5.4. Konvergensen visas genom att kurvan planar ut strax efter 100 element.



Figur 5.4 Konvergensen för maximala spänningen i x-led.

I y-led konvergerade aldrig spänningen, se figur 5.5. Kurvan ökar stadigt uppåt ju fler element på figuren. Det som är värt att notera är att efter 100 element kan man se att kurvan inte lutar lika mycket. I början är kurvan brantare. Detta kan påvisa att spänningen börjar konvergera i y-led. Då de tidigare resultaten konvergerat efter 150 element undersöktes inte hur spänningen ändrades efter 200 element. Konvergensen ansågs vara tillräckligt uppfylld. Vid 152 element var maximala spänningen i y-led 5853,2MPa.



Figur 5.5 Konvergensen för maximala spänningen i y-led.

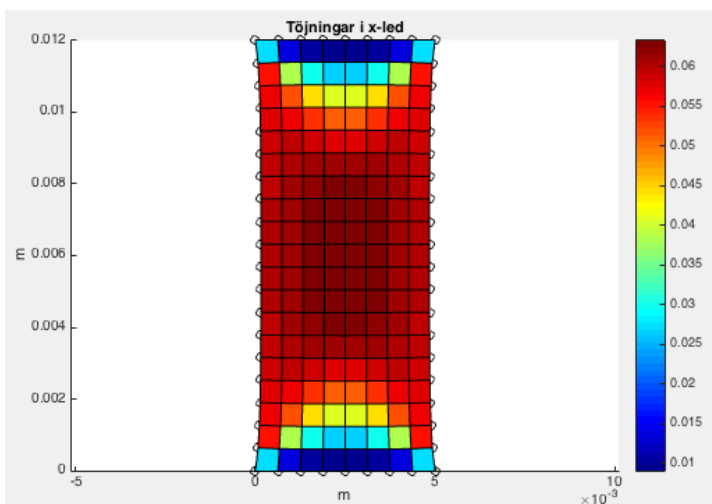
5.2 Tre olika fall med olika materialegenskaper

Tre olika fall analyserades där skillnaden mellan de olika fallen var att E-modulen och Poissons tal ändrades. I analysen undersöktes vad som påverkades när materialparametrarna ändrades. Resultat för maximal töjning och spänning i x- och y-led togs också fram. I det första fallet, referensfallet var E-modulen 210GPa och Poissons tal 0,3. I fall två ändrades E-modulen till 400GPa och i fall tre ändrades Poissons tal till 0,1 medan E-modulen återigen var 210GPa.

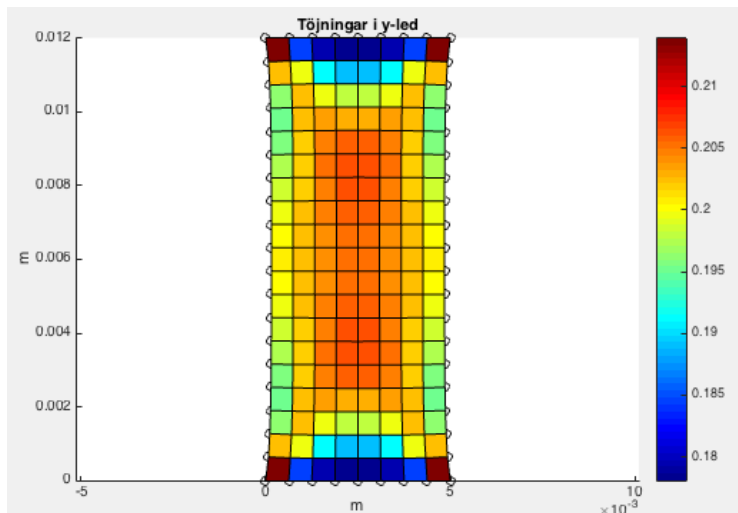
5.2.1 Fall 1 – Referensen

I första fallet var E-modulen 210GPa och Poissons tal 0,3. Maximala töjningen i x-led uppmättes -0,13%. Se figur 5.6. I x-led är töjningen negativ då det bildas en midja i kroppen likt ett dragprov. Störst töjning i x-led sker i mitten av kroppen. Vid randvillkoren för kroppen sker inte så stor töjningen i x-led. Färgskalan till höger i figur 5.6 visar töjning- och spänningsvärden. Rött är de högsta värdena och blått är de lägsta.

I y-led är maximala töjningen 2,66%. Se figur 5.7. Töjningen är betydligt större i y-led än i x-led. Störst töjning sker i de fyra hörnen i figur 5.7. Detta beror på randvillkoren och de skarpa kanterna där en stor töjningskoncentration bildas. På grund av detta är modellen inte så bra i dessa punkter. Vilket i sin tur leder till att störst fel uppskattningar uppkommer där. De höga värdena i kanterna är därför mindre realistiska. I övrigt sker de största töjningarna i y-led i mitten av figuren där midjan bildas.

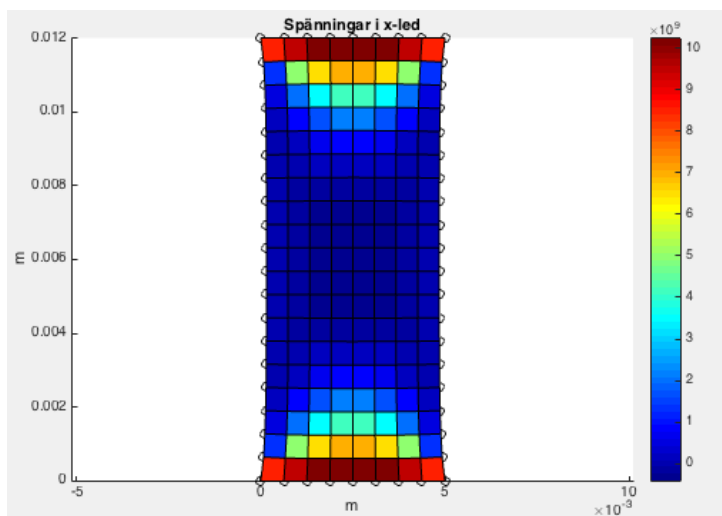


Figur 5.6 Töjningar i x-led för fall 1



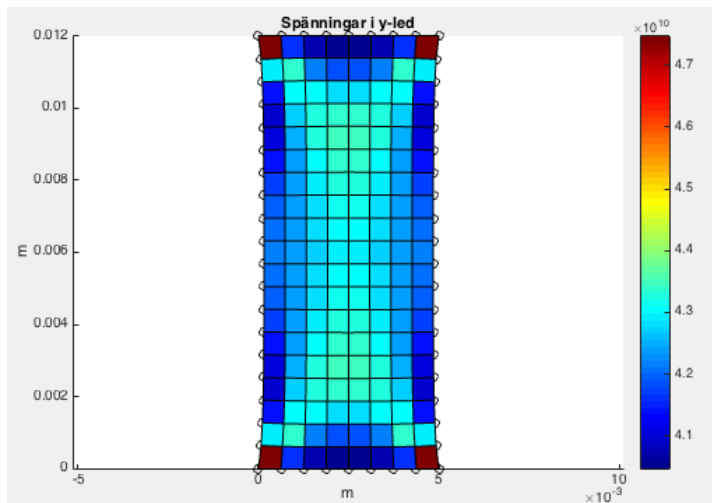
Figur 5.7 Töjningar i y-led för fall 1

Maximala spänningen i x-led uppmättes till 1248,2MPa, se figur 5.8. Störst spänning sker vid randvillkoren för figuren. Längst ner är figuren fast inspänd vilket bidrar till stora spänningar då figuren deformeras 2mm i y-led. Längst upp där förskjutnings villkor verkar uppkommer också stora spänningar. I övrigt är spänningen samma genom större delen av figur 5.8.



Figur 5.8, spänningar i x-led för fall 1

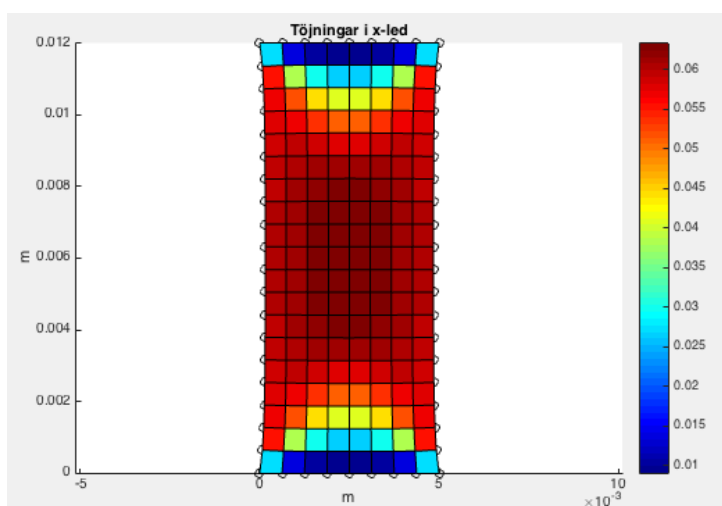
Maximala spänningen i längsgående riktning, y-led är 5883,2MPa, se figur 5.9. Största spänningen sker i ytterkanterna precis som maximala töjningarna i y-led och bör även här bortses ifrån på grund av spänningskoncentrationen vid just randvillkoren och de skarpa hörnen. Större delen av spänningen i kroppen varierar annars inte speciellt mycket.



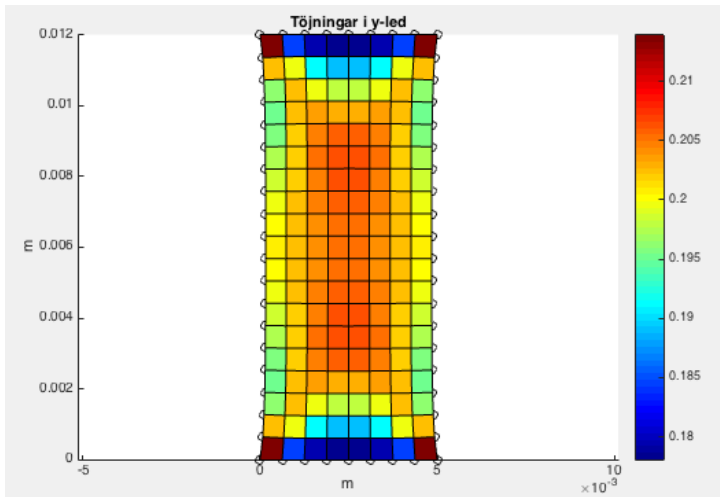
Figur 5.9, spänningar i y-led för fall 1

5.2.2 Fall 2

Genom att ändra E-modulen från 210GPa till 400GPa ökade styvheten i materialet. Detta bidrog till att de ekvivalenta nodkrafterna f ökade, vilket även gjorde att spänningen i x-och y-led ökade. Nodförskjutningarna u förblev detsamma vilket gjorde att töjningarna i x-och y-led inte ändrade sig ifrån fall 1. Se figur 5.10 och 5.11. Störst töjning i x-led är fortfarande i mitten av figuren där midjan bildas. Likadant sker för töjningarna i y-led bortsett från hörnen där störst feluppskattning sker, vilket bidragit till de stora töjningarna. Vid ökad E-modul påverkas alltså inte nodförskjutningar eller töjningar i x-och y-led.

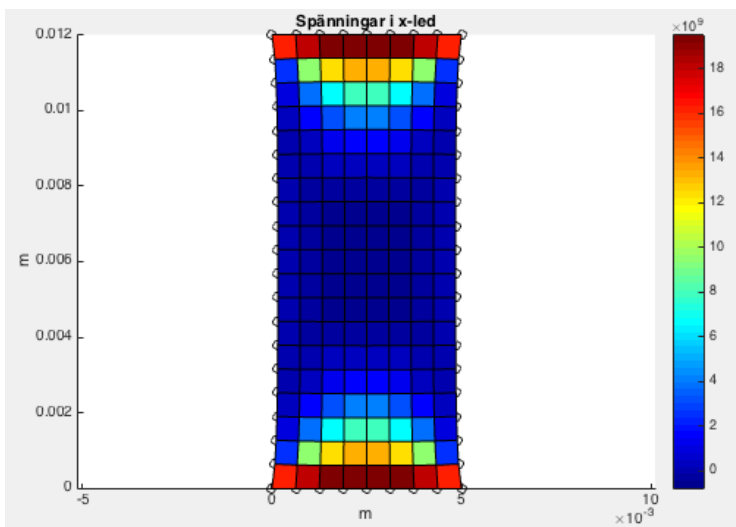


Figur 5.10 Töjningar i x-led för fall 2



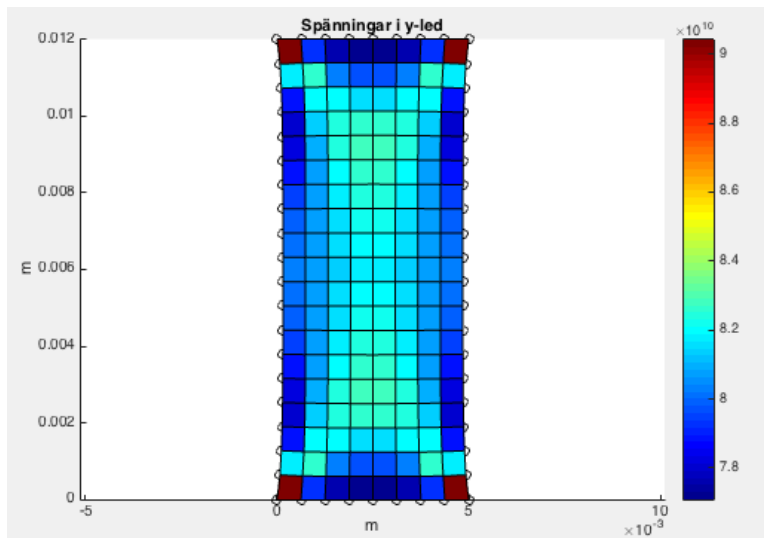
Figur 5.11 Töjningar i y-led för fall 2

Däremot ändrades spänningen när E-modulen förändrades. Vid ökad E-modul blev maximala spänningen i x-led 2378MPa. Spänningarna är fortfarande placerade på samma ställe som för fall1 men har ökat i storlek. Se figur 5.12.



Figur 5.12 Spänningen i x-led för fall 2

Även i y-led ökade maximala spänningen och uppmättes till 11206MPa. På liknande sätt påverkas spänningarna i y-led som i x-led. De ökade men fortfarande placerade på liknande platser som för fall 1. Se figur 5.13.



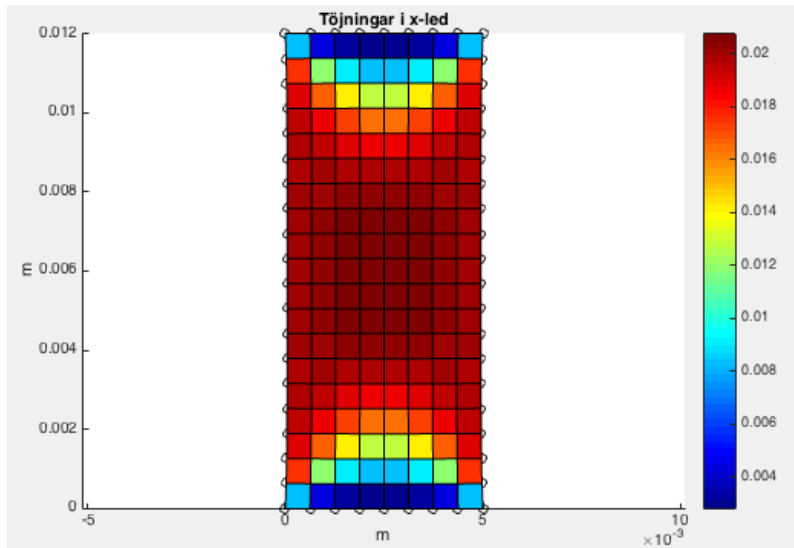
Figur 5.13 spänningar i y-led för fall 2

5.2.3 Fall 3

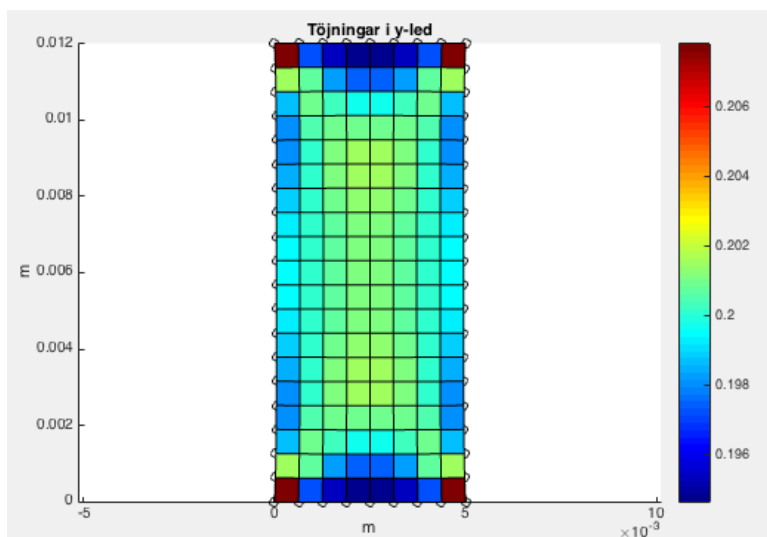
Genom att ändra Poissons tal från 0,3 till 0,1 blir materialet svårare att deformera i tvärgående riktning. Detta påverkade både de ekvivalenta nodkrafterna f och nodförskjutningarna u . Vilket i sin tur gjorde att spänningen och töjningen i tvärgående riktning, x-led minskade. Däremot ändrades inte spänningen och töjningen i längsgående riktning, y-led avsevärt mycket.

Maximala töjningen i x-led är -0,04% vilket är en stor skillnad från fall 1 där töjningen var -0,13% i x-led. Se figur 5.14. Materialet blir svårare att deformera ju mindre Poissons tal är. Vilket leder till mindre midjebildning hos materialet. Då deformationen är väldigt liten syns detta inte i figur 5.14 däremot märks det på värdena.

I y-led blev töjningen också mindre jämfört mot referensen. Maximala töjningen i y-led blev 2,54% jämfört mot 2,66% som tidigare fall gav. Se figur 5.15 där färgen tydligt visar mindre värden.

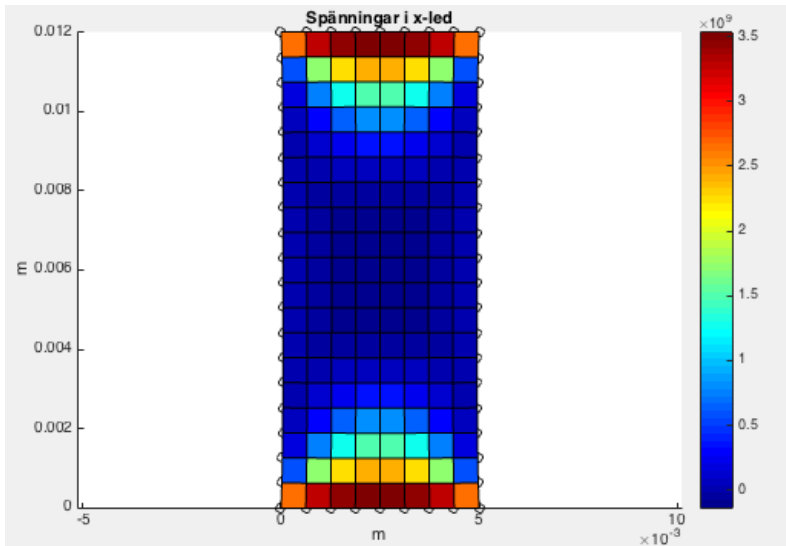


Figur 5.14 Töjningar i x-led för fall 3

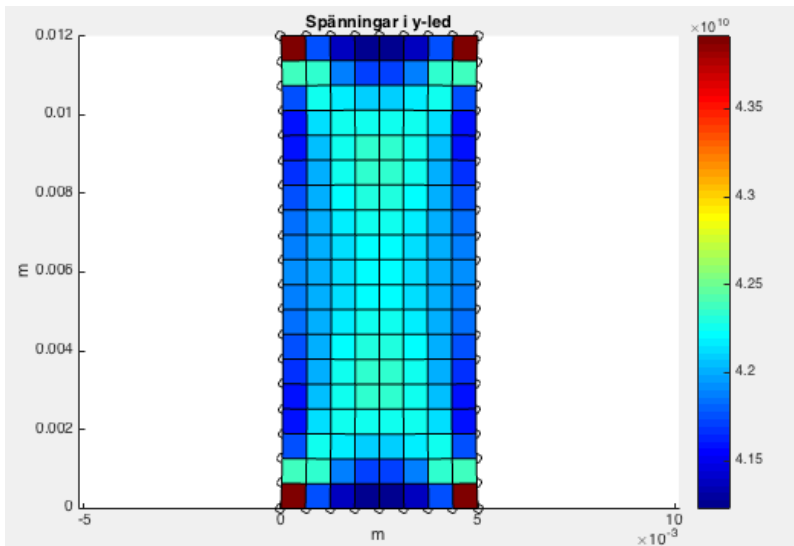


Figur 5.15 Töjningar i y-led för fall 3

Spänningarna påverkas på samma sätt som töjningarna. I x-led påverkas maximala spänningen mer än i y-led när Poissons tal ändrats. Maximala spänningen för fall 3 i x-led är 431,7MPa. Se figur 5.16. Maximala spänningen i y-led är 5472,4MPa. Se figur 5.17. Spänningen i x-led har inte påverkas speciellt mycket jämfört från referensen.



Figur 5.16 spänningar i x-led för fall 3



Figur 5.17 spänningar i y-led för fall 3

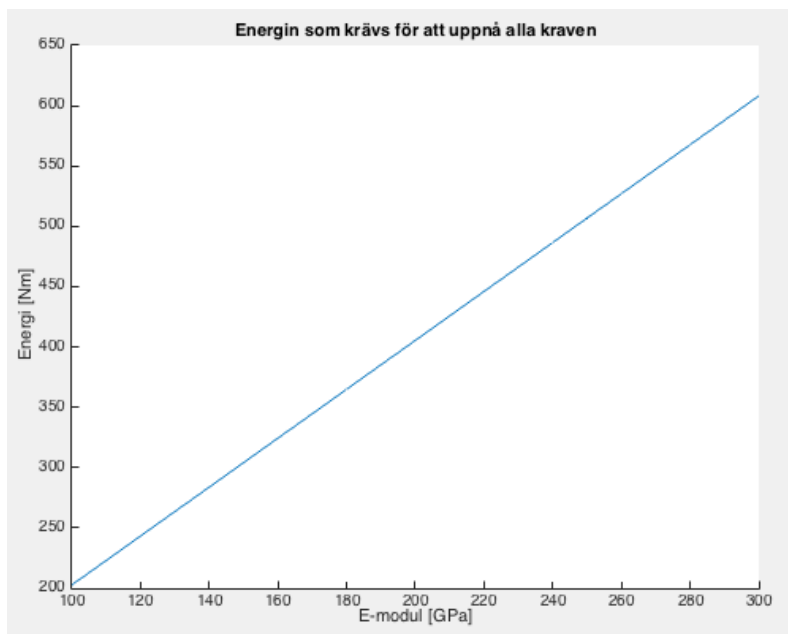
5.3 Jämförelse av energi

För att hitta materialparametrarna jämförs energiskillnader. Energin som tillförts i referensfallet jämförs med kandidaters tillförda energi. Kandidaterna har varierande värden på E-modulen och Poissons tal. Energiskillnaden ställs upp genom två målfunktioner. R_{bc} är skillnaden i energi mellan referensen och energin som krävs för kandidaterna att uppnå samma randvillkor som referensen, se ekvation 9. Den andra målfunktionen R_u är skillnaden i energi mellan referensen och kandidaterna som ska uppfylla samma randvillkor och samma placeringar på

nodförskjutningarna som referensen, se ekvation 10. För att uppfylla målfunktionerna skall båda ekvationerna vara lika med noll. Då har beräkningsmodellen hittat det rätta värdena på materialparametrarna.

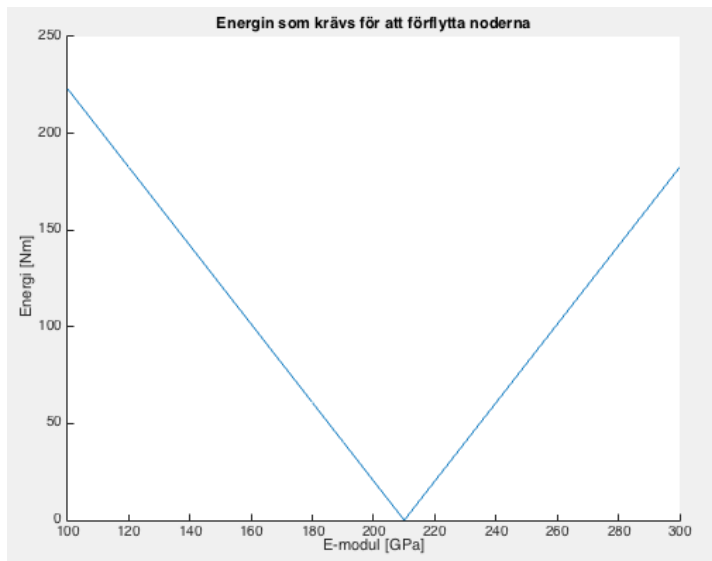
5.3.1 Jämförelse av energin då E-modulen varierar

Först undersöktes hur mycket energi som krävdes för nya kandidater med varierande E-modul till att få samma randvillkor och nodförskjutningar som referensen. E-modulen ändrades till olika värden i ett intervall mellan 100-300GPa. Det visade sig att ju större E-modulen är, ju styvare blir materialet. Då krävs mer energi för att uppnå kraven. Se figur 5.18. Vid E-modulen 210GPa krävs det 371,93Nm för att uppfylla randvillkoren och nodförskjutningarna.



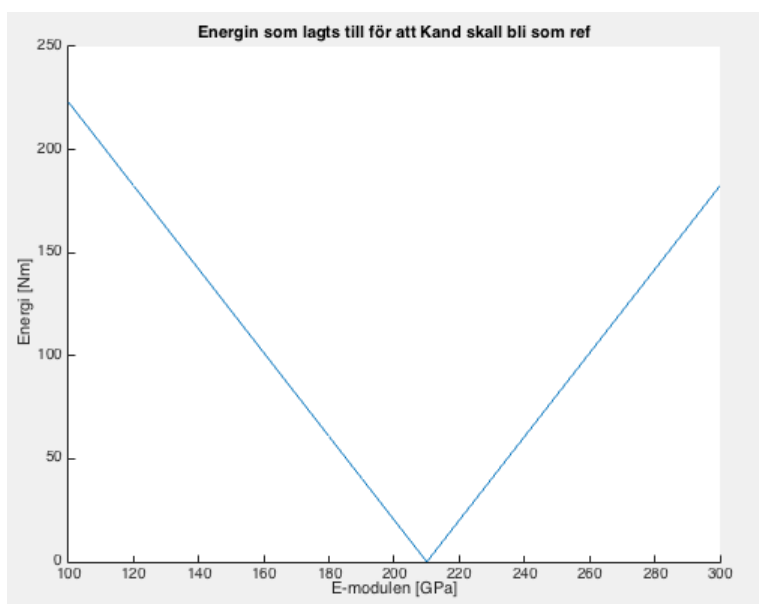
Figur 5.18 Energin som krävs för att uppnå randvillkoren och nodplaceringarna med varierande E-modul

Energiskillnaden mellan referensens energi och tillförda energin för att kandidaterna ska uppfylla samma randvillkor som referensen visas i figur 5.19. Detta är målfunktionen R_{bc} . När E-modulen är 210GPa krävs ingen energi. Detta visar även figur 5.19 då energin är noll vid E-modulen 210GPa. Ju längre ifrån referensens E-modul ju större blir energiskillnaden.



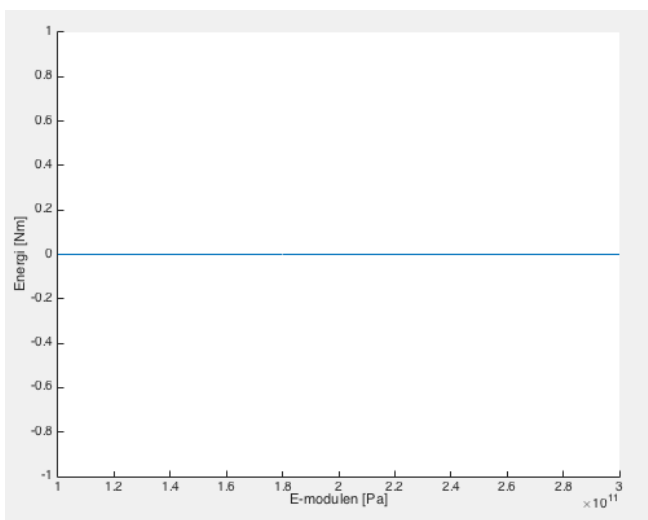
Figur 5.19. Målfunktionen R_{bc}

För att fastställa E-modulen krävs det att båda målfunktionerna är noll. Den andra målfunktionen R_u enligt ekvation 10 visas i figur 5.20. Även här är energiskillnaden noll vid E-modulen 210GPa. De två målfunktionerna är alltså noll vid E-modulen 210GPa. Då båda målfunktionerna är uppfyllda kan beräkningsmodellen fastställa ett värde för E-modulen. Att anmärka är att figuren ser likadan ut som föregående figur 5.19 vilket innebär att ingen ytterligare energi har egentligen lagts till för att förflytta noderna.



Figur 5.20, Målfunktionen R_u

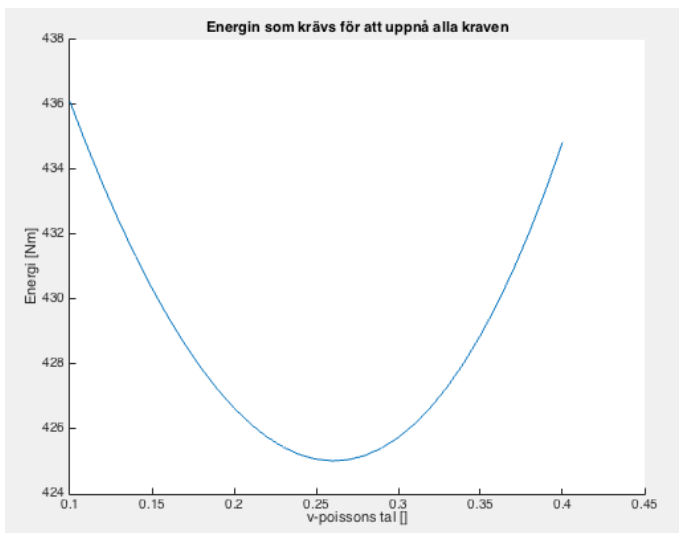
I figur 5.21 visas att skillnaden mellan målfunktionerna R_u och R_{bc} är noll. Detta påvisar att ingen ytterligare energi har lagt till för att förflytta noderna.



Figur 5.21, Energiskillnaden mellan R_u och R_{bc} .

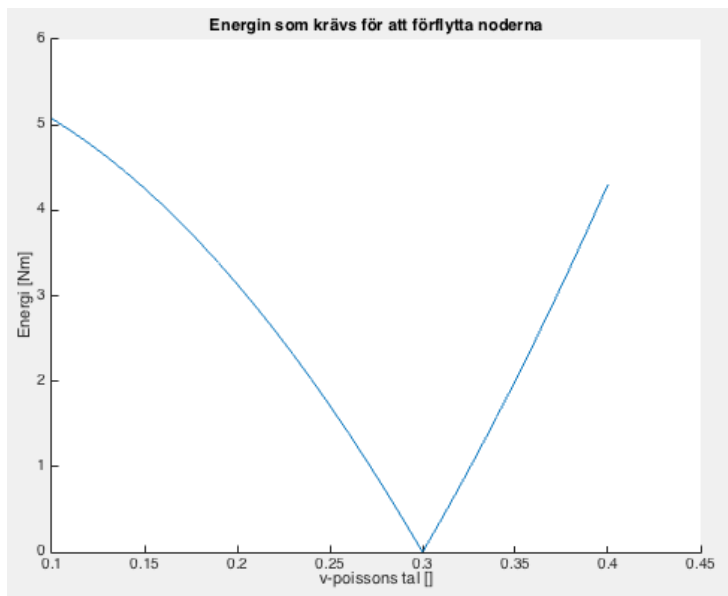
5.3.2 Jämförelse av energin då poissons tal varierar

Energien som krävs för kandidaterna med varierande poissons tal att uppfylla samma randvillkor och samma placeringar på nodförskjutningarna som referensfallet visas i figur 5.22. Vid poissons tal 0,3 krävs 426Nm för att uppfylla kraven. Figur 5.22 visar att ju närmare poissons tal är referensens värde på poissons tal ju mindre energi har tillförts.



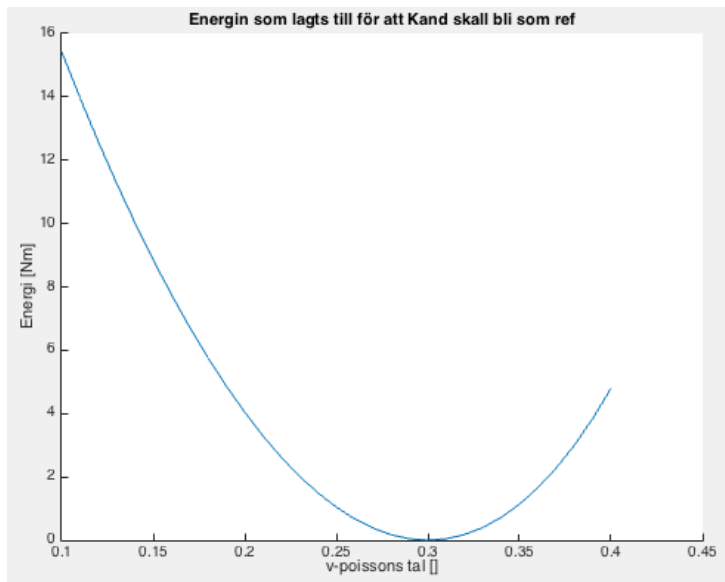
Figur 5.22, Energin som lagts till för kandidaterna med varierande poissons tal att uppnå samma randvillkor och nodplaceringar som referensen

I figur 5.23 visas den ena målfunktionen R_{bc} . Vid poissons tal 0,3 är energiskillnaden noll. Där är det ingen skillnad i energi för att förflytta noderna. Ju längre ifrån referensens poissons tal 0,3 ju större blir energiskillnaden. Skillnaden mäts i absolutbelopp.



Figur 5.23. Målfunktionen R_{bc}

För att fastställa poissons tal krävs även här att båda målfunktionerna är uppfyllda. Vilket innebär att energiskillnaden i båda målfunktionerna skall vara noll. Den andra målfunktionen R_u visas i figur 5.24. Vid poissons tal 0,3 är energiskillnaden noll. Ju längre ifrån referensen poissons tal är ju större blir energiskillnaden. Båda målfunktionerna R_{bc} och R_u är uppfyllda. Beräkningsmodellen har då hittat det rätta värdet på poissons tal, vilket är 0,3.

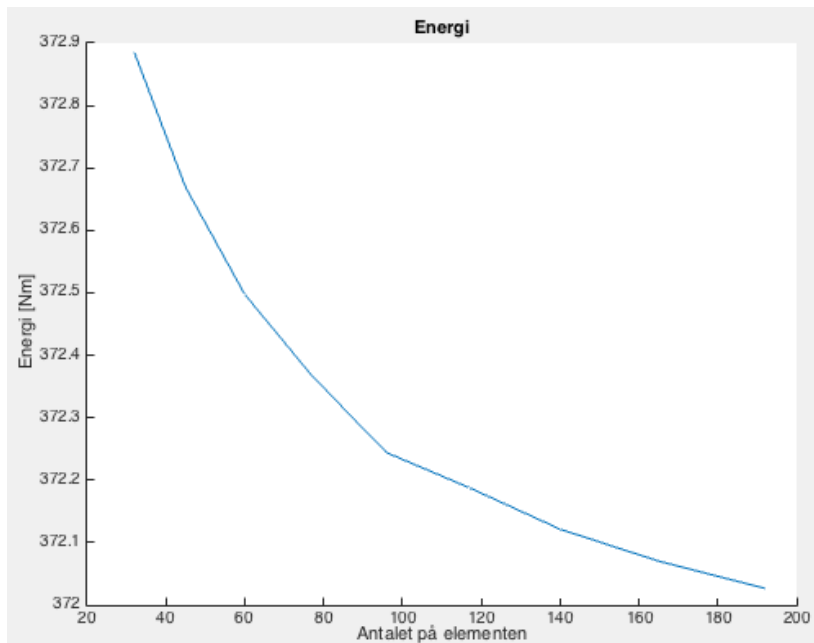


Figur 5.24. Målfunktionen R_u

Att beräkningsmodellen kunde hitta de rätta materialparametrarna som angivits i referensen visar att metodiken fungerar i detta fall. Referensen hade givna värden på E-modulen och poissons tal. Hos kandidaterna varierade materialparametrar. När tillförda energin mellan dessa jämfördes kunde beräkningsmodellen hitta när energiskilladen var noll. Då fanns rätt uppsättning av E-modulen och poissons tal.

5.4 Inverkan av elementstorlek

I jämförelsen med hur energin påverkas genom att variera nätfinheten genom interpolation fås resultatet att energin inte påverkas speciellt mycket. Detta visas i figur 5.25. Energin är ca 372Nm. Energin förändras bara i första decimalen från figuren har 50 element till att den har 200 element. Vid 100 till 200 element blir interpoleringsvärdena mer likvärdiga referensens nodplaceringar och mindre energi krävs. Ju färre element ju svårare blir det att få liknande värden som referensen.



Figur 5.25. Energiskillnad vid ökat antal element i interpoleringen

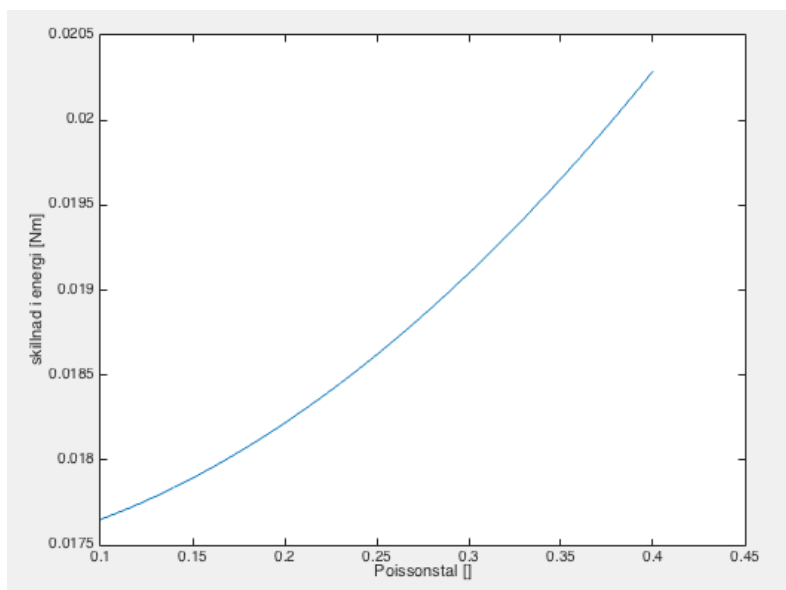
Då energin inte påverkas speciellt mycket beroende på element antalet vid interpolation har det ingen betydande effekt för beräkningsmodellen. Däremot är det viktigt att ha rätt antal element i figuren för att kunna få pålitliga beräkningar på spänningar och töjningar.

5.5 Experimentdata från två olika dragprov

Experimentdata från två dragprov tas in i beräkningsmodellen. För att se om metodiken fungerar då materialparametrarna inte är kända ifrån början. För att fastställa materialparametrarna beräknas målfunktionerna R_{bc} och R_u . Se ekvation 12 och 11. R_{bc} är skillnaden emellan den kända kraften som går in i provstaven och kraften som beräknas med varierande materialparametrar. Kraftskillnaden multipliceras med längden på provstaven för att få en energiskillnad. R_u är energiskillnaden mellan tillförda energin för experimentdatan och för beräkningsmodellen att uppfylla samma nodplaceringar som experimentet. Beräkningsmodellens energi varierar då materialparametrarna varierar.

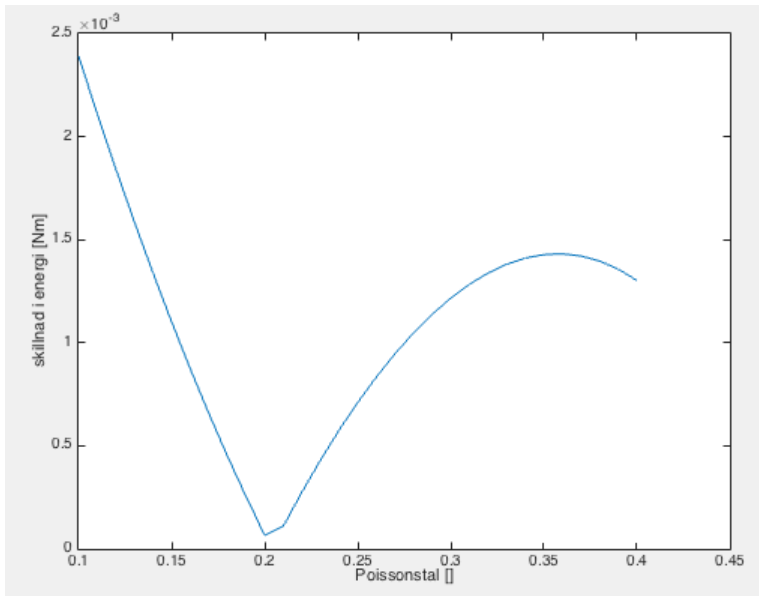
5.5.1 Dragprov 1

Båda målfunktionerna R_{bc} och R_u beräknades för E-modulen och poissons tal för dragprov 1. Målfunktionen R_{bc} ekvation 12 visas i figur 5.26. Figuren visar att ju högre poissons tal är ju mer energi krävs för att uppnå randvillkoren. Energiskillnaden mellan de olika värdena på poissons tal är väldigt liten. Den varierar cirka 0,003Nm mellan poissons tal 0,1-0,4. Den första målfunktionen kan inte bestämma ett värde på poissons tal. Därför behövs den andra målfunktionen R_u .



Figur 5.26 Målfunktionen R_{bc} för poissons tal

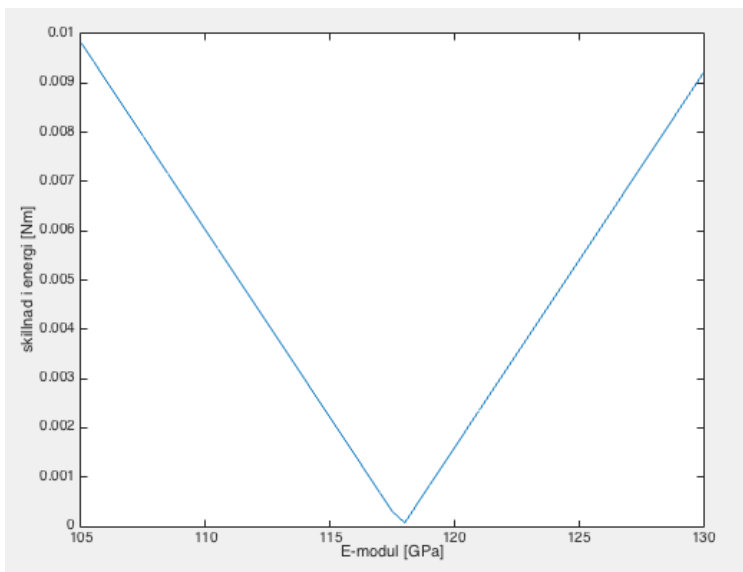
Målfunktionen R_u ekvation 11 visas i figur 5.27. Ett tydligt minimum bildas vid poissons tal 0,2. Beräkningsmodellen får alltså att poissons tal är 0,2 för dragprovet.



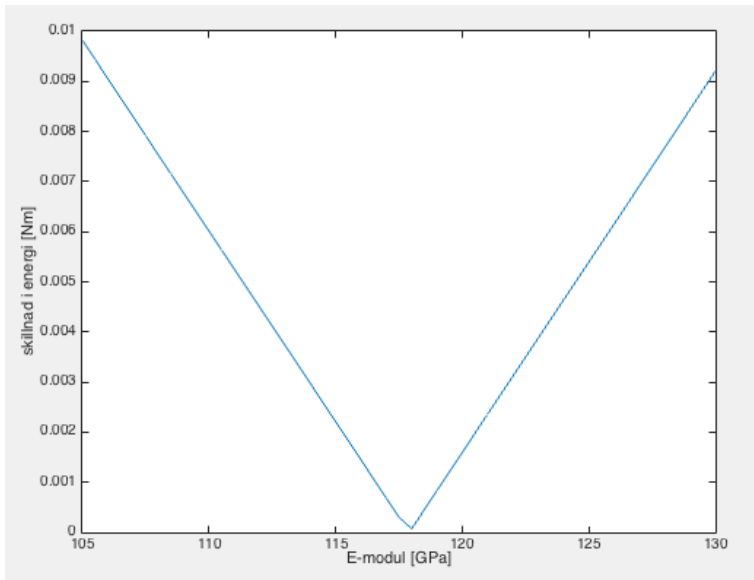
Figur 5.27. Målfunktionen R_u för poissons tal

Målfunktionerna ställs upp på samma sätt för att fastställa ett värde på E-modulen.

Målfunktionen R_{bc} visas i figur 5.28. Vid E-modulen 118GPa är målfunktionen noll. Precis som innan så skiljer sig inte målfunktionen R_{bc} och R_u ifrån varandra utan är identiska. Detta visas i figur 5.29 med målfunktionen R_u för varierande värden på E-modulen.

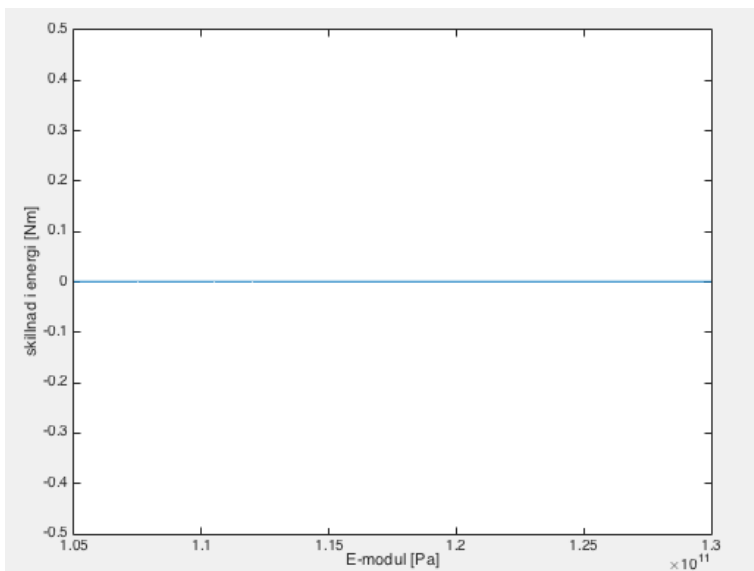


Figur 5.28. Målfunktionen R_{bc} med varierande E-modul



Figur 5.29 Målfunktionen R_u för varierade E-modul

Det krävs alltså ingen extra energi för nodförflyttningen vilket visas i figur 5.30 där skillnaden mellan R_u och R_{bc} visas. Kurvan är ett rakt streck vid energin noll för samtliga E-moduler.

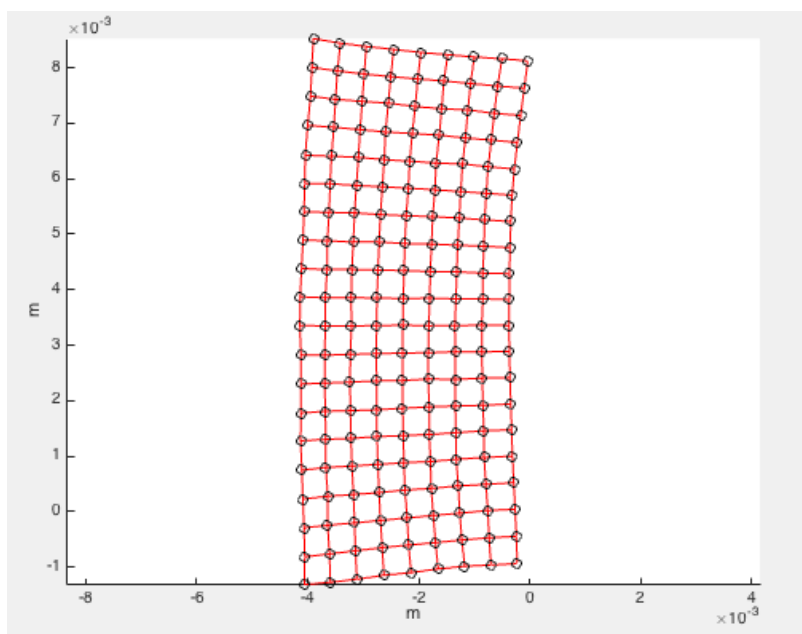


Figur 5.30. Skillnad R_u & R_{bc} för varierande E-modul

Provstaven var av titan som vanligtvis har poissons tal mellan 0,3-0,4 och E-modulen mellan 100-110GPa. Resultaten som givits här är därför inte troliga. Experimentdatan undersöktes och det visade sig att den inte var som den skulle. Maximala förskjutningen i x-led för experimentdatan är 19,9269mm och för y-led 22,7041mm. I ett dragprov ska skillnaden mellan

förskjutningar i x-och y-led vara betydligt större. Förskjutningen i x-led är också positiv. I ett dragprov ska förskjutningen i x-led bli negativ på grund av midjebildningen som sker. Roten till problemet visade sig vara att provstaven rörde sig under dragprovet. Den passade inte perfekt i utrustningen. Detta bidrog till att ett okänt moment påverkade provstaven. Då momentet är okänt kan beräkningsmodellen inte generera pålitliga resultat på materialparametrarna.

Provstavens förskjutningar plottades upp för att se hur momentet påverkade. Detta visas i figur 5.31. Förskjutningarna är förstörade 50 gånger. Provstavskroppen böjs inåt åt höger. Detta visar det okända momentet som uppstod. Kroppen påverkas alltså inte bara av en dragkraft utan också ett moment.

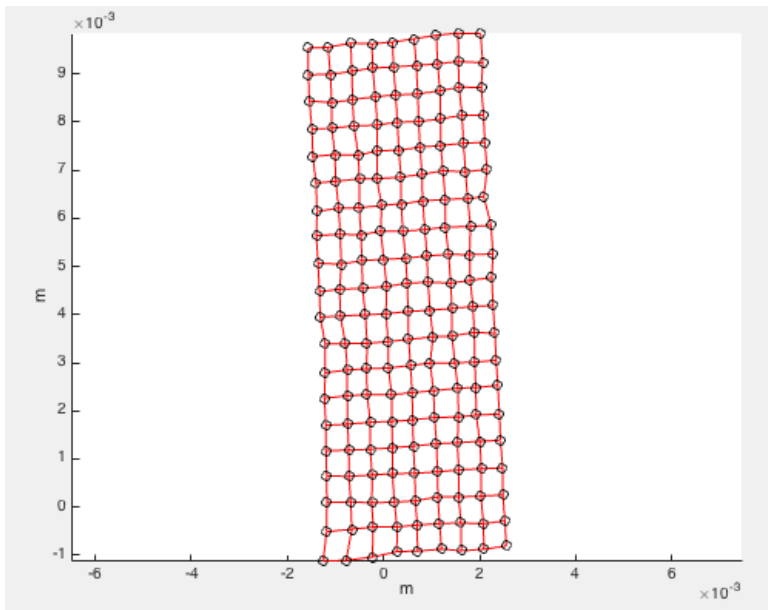


Figur 5.31. Förskjutningen förstörad 50 gånger visar att den utsätts för en dragkraft men också ett önskat moment.

5.5.2 Dragprov 2

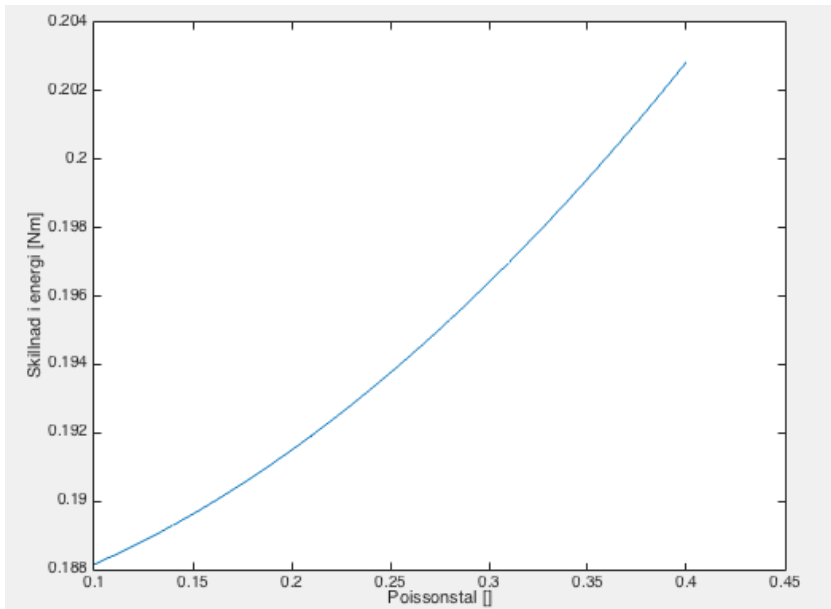
Då inget tillräckligt bra resultat gavs av första experimentdatan togs ett annat dragprov fram. Där passade provstaven perfekt i dragprovsutrustningen. Hur provstaven påverkades av förskjutningarna plottades upp återigen. Se figur 5.32. Förskjutningen är förstörad till 50 gånger så stor. Här ser man att förskjutningarna är betydligt bättre. Förskjutningarna från dragprovet visade också bättre värden. I x-led är maximala förskjutningen $-0,7705\text{mm}$ och i y-led är den

5,1328mm. Att förskjutningen i x-led är negativ och att skillnad mellan värdena i x-och y-led är större påvisar att dessa värden är bättre.

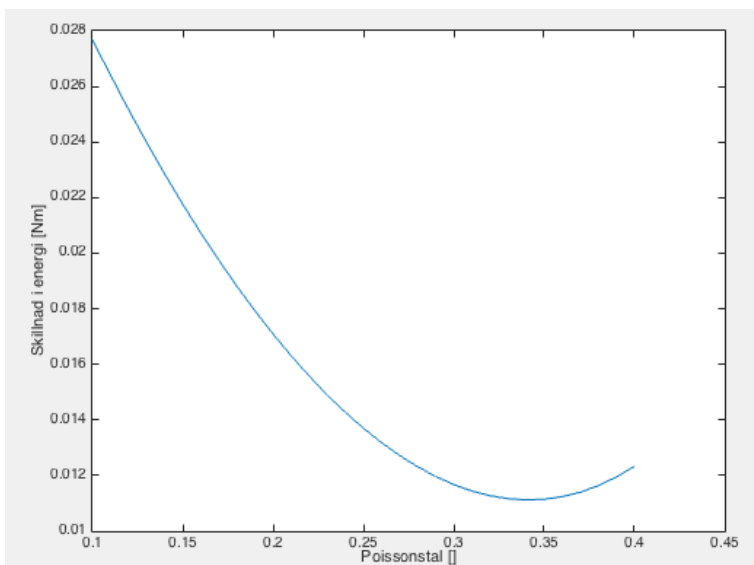


Figur 5.32. Förskjutningen för provstaven förstord 50 gånger så stor.

För att fastställa materialparametrarna beräknades de två målfunktionerna för E-modulen och poissons tal. Målfunktionen R_{bc} för poissons tal visas i figur 5.32. Ju större poissons tal är ju större blir energiskillnaden. Inget specifikt värde på poissons tal kan ges av målfunktionen. För att hitta poissons tal behövs även den andra målfunktionen R_u . Resultatet av målfunktionen R_u visas i figur 5.34. Ett minimum vid poissons tal 0,34 uppstår. Detta är ett väldigt rimligt värde då materialet även här är titan.

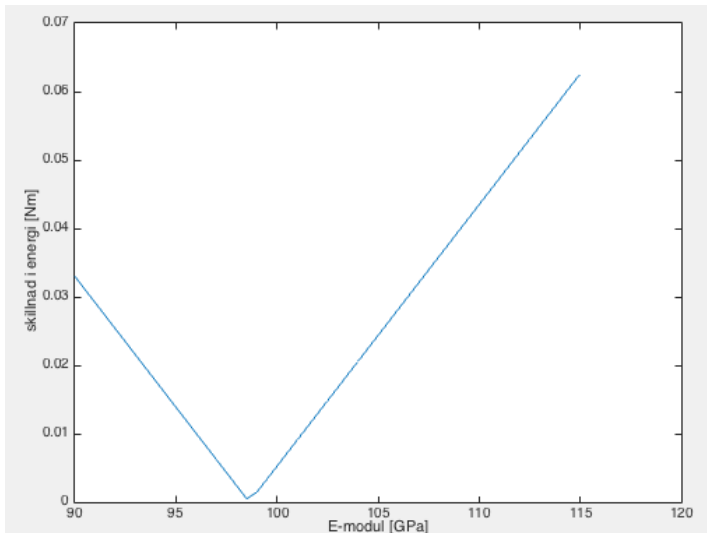


Figur 5.33 Målfunktionen R_{bc} för varierande poissons tal



Figur 5.34. Målfunktionen R_u med varierande poissons tal

På samma sätt beräknas målfunktionerna för E-modulen. Målfunktionen R_{bc} visas i figur 5.35. Ett tydligt minimum uppstår då E-modulen är 98,5GPa. Då tjockleken på denna provstav inte var helt bestämd kan E-modulen variera inom ett visst intervall. Provstavens tjocklek var mellan 2,5-3mm. Detta ger ett intervall mellan 90-105GPa hos E-modulen. Ju tjockare provstaven är desto mindre E-modul fås. I figuren är tjockleken 2,7mm. Målfunktionen R_u ger inget nytt resultat utan samma kurva och samma minimum värde ges för varierande E-modul.

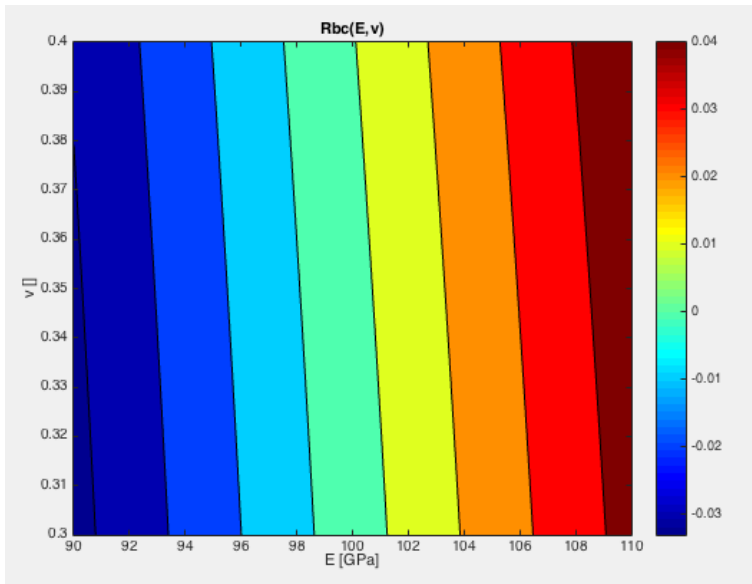


Figur 5.35. Målfunktionen R_{bc} med varierande E-modul

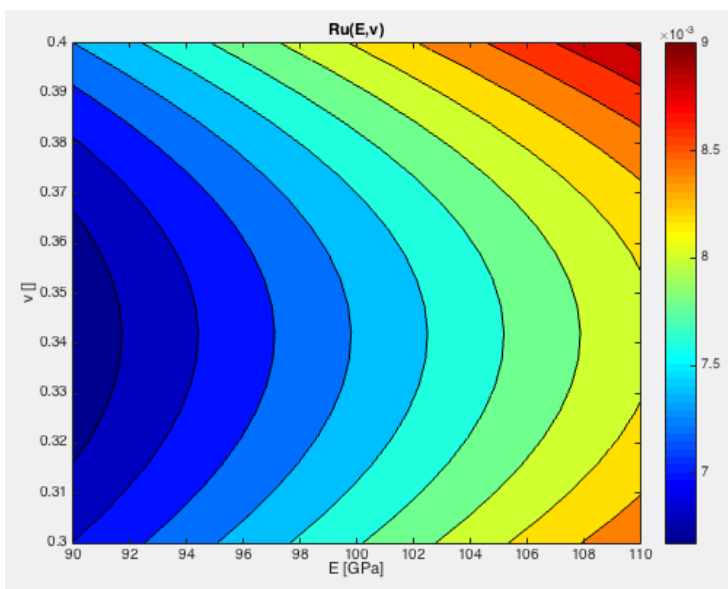
Kurvorna som tagits fram kan fastställa rimliga värden på materialparametrarna. Något som måste tas hänsyn till är dock att när en av materialparametrarna varierades var den andra konstant. När Poissons tal varierades var E-modulen konstant 70 GPa och när E-modulen varierades var Poissons tal konstant 0,3. Detta kan ha en liten inverkan på resultaten. För att hitta en lämplig uppsättning på parametrarna beräknades målfunktionerna med både varierad Poissons tal och E-modul. Målfunktionerna R_{bc} och R_u visas i figur 5.36 och 5.37. Båda målfunktionerna behöver vara noll för att uppfyllas. På x-axeln varierar E-modulen och på y-axeln varierar E-modulen.

I figur 5.36 visas målfunktionen R_{bc} . Färgerna i figuren motsvarar värden på energiskillnaden. Tillhörande värden visas i färgskalan till höger i figuren. Ett intervall på E-modulen kan fås då energiskillnaden är noll. Detta visas med den ljusgröna färgen i figuren. Intervallet för E-modulen ligger mellan 98,5-101 GPa. För att fastställa Poissons tal behövs målfunktionen R_u .

Målfunktionen visas i figur 5.37. Även här är E-modulen på x-axeln och Poissons tal på y-axeln. Kurvorna och färgerna i figuren motsvarar målfunktionen. Ett intervall på Poissons tal mellan 0,315-0,365 kan fås. Detta visas med den mörkblåa färgen längst till vänster i figuren. Där är energiskillnaden som minst.



Figur 5.36 Målfunktionen R_{bc} med varierande poissons tal och E -modul



Figur 5.37 Målfunktionen R_u med varierande poissons tal och E -modul

Då både E -modulen och poissons tal varierar kan rimliga intervall fås. Detta påvisar att metodiken för att fastställa materialparametrar fungerar även då indatan kommer från ett verkligt experiment.

6. SLUTSATS

Arbetet har handlat om att skapa en metodik för att fastställa materialparametrar. Detta genom att skapa en beräkningsmodell i Matlab med hjälp av CalFem. Beräkningsmodellen skulle uppvisa samma beteende som ett verkligt dragprov. De okända materialparametrarna som skulle fastställas var E-modulen och poissons tal.

Metoden utvärderades slutligen med hjälp av data från två olika experiment. Förskjutningar och dragkraften i experimenten var kända. De resultat som framkom ur dragprov 1 var att provkroppen hade en E-modul på 118GPa och att poissons tal var 0,2. Den provstav som utsattes för provningen var av titan. Vilket vanligtvis har en E-modulen mellan 100-110GPa och poissonstal mellan 0,3-0,4. Detta betyder att de resultat som framkom ur experiment 1 inte stämmer överens med verkligheten. Felkällan som skapade de felaktiga resultaten var att provstaven rörde sig i utrustningen under dragprovet. Detta gjorde att provstaven påverkades inte bara av en dragkraft utan även ett moment. Detta bidrog till att förskjutningarna i provet blev felaktiga.

I dragprov 2 var provstaven bättre dimensionerad till dragprovutrustningen. Då provstaven inte utsattes för ett lika stort eller inget okänt moment blev resultaten trovärdiga. För provstaven av titan kunde E-modulen fastställas inom ett intervall mellan 98,5-101GPa och poissons tal mellan 0,315-0,365.

Genom att plotta upp båda dragproven med överdriven deformation kunde det tydligt ses hur dragproven betedde sig. I första experimentet kunde ett moment tydligt ses påverka provstaven. I det andra experimentet belastades provstaven med drag och inget större moment verkade uppstå. Detta besvarar en av frågeställningarna från inledningen; om beräkningsmodellen skulle kunna bete sig som ett verkligt dragprov. De andra frågeställningen handlade om trovärdiga värden på materialparametrarna kunde fås. I andra experimentet gavs trovärdiga intervall för E-modulen och poissons tal. Vilket visar att metodiken som användes fungerar och att det är möjligt att fastställa materialparametrar utifrån en beräkningsmodell.

För att undersöka vidare hur bra metodiken är behövs fler experiment testas i beräkningsmodellen. För att skapa en så trovärdig metod och beräkningsmodell som möjligt behövs fler tester med olika material genomföras.

En felkälla som kan påverka trovärdigheten i de resultat som tagits fram kan vara en okänd parameter. Exempelvis att ett moment påverkat provstaven. En annan felkälla kan vara att kraften som går in i provstaven påverkar hela kroppen och inte bara det område som analyseras i beräkningsmodellen. Kraften ingår i energiekvationerna och införs då fel kraft påverkas slutresultatet.

6.1 Fortsatt arbete

Förslag på fortsatt arbete för projektet är att testa beräkningsmodellen med data från flera dragprov med andra material. För att undersöka om beräkningsmodellen fungerar på fler experiment som kan antas ha en linjärelastisk materialmodell. Belastningsfallen skulle även kunna få svårare geometrier och utsättas för andra randvillkor. Metodiken för att fastställa materialparametrar skulle kunna undersökas om det kan tillämpas på flera experiment. Då skulle även flera materialparametrar kunna fastställas.

REFERENSLISTA

- [1] T. Dahlberg, *Teknisk hållfasthetslära*, 3:e upplagan, 2001, ISBN-9789144019208
- [2] S. Sunnersjö, *Fem i praktiken – En introduktion till finita elementmetodens praktiska tillämpning*, 2:a upplagan, 1999, ISBN-9789175485416
- [3] G. R Liu, S. S Quek, *Finite Element Method: A practical Course*, 2003, Butterworth-Heinemann, ISBN-9780080983561
- [4] <http://www.3ds.com/products-services/simulia/products/abaqus>, 2015-03-10
- [5] <http://www.ansys.com>, 2015-03-10
- [6] R Grahn, P-Å Jansson, *Mekanik*, Studentlitteratur, 3:e upplagan, 1997, ISBN-9789144085784
- [7] <http://www.math.chalmers.se/Math/Grundutb/CTH/tma262/0304/numerikZ1.pdf>, 2015-04-16
- [8] <http://academic.eb.com/EBchecked/topic/291584/interpolation>, 2015-04-16
- [9] P. Jönsson, *MATLAB beräkningar inom teknik och naturvetenskap*, 3:e upplagan, 2012, ISBN-9789144069265
- [10] http://www.solid.lth.se/fileadmin/hallfasthetslara/utbildning/kurser/FHL064_FEM/calfem34.pdf, 2015-02-02
- [11] <http://dixon.hh.se/bertil/Kurser/Common/FEMgk/Notes/kompendiumA4.pdf>, 2015-04-14
- [12] <http://www.correlatedsolutions.com/digital-image-correlation>, 2015-05-12

BILAGOR

Bilaga 1

```

%% första statistiskt belastningsfall
clc
clf
clear all
close all

% figurens egenskaper: E-modul, poissons tal, tjocklek
% ptype anger om figuren har plan töjning/spännings tillstånd
ptype=1;
E=210e9;
v=0.3;
t=1;

% Skapar materialmatrisen
D=hooke(ptype,E,v);
ep=[ptype t];

% Ändring av storlek på figuren
% ex1 & ex2 är första punkten ner till vänster i elementen
% ey1 & ey2 är sista punkten upp till höger i elementen
ex1=[0 5 5 0];
ey1=[0 0 5 5];
ex2=[0 5 5 0];
ey2=[5 5 10 10];

% skapar elementstyhetsmatriser
Ke1=plane(ex1,ey1,ep,D);
Ke2=plane(ex2,ey2,ep,D);

% anger vilka frihetsgrader som hör till varje element
edof=[1 1 2 3 4 7 8 5 6;2 5 6 7 8 11 12 9 10];

f=zeros(12,1);
K=zeros(12);

% lägger in de lokala elementstyhetsmatriserna i den globala
% styvhetsmatrisen
K=assem(edof(1,:),K,Ke1)
K=assem(edof(2,:),K,Ke2)

% anger randvillkor
bc=[1 0;2 0;3 0;4 0;9 0;10 2;11 0;12 2];

% löser nodförskjutningarna & de ekvivalenta nodkrafterna
[u,f]=solveq(K,f,bc)

% plottar figuren
figure(1)
plot(ex1,ey1,'k')
axis([-5 10 -5 15])
hold on
plot(ex2,ey2,'k')
hold on

```

```

plot([0 0],[0 10],'k')

%plotar förskjutningen
plot([0 5],[0 0],'r') %plottar vågräta linjen längst ner
plot([0+u(5) 5+u(7)],[5+u(6) 5+u(8)'],'r') %plottar vågräta linjen i mitten
plot([0+u(9) 5+u(11)],[10+u(10) 10+u(12)'],'r') %plottar vågräta linjen längst
%upp
plot([0 0+u(5)],[0 5+u(6)'],'r') %plottar lodräta linjen mellan punkt 1 o 4
plot([5 5+u(7)],[0 5+u(8)'],'r') %plottar lodräta linjen mellan punkt 2 o 3
plot([0+u(5) 0+u(9)],[5+u(6) 10+u(10)'],'r') %plottar lodräta linjen mellan
%punkt 4 o 6
plot([5+u(7) 5+u(11)],[5+u(8) 10+u(12)'],'r') %plottar lodräta linjen mellan
%punkt 3 o 5

% nodförskjutningarna för varje element
ed1=u(1:8)';
ed2=u(5:12)';

% beräknar spänning & töjning i x,y,z-riktning för varje element
[es1,et1]=planqs(ex1,ey1,ep,D,ed1)
[es2,et2]=planqs(ex2,ey2,ep,D,ed2)

```

Bilaga 2

```

%% Referensen

clc
clf
clear all
close all

% figurens egenskaper: E-modul, poissons tal, tjocklek
% ptype anger om figuren har plan tøjning/spännings tillstånd
ptype=1;
E=210e9;
v=0.3;
t=0.002;

% Ändring av storlek på figuren
% p1 är första punkten ner till vänster
% p2 är sista punkten upp till höger
p1=[0 0];
p2=[0.004 0.0080];

% Hur fint figuren skall meshas
% nelx är antal noder i x-led
% nely är antal noder i y-led
nelx=8;
nely=19;

% ndofs - antal frihetsgrader i varje nod, nen - antalet noder i varje element
ndofs=2;
nen=4;

% meshar upp figuren
% B1 anger de frihetsgrader som existerar på nedre sidan av figuren,
% B2 anger frihetsgraderna för högra sidan, B3 - ovansidan, B4 - vänstra sidan
[Edof, Ex, Ey, B1, B2, B3, B4, xq, yq]=quad_mesh(p1, p2, nelx, nely, ndofs);

% Anger hur många frihets grader figuren har
fg=B3(1, end);

% Randvillkoren skapas som säger att figuren skall vara
% fast i nedre kanten medans ovansidan dras uppåt 2 mm
[bc]=randvillkor(B1, B3);

% Skapar materialmatrisen
D=hooke(ptype, E, v);

% Skapar matriser som behövs i planqe-funktionen
% K - styvhetsmatris, f - ekvivalenta nodkraftsmatris
ep=[ptype t];
K=zeros(fg, fg);
f=zeros(fg, 1);

% skapar först elementstyvhetsmatriser och därefter placeras

```

```

% i den globala styvhetsmatrisen
for i=1:length(Ex(:,1))
    Ke=plane(Ex(i,:),Ey(i,:),ep,D);
    K=assem(Edof(i,:),K,Ke);
end

% Löser nodförskjutningarna - u och ekvivalenta nodkrafterna - f
[u,f]=solveq(K,f,bc)

% Element förskjutningsvektor skapas
[ed]=extract(Edof,u);

% plottar upp figuren före och efter deformationen
% plotpar beskriver hur du vill att figurer skall ritas upp
% elnum anger elementnummer
plotpar=[1 1 1];
elnum=Edof(:,1);

figure (1)
eldraw2(Ex,Ey,plotpar,elnum);
[sfac,x,y]=eldisp2(Ex,Ey,ed,[1 4 1],1);
hold on
title('Förskjutningar')
xlabel('mm')
ylabel('mm')

% placerar nodförskjutningar i x-och y-led i varsin matris
Xq=xq(:);
Yq=yq(:);

for i=1:size(Xq)
    uy(i)=u(2*i);
    ux(i)=u(2*i-1);
end

% placerar nodförskjutningar i x-och y-led i varsin elementmatris
% och beräknar resultanten av nodförskjutningar i x-och y-led för varje nod
uX=ux';
uY=uy';

for j=1:length(ed(:,1))
    for i=1:4
        edx(j,i)=ed(j,2*i-1);
        edy(j,i)=ed(j,2*i);
        ED(j,i)=abs(sqrt(edy(j,i).^2+edx(j,i).^2));
    end
end

% summerar nodförskjutningen i varje element
for i=1:length(ED(:,1))
    M(i)=sum(ED(i,:));
end

% plottar summan av nodförskjutningarna för varje element med en
% motsvarande färg och riktningspilar för hur noderna förskjuts

```



```

figure(6)
eldraw2(Ex,Ey,plotpar);
hold on
fill(Ex',Ey',M)
colorbar
colormap('jet')
hold on
quiver(Xq,Yq,uX,uY,1,'c');

% beräknar spänningar och töjningar i elementen
s=zeros(Edof(end,1),3);
t=zeros(Edof(end,1),3);

for i=1:length(Ex(:,1))
    [es,et]=planqs(Ex(i,:),Ey(i,:),ep,D,ed(i,:));
    s(i,:)=es;
    t(i,:)=et;
end

Sx=s(:,1)';
Tx=abs(t(:,1)');

Sy=s(:,2)';
Ty=abs(t(:,2)');

X=x';
Y=y';

% plottar spänningen i x-led
figure(2)
title('Spänningar i x-led')
xlabel('mm')
ylabel('mm')
eldraw2(X,Y,plotpar)
hold on
fill(x,y,Sx)
colormap('jet')
colorbar

% plottar töjningar i x-led
figure(3)
title('Töjningar i x-led')
xlabel('mm')
ylabel('mm')
eldraw2(X,Y,plotpar)
hold on
fill(x,y,Tx)
colormap('jet')
colorbar

% plottar spänningen i y-led
figure(4)
title('Spänningar i y-led')
xlabel('mm')
ylabel('mm')

```

```
eldraw2(X,Y,plotpar)
hold on
fill(x,y,Sy)
colormap('jet')
colorbar

% plottar töjningar i y-led
figure(5)
title('Töjningar i y-led')
xlabel('mm')
ylabel('mm')
eldraw2(X,Y,plotpar)
hold on
fill(x,y,Ty)
colormap('jet')
colorbar

xqr=xq;
yqr=yq;

% beräknar referensens energi och sparar data
WR=(u'*K*u)/2
save('DataR.mat','u','Ex','Ey','ep','ptype','Edof','fg','WR','bc','xqr','yqr',
'K','B3','B1')
save('Data2.mat','B1','B3','ptype','fg','Ex','Ey','ep','Edof')
```

Bilaga 3

```
function [bc]=randvillkor(B1,B3)
% Randvillkor som bildar matris bc som skall in i solveq
M=length(B1(:));
N=length(B3(:));
k=M+N;
n=length(B3);

bc=zeros(k,2);
bc(1:n,1)=B3(1:end,2);
bc(1:n,2)=2;
bc(n+1:N,1)=B3(1:end,1);
bc(N+1:end,1)=B1(:);
end
```

Bilaga 4

Antal element	Max nodförskjutning [mm]	Max töjning x-led []	Max töjning y-led []	Max spänning x-led [MPa]	Max spänning y-led [MPa]
2	1	-0,50%	2,50%	568,4	5420,5
4	1	-0,49%	2,50%	596,4	5428,9
8	1,5	-0,40%	2,56%	767,7	5355,8
18	1,7	-0,31%	2,59%	902,9	5470,5
27	1,8	-0,24%	2,58%	1040,9	5456,1
36	1,7	-0,32%	2,57%	903,5	5505,5
42	1,7	-0,29%	2,58%	950,5	5532,1
48	1,8	-0,27%	2,58%	993,7	5558,5
54	1,8	-0,25%	2,58%	1032,7	5583,7
64	1,8	-0,26%	2,58%	996,2	5614,1
72	1,8	-0,24%	2,58%	1034,5	5647,6
84	1,9	-0,17%	2,57%	1171,6	5686,2
96	1,8	-0,19%	2,60%	1125,7	5736,8
100	1,8	-0,23%	2,61%	1069,2	5731,7
112	1,9	-0,17%	2,62%	1170,8	5786,5
126	1,9	-0,17%	2,64%	1173,5	5826,7
133	1,9	-0,13%	2,62%	1252	5823,3
140	1,9	-0,12%	2,63%	1263,2	5836,5
144	1,9	-0,14%	2,65%	1235,7	5866,5
152	1,9	-0,13%	2,66%	1248,2	5853,2
154	1,9	-0,11%	2,64%	1282,7	5860,3
160	1,9	-0,12%	2,66%	1259,6	5898,7
168	1,9	-0,12%	2,67%	1269,9	5913,2
171	1,9	-0,13%	2,68%	1250,1	5935,8
180	1,9	-0,12%	2,69%	1261,3	5953,5
189	1,9	-0,12%	2,69%	1271,6	5970
192	1,9	-0,10%	2,68%	1296,2	5951,3
200	1,9	-0,12%	2,71%	1259	6002,1

Bilaga 5

```

%% Energiekvation poissons tal, kandidater

clc
clear all
close all
clf

% laddar data från referensen
load('DataR.mat')

% anger intervall på E-modulen och poissons tal
E=[150e9,210e9,250e9];
v=[0.1:0.01:0.4];

% beräknar energiekvationerna för varierande poissons tal
w=zeros(1,length(E));

for k=1:length(E)
    for i=1:length(v)
        D=hooke(ptype,E(k),v(i));
        K=zeros(fg,fg);
        for j=1:length(Ex(:,1))
            Ke=plane(Ex(j,:),Ey(j,:),ep,D);
            K=assem(Edof(j,:),K,Ke);
        end
        f=zeros(fg,1);
        [u1,f]=solveq(K,f,bc);
        % energiekvationer, w - energin för att ha samma randvillkor som
        % referensen, wk - energin för att ha samma randvillkor &
        % nodplacering som referensen
        wk(i)=(u1'*K*u1)/2;
        w(i)=(u'*K*u)/2;
    end
    % målfunktioner
    % Rnod - energiskillnad mellan w & wk, Ru - energiskillnad mellan w och
    % referensens energi, Rbc - energiskillnad mellan wk & referensens
    Rnod=(w-wk);
    Ru=(w-WR);
    Rbc=abs(wk-WR);

    % plottar energiekvationerna & målfunktionerna
    figure(1)
    D=['b','r','g'];
    subplot(1,3,k)
    title('Energien som krävs för att uppnå alla kraven')
    xlabel('v-poissons tal')
    hold on
    ylabel('Energi W')
    plot(v,w,D(k))
    hold on

    figure(2)
    subplot(1,3,k)

```

```
title('Energien som krävs för att förflytta noderna')
xlabel('v-poissons tal')
hold on
ylabel('Energien W')
plot(v,Rbc,D(k))
hold on

figure(3)
subplot(1,3,k)
title('Energien som lagts till för att Kand skall bli som ref')
xlabel('v-poissons tal')
hold on
ylabel('Energien W')
plot(v,Ru,D(k))
hold on
end
```

Bilaga 6

```

%% Energi ekvation E-modul, kandidater

clc
clear all
close all
clf

% laddar data från referensen
load('DataR.mat')
v=0.3;

% anger intervall på E-modulen och poissons tal
E=[100e9:10e9:300e9];
w=zeros(1,length(E));

% beräknar energi ekvationerna för varierande E-modul
for i=1:length(E)
    D=hooke(ptype,E(i),v);
    K=zeros(fg,fg);
    for j=1:length(Ex(:,1))
        Ke=plane(Ex(j,:),Ey(j,:),ep,D);
        K=assem(Edof(j,:),K,Ke);
    end
    f=zeros(fg,1);
    [u1,f]=solveq(K,f,bc);
    % energi ekvationer, w - energin för att ha samma randvillkor som
    % referensen, wk - energin för att ha samma randvillkor &
    % nodplacering som referensen
    wk(i)=(u1'*K*u1)/2;
    w(i)=(u'*K*u)/2;
end

% målfunktioner
% Rnod - energiskillnad mellan w & wk, Ru - energiskillnad mellan w och
% referensens energi, Rbc - energiskillnad mellan wk & referensens
Rnod=(w-wk);
Ru=abs(w-WR);
Rbc=abs(wk-WR);

% plottar energi ekvationerna & målfunktionerna
figure(1)
title('Energien som krävs för att uppnå alla kraven')
xlabel('E-modul')
hold on
ylabel('Energien W')
plot(E,w)

figure(2)
title('Energien som krävs för att förflytta noderna')
xlabel('E-modul')
hold on
ylabel('Energien W')
plot(E,Rbc)

```

```
figure(3)
title('Energien som lagts till för att Kand skall bli som ref')
xlabel('E-modulen')
hold on
ylabel('Energien W')
plot(E,Ru)
hold on
```


Bilaga 7

```

%% Energi skillnad vid interpolering

clc
clf
clear all
close all

% laddar data från referensen
load('DataR.mat')
ur=u;

% figurens egenskaper: E-modul, poissons tal, tjocklek
% ptype anger om figuren ska ha plan töjning/spännings tillstånd
ptype=1;
E=210e9;
v=0.3;
t=0.002;

% Ändring av storlek på figuren
% p1 är första punkten ner till vänster
% p2 är sista punkten upp till höger
p1=[0 0];
p2=[0.035 0.080];

% Hur fint figuren skall meshas
% nelx är antal noder i x-led
% nely är antal noder i y-led
nelx=[4:1:12]
nely=[8:1:17]

for j=1:length(nelx)
    % ndofs - antal frihetsgrader i varje nod, nen - antalet noder i varje
    element
        ndofs=2;
        nen=4;

        % meshar upp figuren
        % B1 anger de frihetsgrader som existerar på botten sidan av figuren,
        % B2 anger för högra sidan, B3 anger ovansidan
        % B4 anger den vänstra sidan.
        [Edof,Ex,Ey,B1,B2,B3,B4,xq,yq]=quad_mesh(p1,p2,nelx(j),nely(j),ndofs);

        %Anger hur många frihets grader figuren har
        fg=B3(1,end);

        % Randvillkoren skapas som säger att figuren skall vara
        % fast i botten läget medans ovansidan dras uppåt 2 mm
        [bc]=randvillkor(B1,B3);

        % Skapar material matrisen
        D=hooke(ptype,E,v);
        ep=[ptype t];

```

```

K=zeros(fg,fg);
f=zeros(fg,1);

% skapar först element styvhets matrisen och där efter den globala
% styvhets matrisen
for i=1:length(Ex(:,1))
    Ke=plane(Ex(i,:),Ey(i,:),ep,D);
    K=assem(Edof(i,:),K,Ke);
end

% Får fram hur varje frihetsgrad ändrar riktning u och kraft vektorn f.
[u,f]=solveq(K,f,bc)

Xq=xq(:);
Yq=yq(:);

% placerar referensens förskjutningar i x-och y-led
ux=ur(1:2:end);
uy=ur(2:2:end);

% interpolerar nodförskjutningar för beräkningsmodellen
% utifrån referensen
vx=griddata(xqr(:),yqr(:),ux,Xq,Yq)
vy=griddata(xqr(:),yqr(:),uy,Xq,Yq)

% bildar en deformationsmatris med de interpolerade värdena
U=zeros(fg,1);
U(1:2:end)=vx
U(2:2:end)=vy

% Energiekvationer,
% wk1 - energin för att ha samma randvillkor som referens
% wk2 - energin för att ha samma randvillkor & placering som referens
wk1(j)=(u'*K*u)/2;
Wk2(j)=(U'*K*U)/2;

% antal element
h(j)=nelx(j)*nely(j);
end

% plottar energiekvationen wk2 beroende på antal element
figure(1)
title('Energi')
xlabel('Antalet på elementen')
hold on
ylabel('Energi W')
plot(h,Wk2)

```

Bilaga 8

```

%% interpolering för dragprov 1

clc
clf
clear all
close all

% laddar värden från dragprov 1 och från referensen
load('xyz0.txt')
load('xyz49.txt')
load('DataR.mat')

% anger facettstorlek, steglängd, skalfaktor [px -> mm]
% c - avståndet från centrum på facetten till kanten
fs=19;
sl=11;
sfak=(3.56/197.56);
c=sfak*9.5;

% beräknar facettförskjutningarna och placerar dessa i x-och y-led
udata=(xyz49-xyz0)./1000;
ux=udata(:,1);
uy=udata(:,2);

% placerar x-och y-koordinater i hörnen för facetterna
xq=(xyz0(:,1)./1000)-((min(xyz0(:,1))-c)/1000);
yq=(xyz0(:,2)./1000)-((min(xyz0(:,2))-c)/1000);

% interpolerar nodförskjutningar för beräkningsmodellen
% utifrån facettförskjutningarna
vx=griddata(xq,yq,ux,xqr(:),yqr(:),'V4');
vy=griddata(xq,yq,uy,xqr(:),yqr(:),'V4');

% bildar en deformationsmatris med de interpolerade värdena
U=zeros(fg,1);
U(1:2:end)=vx;
U(2:2:end)=vy;

% beräknar energin för de interpolerade värdena
Wm=(U'*K*U)/2;

% placerar facetternas koordinater i x-och y-led
% och plottar de odeformerade facetterna
for z=1:length(xq)
    XQ1(z)=xq(z)-(c/1000);
    XQ2(z)=xq(z)+(c/1000);
end

for n=1:length(xq)
    Xp(n,1)=XQ1(n);
    Xp(n,4)=XQ1(n);
    Xp(n,2)=XQ2(n);

```

```
    Xp(n,3)=XQ2(n);
end

for z=1:length(yq)
    YQ1(z)=yq(z)-(c/1000);
    YQ2(z)=yq(z)+(c/1000);
end

for n=1:length(yq)
    Yp(n,1)=YQ1(n);
    Yp(n,2)=YQ1(n);
    Yp(n,3)=YQ2(n);
    Yp(n,4)=YQ2(n);
end

figure(1)
eldraw2(Xp,Yp,[1 4 1]);

% plottar de deformerade facetterna
% ed - beräknar en deformationen för varje element
[ed]=extract(Edof,U);
figure(2)
[sfac]=eldisp2(Ex,Ey,ed,[1 4 1],50);

% sparar data för dragprov 1
save('case.mat','U','vx','vy','ux','uy')
```

Bilaga 9

```

%% Energiekvation E-modulen, dragprov 1

clc
clear all
close all
clf

% laddar data från referensen & interpoleringen
load('DataR2.mat')
load('case.mat')
format long

% anger intervall på E-modulen och poissons tal
E=[70e9:5e9:150e9];
v=[0.3];

w=zeros(1,length(E));

% anger randvillkor
[BC]=randvillkorEXP(B1,B3)

% Kraften från dragprovet
Fexp=3.554688e3;
Fexp2=3.35566e3;

% beräknar energiekvationerna för varierande E-modul
for k=1:length(E)
    for i=1:length(v)
        D=hooke(pstype,E(k),v(i));
        K=zeros(fg,fg);
        for j=1:length(Ex(:,1))
            Ke=plange(Ex(j,:),Ey(j,:),ep,D);
            K=assem(Edof(j,:),K,Ke);
        end
        f=zeros(fg,1);
        [u,f]=solveq(K,f,BC);
        % summerar de ekvivalenta nodkrafterna i y-led, raden längst upp
        Fnod(k)=sum(f(344:2:end));
        % energiekvationer, wbc - energin för att ha samma randvillkor som
        % referensen, wu - energin för att ha samma randvillkor &
        % nodplacering som referensen
        wbc(k)=(u'*K*u)/2;
        wu(k)=(U'*K*U)/2;
        % beräknar målfunktionen Rbc, energiskillnad mellan känd kraft &
        % kraften med varierande E-modul, multiplicerat med längden på
        % provstaven
        Rbc(k)=abs(Fnod(k)-Fexp2)*(U(354)-U(10));
    end
end

% beräknar målfunktionen Ru, energiskillnad mellan wu & wbc
Ru=abs(wu-wbc);

```

```
% plottar energiekvationerna & målfunktionerna
figure(1)
plot(E,Ru)
axis([70e9 150e9 -0.5 0.5])
xlabel('E-modul')
ylabel('skillnad i energi')

figure(2)
plot(E,Rbc)
xlabel('E-modul')
ylabel('skillnad i energi')

figure(3)
plot(E,Fnod)
```

Bilaga 10

```

%% Energi ekvation Poissons tal, dragprov 1
clc
clear all
close all
clf

% laddar data från referensen & interpoleringen
load('DataR2.mat')
load('case.mat')
format long

% anger intervall på E-modulen och poissons tal
E=[70e9,100e9,150e9];
v=[0.1:0.01:0.4];

w=zeros(1,length(E));

% anger randvillkor
[BC]=randvillkorEXP(B1,B3)

% Kraften från dragprovet
Fexp=3.554688e3;
Fexp2=3.35566e3

% beräknar energi ekvationerna för varierande E-modul
for k=1:length(E)
    for i=1:length(v)
        D=hooke(ptype,E(k),v(i));
        K=zeros(fg,fg);
        for j=1:length(Ex(:,1))
            Ke=plange(Ex(j,:),Ey(j,:),ep,D);
            K=assem(Edof(j,:),K,Ke);
        end
        f=zeros(fg,1);
        [u,f]=solveq(K,f,BC);
        % summerar de ekvivalenta nodkrafterna i y-led, raden längst upp
        Fnod=sum(abs(f(344:2:end)));
        % beräknar målfunktionen Rbc, energiskillnad mellan känd kraft &
        % kraften med varierande poissons tal, multiplicerat med längden på
        % provstaven
        Rbc(i)=(Fnod-Fexp)*(U(354)-U(10));
        % energi ekvationer, wbc - energin för att ha samma randvillkor som
        % referensen, wu - energin för att ha samma randvillkor &
        % nodplacering som referensen
        wbc(i)=(u'*K*u)/2;
        wu(i)=(U'*K*U)/2;
    end
end

% beräknar målfunktionen Ru, energiskillnad mellan wu & wbc
Ru=abs(wu-wbc);

% plottar energi ekvationerna & målfunktionerna

```

```
figure(1)
plot(v,Ru)
xlabel('Poissonstal')
ylabel('skillnad i energi')
```

```
figure(2)
plot(v,Rbc)
xlabel('Poissonstal')
ylabel('skillnad i energi')
```


Bilaga 11

```
function [BC]=randvillkorEXP(B1,B3)
% Funktionen randvillkorEXP bildar matris BC som skall in i solveq

% laddar data från referens & interpolering
load('DataR.mat')
load('case.mat')

M=length(B1(:));
N=length(B3(:));
k=M+N;
bc1=zeros(M,2);
bc2=zeros(M,2);
n=length(B3);

% lägger in randvillkor för nedre & övre noderna i figuren
bc1(1:n,1)=B3(1:end,1);
bc1(n+1:N,1)=B3(1:end,2);
X=vx(172:180);
Z=vy(172:180);
bc1(1:n,2)=X;
bc1(n+1:N,2)=Z;

bc2(1:n,1)=B1(1:end,1);
bc2(n+1:end,1)=B1(1:end,2);
bc2(1:n,2)=vx(1:9,1);
bc2(n+1:end,2)=vy(1:9,1);

BC=[bc1;bc2];
end
```

Bilaga 12

```

%% Interpolering för dragprov 2

clc
clf
clear all
close all

% laddar data från dragprov 2 och från referensen
load('Txyz0.txt')
load('Txyz225.txt')
load('DataR.mat')

% anger facettstorlek, steglängd, skalfaktor [px -> mm]
% c - avståndet från centrum på facetten till kanten
fs=15;
sl=13;
sfak=(3.58/251);
c=sfak*7.5;

% beräknar facettförskjutningarna och placerar dessa i x-och y-led
udata=(Txyz225-Txyz0)./1000;
ux=udata(:,1);
uy=udata(:,2);

% placerar x-och y-koordinater i hörnen för facetterna
xq=(Txyz0(:,1)./1000)-((min(Txyz0(:,1))-c)/1000);
yq=(Txyz0(:,2)./1000)-((min(Txyz0(:,2))-c)/1000);

% interpolerar nodförskjutningar för beräkningsmodellen
% utifrån facettförskjutningarna
vx=griddata(xq,yq,ux,xqr(:),yqr(:),'V4');
vy=griddata(xq,yq,uy,xqr(:),yqr(:),'V4');

% bildar en deformationsmatris med de interpolerade värdena
U=zeros(fg,1);
U(1:2:end)=vx;
U(2:2:end)=vy;

% beräknar energin för de interpolerade värdena
Wm=(U'*K*U)/2;

% placerar facetternas koordinater i x-och y-led
% och plottar de odeformerade facetterna
for z=1:length(xq)
    XQ1(z)=xq(z)-(c/1000);
    XQ2(z)=xq(z)+(c/1000);
end

for n=1:length(xq)
    Xp(n,1)=XQ1(n);
    Xp(n,4)=XQ1(n);
    Xp(n,2)=XQ2(n);

```

```
    Xp(n,3)=XQ2(n);
end

for z=1:length(yq)
    YQ1(z)=yq(z)-(c/1000);
    YQ2(z)=yq(z)+(c/1000);
end

for n=1:length(yq)
    Yp(n,1)=YQ1(n);
    Yp(n,2)=YQ1(n);
    Yp(n,3)=YQ2(n);
    Yp(n,4)=YQ2(n);
end

figure(1)
eldraw2(Xp,Yp,[1 4 1]);

% plottar de deformerade facetterna
% ed - beräknar en deformationen för varje element
[ed]=extract(Edof,U);
figure(2)
[sfac]=eldisp2(Ex,Ey,ed,[1 4 1],50);

% sparar data för dragprov 2
save('case2.mat','U','vx','vy','ux','uy')
```

Bilaga 13

```

%% Energi ekvation E-modulen, dragprov 2
clc
clear all
close all
clf

% laddar data från referensen & interpoleringen
load('DataR2.mat')
load('case2.mat')
format long

% anger intervall på E-modulen och poissons tal
E=[70e9:5e9:150e9];
v=[0.3];

w=zeros(1,length(E));

% anger randvillkor
[BC]=randvillkorEXP(B1,B3)

% Kraften från dragprovet
Fexp=7.15e3;

% beräknar energi ekvationerna för varierande E-modul
for k=1:length(E)
    for i=1:length(v)
        D=hooke(ptype,E(k),v(i));
        K=zeros(fg,fg);
        for j=1:length(Ex(:,1))
            Ke=plange(Ex(j,:),Ey(j,:),ep,D);
            K=assem(Edof(j,:),K,Ke);
        end
        f=zeros(fg,1);
        [u,f]=solveq(K,f,BC);
        % summerar de ekvivalenta nodkrafterna i y-led, raden längst upp
        Fnod(k)=sum(f(344:2:end));
        % energi ekvationer, wbc - energin för att ha samma randvillkor som
        % referensen, wu - energin för att ha samma randvillkor &
        % nodplacering som referensen
        wbc(k)=(u'*K*u)/2;
        wu(k)=(U'*K*U)/2;
        % beräknar målfunktionen Rbc, energiskillnad mellan känd kraft &
        % kraften med varierande E-modul, multiplicerat med längden på
        % provstaven
        Rbc(k)=abs(Fnod(k)-Fexp)*(U(354)-U(10));
    end
end

% beräknar målfunktionen Ru, energiskillnad mellan wu & wbc
Ru=abs(wu-wbc);

% plottar energi ekvationerna & målfunktionerna
figure(1)

```

```
plot(E,Ru)
axis([70e9 150e9 -0.5 0.5])
xlabel('E-modul')
ylabel('skillnad i energi')
```

```
figure(2)
plot(E,Rbc)
xlabel('E-modul')
ylabel('skillnad i energi')
```

```
figure(3)
plot(E,Fnod)
xlabel('E-modul')
ylabel('skillnad i kraft')
```

Bilaga 14

```

%% Energi ekvation Poissons tal, dragprov 2
clc
clear all
close all
clf

% laddar data från referensen & interpoleringen
load('DataR2.mat')
load('case2.mat')
format long

% anger intervall på E-modulen och poissons tal
E=[70e9,100e9,150e9];
v=[0.1:0.01:0.4];

w=zeros(1,length(E));

% anger randvillkor
[BC]=randvillkorEXP( B1,B3 )

% Kraften från dragprovet
Fexp=7.15e3;

% beräknar energi ekvationerna för varierande poissons tal
for k=1:length(E)
    for i=1:length(v)
        D=hooke(ptype,E(k),v(i));
        K=zeros(fg,fg);
        for j=1:length(Ex(:,1))
            Ke=plange(Ex(j,:),Ey(j,:),ep,D);
            K=assem(Edof(j,:),K,Ke);
        end
        f=zeros(fg,1);
        [u,f]=solveq(K,f,BC);
        % summerar de ekvivalenta nodkrafterna i y-led, raden längst upp
        Fnod=sum(abs(f(344:2:end)));
        % beräknar målfunktionen Rbc, energiskillnad mellan känd kraft &
        % kraften med varierande poissons tal, multiplicerat med längden på
        % provstaven
        Rbc(i)=(Fnod-Fexp)*(U(354)-U(10));
        % energi ekvationer, wbc - energin för att ha samma randvillkor som
        % referensen, wu - energin för att ha samma randvillkor &
        % nodplacering som referensen
        wbc(i)=(u'*K*u)/2;
        wu(i)=(U'*K*U)/2;
    end
end

% beräknar målfunktionen Ru, energiskillnad mellan wu & wbc
Ru=abs(wu-wbc);

% plottar energi ekvationerna & målfunktionerna
figure(1)

```

```
plot(v,Ru)
xlabel('Poissonstal')
ylabel('Skillnad i energi')
```

```
figure(2)
plot(v,Rbc)
xlabel('Poissonstal')
ylabel('Skillnad i energi')
```

Bilaga 15

```

%% Målfunktionerna Rbc & Ru för poissons tal & E-modulen
clc
clear all
close all
clf

% Anger min & max för E-modulen, och steglängd
E_min=90e9;
E_max=110e9;
N_E=20;

% Anger min & max för Poissons tal, och steglängd
v_min=0.3;
v_max=0.4;
N_v=20;

% Skapar ett nät av varierande värden för E-modulen & poissons tal
[E,v]=meshgrid(linspace(E_min,E_max,N_E), linspace(v_min,v_max,N_v))

% skapar Rbc & Ru matriser
R_u=zeros(N_E,N_v);
R_bc=zeros(N_E,N_v);

% Anropar funktion som utvärderar en kandidat
for Ei=1:N_E
    for vi=1:N_v
        [R_u(Ei,vi), R_bc(Ei,vi)]=evaluate_candidate(E(Ei,vi), v(Ei,vi))
    end
end

% Plottar målfunktionerna Rbc & Ru med varierande värden för
% poissons tal och E-modulen
figure(1); clf
contourf(E*1e-9,v,R_u)
title('Ru(E,v)')
xlabel('E [GPa]')
ylabel('v []')
hold on

figure(2); clf
contourf(E*1e-9,v,R_bc)
title('Rbc(E,v)')
xlabel('E [GPa]')
ylabel('v []')

```


Bilaga 16

```

function [R_u, R_bc] = eveluate_candidate(E,v)
% funktionen som utvärderar för olika värden på
% E-modulen och poissons tal

% laddar data från bestämt dragprov
load('DataR2.mat')
load('case2.mat')

% skapar en energimatris
w=zeros(1,length(E));

% bestämmer randvillkoren
[BC]=randvillkorEXP( B1,B3 )

% Kraftten som går in i utrustningen
Fexp=7.15e3;

% skapar materialmatrisen D och styvhetsmatrisen K
D=hooke(ptype,E,v);
K=zeros(fg,fg);

for j=1:length(Ex(:,1))
    Ke=planqe(Ex(j,:),Ey(j,:),ep,D);
    K=assem(Edof(j,:),K,Ke);
end
f=zeros(fg,1);

% beräknar nodförskjutningen och ekvivalenta nodkraften
% och summerar ekvivalenta nodkraften
[u,f]=solveq(K,f,BC);
Fnod=sum(abs(f(344:2:end)));

% beräknar energiekvationerna Wbc & Wu
wbc=(u'*K*u)/2;
wu=(U'*K*U)/2;

% beräknar målfunktionerna Rbc & Ru
R_u=abs(wu-wbc);
R_bc=abs((Fnod-Fexp)*(U(354)-U(10)));
end

```