

FHL066 P1

Edvard Johansson, tfy12ejo, 940211-0614 Oskar Smedman, tfy12osm, xxxxxx-xxxx

November 2015

1 Introduktion

För att kunna knyta an den finita elementmetoden till verkligheten är kännedom om olinjäriteter något mycket väsentligt, då dessa uppstår vid större deformationer.

2 Teori

Vid Newton-iterering används residualen, \mathbf{r} , för att bedöma när itereringen är klar. Residualen vid något steg $n + 1$ tecknas som

$$\mathbf{r}_{n+1} = \mathbf{f}_{int}^{n+1} - \mathbf{f}_{ext}^{n+1} \quad (1)$$

Här är \mathbf{f}_{int}^{n+1} den interna kraften och \mathbf{f}_{ext}^{n+1} är den externa kraften.

För den interna kraften i en stång gäller, se [Krenk, 2009], att

$$\mathbf{q}_A = -EA\epsilon_G \frac{1}{l_0} \mathbf{x} \quad \mathbf{q}_B = EA_0\epsilon_G \frac{1}{l_0} \mathbf{x}, \quad (2)$$

där \mathbf{q}_A och \mathbf{q}_B är krafterna i ändpunkterna \mathbf{x}_A respektive \mathbf{x}_B , E är elasticitetsmodulen för materialet, A_0 tvärsnittsarean, l_0 ursprungslängden för en viss stång, och ϵ_G är Green-töjningen för stängen; då vi betraktar endimensionella stänger bestäms längdändringen med ett enda tal. Denna Green-töjning relaterar kan skrivas som

$$\epsilon_G = \frac{1}{l_0} (\mathbf{x}_0^T \mathbf{u} + \frac{1}{2} \mathbf{u}^T \mathbf{u}). \quad (3)$$

I vår uppgift används två olika konstitutiva samband. Först ett linjärt, med

$$S = E\epsilon_G, \quad (4)$$

och sedan ett olinjärt

$$S = E \frac{\ln(2\epsilon_G + 1)}{4\sqrt{2\epsilon_G + 1}}, \quad (5)$$

där S i båda fallen är den andra Piola–Kirchhoff-spänningen, definierad som kvoten mellan den interna normalkraften N och A_0 .

vidare gäller att

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{u} = \mathbf{x}_B - \mathbf{x}_A + \mathbf{u}_B - \mathbf{u}_A. \quad (6)$$

Här är

$$\mathbf{x}_A = [x_A, y_A, z_A]^T \quad \mathbf{u}_A = [a_{xA}, a_{yA}, a_{zA}]^T, \quad (7)$$

och motsvarande för \mathbf{x}_B och \mathbf{u}_B . Vid beräkning av den totala interna kraften assembleras bidragen från alla noder i en global vektor. För en stång (och två noder) kan den interna kraften därför med fördel uttryckas som

$$\tilde{\mathbf{q}}^T = [\mathbf{q}_A^T, \mathbf{q}_B^T] \quad (8)$$

vilket gör att kraften för ett element kan skrivas som

$$\mathbf{f}_{int}^{e,n+1} = \frac{AS}{l_0} \begin{bmatrix} \mathbf{x}_A + \mathbf{u}_A - \mathbf{x}_B - \mathbf{u}_B \\ \mathbf{x}_B + \mathbf{u}_B - \mathbf{x}_A - \mathbf{u}_A \end{bmatrix} \quad (9)$$

vilket är precis ekvation (1) i projektbeskrivningen.

Nu har vi sett hur den interna kraften i ekvation (1) kan tecknas. Den yttre kraften beror självklart på situationen. I det här projektet beror den inte på förskjutningarna. Eftersom den inre kraften beror på förskjutningen kan den tecknas som $\mathbf{r} = \mathbf{r}(\mathbf{u})$

För att se hur krafterna förändras i varje steg kan residualen för en liten förskjutning taylorutvecklas, kring det föregående jämviktsläget, där förskjutningen är $\tilde{\mathbf{u}}$:

$$\mathbf{r}(\tilde{\mathbf{u}} + \delta\tilde{\mathbf{u}}) = \mathbf{r}(\tilde{\mathbf{u}}) + \frac{\partial \mathbf{f}_{int}(\tilde{\mathbf{u}})}{\partial \tilde{\mathbf{u}}} \delta\tilde{\mathbf{u}} + \frac{\partial \mathbf{f}_{ext}(\tilde{\mathbf{u}})}{\partial \tilde{\mathbf{u}}} \delta\tilde{\mathbf{u}} + O(\delta\tilde{\mathbf{u}}^2) \quad (10)$$

I ekvation (10) gäller att $\mathbf{r}(\tilde{\mathbf{u}}) = 0$; för i ett jämviktsläge är residualen ju noll. Vidare beror den externa kraften som sagt inte på förskjutningar, varför differentialen av den termen blir noll. I Taylorutvecklingen försummas även termer av ordningen $O(\delta\tilde{\mathbf{u}}^2)$ och högre. Efter en liten förändring av $\tilde{\mathbf{u}}$ blir nu

$$\mathbf{r}(\tilde{\mathbf{u}} + \delta\tilde{\mathbf{u}}) = \frac{\partial \mathbf{f}_{int}(\tilde{\mathbf{u}})}{\partial \tilde{\mathbf{u}}} \delta\tilde{\mathbf{u}}. \quad (11)$$

I ekvation (11) definierar vi den första faktorn som den tangentiella styvhetsmatrisen, $\mathbf{K} = \frac{\partial \mathbf{f}_{int}(\tilde{\mathbf{u}})}{\partial \tilde{\mathbf{u}}}$.

Innan denna beräknas påminnes läsaren om våra konstitutiva lagar: spänningen i en stång beror på töjningen.

Eftersom $\tilde{\mathbf{u}}^T = [\mathbf{u}_A^T, \mathbf{u}_B^T]$ innebär det att

$$\mathbf{K} = \frac{\partial \mathbf{f}_{int}(\tilde{\mathbf{u}})}{\partial \tilde{\mathbf{u}}} = \frac{\partial \mathbf{f}_{int}([\mathbf{u}_A^T, \mathbf{u}_B^T]^T)}{\partial [\mathbf{u}_A^T, \mathbf{u}_B^T]^T} = \begin{bmatrix} \frac{\partial \mathbf{f}_{int}([\mathbf{u}_A^T, \mathbf{u}_B^T]^T)}{\partial \mathbf{u}_A^T} & \frac{\partial \mathbf{f}_{int}([\mathbf{u}_A^T, \mathbf{u}_B^T]^T)}{\partial \mathbf{u}_B^T} \end{bmatrix} \quad (12)$$

Med hjälp av ekvation (9) blir styvhetsmatrisen nu

$$\mathbf{K} = \frac{A}{l_0} \begin{bmatrix} \mathbf{x}_A + \mathbf{u}_A - \mathbf{x}_B - \mathbf{u}_B \\ \mathbf{x}_B + \mathbf{u}_B - \mathbf{x}_A - \mathbf{u}_A \end{bmatrix} \frac{\partial S(\tilde{\mathbf{u}})}{\partial \tilde{\mathbf{u}}} + \frac{AS(\tilde{\mathbf{u}})}{l_0} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix}. \quad (13)$$

För beräkning av andra termen i ekvation (13) har det använts att

$$\frac{AS}{l_0} \frac{\partial}{\partial \tilde{\mathbf{u}}} \begin{bmatrix} \mathbf{x}_A + \mathbf{u}_A - \mathbf{x}_B - \mathbf{u}_B \\ \mathbf{x}_B + \mathbf{u}_B - \mathbf{x}_A - \mathbf{u}_A \end{bmatrix} = \frac{AS}{l_0} \begin{bmatrix} \frac{\partial \mathbf{x}_A + \mathbf{u}_A - \mathbf{x}_B - \mathbf{u}_B}{\partial \mathbf{u}_A} \\ \frac{\partial \mathbf{x}_B + \mathbf{u}_B - \mathbf{x}_A - \mathbf{u}_A}{\partial \mathbf{u}_A} \end{bmatrix} = \frac{AS}{l_0} \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \quad (14)$$

och omvänt för $\frac{\partial \mathbf{f}_{int}([\mathbf{u}_A^T, \mathbf{u}_B^T])}{\partial \mathbf{u}_B}$.

I ekvation 13 kan första termen variera, beroende på hur spänningen definieras, se ekvation 4 och 5. Gemensamt för de båda är att

$$\frac{\partial S(\tilde{\mathbf{u}})}{\partial \tilde{\mathbf{u}}} = \frac{\partial S}{\partial \epsilon_G} \frac{\partial \epsilon_G}{\partial \tilde{\mathbf{u}}} \quad (15)$$

I ekvation 15 gäller för den andra faktorn (se ekvation (2.25) i [Krenk, 2009]):

$$\frac{\partial \epsilon_G}{\partial \tilde{\mathbf{u}}} = \frac{\tilde{\mathbf{x}}^T}{l_0^2} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} -\mathbf{I} \\ \mathbf{I} \end{bmatrix} \frac{\mathbf{x}_B - \mathbf{x}_A + \mathbf{u}_B - \mathbf{u}_A}{l_0^2} \quad (16)$$

Om ekvation 4 används för S erhålls

$$\frac{\partial S}{\partial \epsilon_G} = E, \quad (17)$$

vilket gör att

$$\mathbf{K} = \frac{A}{l_0} \begin{bmatrix} \mathbf{x}_A + \mathbf{u}_A - \mathbf{x}_B - \mathbf{u}_B \\ \mathbf{x}_B + \mathbf{u}_B - \mathbf{x}_A - \mathbf{u}_A \end{bmatrix} E \frac{\tilde{\mathbf{x}}^T}{l_0^2} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix} + \frac{AS(\tilde{\mathbf{u}})}{l_0} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix} \quad (18)$$

om ekvation 5 i stället används erhålls

$$\frac{\partial S}{\partial \epsilon_G} = \frac{E}{4} \left(\frac{1}{2\epsilon_G + 1} \cdot \frac{2}{\sqrt{2\epsilon_G + 1}} - \frac{\ln(2\epsilon_G + 1)}{(2\epsilon_G + 1)^{3/2}} \right). \quad (19)$$

Så att dess styvhetsmatris slutligen blir

$$\mathbf{K} = \frac{A}{l_0} \begin{bmatrix} \mathbf{x}_A + \mathbf{u}_A - \mathbf{x}_B - \mathbf{u}_B \\ \mathbf{x}_B + \mathbf{u}_B - \mathbf{x}_A - \mathbf{u}_A \end{bmatrix} \frac{\partial S}{\partial \epsilon_G} \frac{\tilde{\mathbf{x}}^T}{l_0^2} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix} + \frac{AS(\tilde{\mathbf{u}})}{l_0} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix} \quad (20)$$

Där $\frac{\partial S}{\partial \epsilon_G}$ är det som beskrivs i ekvation (19).

2.1 Kraftstyrd last

Vid studie av någon struktur kan det vara intressant att se hur denna beter sig under inverkan av olika krafter. Detta kan göras numeriskt genom att utgå från ett jämviktsläge och sedan söka ett nytt jämviktsläge som borde vara någonstans i närheten. Till en början undersöks strukturen med hjälp av en enklare stegningsmetod, nämligen kraftstyrning. Det innebär att vi tvingar kraften att ha ökat med ett givet värde mellan två steg. Tyvärr kan denna enkla metod ge problem: om vi föreskriver en kraftökning på särskilda strukturer kommer det att uppstå en tvärknäppning (eng. *snap-through*), en diskontinuerlighet i grafen, vilket yttrar sig som spikraka streck: stor förskjutningsförändring för en liten kraftökning. Bortsett från denna anomali är grafen dock korrekt; punkterna därutöver ligger på jämviktskurvan.

2.2 Newton-iteration

Då man bestämt hur man vill stega måste man faktiskt hitta en punkt på kurvan som uppfyller jämviktsvillkoret $\mathbf{r} = \mathbf{0}$. Detta kan göras med hjälp av Newton-iteration. Residualen beror i statiken endast på nodförskjutningar och yttre kraft. Från (11) följer, för ett förskjutningsinkrement $\delta\mathbf{u}$, att

$$\mathbf{K}\delta\mathbf{u} = \mathbf{r}. \quad (21)$$

Till varje ny uppsättning storheter (spänning, töjning, o.s.v.) under ett iterationssteg hör en ny residual \mathbf{r} . Detta innebär att man före beräkningen enligt (21) befinner sig precis på avståndet \mathbf{r} ifrån jämviktsläget i kraftled; i praktiken finner man nästa jämviktsläge genom att studera tangenten i nuvarande jämviktsläge. Därefter inses att man kan finna nollstället i dennes riktning. Utgående från förskjutningen i detta föregående jämviktsläge adderas nu ytterligare en förskjutning som ett slags linjär extrapolering. Det tål att upprepas att residualen inte är linjär, vilket gör att man oftast inte träffar rätt—den nya förskjutningen som man testade stämde inte helt och hållet. Dock har man nu ett nytt läge allt närmare det eftertraktade att utgå ifrån. Härifrån upprepas samma procedur, och till slut befinner man sig godtyckligt nära jämviktsläget, se figur 1.7 i [Krenk, 2009].

Det finns en viktig detalj att betrakta då man beräknar residualen. Den yttre lastvektorn \mathbf{f} vi i praktiken matar in motsvarar inte övergripande de sanna krafterna, utan bara i de noder med naturliga randvillkor, d.v.s. där den yttre pålagda kraften är känd; där den är okänd sätts helt enkelt värdet 0. Dessa noder har i stället har *väsentliga* randvillkor, d.v.s. känd förskjutning, och då vi itererar fram lösningar tvingar vi dessa att vara uppfyllda från början. Det innebär att vi enbart behöver betrakta residualen för de frihetsgrader som inte har väsentliga randvillkor. Det kan implementeras genom att man fixerar residualens värde till 0 för dessa frihetsgrader; de kommer ju att uppfylla randvillkoren oberoende av det.

2.3 Crisfields metod

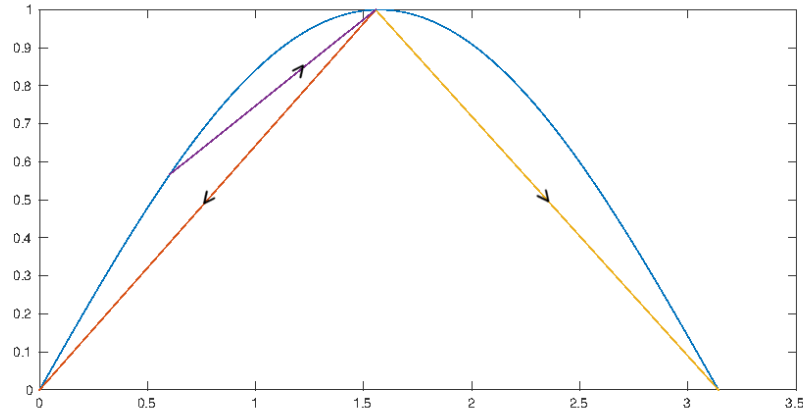
För att hantera de så kallade tvärknäppningsfenomenen kan en kurvföljande algoritm användas, i det här fallet Crisfields metod. Denna metod påbjuder att steget ligger på en tänkt sfär i kraft-förskjutningsrymden, och det enda tillåtna målet är skärningen mellan denna sfär och jämviktskurvan. Denna sfär har sin mittpunkt i det nuvarande jämviktsläget, och dess radie avgörs av vissa parametrar. Se figur 8.12 i [Krenk, 2009] för en illustration av sfären och dess skärning.

Med hjälp av denna metod anpassas ömsesidigt kraften och förskjutningen i varje steg så att ett närliggande jämviktsläge hittas. Alltså erhålls en kontinuerlig graf, till skillnad från den i fallet med kraftstyrning.

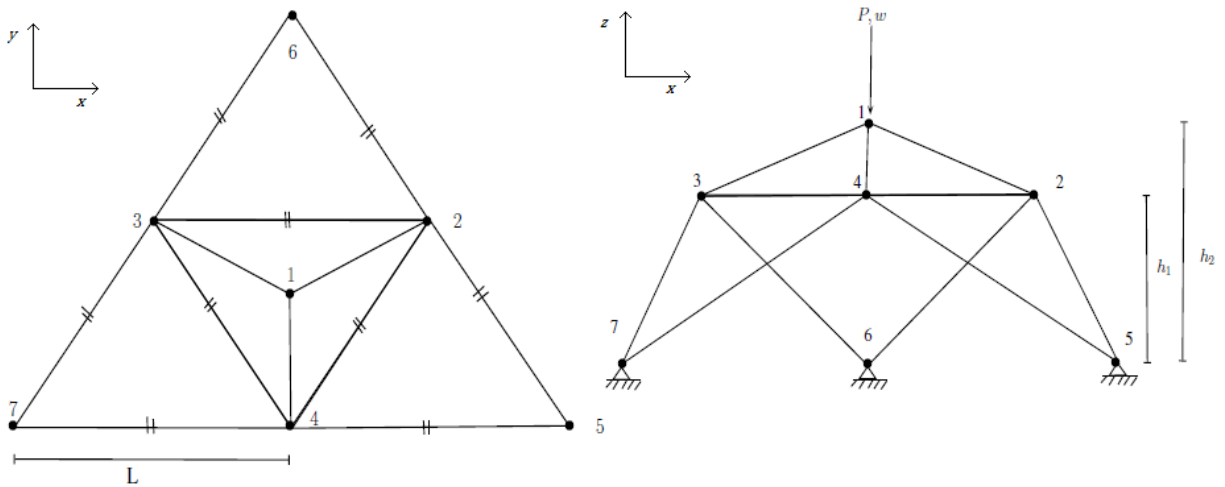
Här inses dock ett problem. Skärningen av en cirkel med mittpunkt på en graf, och grafen själv kommer alltid ha två lösningar. Det finns en eftertraktad lösning, "framåt" på kurvan, och en mindre önskvärd, gammal lösning, som är "bakåt" på kurvan. För att skilja dessa åt får man studera och jämföra med tidigare beräkningar. Med tillräckligt kort steglängd kan den korrekta lösningen alltid antas vara den vars vinkel i jämförelse med det förra steget är så stor som möjligt. I ett steg kan ju vinkeln göras godtyckligt nära 180° , om bara steglängden blir tillräckligt liten (se figur 1).

3 Problemformulering

Vi betraktar ett fackverk uppbyggt av stänger, se figur 2.



Figur 1: Här kan man se ett exempel på vad som menas med att vinkeln ska maximeras. Den korrekta grafen är den blåa, vi diskretiserar den, den lila linjen visar föregående steget. I den nya positionen kan både det röda respektive gula strecket anses vara godtagbara lösningar, den som väljes är den gula linjen. Detta eftersom vinkeln mellan den lila och gula linjen är större än vinkeln mellan den lila och röda linjen.



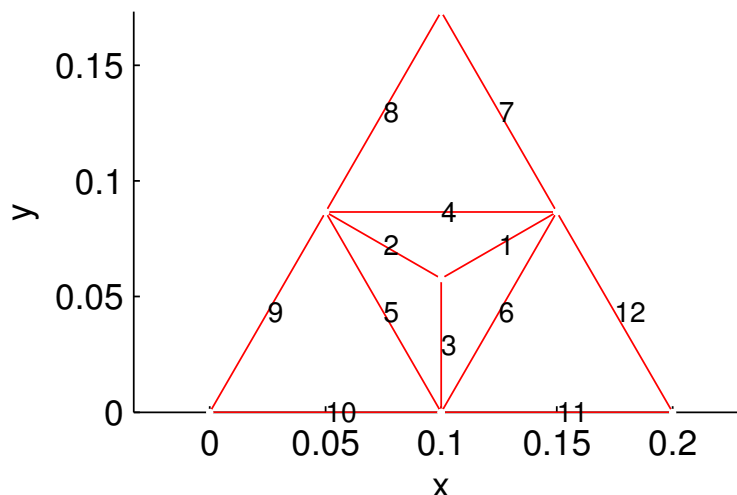
Figur 2: Fackverk. Dimensionerna ges av $L = 100$ mm, $h_1 = 20$ mm, $h_2 = 30$ mm.

Stängerna i fackverket karakteriseras av tvärsnittsarean $A_0 = 1 \text{ mm}^2$ och elasticitetsmodulen $E = 10 \text{ MPa}$. I denna uppgift undersöks det hur denna struktur svarar på belastning i form av en nedåt riktad kraft P i översta punkten, med eller utan mindre störningar. Mer specifikt:

- Ett linjärt samband mellan spänningar och töjningar antas, enligt (4).
- Kraftstyrd belastning tillämpas, och kraft-förskjutningskurvan hämtas m.h.a. Newton-Raphson-iterering.
- Crisfields metod används i stället för kraftstyrning.
- Crisfields metod används igen, men med ett olinjärt konstitutivt samband enligt (5).
- Ett antal störningar införs för att påtvinga en annan jämviktsväg än den symmetriska.

4 Implementering

För att kunna få några beräkningar gjorda måste systemet modelleras inuti vår goda vän datorn. Först och främst krävs beskrivningar av stängerna, i form av funktioner som tillåter beräkning av styvheter, töjningar, förskjutningar, spänningar, med mera. Efter att det genomförts måste alla stänger kopplas ihop på lämpligt vis, så att de interagerar på ett vettigt sätt.



Figur 3: Elementnumrering.

För att konstruera systemet i ett Matlab-program skapas Ortsvektorer för vardera nodens position, varefter alla noder numreras, och en vektor relaterar nodnummer till koordinater. Noderna kan sedan användas för att beskriva stänger, då varje stång har en start- och en slutpunkt. Båda dessa är noder – det är ju så stängerna sitter kopplade till varandra. Vid senare beräkningar kommer styvhetsmatriser beräknas för ett element, alltså en stång, i sänder. Denna elementstyvhetsmatris infogas sedan till den globala styvhetsmatrisen, för hela systemet. För att hålla koll på detta används en matris som håller relaterar stänger till element, vilket sedan möjliggör ihopkoppling av alla element till ett system. Numreringen av element kan ses i figur 3. Detta är i enlighet med grundkursen i FEM.

När geometrin implementerats ordnas lämpligen randvillkoren. Dessa utgörs av strukturens tre ben som är fastspända i marken, samt den pålagda kraften, som är känd i alla andra noder. Därefter sparas också materialparametrar, såsom elasticitetsmodul och tvärsnittsarea för stängerna. När detta är klart kan beräkningar utföras på stängerna.

Problemet som vi möter är olinjärt, främst för att geometrin för systemet ändras då det belastas. Vid de olika FEM-beräkningarna på systemet kommer det dock stegvis att behandlas som ett linjärt problem, med Newton-iteration för att kompensera för de olinjära bitarna. Detta innebär att metoder från grundkursen inom FEM, trots dess begränsning till linjära system, kan tillämpas på systemet.

I detta fallet löses problemet i den så kallade totala Lagrange-formuleringen, det vill säga med utgångspunkt i den odeformerade kroppen för att uttrycka storheter, och med dessa utföra beräkningar, till skillnad från exempelvis den uppdaterade Lagrange-formuleringen. Algoritmerna som implementerades beskrivs i Algorithm 2.1 [Krenk, 2009] samt Box 3.2. i [Ristinmaa, 2008].

Nämnvärt är att det finns flera olika jämviktswägar. Utan vidare åtgärder går systemet den symmetriska vägen. För att sedan undersöka alternativa jämviktswägar införs mindre störningar: först i geometrin, genom att flytta en nod någorlunda. Senare införs i stället en störning i lasten, i vårt fall medelst en lätt snedvinkling av den pålagda kraften så att den "spiller över" i en annan frihetsgrad. Med valfri störning tillämpad körs programmet i vanlig ordning.

Stegandet i Crisfields metod är beroende av en parameter ψ . Då man betraktar stegvillkoret

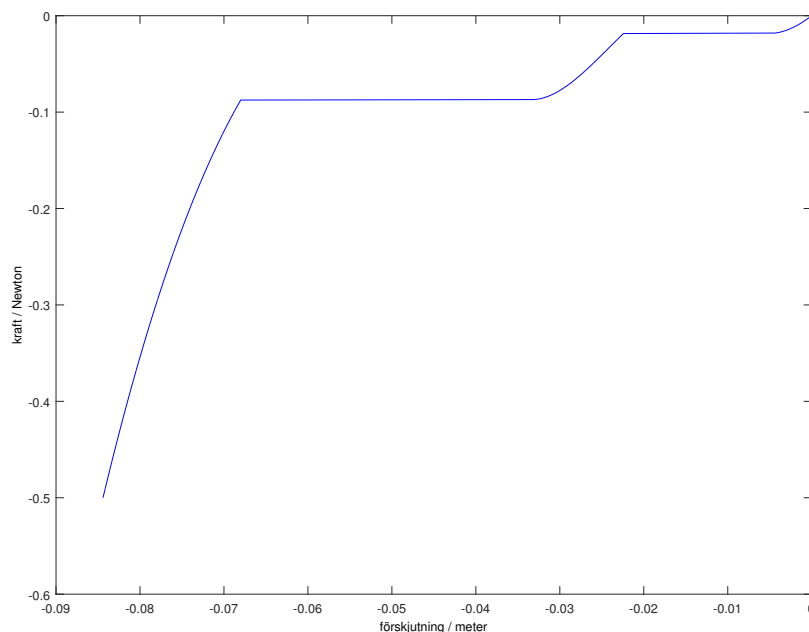
$$\Delta \mathbf{a}^T \Delta \mathbf{a} + \psi \Delta \mathbf{f}^T \Delta \mathbf{f} = l^2 \quad (22)$$

och tolkar det som en hypersfär i kraft–förskjutningsrymden utläses att ψ lämpligen väljs så att termerna $|\Delta \mathbf{a}|$ och $\sqrt{\psi}|\Delta \mathbf{f}|$ hamnar i samma storleksordning. Efter försök visar det sig passande med exempelvis $\psi \approx 0,023 \text{ m}^2 \cdot \text{N}^{-2}$, genom att studera tangenten initialt.

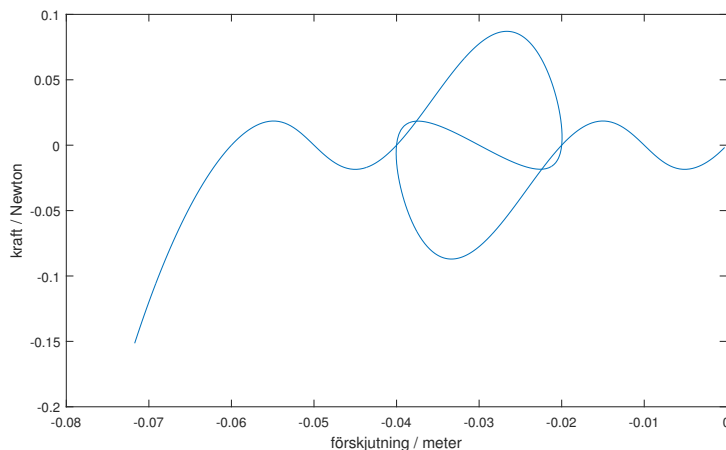
Då man byter från en konstitutiv lag från en annan är det viktigt att tänka på att det inte bara är spänningarna som beräknas på annat sätt, utan även tangentstyvheten \mathbf{K} , vars utseende ju påverkas av vald lag.

5 Resultat

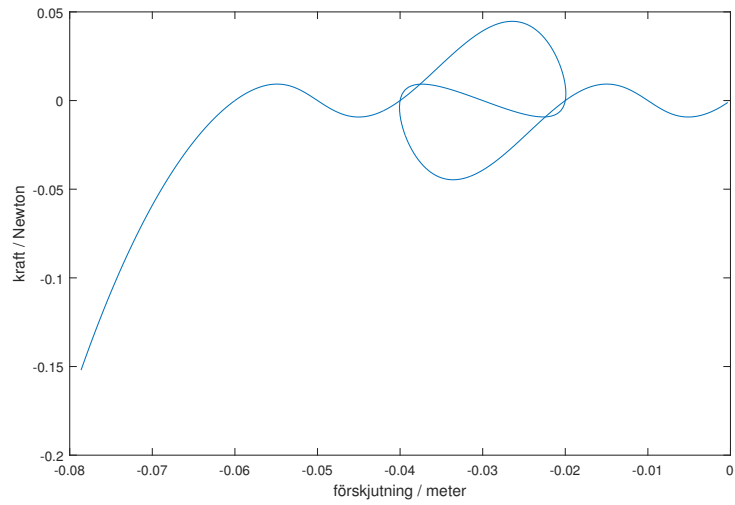
”[The report should contain] results from the calculations in form of illustrative figures and tables”



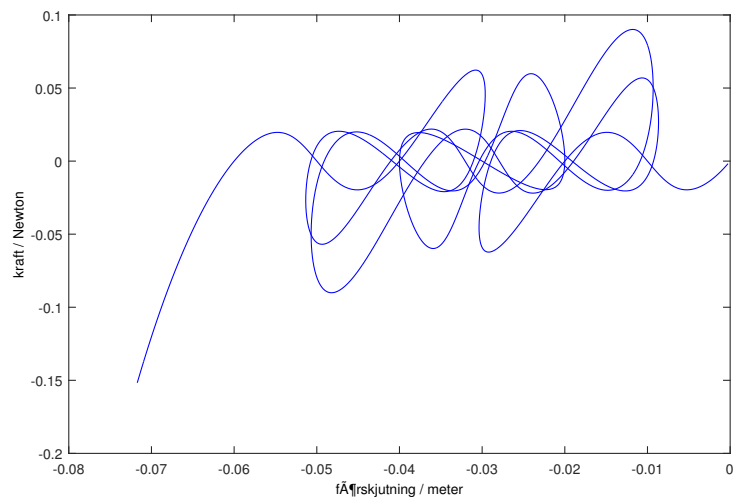
Figur 4: Kraft–förskjutningskurva hämtad m.h.a. kraftstyrning. Märk väl de två diskontinuiteterna.



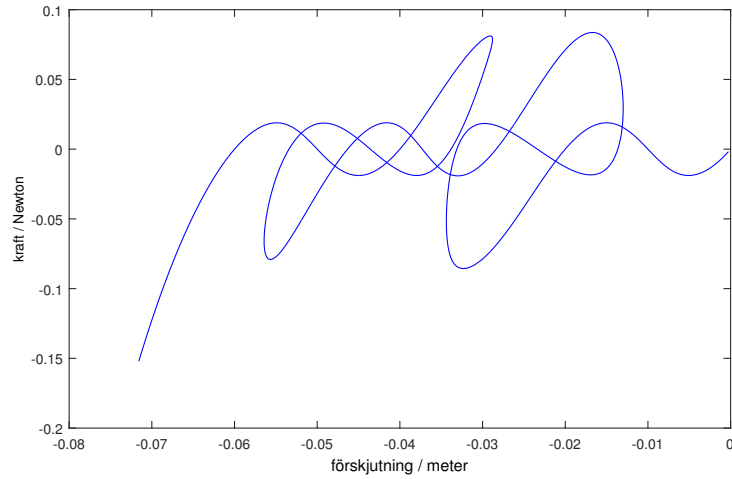
Figur 5: Kraft–förskjutningskurva hämtad m.h.a. Crisfields metod.



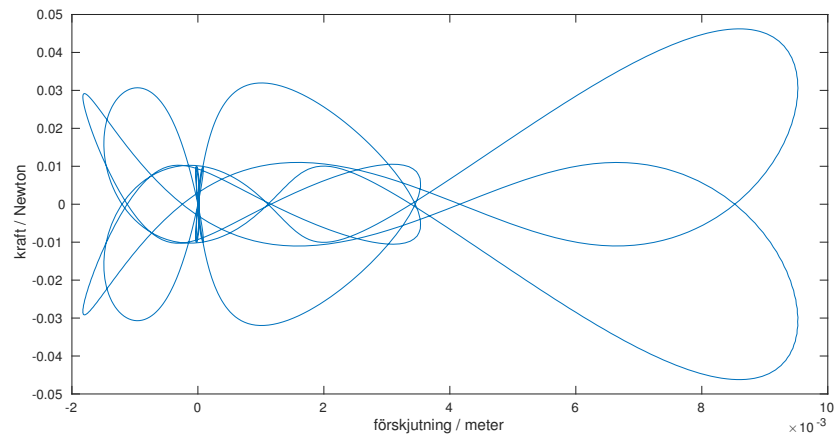
Figur 6: Kraft–förskjutningskurva hämtad m.h.a. Crisfields metod, för en olinjär konstitutiv lag.



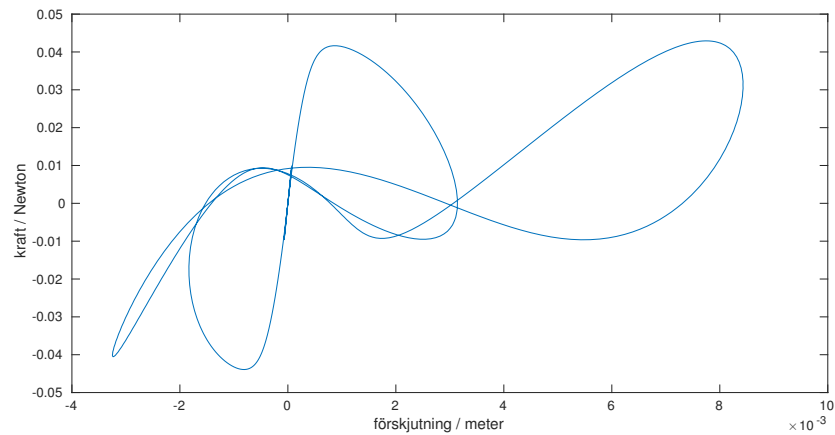
Figur 7: Kraft–förskjutningskurva då strukturens toppunkt flyttats 0,012 m is negativt y -led.



Figur 8: Kraft-förskjutningskurva då kraften vinklats 12° i positiv y -led.



Figur 9: Kraft-förskjutningskurva i y -led då strukturens toppunkt flyttats 0,012 m is negativt y -led.



Figur 10: Kraft-förskjutningskurva i y -led då kraften vinklats 12° i positiv y -led.

perturb_switch	0	1	2
u_x	0	0	0
u_y	0	0	0
u_z	-0,0600	-0,0601	-0,0601

Tabell 1: Tabellen beskriver förskjutningar i x -, y - respektive z -led i den översta noden för olika förutbestämda störningar (genom att ändra värdet på `perturb_switch`). Notera att värdena aktivt har valts kring läget där $u_z = -0,06$, samtliga angivna i meter. Detta läge motsvarar att strukturen vänts ”upp-och-ned”.

6 Diskussion

I figur 4 med dess tvåfaldiga tvärknäppning syns tydligt bristerna med kraftstyrning, jämfört med figur 5 som ter sig mer realistiskt. Figur 6 låter oss få veta att den olinjära lag vi använt inte påverkar det allmänna utseendet på jämviktskurvan något dramatiskt.

Beroende på vilken fysikalisk situation en konstruktion utsätts för, och vilken information som är relevant, bör man överväga huruvida kraftstyrning respektive Crisfield-metoden ska användas. Om man exempelvis endast planerar att påverka systeme med en (approximativt konstant) kraft, i en riktning, så kan med fördel kraftstyrning användas. Denna ger då all information som är relevant för systemet, var tvärknäppning sker och hur systemet beter sig i övrigt. Informationen som Crisfield-metoden hade kunnat ge en är ändå inte intressant och innebär ytterligare beräkningar. Till råga på detta var vår upplevelse att det var enklare att implementera en kraftstyrd lösare.

Något som är intressant att studera i figur 5 är hur många jämviktslägen som finns för en viss kraft. För krafter utanför intervallet $0,05 \text{ N} \gtrsim F \gtrsim -0,05 \text{ N}$ finns inte fler än tre jämviktslägen för given kraft. Men innanför det intervallet finns flera jämviktslägen för vissa krafter, systemet kan alltså vara i ”balans” för samma kraft trots att dess struktur skiljer sig mellan de olika jämviktslägena.

I figurerna 7 och 8 kan man se vissa likheter med det symmetriska fallet i början av kurvan, varefter de börjar bete sig annorlunda, något vi anser vara mycket intressant. För jämviktsläget vid symmetri finns den ”kortaste” jämviktsvägen. En förklaring till detta skulle kunna vara precis symmetrin. Eftersom strukturen är helt symmetrisk kommer förskjutningen också att bete sig helt symmetriskt, vilket förmodligen innebär mindre möjlighet till att grafen ”slår knut” på sig själv. Det verkar också som att figur 7 är symmetrisk i z -led, detta känns rimligt, konstruktionen blir ju bara en spegling av sig själv då den tryckts helt igenom (alltså motsvarigheten till tvärknäppning). I det fallet finns dock ett större antal knutar. Detta skulle kunna vara tack vare asymmetrin i geometrin. Tack vare den kan flera olika strukturer uppnås. Man skulle kunna se det som att kravet på symmetri sätter ett extra villkor på hur strukturen uppför sig. Då detta krav sedan förkastas kan man vänta sig ännu fler möjligheter. Även om det inte är uppenbart då enbart toppnoden betraktas måste man inse att resten av strukturen skulle kunna uppföra sig på många fler vis, vilket kan förklara den mer komplexa jämviktskurvan; för det asymmetriska fallet finns åtminstone en stång som beter sig helt annorlunda än de övriga, varför det är rimligt att asymmetrin skulle ge fler krokar.

Grafen för 8 är däremot definitivt osymmetrisk. Detta är inte oväntat. När ingen last påverkar systemet är det nämligen helt symmetriskt; trycker vi därefter på systemet med en sned last kan detta intuitivt tänkas förskjuta noderna något i lastens riktning. Detta innebär att systemet aldrig kommer återfås i en speglad struktur.

I ursprungsläget befinner sig den översta noden i läge $[0, \delta y, 0.03]$ (refereras till som startläge). Det finns ytterligare ett jämviktsläge i punkten $[0, \delta y, -0.03]$ (detta läge refereras till som det speglade läget) - se tabell 1. (Notera att $\delta y = 0$ så länge geometrin inte har störts något.) Vi beslöt oss för att studera detta läge då z -koordinaten var av samma belopp, men negativt tecken, medan de övriga koordinaterna var desamma, alltså då noden borde ha ”spegplats”. Som kan studeras i 1 förekommer endast förskjutning i z -led för just detta läge. Studeras figurerna 10 och 9 kan det konstateras att nodens y -förskjutning varierar mycket för de båda strukturerna med störningar, men nästan inget för den symmetriska strukturen (denna graf fick inte plats på de 10 sidorna). Det var väntat att det ostörda systemet skulle ha mycket små avvikelser från att förskjutningen i y -led var 0, eftersom den är symmetrisk. Det var också väntat att figur 9 skulle vara symmetrisk, med tanke på tabell 1 - det är ingen förskjutning i någon annan led än z -led då strukturen ”spegplats”. Eftersom kraften endast verkar i z -led måste vägen vara densamma för

negativa respektive positiva z - alltså måste den vara symmetrisk. För att övertyga sig om detta kan man tänka sig situationen då strukturen är i jämviktsläget svarande mot att den är helt tillplattad - vad händer då en liten kraft (i positiv respektive negativ z -led) påverkar strukturen på nod 1, samma sak kan göras för startläge och det "speglade" läget (här pekar kraften mot $z=0$). I båda dessa fallen blir situationerna desamma - helt symmetriska. Detta innebär att symmetrisk rörelse kommer ske, och likande resonemang kan göras i den nya punkten. Även för den sneda lasten återgår förskjutningen i y -led för nod 1 till 0 för det speglade läget. Denna graf är dock inte symmetrisk, vilket har med strukturen att göra. Om man här försöker tänka på strukturens beteende i ursprungsläget, respektive det speglade läget inser man en betydande skillnad. Låt oss studera vinkeln mellan element 3 (i figur 3) och kraften som påverkar systemet i startläge och det "speglade" läget. I startläget är denna vinkeln större än i det "speglade" läget, detta gör att grafen inte är symmetrisk. Notera också att förskjutningarna i x -led studerades för de tre situationerna. För samtliga av dessa var avvikelserna mycket små, detta är såklart väntat eftersom allt är helt symmetriskt i x -led.

6.1 Programmetts prestanda

Något som hade kunnat förbättra programmetts prestanda hade varit att beräkna \mathbf{K} mycket mer sällan. Vid problem, d.v.s. då denna är näst singulär, beräknas inte något nytt $\delta \mathbf{u}$; först beräknas \mathbf{K} på nytt.

Nackdelen med det är att om man ändå ska räkna ut \mathbf{K} hade man kunnat göra det under tiden som man redan itererade över elementen, för att beräkna residualen. Hursomhelst kan man spara otroligt mycket kraft på att göra detta—det jobbigaste vår dator vet är beräkningar av \mathbf{K} . I vårt fall är det dock föga beklagligt, med tanke på systemets ringa storlek.

Anledningen till att det är okej för oss att beräkna \mathbf{K} mer sällan är för att styvhetsmatrisen inte förändras så mycket för ett litet beräkningssteg. Jakobianen pekar i någorlunda korrekt riktning kommer det ändå finnas en chans att den kan användas. Newton-iteration används för att iterera fram de nya jämviktslägena. Något som är trevligt med denna är att den konvergerar mot en korrekt lösning om den konvergerar över huvud taget; den kan alltså inte hamna i fel lösning! Konvergerar det inte mot en lösning förfinas \mathbf{K} och eventuellt steglängden, varefter det testas på nytt.

Att Newton-iteration används innebär också att lösningen måste vara mycket stabil. Det globala felet ökas inte i beräkningsstegen, eftersom Newton-iterationen för varje beräkning uppfyller att dess avvikelse är mindre än någon tolerans.

Som tidigare nämnts hade en annan förbättring av programmet kunnat vara att använda uppdaterad Lagrange-formulering i stället för den totala Lagrange-formuleringen. Ekvationerna blir då formulerade på ett tydligt sätt då de uttrycks i hur strukturen faktiskt ser ut, men priset man betalar är just det att man måste uppdatera geometrin.

7 Appendix

7.1 Funktionen bar

Givet koordinaterna $[\mathbf{x}_A \quad \mathbf{x}_B]$, lagrade i `ec`, returnerar denna enkla funktion stängen skriven som en vektor, d.v.s. \mathbf{x}_0 .

```
function x0 = bar(ec)
x0 = ec(:,2)-ec(:,1);
```

7.2 Funktionen bar_def

Likt `bar` skriver denna funktion stängen på vektorform, men efter den deformerats, d.v.s. \mathbf{x} . Därför behöver den även nodförskjutningarna $\mathbf{a}^T = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_6]$.

```
function x = bar_def(ec,ed)
a = ec + [ed(1:3), ed(4:6)];
x = a(:,2)-a(:,1);
```

7.3 Funktionen bar3gs

Denna funktion beräknar kraften **es** och töjningen **ee** i en deformerad stång, givet koordinaterna, förskjutningarna och materialparametrarna $\mathbf{ep} = [E \ A_0]$, under den konstitutiva lagen (4).

```
function [es,ee] = bar3gs(ec,ep,ed)

E=ep(1); A0 = ep(2);
L0 = norm(bar(ec));      % Odeformerad längd
L = norm(bar_def(ec,ed)); % Deformerad längd

ee = (L^2-L0^2)/(2*L0^2);
es=E*A0*ee;
```

7.4 Funktionen bar3ge

Beräknar tangentstyvhetismatrisen **K** för ett stångelement, givet dess koordinater, materialparametrar, förskjutningar samt normalkraft **es**, under den linjära konstitutiva lagen (4); se avsnitt 2 i denna rapport och avsnitt 2.3 i [Krenk, 2009].

```
function Ke=bar3ge(ec,ep,ed,es)

u=(ed(4:6)-ed(1:3))';
x0 = bar(ec);
l0=norm(x0);
E=ep(1); A0 = ep(2);

M=x0*x0';
K0 = E*A0/l0^3*[M -M; -M M];

U=x0*u'+u*x0'+u*u';
Ku = E*A0/l0^3*[U -U; -U U];

Ksigma = es/l0*[eye(3) -eye(3); -eye(3) eye(3)];

Ke = K0 + Ku + Ksigma;
```

7.5 Funktionen bar3gf

Beräknar den interna kraftvektorn i ett stångelement, givet koordinater, förskjutningar och normalkraft.

```
function ef=bar3gf(ec,ed,es)
L0=norm(bar(ec));

f_B = es/L0*bar_def(ec,ed);
f_A = -f_B;
ef = vertcat(f_A,f_B)';
```

7.6 Funktionen bar3gs_log

Har samma funktion som bar3gs, men antar i stället den olinjära lagen (5). Märk väl att normalkraften $N = SA_0$.

```
function [es,ee] = bar3gs_log(ec,ep,ed)
E=ep(1); A0 = ep(2);
L0 = norm(bar(ec));
L = norm(bar_def(ec,ed));
ee = (L^2-L0^2)/(2*L0^2);

es=A0*(E*log(2*ee+1))/(4*sqrt(2*ee+1));
```

7.7 Funktionen bar3ge_log

Fungerar som bar3ge, men antar den olinjära lagen (5). Kräver dessutom Greentöjningen eps som indata.

```
function Ke=bar3ge_log(ec,ep,ed,es,eps)

x0 = bar(ec);
l_0=norm(x0);
E = ep(1); A_0=ep(2);
x = bar_def(ec,ed);
dedu = [x',-x']/(l_0^2);
dxdu = [eye(3), -eye(3); -eye(3), eye(3)];

dsde = E/4*(2*1/(2*eps+1)*1/(sqrt(2*eps+1))+2*log(2*eps+1)*(-1/2)*(2*eps+1)^(-3/2));

S = es/A_0;

Ke = (A_0/l_0)*(dsde*[x;-x]*dedu+S*dxdu);
```

7.8 Funktionen node_dof

Tar ett nodnummer som parameter och returnerar dess index i edof-matrisen.

```
function dof = node_dof(i)

dof = [3*i-2 3*i-1 3*i];
```

7.9 Funktionen calc_a

Beräknar de olika a-värdena som används i crisf.m. Inparametrar är kraften P, förändring av \mathbf{a}_i , förändring av lambda, förändring av \mathbf{a} -komponenten \mathbf{a}_G , förändring av förändring av \mathbf{a} -komponenten \mathbf{a}_P , längden l i ekvation (22), samt Ψ i samma ekvation.

```
function [a1 a2 a3 a4 a5] = calc_a(P, delta_a, delta_lambda, da_P, da_G, l, psy)
a1 = da_P'*da_P + psy*P'*P;
a2 = 2*da_P'*(delta_a+da_G) + 2*psy*delta_lambda*P'*P;
a3 = (delta_a+da_G)'*(delta_a+da_G)+psy*delta_lambda^2*P'*P-l^2;
a4 = delta_a'*(delta_a+da_G);
a5 = delta_a'*da_P;
end
```

7.10 Skriptet data

Detta skript matar in de relevanta storheterna och materialparametrarna.

```
E = 10e6; %E-modul, Pa  
A = 1e-6; %area, m^2
```

```
%Geometri  
L = 100e-3; %m  
h_1 = 20e-3; %m  
h_2 = 30e-3; %m
```

```
ep=[E A];
```

7.11 Skriptet nodes

Detta skript ordnar geometrin samt andra väsentliga storheter.

```
data;
% n1... n7 är nodernas ortsvektorer
n1 = [L;L*tand(30);h_2]; %y-koordinaten fås genom en tänkt bisektris från
%nod 7 till nod 1
n2 = [L*3/2; L*sind(60);h_1];
n3 = [L*cosd(60); L*sind(60);h_1];
n4 = [L;0;h_1];
n5 = [2*L;0;0];
n6 = [L; 2*L*sind(60);0];
n7 = [0;0;0];

% Inför störning genom att förflytta toppunkten något
if (perturb_switch == 1)
    skewing=0.012;
    n1 = [L; L*tand(30)-skewing; h_2];
end

coord = [n1'; n2'; n3'; n4'; n5'; n6'; n7'];
coord0=coord;
% Relaterar stångelement till noderna
Enod = [1 1 2;
        2 1 3;
        3 1 4;
        4 2 3;
        5 3 4;
        6 2 4;
        7 2 6;
        8 3 6;
        9 3 7;
        10 4 7;
        11 4 5;
        12 2 5];

Edof=[Enod(:,1), node_dof(Enod(:,2)),node_dof(Enod(:,3))];
nelm = length(Enod);
nnod = length(coord);
ndof = nnod*3;
a=zeros(ndof,1);

top_dof=3; %z-förskjutning i nod 1
P_end=-1.5e-1; %slutgiltig kraft

nbr_steps=1000; % antal steg (Newton-Raphson)

f=zeros(ndof,1);
df=f;
df(top_dof)=P_end/nbr_steps;

K=zeros(ndof);

% Ordnar randvillkor (se CALFEM)
bc=[];
for i=5:7
    bc = [bc; node_dof(i)', zeros(3,1)];
end
```

7.12 Skriptet newt

Belastar strukturen med konstant lastökning varje steg.

```
perturb_switch=0; % stänger av störningen
nodes;

TOL=norm(df)*1e-3;

plot_a = zeros(nbr_steps, 1);
plot_f = zeros(nbr_steps, 1);

for n = 1:nbr_steps
    f = f + df;
    K=0*K;

    % Se nedan
    for j = 1:nelm
        index_dof=Edof(j,2:end);
        index_nod=Enod(j,2:end);
        ec=coord0(index_nod,:)' ;
        ed=a(index_dof);
        [es,~]=bar3gs(ec,ep,ed);
        Ke = bar3ge(ec,ep,ed,es);
        K(index_dof,index_dof)=K(index_dof,index_dof)+Ke;
    end
    res=TOL+1; % Tvingar programmet att köra loopen minst en gång
    count=0;

    while res > TOL
        count = count+1;
        r = f;
        K=K*0;
        % Assemblerar styvhetsmatris samt residual
        for j = 1:nelm
            % Plockar ut de frihetsgrader resp. noder stängen gränsar till
            index_dof=Edof(j,2:end);
            index_nod=Enod(j,2:end);
            ec=coord0(index_nod,:)' ;
            ed=a(index_dof);
            [es, ee] = bar3gs(ec,ep,ed);
            ef=bar3gf(ec,ed,es);
            r(index_dof) = r(index_dof) - ef;
            Ke = bar3ge(ec,ep,ed,es);
            K(index_dof,index_dof)=K(index_dof,index_dof)+Ke;
        end
        da = solveq(K, r, bc);
        a=a+da;
        r(bc(:,1))=0;
        res = norm(r);

        for k=1:3
            % Lagrar koordinaterna för det deformerade systemet
            coord0(:,k) = coord0(:,k)+a(k:3:(end+k-3));
        end
    end
end
```



```

        plot_f(n) = f(top_dof);
        plot_a(n) = a(top_dof);
    end
    %%
    plot((plot_a), (plot_f))
    xlabel('förskjutning / meter')
    ylabel('kraft / Newton')

```

7.13 Skriptet crisf

Löser problemet, med eller utan störningar, med Crisfields metod.

```

% Tillåter olika störningar
%0. ingen störning
%1. störning i geometri
%2. störning i last
perturb_switch = 0;

nodes;

plot_a = [];
plot_f = [];

lambda=0;

P = zeros(ndof,1);
P_end=-.15;
if perturb_switch == 2
    P(top_dof) = P_end*cosd(12);
    P(top_dof-1) = P_end*sind(12);
else
    P(top_dof) = P_end;
end
G = P*0;
f_int=P*0;

l=4e-4; % steglängd, 'radie' för hypersfären
l_0=1;
TOL = 1*1e-3;

psy=1/44;
n=0;

old_a = a;

% Växlar mellan de två konstitutiva lagarna
% 0. olinjära
% 1. linjära
USE_HOOKE = 0;
% Tillåter kontinuerlig grafitning efter varje laststeg
continuous_plot = 0;

%Då lambda = 1 har den slutgiltiga kraften uppnåtts.
while lambda < 1

```

```

n=n+1

a_i = a;
lambda_i = lambda;

res = TOL+1;
i = 1;
G=G*0;

while (res > TOL)

    K=K*0;
    f_int=f_int*0;

    for j = 1:nelm
        index_dof=Edof(j,2:end);
        index_nod=Enod(j,2:end);
        ec=coord0(index_nod,:);

        ed=a_i(index_dof);
        % beroende på vilken konstitutiv lag som används
        % ser beräkningarna olika ut
        if USE_HOOKE
            [es,~]=bar3gs(ec,ep,ed);
            Ke = bar3ge(ec,ep,ed,es);
        else
            [es,eps]=bar3gs_log(ec,ep,ed);
            Ke = bar3ge_log(ec,ep,ed,es,eps);
        end
        K(index_dof,index_dof)=K(index_dof,index_dof)+Ke;
        ef=bar3gf(ec,ed,es);
        f_int(index_dof) = f_int(index_dof) + ef;
    end

    da_G = solveq(K,-G,bc);
    da_P = solveq(K,P,bc);

    delta_a=a_i-a;
    delta_lambda=lambda_i-lambda;
    [a1, a2, a3, a4, a5] = calc_a(P, delta_a, delta_lambda, da_P, da_G, l, psy);

    if i > 1

        p=a2/a1;          q=a3/a1;
        d_lambda_test=-p/2+sqrt((p/2)^2-q)*[1; -1];
        if (a4+a5*d_lambda_test(1) > a4+a5*d_lambda_test(2))
            d_lambda_test=d_lambda_test(1);
        else
            d_lambda_test=d_lambda_test(2);
        end
    else
        delta_a_n = a - old_a;
        s=sign(delta_a_n'*da_P);
        if (n == 1)

```

```

        % tvingar 'framåt' allra första steget
        s = 1;
    end
    d_lambda_test=s*sqrt(-a3/a1);

    s=1;
end
if ~isreal(d_lambda_test)
    % Korrigerar steglängd då denna är för stor
    l = l/2;
    continue;
else
    d_lambda = d_lambda_test;
    l = min(l*2,l_0);
end

    % godkänner de nya storheterna
    lambda_i = lambda_i + d_lambda;

    a_i = a_i + da_G + da_P*d_lambda;

    G=f_int-lambda_i*P;
    G(bc(:,1)) = 0;
    res=norm(G);
    i=i+1;

end
old_a = a;
a=a_i;

lambda=lambda_i;

plot_f(n) = lambda*P_end;
plot_a(n) = a(top_dof);

if continuous_plot
    for k=1:3
        coord(:,k) = coord0(:,k)+a(k:3:(end+k-3));
    end
    [Ex,Ey,Ez]=coordxtr(Edof,coord,node_dof((1:nnod)'),2);
    clf;
    eldraw3(Ex,Ey,Ez,[1 4 1]);
    pause;
end
end

plot(plot_a, plot_f)
xlabel('förskjutning / meter')
ylabel('kraft / Newton')
A = zeros(length(plot_a), 3);
F = zeros(length(plot_f), 3);

for k=1:3
    % sparar undan koordinaterna för det deformerade systemet

```

```

        coord(:,k) = coord0(:,k)+a(k:3:(end+k-3));
    end
%%
[Ex,Ey,Ez]=coor dxtr(Edof,coord,node_dof((1:nnode)'),2);
eldraw3(Ex,Ey,Ez,[1 4 1]);

```

Referenser

- S. Krenk. *Non-linear Modeling and Analysis of Solids and Structures*. Cambridge University press, 2009.
- M. Ristinmaa. *Introduction to non-linear Finite element method*. Division of Mechanics, Lund University, 2008.