

Slaapbank



Auteurs: Christiaan de Wit 1016180
Linda de Graaf 1016016
e-mail: c.dewit@its.tudelft.nl
lindadegraaf@hotmail.com
Vak: Modelbouw III
Vakcode: WI3093
Docenten: C. Korving
C. Vuik
Datum: juni 2003

Inhoud:

1	INLEIDING & PROBLEEMSTELLING	3
2	MODEL I.....	4
2.1	Vereenvoudigingen en Aannamen.....	4
2.2	Model opstellen.....	4
2.2.1	Krachtenevenwicht	4
2.3	Resultaten en Interpretatie	6
3	MODEL II	7
3.1	Vereenvoudigingen en Aannamen.....	7
3.2	Model opstellen.....	7
3.2.1	Coördinatentransformatie	8
3.2.2	Momentenevenwicht.....	8
3.2.3	Differentiaalvergelijking	11
3.2.4	Randvoorwaarden	11
3.2.5	Terugtransformatie (Gerepeteerde Trapeziumregel)	12
3.3	Breken van de balk (Spanningstensor)	12
3.4	Vereenvoudigen tot een lineair probleem zonder inklemming.....	13
3.4.1	Analytisch oplossen	13
3.4.2	Resultaten en interpretatie (Analytische oplossing)	14
3.4.3	Numeriek oplossen.....	15
3.4.4	Resultaten en interpretatie (Discrete oplossing)	16
3.5	Vereenvoudigen tot een niet-lineair probleem zonder inklemming	17
3.5.1	Numeriek oplossen.....	17
3.5.2	Resultaten en interpretatie.....	18
3.6	Volledige model met inklemming	19
3.6.1	Numeriek oplossen.....	19
3.6.2	Resultaten en interpretatie.....	19
4	TOETSING AAN DE WERKELIJKHEID & CONCLUSIE	23
5	BIJLAGEN – MATLAB CODE	24
5.1	main.m.....	24
5.2	init.m	24
5.3	transf.m	24
5.4	anal.m.....	25
5.5	discr.m.....	25
5.6	pic_Psin.m.....	25
5.7	Qsearch.m	26
5.8	pic_Qcos.m.....	26
5.9	sigma_xx.....	26
5.10	plot_xy.....	27
6	SYMBOLENLIJST	28
7	LITERATUURLIJST.....	29

1 Inleiding & Probleemstelling

Een vaak voorkomend probleem op feestjes is een tekort aan zitplaatsen. Een jaar geleden op mijn (Christiaan) verjaardagsfeestje liep ik tegen hetzelfde probleem op. Ik had veel te veel mensen uitgenodigd, en veel te weinig plaats. Ook van mijn slaapbank, die ik had ingeklapt, werd vol op gebruik gemaakt. Echter, mijn slaapbank beschikte over een lattenbodemplank, wat natuurlijk in theorie niet oneindig belastbaar is. Dat dit in werkelijkheid ook zo is, heb ik aan den lijve ondervonden toen in de loop van de avond een gezette vriend van mij zijn kans zag om ook eens op die heerlijk zachte bank neer te ploffen, met als resultaat dat de latjes die in het frame van de slaapbank waren geklemd spoedig omklapten. Dit kwam de stevigheid van het zitoppervlak zeker niet ten goede, daar de latjes nu nog minder kracht konden weerstaan, zodat menig latje uiteindelijk uit het frame verdween en mijn vrienden met een harde smak op de grond terecht kwamen.

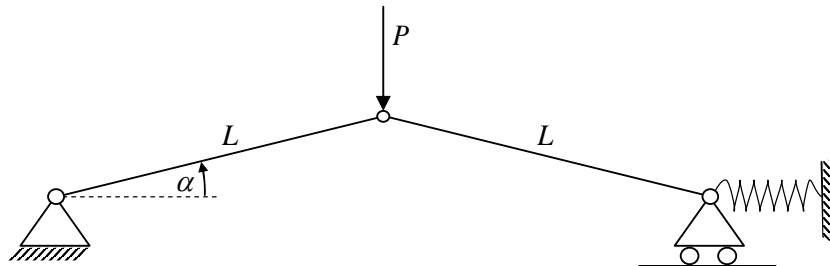
Wij vroegen ons toen af hoeveel zo'n latje nu eigenlijk aan gewicht kon hebben. Ben jij ook nieuwsgierig en heb je ook een lattenbodemplank, dan is dit werkstuk zeker de moeite waard om te lezen.



2 Model I

2.1 Vereenvoudigingen en Aannamen

In dit eerste model interpreteren we een symmetrisch gekromde houten lat op de volgende manier.



Figuur 1: Schematische opstelling Model I

In dit vereenvoudigde model spelen het scharnier (tussen de twee rechte onvervormbare balken met gelijke lengte) en de veer de rol van de buigzaamheid van de in werkelijkheid *gekromde houten lat*. De kromming van de lat in rust (zonder last) komt in dit model neer op het kiezen van een constante waarde voor α_0 . Omdat in werkelijkheid de latjes licht gekromd zijn nemen we aan dat α_0 en α klein zijn.

De puntlast P , zoals aangegeven in *figuur 1*, representeert een deel van het gewicht van een persoon dat bijv. op het matras zit. Dit matras, dat op de lattenbodem ligt, zorgt ervoor dat de kracht wordt verdeeld over de verschillende latjes van de lattenbodem.

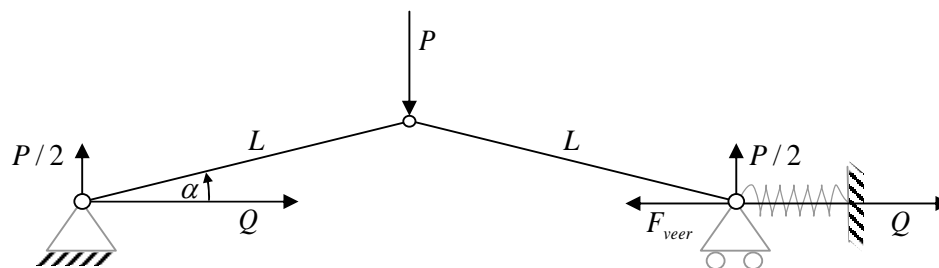
We onderzoeken onder welke puntlast een enkel 'latje' bezwijkt/doorschiet.

2.2 Model opstellen

Om te onderzoeken hoe ons model reageert op verschillende waarden van P zullen we eerst een verband moeten vinden tussen P en α , omdat dit onze enige variabelen zijn in het model. Het ligt voor de hand om eerst te proberen de krachtenevenwichten op te stellen.

2.2.1 Krachtenevenwicht

Het horizontale en verticale krachtenevenwicht vinden we uit *figuur 2* hieronder:



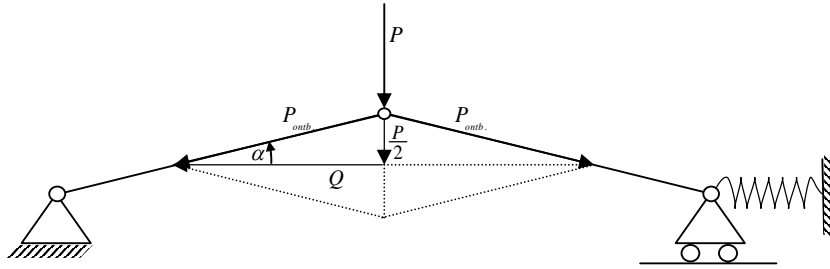
Figuur 2: Krachtenevenwicht

Horizontale krachtenevenwicht: $2Q - F_{veer} = 0$

Verticale krachtenevenwicht: $2(P/2) - P = 0$

We zien dat het verticale krachtenevenwicht ons niet veel informatie verschaft en gaan alleen verder met het horizontale krachtenevenwicht.

Omdat de twee balken even lang zijn kunnen we P op de volgende manier ontbinden:



Figuur 3: Ontbinding van P

Uit deze *figuur 3* vinden we de volgende vergelijkingen:

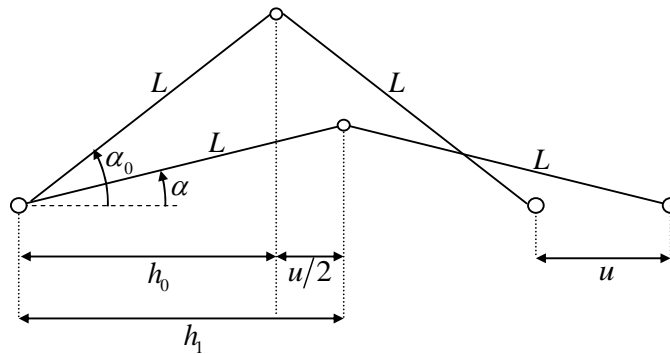
$$\left. \begin{aligned} \sin \alpha &= \frac{P/2}{P_{ontb.}} \Leftrightarrow P_{ontb.} = \frac{P}{2 \sin \alpha} \\ \cos \alpha &= \frac{Q}{P_{ontb.}} \Leftrightarrow Q = P_{ontb.} \cos \alpha \end{aligned} \right\} \Rightarrow Q = \frac{P \cos \alpha}{2 \sin \alpha}$$

Verder geldt er voor de veer: $F_{veer} = C_{veer}u$, met $C_{veer} \in \mathbb{R}$

We vinden nu voor het horizontale krachterevenwicht de volgende vergelijking:

$$\frac{P \cos \alpha}{\sin \alpha} - C_{veer}u = 0$$

We gaan nu u proberen uit te drukken in α m.b.v. onderstaand figuur:



Figuur 4: Relatie tussen α en u

Omdat $2h_1 = 2h_0 + u \Rightarrow h_1 - h_0 = \frac{u}{2}$

Verder geldt:

$$\left. \begin{aligned} \cos \alpha &= \frac{h_1}{L} \\ \cos \alpha_0 &= \frac{h_0}{L} \end{aligned} \right\} \Rightarrow \cos \alpha - \cos \alpha_0 = \frac{h_1 - h_0}{L} = \frac{u}{2L}$$

Dus: $u = 2L(\cos \alpha - \cos \alpha_0)$

We vinden nu voor het horizontale krachterevenwicht

$$P \frac{\cos \alpha}{\sin \alpha} = C_{veer} u = 2C_{veer} L (\cos \alpha - \cos \alpha_0)$$

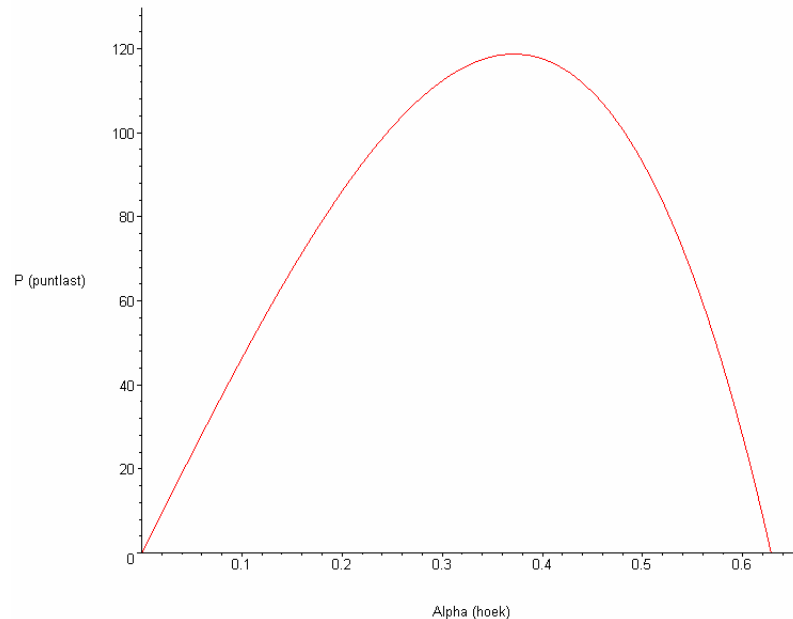
$$P = 2C_{veer} L \left(\sin \alpha - \frac{\sin \alpha \cos \alpha_0}{\cos \alpha} \right)$$

$$P(\alpha) = 2C_{veer} L \sin \alpha \left(1 - \frac{\cos \alpha_0}{\cos \alpha} \right)$$

NB: Er geldt natuurlijk dat $P(\alpha_0) = 0$

2.3 Resultaten en Interpretatie

We plotten P als functie van α voor $C_{veer} = 4000$, $L = 0,3104$ en $\alpha_0 = 0,2\pi$, daar we alleen geïnteresseerd zijn in het globale verloop van $P(\alpha)$.



Figuur 5: Grafiek Model I

In *figuur 5* zien we het verband tussen de puntlast P en de hoek α . We zien dat voor $P = 0$ geldt $\alpha = \alpha_0 = 0,2\pi \approx 0,6283$. Voor toenemende P zien we dat de hoek α steeds kleiner wordt, totdat $P \approx 120$. Vanaf dit kritieke punt zien we dat de druk P niet meer opgeschroefd hoeft te worden om de hoek α te verkleinen. Hieruit concluderen we dat de lat doorklapt, of dat deze eerder al gebroken is ten gevolge van de puntlast P .

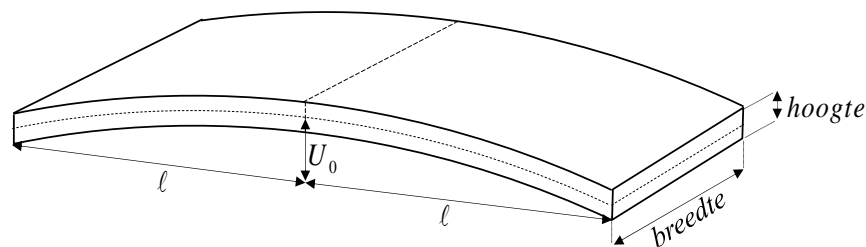
Praktisch vertelt *dit* model ons dat een latje onderhevig aan een gewicht van ± 12 kg

$\left(m = \frac{F}{g} \approx \frac{120}{10} = 12 \text{ kg, met } g \approx 10 \text{ m/s}^2 \right)$ doorklapt, danwel al eerder gebroken is.

3 Model II

3.1 Vereenvoudigingen en Aannamen

We gaan het model nu iets complexer maken en proberen het omslagpunt te bepalen van een, met kracht belaste, gekromde balk. Onze balk is een latje van een lattenbodem uit een slaapbank, dus de balk is slank. We modeleren de balk als een lijn die door het midden van de balk loopt, ook wel de elastische lijn genoemd. Voor ons model hebben we gegevens nodig die overeenstemmen met de werkelijkheid. Deze gegevens hebben we verkregen door drie latjes, van de slaapbank op de titelpagina, te meten en deze gegevens te middelen. We meten de hoogte, breedte, uitwijking (U_0) en horizontale lengte (ℓ) van de latjes, zoals in *figuur 6* is afgebeeld. Hierbij wordt alles vanaf de elastische lijn gemeten. De elasticiteitsmodulus van hout bedraagt ongeveer $2 \cdot 10^{10}$ N/m² (zie BINAS).



Figuur 6: Lat specificaties

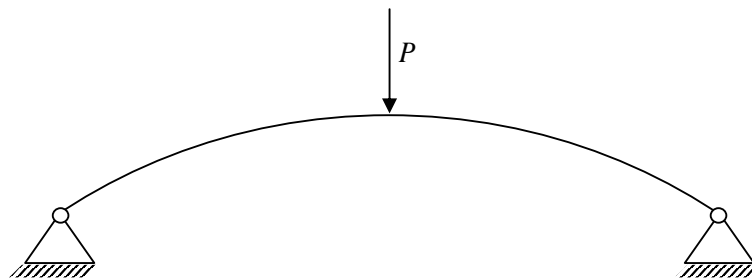
De elastische lijn heeft natuurlijk geen hoogte en breedte. Toch geeft dit geen problemen voor ons model aangezien de hoogte en breedte alleen belangrijk zijn voor het oppervlaktemoment $I = \frac{1}{12} \times \text{breedte} \times \text{hoogte}^3$.

We veronderstellen dat de lat is ingeklemd tussen twee scharnieren. Dit houdt in dat de balk bij de uiteinden geen moment kan opnemen en dat de uiteinden zich niet kunnen verplaatsen. Verder nemen we aan dat er op het midden van de lat een verticale puntlast wordt uitgeoefend.

De ruimtelijke afmetingen van de balk alsmede de elasticiteitsmodulus E en het oppervlaktemoment I zijn opgenomen in de MATLAB code (zie bijlage `init.m`).

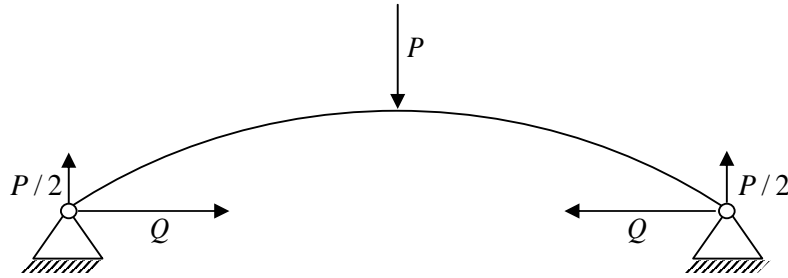
3.2 Model opstellen

We oefenen een puntlast P uit op het midden van de, tussen scharnieren ingeklemde, balk:



Figuur 7: Schematische opstelling Model II

Ten gevolge van de puntlast P wordt er door de scharnieren een reactiekracht uitgeoefend. De reactiekracht is in verticale richting $\frac{P}{2}$ N (verticaal krachtenevenwicht). In horizontale richting is deze afhankelijk van de kromming van de balk. We noemen de horizontale reactiekracht Q . In *figuur 8* is het krachtenevenwicht schematisch weergegeven.

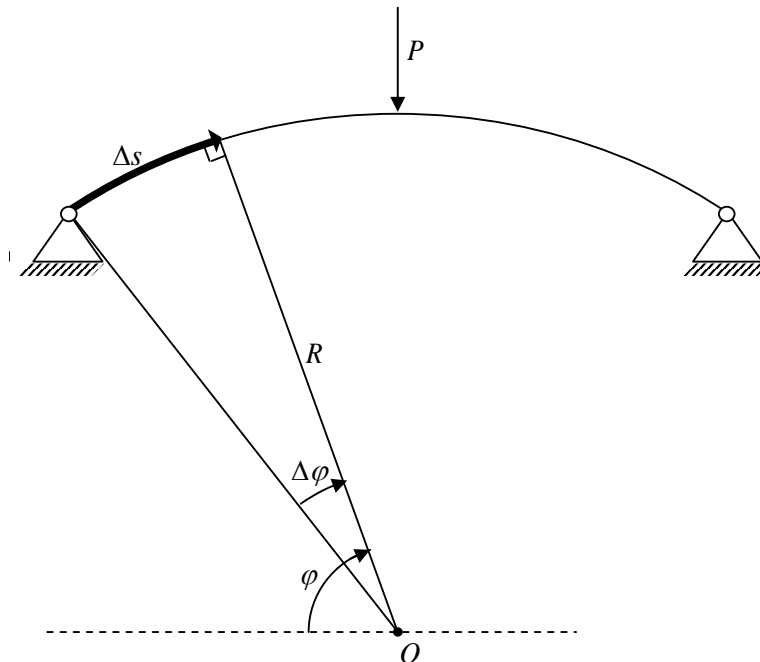


Figuur 8: Krachtenevenwicht

3.2.1 Coördinatentransformatie

We passen een coördinaten transformatie toe naar hoek φ en booglengte s . Hierbij geldt dat de hoek maal de kromtestraal gelijk is aan de booglengte, dus $\Delta\varphi \cdot R = \Delta s$ (zie *figuur 9*) ofwel:

$$R = \frac{ds}{d\varphi}$$



Figuur 9: Coördinatentransformatie

3.2.2 Momentenevenwicht

We beperken ons tot de linkerhelft van de balk, aangezien de balk symmetrisch is en aan beide kanten op dezelfde manier wordt ingeklemd. Het halveren van ons probleem heeft tot gevolg dat ook de puntlast P gehalveerd dient te worden, zodat op de linkerhelft van de balk slechts een puntlast van $P/2$ werkt (de overige $P/2$ werkt op de rechterhelft).

Voor een rechte op buiging belaste balk weten we uit de elastostatica dat het volgende verband geldt tussen het buigend moment M en de bijbehorende lokale kromtestraal R :

$$M = \frac{EI}{R}$$

Bij ons probleem echter, heeft de balk al een kromming R_0 voordat er krachten op werken. De balk heeft dus in rust wel een kromming, maar geen moment. Om hieraan te voldoen dienen we bovenstaande formule aan te passen.

In plaats van $\frac{1}{R}$ moet er nu $+\left(\frac{1}{R_0} - \frac{1}{R}\right)$ of $-\left(\frac{1}{R_0} - \frac{1}{R}\right)$ in de formule worden gesubstitueerd, zodat er in rust (als $R = R_0$) inderdaad nog geen moment is. We weten alleen nog niet of we de positieve danwel negatieve term moeten substitueren. Dit hangt af van de keuze van de positieve richting van het moment. We kiezen het moment met de klok mee positief. Het moment M in bovenstaande formule is dus positief. Verder zijn de elasticiteitsmodules E en het oppervlaktemoment I natuurlijk positief. Dus $\pm\left(\frac{1}{R_0} - \frac{1}{R}\right)$ moet positief zijn om aan de gelijkheid te voldoen. Als de balk doorbuigt wordt de kromtestraal groter, ofwel er geldt altijd dat $R > R_0$. Hieruit volgt dat $\frac{1}{R} < \frac{1}{R_0}$, ofwel $\frac{1}{R_0} - \frac{1}{R} \geq 0$. Door nu dus de positieve term te substitueren vinden we de volgende formule:

$$M = EI \left(\frac{1}{R_0} - \frac{1}{R} \right)$$

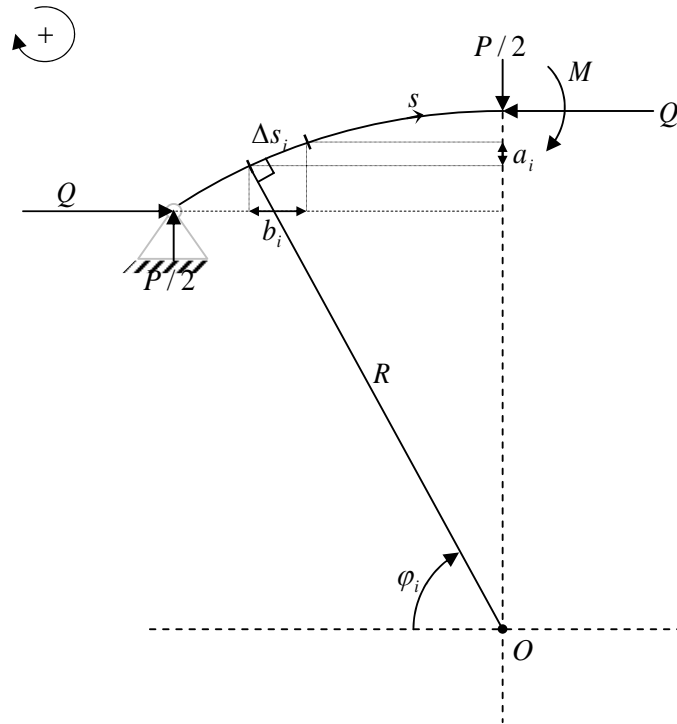
We passen de coördinatentransformatie toe, waardoor R wordt vervangen door $\frac{ds}{d\varphi}$:

$$M = EI \left(\frac{1}{R_0} - \frac{1}{R} \right) = EI \left(\frac{1}{R_0} - \frac{d\varphi}{ds} \right) \quad (1.1)$$

Nu is het de bedoeling om een tweede uitdrukking voor het moment M te vinden. Het moment ontstaat door de krachten die op de balk uitgeoefend worden. Het totale moment wordt gegeven door de som van alle 'kracht maal armen'.

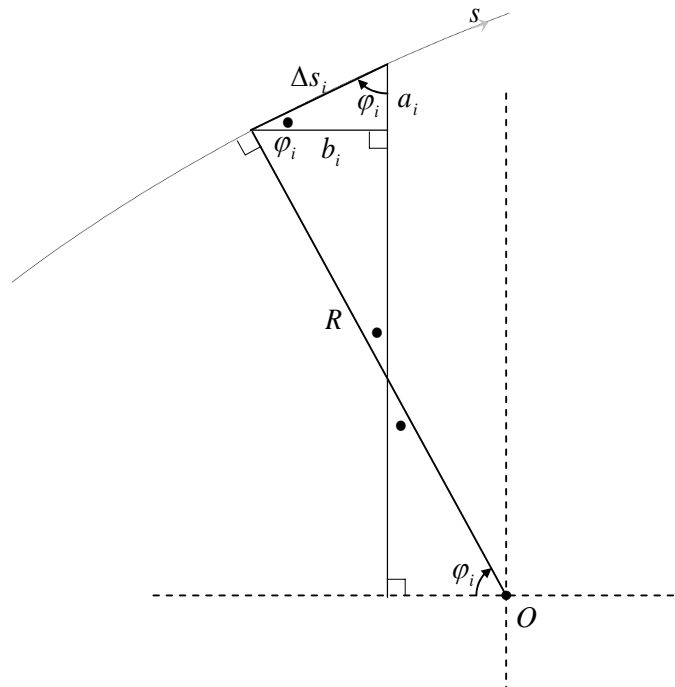
Moment per stukje van de balk

We verdelen nu de boog in stukjes Δs_i . Op Δs_i werken twee krachten, namelijk Q en $P/2$. Het moment op Δs_i is gelijk aan $-a_i \cdot Q + b_i \cdot P/2$, aangezien a_i en b_i de loodrechte armen op de krachten zijn (zie *figuur 10*).



Figuur 10: Moment van Δs_i

We proberen een expliciete uitdrukking te vinden voor a_i en b_i (zie *figuur 11*).



Figuur 11: Hoekrelaties

Zoals uit de bovenstaande figuur blijkt is de hoek φ_i bij het middelpunt O hetzelfde als de hoek tussen de lijnen Δs_i en a_i . Hierdoor weten we dat $a_i = \Delta s_i \cdot \cos \varphi_i$ en $b_i = \Delta s_i \cdot \sin \varphi_i$. Dus op een stukje Δs_i van de balk geldt $M = -Q \cdot \Delta s_i \cdot \cos \varphi_i + P/2 \cdot \Delta s_i \cdot \sin \varphi_i$.

Moment op plaats s op de balk

We bepalen het moment op plaats s op de balk. We delen de balk op in stukjes delta Δs_i en tellen de momenten van de stukjes tot en met s bij elkaar op:

$$M = -Q \cdot \sum_0^s \Delta s_i \cdot \cos \varphi_i + P/2 \cdot \sum_0^s \Delta s_i \cdot \sin \varphi_i$$

Uit de analyse weten we dat deze som convergeert, zodat:

$$M = -Q \int_0^s \cos \varphi \, ds + P/2 \int_0^s \sin \varphi \, ds \quad (1.2)$$

Door nu formule (1.1) en (1.2) te combineren kunnen we M elimineren en een differentiaalvergelijking opstellen voor ons probleem.

3.2.3 Differentiaalvergelijking

Uit (1.1) en (1.2) volgt:

$$-EI \left(\frac{d\varphi}{ds} - \frac{1}{R_0} \right) = \frac{1}{2} P \int_0^s \sin \varphi \, ds - Q \int_0^s \cos \varphi \, ds$$

Door differentiatie naar s ontstaat er een tweede orde niet-lineaire differentiaalvergelijking:

$$-EI \frac{d^2\varphi}{ds^2} = \frac{1}{2} P \sin \varphi - Q \cos \varphi \quad (1.3)$$

3.2.4 Randvoorwaarden

Om deze tweede orde differentiaalvergelijking op te lossen hebben we normaal gesproken twee randvoorwaarden nodig. Wij hebben echter in onze differentiaalvergelijking nog een onbekende kracht Q . Zodat we drie randvoorwaarden nodig hebben.

Randvoorwaarde 1: Scharnier

De balk is ingeklemd tussen scharnieren, dus bij elke scharnier is het moment gelijk aan nul, ofwel op $s = 0$ geldt $M = 0$:

$$M|_{s=0} = EI \left(\frac{1}{R_0} - \frac{d\varphi}{ds} \Big|_{s=0} \right) = 0$$

De eerste randvoorwaarde is dus:

$$\frac{d\varphi}{ds} \Big|_{s=0} = \frac{1}{R_0}$$

Randvoorwaarde 2: Horizontale fixatie middelpunt balk

De balk heeft een booglengte L en op het midden van de balk werkt een verticale kracht ter grote van $\frac{1}{2}P$. Het midden gaat dus verticaal naar beneden en verschuift nooit naar links of naar rechts. Ofwel als tweede randvoorwaarde geldt:

$$\varphi|_{s=\frac{1}{2}L} = \frac{1}{2}\pi$$

Randvoorwaarde 3: Inklemming

De balk is aan de linker- en de rechter-kant ingeklemd. De uiteinden van de balk blijven dus keurig op hun plaats. Tevens weten we uit de vorige randvoorwaarde dat ook het middelpunt van

de balk niet horizontaal verschuift. De arm van de verticale kracht $\frac{1}{2}P$ op $s = \frac{1}{2}L$ heeft dus constante lengte ℓ . Ofwel, de derde randvoorwaarde is nu:

$$\ell = \int_0^{L/2} \sin \varphi ds \quad (1.4)$$

3.2.5 Terugtransformatie (Gerepeteerde Trapeziumregel)

Aan het begin van dit hoofdstuk hebben we een transformatie uitgevoerd naar hoek φ en booglengte s . Als we echter een plaatje willen plotten zegt het ons niets als we s tegen φ uitzetten. Dus we willen graag terugtransformeren naar x en y zodat we bij het plotten de balk herkennen en kunnen zien wat er gebeurt. We nemen als x -as de horizontale as en als y -as de verticale as. Uit het momentenevenwicht weten we al het verband tussen x , y , φ en s :

$$\begin{aligned} x(s) &= \int_0^s \sin \varphi(s) ds \\ y(s) &= \int_0^s \cos \varphi(s) ds \end{aligned} \quad (1.5)$$

Deze integralen kunnen we niet analytisch uitrekenen. Dus gaan we ze numeriek bepalen. We delen $L/2$ met behulp van $n+1$ punten op in n stukjes met stapgrootte $h = L/(2n)$. Het eerst punt op $s = 0$ noemen we 0 en het laatste punt op $s = L/2$ noemen we n . We gebruiken nu de gerepeteerde trapeziumregel, die een nauwkeurigheid heeft van $O(h^2)$ (zie [NMD]):

$$\int_a^b f(x)dx = h \left[\frac{1}{2} f(x_0) + f(x_1) + \dots + \frac{1}{2} f(x_n) \right] + O(h^2)$$

Dus:

$$x(1) = h \left[\frac{1}{2} \sin \varphi(s_0) + \frac{1}{2} \sin \varphi(s_1) \right]$$

en

$$x(i) = h \left[\frac{1}{2} \sin \varphi(s_0) + \sin \varphi(s_1) + \sin \varphi(s_2) + \dots + \frac{1}{2} \sin \varphi(s_i) \right]$$

$$y(i) = h \left[\frac{1}{2} \cos \varphi(s_0) + \cos \varphi(s_1) + \cos \varphi(s_2) + \dots + \frac{1}{2} \cos \varphi(s_i) \right]$$

Echter, nu hebben de vectoren x en y een punt minder dan de vector s . Ze hebben namelijk geen x_0 en y_0 . We lossen dit op door de punten $x_0 = 0$ en $y_0 = 0$ toe te voegen. Waarom juist nul komt later ter sprake. We kunnen nu een procedure programmeren die zorgt voor de terugtransformatie (zie bijlage `transf.m`).

De rest van dit werkstuk is gewijdt aan het oplossen van de differentiaalvergelijking (1.3) met behulp van de randvoorwaarden en het in beeld brengen van de resultaten.

3.3 Breken van de balk (Spanningstensor)

De balk hoeft niet altijd door te slaan. Als er maar genoeg spanning op komt te staan kan de balk ook breken. We kunnen aan de spanningstensor σ_{xx} zien of de balk gebroken is of niet. De balk breekt als $\sigma_{xx} \approx 1 \cdot 10^7 \text{ N/m}^2$. We gaan nu proberen σ_{xx} te bepalen. Volgens [MII] blz. 32 geldt er dat $\sigma_{xx} = -Ey\eta''$ en $M = -EI\eta''$. Hierin heeft y een andere betekenis als we gewend zijn. Er geldt namelijk dat $y = \frac{1}{2} \times \text{hoogte}$ van de balk. We gaan nu η'' elimineren zodat:

$$\sigma_{xx} = \frac{E \cdot \text{hoogte} \cdot M}{2EI} = \frac{1}{2} \frac{\text{hoogte} \cdot M}{I}$$

De enige onbekende in bovenstaande vergelijking is het moment M . We kunnen nu M op elke plaats in de balk op twee manieren berekenen, namelijk m.b.v. (1.1) of m.b.v. (1.2).

Method 1

Uit (1.1) volgt:

$$M|_{s=i} = EI \left(\frac{1}{R_0} - \frac{1}{R|_{s=i}} \right) = EI \left(\frac{1}{R_0} - \frac{d\varphi}{ds} \Big|_{s=i} \right), \text{ met } \frac{d\varphi}{ds} \Big|_{s=i} = \frac{u_{i+1} - u_{i-1}}{2h} + O(h^3)$$

Method 2

Uit (1.2) volgt:

$$M|_{s=i} = -Q \int_0^{s=i} \cos \varphi ds + P/2 \int_0^{s=i} \sin \varphi ds$$

Het numeriek benaderen van bovenstaande uitdrukking geschiedt wederom met behulp van de gerepeteerde trapeziumregel (zie bijlage sigma_xx.m).

3.4 Vereenvoudigen tot een lineair probleem zonder inklemming

We beginnen met een eenvoudiger probleem. We substitueren $\varphi = \frac{1}{2}\pi$, daar de balk zeer licht gekromd is en φ dus over de gehele balk ongeveer gelijk is aan $\frac{1}{2}\pi$. De op te lossen differentiaalvergelijking wordt nu:

$$-EI \frac{d^2 \varphi}{ds^2} = \frac{1}{2} P$$

3.4.1 Analytisch oplossen

We hebben dus de vergelijking $-EI \frac{d^2 \varphi}{ds^2} = \frac{1}{2} P$. Deze kunnen we analytisch oplossen, met behulp van de randvoorwaarden. De laatste voorwaarde kunnen wij niet gebruiken omdat deze gebaseerd is op het feit dat de lat ingeklemd is, aangezien er geen kracht Q is (zodat de uiteinden niet ingeklemd zijn en vrij kunnen bewegen). We schrijven de vergelijking als volgt:

$$\frac{d^2 \varphi}{ds^2} = -\frac{P}{2EI}$$

Na integreren ontstaat er de formule:

$$\frac{d\varphi}{ds} = -\frac{P}{2EI} s + C, \text{ met } C \text{ een constante.}$$

Met behulp van de eerste randvoorwaarde bepalen we de constante C :

$$\frac{d\varphi}{ds} \Big|_{s=0} = -\frac{P}{2EI} \cdot 0 + C = C = \frac{1}{R_0}$$

Invullen van de gevonden waarde van de constante C geeft:

$$\frac{d\varphi}{ds} = -\frac{P}{2EI} s + \frac{1}{R_0}$$

Na nogmaals te integreren ontstaat er:

$$\varphi = -\frac{P}{4EI} s^2 + \frac{1}{R_0} s + D, \text{ met } D \text{ weer een constante.}$$

Nu gebruiken we de tweede randvoorwaarde:

$$\varphi|_{s=\frac{1}{2}L} = -\frac{P}{4EI} \cdot \frac{1}{4}L^2 + \frac{1}{R_0} \cdot \frac{1}{2}L + D = -\frac{PL^2}{16EI} + \frac{L}{2R_0} + D = \frac{1}{2}\pi \Rightarrow D = \frac{PL^2}{16EI} - \frac{L}{2R_0} + \frac{1}{2}\pi$$

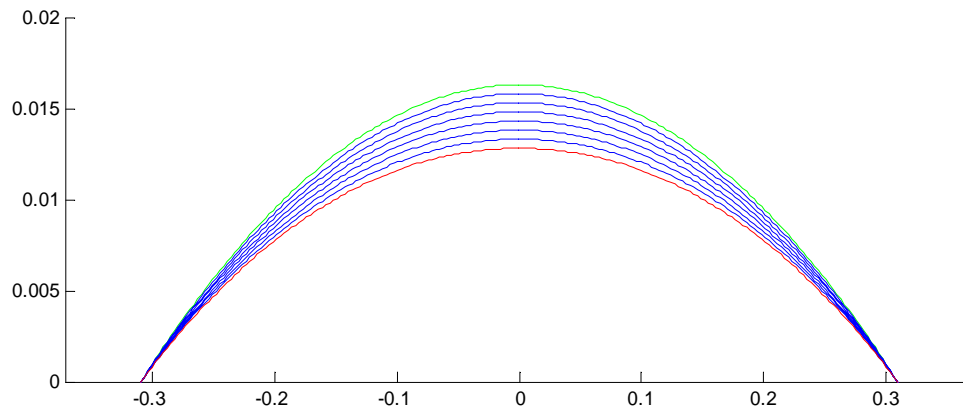
De analytische oplossing is dus:

$$\varphi(s) = -\frac{P}{4EI}s^2 + \frac{1}{R_0}s + \frac{PL^2}{16EI} - \frac{L}{2R_0} + \frac{1}{2}\pi$$

3.4.2 Resultaten en interpretatie (Analytische oplossing)

De plots in dit verslag komen als volgt tot stand. We beginnen elke plot met een plot van de balk in rust ($P = 0$) en verhogen de puntlast P steeds met 10 N, totdat de balk doorklapt of gebroken is. In de plots wordt de eenheid van de x -as en de y -as steeds in meters weergegeven. Omdat de kromming van de balk en de vervorming ten gevolge van de puntlast P nihil zijn, heeft de y -as een andere schaalverdeling dan de x -as. De groene grafiek in elke plot geeft de balk in rust ($P=0$) weer, de rode grafiek geeft aan dat de balk gebroken of doorgeklapt is. Alle tussenliggende situaties zullen in het blauw worden weergegeven. De specificaties van de balk zijn verder voor alle plots als in bijlage `init.m`, tenzij iets anders staat vermeld.

We plotten de teruggetransformeerde analytische oplossing m.b.v. MATLAB (zie bijlage `ana1.m`).



Grafiek 1: Analytische oplossing (zonder inklemming) voor $P=0:10:70$

P in N	σ_{xx} in N/m^2
0	0
10	1.55e+006
20	3.10e+006
30	4.65e+006
40	6.20e+006
50	7.75e+006
60	9.31e+006
70	1.09e+007

Tabel 1

We zien dat de balk bij een puntlast tussen de 60 en 70 N breekt. Dus volgens dit analytische model, zonder inklemming, breekt de lat bij een belasting van ongeveer 7 kg.

3.4.3 Numeriek oplossen

We kunnen de differentiaalvergelijking ook numeriek oplossen, met een nauwkeurigheid van $O(h^2)$ (zie blz 56 en 57 [NMD]). Alleen op $s = 0$ is de nauwkeurigheid van orde $O(h)$.

Discretiseren

We verdelen de balk in n stukjes van lengte $h = \frac{L}{2n}$. Op de balk geldt nu:

$$\left. \frac{d^2 \varphi}{ds^2} \right|_{s=ih} = \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} + O(h^2) = -\frac{P}{2EI} \quad \text{met } u_i = \varphi(ih)$$

Voor $i = 0 \dots (n-2)$ volgt:

$$\begin{aligned} i = 0 & \quad \frac{1}{h^2} \begin{pmatrix} u_{-1} & -2u_0 & +u_1 \end{pmatrix} = -\frac{P}{2EI} \\ i = 1 & \quad \frac{1}{h^2} \begin{pmatrix} u_0 & -2u_1 & +u_2 \end{pmatrix} = -\frac{P}{2EI} \\ i = 2 & \quad \frac{1}{h^2} \begin{pmatrix} u_1 & -2u_2 & +u_3 \end{pmatrix} = -\frac{P}{2EI} \\ & \quad \vdots \\ i = (n-2) & \quad \frac{1}{h^2} \begin{pmatrix} u_{n-3} & -2u_{n-2} & +u_{n-1} \end{pmatrix} = -\frac{P}{2EI} \\ i = (n-1) & \quad \frac{1}{h^2} \begin{pmatrix} u_{n-2} & -2u_{n-1} & +u_n \end{pmatrix} = -\frac{P}{2EI} \end{aligned}$$

Op de randen hebben we nu een probleem, aangezien we niet weten wat u_{-1} en u_n zijn. Hiervoor gebruiken we de randvoorwaarden. Uit randvoorwaarde 1 volgt:

$$\left. \frac{d\varphi}{ds} \right|_{s=0} = \frac{1}{R_0} \Rightarrow \left. \frac{d\varphi}{ds} \right|_{s=0} = \frac{u_1 - u_{-1}}{2h} + O(h^3) = \frac{1}{R_0} \Rightarrow u_{-1} \approx u_1 - \frac{2h}{R_0}$$

Randvoorwaarde 2 geeft:

$$\varphi|_{s=\frac{1}{2}L} = \frac{1}{2}\pi \Rightarrow \varphi|_{s=\frac{1}{2}L} = u_n = \frac{1}{2}\pi$$

Substitutie van u_{-1} en u_n geeft:

$$\begin{aligned} i = 0 & \quad \frac{1}{h^2} \begin{pmatrix} -u_0 & +u_1 \end{pmatrix} = -\frac{P}{4EI} + \frac{1}{R_0 h} \\ i = 1 & \quad \frac{1}{h^2} \begin{pmatrix} u_0 & -2u_1 & +u_2 \end{pmatrix} = -\frac{P}{2EI} \\ i = 2 & \quad \frac{1}{h^2} \begin{pmatrix} u_1 & -2u_2 & +u_3 \end{pmatrix} = -\frac{P}{2EI} \\ & \quad \vdots \\ i = (n-2) & \quad \frac{1}{h^2} \begin{pmatrix} u_{n-3} & -2u_{n-2} & +u_{n-1} \end{pmatrix} = -\frac{P}{2EI} \\ i = (n-1) & \quad \frac{1}{h^2} \begin{pmatrix} u_{n-2} & -2u_{n-1} \end{pmatrix} = -\frac{P}{2EI} - \frac{\pi}{2h^2} \end{aligned}$$

Nu kunnen we dit in matrix notatie schrijven:

$$\frac{1}{h^2} \begin{pmatrix} -1 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & 1 & -2 & 1 \\ & & & & & 1 & -2 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{n-2} \\ u_{n-1} \end{pmatrix} = \begin{pmatrix} -P/(4EI) + 1/(R_0h) \\ -P/(2EI) \\ -P/(2EI) \\ \vdots \\ -P/(2EI) \\ -P/(2EI) - \pi/(2h^2) \end{pmatrix}$$

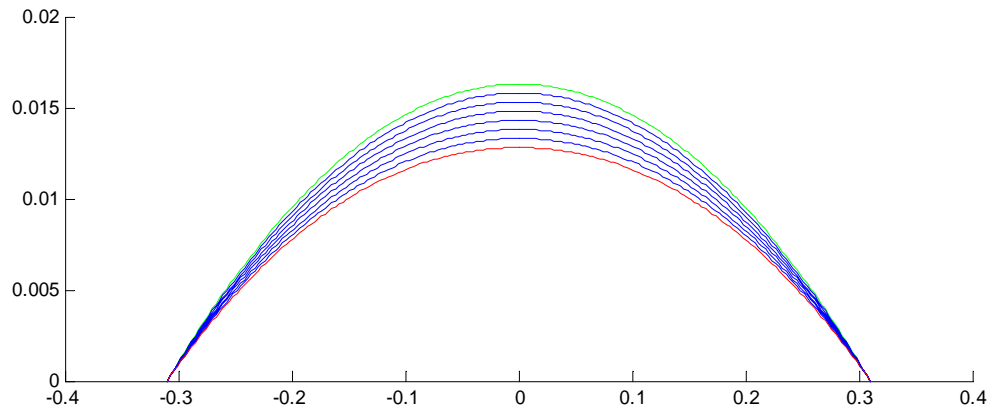
Dit is van de vorm:

$$\mathbf{A}\mathbf{u} = \mathbf{f}$$

We kunnen bovenstaand matrixvectorproduct oplossen met behulp van MATLAB.

3.4.4 Resultaten en interpretatie (Discrete oplossing)

We plotten de teruggetransformeerde discrete oplossing m.b.v. MATLAB (zie bijlage `discr.m`).



Grafiek 2: Discrete oplossing (zonder inklemming) voor $P=0:10:70$

P in N	σ_{xx} in N/m^2
0	0
10	1.55e+006
20	3.10e+006
30	4.65e+006
40	6.20e+006
50	7.75e+006
60	9.31e+006
70	1.09e+007

Tabel 2

De gevonden discrete oplossing is vrijwel identiek aan de analytische oplossing, wat natuurlijk ook te verwachten was, daar het probleem (de differentiaalvergelijking) dezelfde betreft. Deze resultaten bevestigen dat de discretisatie (en de implementatie hiervan) hoogstwaarschijnlijk correct zijn, zodat we dit als basis kunnen gebruiken voor het oplossen van de niet-lineaire differentiaalvergelijking (1.3).

3.5 Vereenvoudigen tot een niet-lineair probleem zonder inklemming

We hebben in de vorige paragraaf een redelijk eenvoudig probleem opgelost. We gaan nu het probleem iets ingewikkelder maken door niet meer $\varphi = \frac{1}{2}\pi$ te substitueren. Er wordt wel aangenomen dat de balk *niet* ingeklemd is door. Dit wordt gerealiseerd door Q nul te stellen, zodat de term $-Q \cos \varphi$ weg valt. De op te lossen differentiaalvergelijking wordt nu:

$$-EI \frac{d^2 \varphi}{ds^2} = \frac{1}{2} P \sin \varphi$$

Dit is een niet-lineair probleem, wat niet meer analytisch op te lossen is. In de volgende paragraaf bespreken we het numeriek oplossen van deze niet-lineaire differentiaalvergelijking (zonder inklemming).

3.5.1 Numeriek oplossen

We gaan de differentiaalvergelijking numeriek oplossen.

Discretiseren

Discretiseren gaat analoog aan het lineaire model:

$$\frac{1}{h^2} \begin{pmatrix} -1 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & -2 & 1 \\ & & & & & 1 & -2 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{n-2} \\ u_{n-1} \end{pmatrix} = \begin{pmatrix} -P/(4EI) \cdot \sin \varphi + 1/(R_0 h) \\ -P/(2EI) \cdot \sin \varphi \\ -P/(2EI) \cdot \sin \varphi \\ \vdots \\ -P/(2EI) \cdot \sin \varphi \\ -P/(2EI) \cdot \sin \varphi - \pi/(2h^2) \end{pmatrix}$$

Niet-lineaire term verwerken (Picard iteratie)

De niet lineaire term aan de rechterzijde verwerken we met behulp van *Picard iteratie* (zie blz. 87 van [NMD]). We kiezen een startvector \mathbf{u}_k en substitueren deze voor φ in de rechterzijde, waarna een nieuwe oplossing \mathbf{u}_{k+1} wordt verkregen door het matrixvectorproduct op te lossen. Deze nieuwe oplossing wordt vervolgens weer als startwaarde gebruikt voor een nieuwe *Picard iteratie*. Schematisch kunnen we dit als volgt noteren:

$$\mathbf{A} \mathbf{u}_{k+1} = \mathbf{f}(\mathbf{u}_k)$$

Voor startwaarde \mathbf{u}_0 kiezen we de discrete oplossing van het lineaire probleem. Per iteratie bepalen we de term:

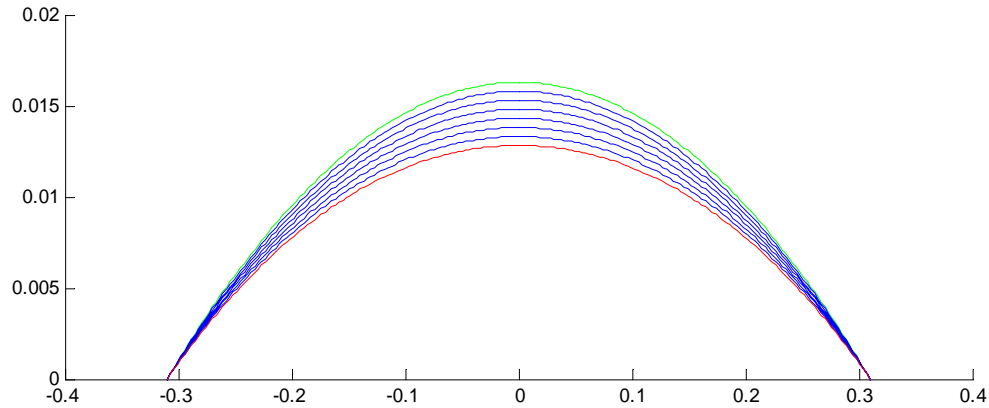
$$\|\mathbf{u}_{k+1} - \mathbf{u}_k\|_{\infty}$$

Deze uitdrukking representeert de afstand tussen de nieuwe en de oude oplossing. We beëindigen het iteratie proces wanneer deze afstand kleiner is dan een bepaalde tolerantie (bijv. 1×10^{-16} , de precisie van de computer).

Bovenstaand proces is geïmplementeerd in MATLAB.

3.5.2 Resultaten en interpretatie

We plotten de teruggetransformeerde discrete oplossing van het niet-lineaire probleem (zonder inklemming) m.b.v. MATLAB (zie bijlage pic_Psin.m).



Grafiek 3: Discrete oplossing van het niet-lineaire probleem (zonder inklemming) voor $P=0:10:70$

P in N	σ_{xx} in N/m^2
0	0
10	1.55e+006
20	3.10e+006
30	4.65e+006
40	6.20e+006
50	7.75e+006
60	9.31e+006
70	1.09e+007

Tabel 3

De gelijkenis met voorgaande resultaten verklaren we aan de hand van de, al eerder genoemde, zeer lichte kromming van de balk. Deze zeer lichte kromming komt overeen met een zéér grote kromtestraal, wat als gevolg heeft dat de hoek φ over de gehele balk vrijwel constant is. Immers, voor een balk met constante booglangte L en grote kromtestraal R_0 geldt dat het verschil tussen de hoek aan het begin en het einde van de balk gelijk is aan:

$$(\varphi(L) - \varphi(0))R_0 = L \Rightarrow \varphi(L) - \varphi(0) = \frac{L}{R_0} \approx 0 \quad \text{met } R_0 \text{ zéér groot}$$

Bovenstaande, gecombineerd met randvoorwaarde 2 geeft:

$$\left. \begin{array}{l} \varphi(s) \approx \text{constant} \\ \varphi(\frac{1}{2}L) = \frac{1}{2}\pi \end{array} \right\} \Rightarrow \varphi(s) \approx \frac{1}{2}\pi \quad \text{met } 0 \leq s \leq L$$

Hieruit volgt dat het niet-lineaire probleem vrijwel gelijk is aan het lineaire probleem:

$$-EI \frac{d^2\varphi}{ds^2} = \frac{1}{2}P \sin \varphi(s) \approx \frac{1}{2}P \sin\left(\frac{1}{2}\pi\right) = \frac{1}{2}P$$

NB: De nihiele vervorming van de balk ten gevolge van de puntlast P zal de kromtestraal eventueel nog kunnen verhogen, wat onze voorgaande conclusie niet beïnvloed.

3.6 Volledige model met inklemming

We beschouwen het volledige probleem:

$$-EI \frac{d^2 \varphi}{ds^2} = \frac{1}{2} P \sin \varphi(s) - Q \cos \varphi(s)$$

3.6.1 Numeriek oplossen

We gaan de differentiaalvergelijking numeriek oplossen.

Discretiseren

We discretiseren weer analoog aan voorgaande problemen:

$$\frac{1}{h^2} \begin{pmatrix} -1 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{n-2} \\ u_{n-1} \end{pmatrix} = \begin{pmatrix} -P/(4EI) \cdot \sin \varphi + Q/2EI \cdot \cos \varphi + 1/(R_0 h) \\ -P/(2EI) \cdot \sin \varphi + Q/EI \cdot \cos \varphi \\ -P/(2EI) \cdot \sin \varphi + Q/EI \cdot \cos \varphi \\ \vdots \\ -P/(2EI) \cdot \sin \varphi + Q/EI \cdot \cos \varphi \\ -P/(2EI) \cdot \sin \varphi + Q/EI \cdot \cos \varphi - \pi/(2h^2) \end{pmatrix}$$

Niet-lineaire term verwerken (Picard iteratie) en bepaling van de juiste Q

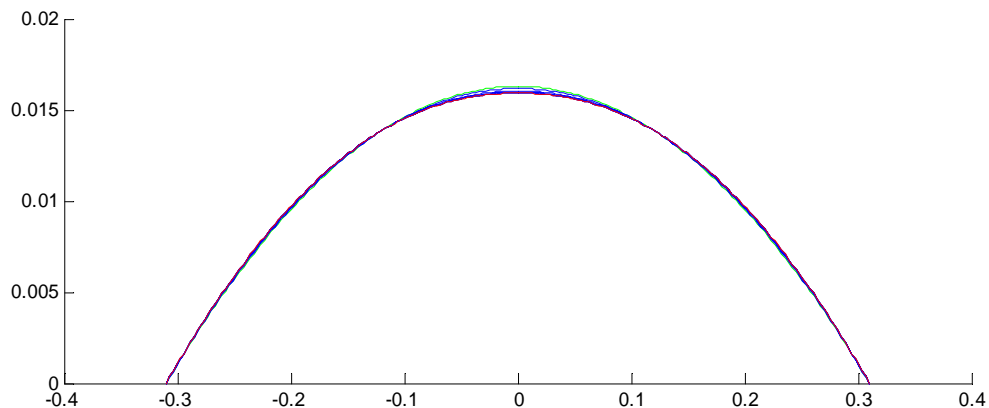
Analoog aan voorgaand model, passen we weer *Picard iteratie* toe. We kunnen zo het probleem voor willekeurige Q oplossen. De kracht Q is nog niet bepaald. We gebruiken onze derde randvoorwaarde om deze extra variabele te bepalen. Door (1.4) en (1.5) te combineren ontstaat er:

$$\ell = \int_0^{L/2} \sin \varphi ds = x(L/2) = x(nh)$$

Ofwel de n -de coördinaat van vector \mathbf{x} moet gelijk zijn aan ℓ . Met een pseudo binair zoekalgoritme bepalen we een Q en lossen het probleem op. Vervolgens transformeren we terug en controleren of de n -de coördinaat van \mathbf{x} gelijk is aan ℓ . Indien dit niet het geval is past het pseudo binair zoekalgoritme de waarde van Q aan, totdat een bepaalde precisie (tolerantie) is voldaan (zie bijlage `Qsearch.m` en `pic_Qcos.m`).

3.6.2 Resultaten en interpretatie

We plotten de teruggetransformeerde discrete oplossing van het niet-lineaire volledige probleem met inklemming m.b.v. MATLAB (zie bijlage `Qsearch.m` en `pic_Qcos.m`).



Grafiek 4a: Discrete oplossing van het niet-lineaire probleem met inklemming voor $P=0,100,200,250:10:280$

P in N	Q in N	σ_{xx} in N/m ²
0	0	0
100	7.43e+002	3.48e+006
200	1.49e+003	7.12e+006
250	1.86e+003	9.01e+006
260	1.94e+003	9.39e+006
270	2.01e+003	9.78e+006
280	2.09e+003	1.02e+007

Tabel 4

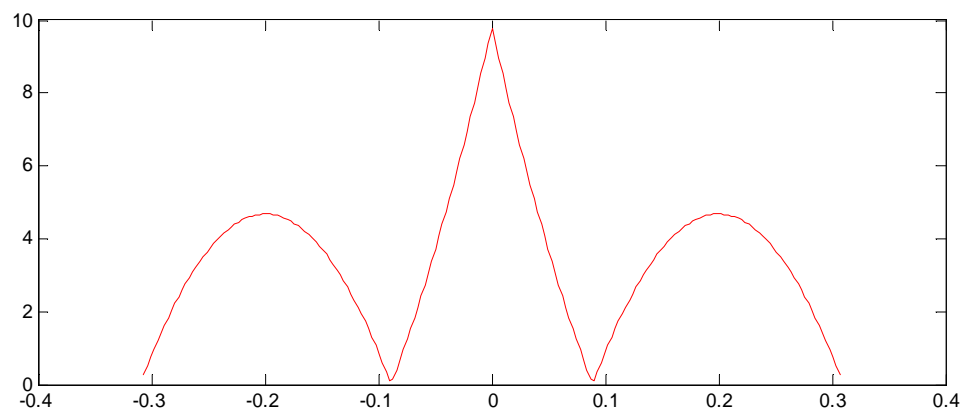
We zien, in contrast met voorgaande resultaten, dat de balk in dit volledige model pas breekt bij een puntlast van ongeveer 280 N. Wanneer de balk ingeklemd wordt kan deze dus hoger worden belast dan wanneer de balk niet ingeklemd wordt (vergelijk 28 kg *met* inklemming en 7 kg *zonder* inklemming). De kracht Q heft waarschijnlijk een deel van de puntlast P op, zodat er minder kracht voor de balk overblijft, zodat deze pas later breekt. Verder valt op, dat de balk met inklemming veel minder vervorming vertoont.

De balk breekt wanneer σ_{xx} te groot wordt (hier bij $P = 280$ N). Zoals we al eerder hebben gezien geldt:

$$\sigma_{xx} = |M| \frac{\text{hoogte}}{2I}$$

Waar breekt de balk?

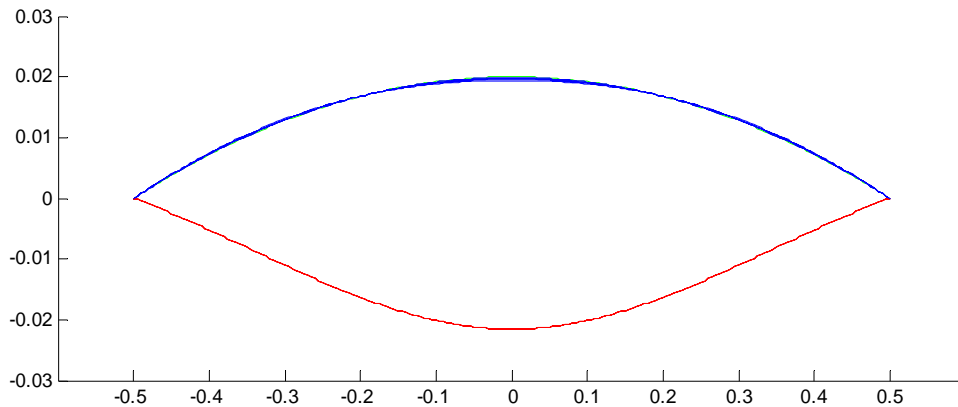
We willen nu het punt gaan bepalen waar de balk zal gaan breken. Dit zal het punt zijn waar het moment in de balk in absolute waarde het grootst is. We plotten dus de absolute waarde van het moment in de balk (voor $P = 280$ N, zodat de balk nog net niet gebroken is) om te bepalen waar deze maximaal is (zie *grafiek 4b* hieronder).



Grafiek 4b: Absolute waarde van het moment in de balk van het niet-lineaire probleem met inklemming voor $P=270$

We zien dat het moment in het midden van de lat, waar de puntlast werkt, een globaal maximum heeft, zodat de balk daar zal gaan breken bij nog hogere belasting. Verder zien we dat er nog een lokaal maximum is bij $\pm 0,2$ m. Het valt ons op dat bij deze twee maxima de uitwijking (vervorming) van de balk (onderhevig aan een puntlast P) ook maximaal is (zie *grafiek 4a*). Wellicht is het zo dat de balk altijd breekt op het punt waar de uitwijking van de neurale lijn (vervorming) maximaal is.

In de volgende plot hebben we de lengte ($\ell = 0,5 \text{ m}$) en de uitwijking ($U_0 = 0,02 \text{ m}$) zodanig aangepast zodat de lat doorklapt, voordat hij kan breken.



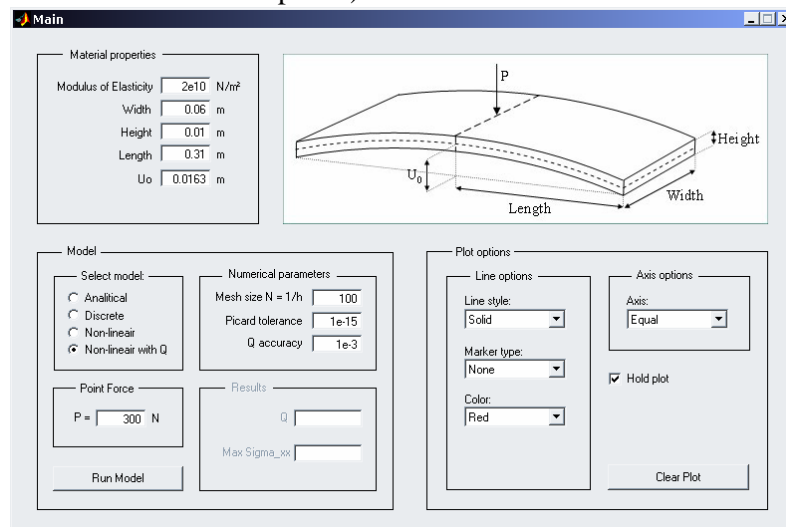
Grafiek 5: Discrete oplossing van het niet-lineaire probleem met inklemming voor $P=0:25:100,101$

P in N	Q in N	σ_{xx} in N/m^2
0	0	0
25	$2.44\text{e}+002$	$1.40\text{e}+006$
50	$4.89\text{e}+002$	$2.85\text{e}+006$
75	$7.35\text{e}+002$	$4.36\text{e}+006$
100	$9.81\text{e}+002$	$5.94\text{e}+006$
101	$9.79\text{e}+002$	$4.64\text{e}+007$

Tabel 5

We zien dat de balk bij 10,1 kg is doorgeklapt. Verder merken we op dat het probleem numeriek instabiel wordt bij een kritieke belasting van $\pm 100 \text{ N}$.

Verder hebben we nog een grafische gebruikersinterface gemaakt, zodat er makkelijk en snel resultaten kunnen worden verkregen voor verschillende instellingen. De code is niet opgenomen in de bijlagen, maar is wel te vinden op de bijgevoegde diskette. De interface is op te starten door in MATLAB het bestand `main_gui.m` te runnen (Matlab 6.5 voor Windows OS aanbevolen met een minimale resolutie van 1024×768 pixels).



Figuur 12: Voorbeeld van de grafische gebruikersinterface `main_gui.m`

4 Toetsing aan de werkelijkheid & Conclusie

De gegevens van onze balk kwamen overeen met een latje uit de slaapbank van Christiaan. De latjes in de slaapbank waren echter niet gebroken, maar doorgeklapt, nadat de gezette vriend op de bank was neergeploft. In ons model *braken* de latjes al voordat ze doorklapten (bij een belasting van 28 kg). Ons model komt dus niet overeen met de werkelijkheid. Dit komt onder andere doordat ons latje theoretisch ingeklemd is en de latjes in de slaapbank aan beide uiteinden met een stuk plastic in het frame waren gemonteerd, zodat deze niet volledig ingeklemd waren, maar nog speling hadden. Verder lag er natuurlijk nog een matras op de lattenbodem, zodat de krachten in realiteit niet als puntlasten op de middens van de latjes werkten. Ook bij de gekozen elasticiteitsmodulus van $2 \times 10^{10} \text{ N/m}^2$ (welke volgens BINAS overeenkomt met vurenhout) kunnen vraagtekens worden gezet. Wanneer we de houtsoort van de latjes uit de slaapbank nader bekijken, zien we namelijk dat het multiplex is (op elkaar gelijmde houtsoorten). De elasticiteitsmodulus hiervan is ons onbekend.

De algemene conclusie die we uit dit hele verhaal kunnen trekken is echter simpel:

Wanneer je een gezette vriend op je slaapbank laat neerploffen, kunnen er theoretisch gezien latjes van de lattenbodem breken (zie dit verslag), danwel latjes doorklappen (empirisch bepaald op Christiaan's feestje). In beide gevallen is je slaapbank kapot.



5 Bijlagen – MATLAB code

5.1 main.m

```
% Main program

clear % Clear Workspace
init % Initialize material properties

%***** Set-Up *****
method = 4; % Choose method:
            % 1 = Analytical      -EI(phi"(s)) = P/2
            % 2 = Discrete        -EI(phi"(s)) = P/2
            % 3 = Non Linearair   -EI(phi"(s)) = P/2*sin(phi(s))
            % 4 = Non Linearair with Q -EI(phi"(s)) = P/2*sin(phi(s)) - Q*cos(phi(s))
P_range = 0:50:300; % Specify P value(s) (Point Force)
n = 100; % Define number of steps: n
pic_tol = 1e-16; % Define the tolerance for speed of convergence for Picard
Q_tol = 1e-3; % Define Q accuracy (used in Qsearch)
%*****

h=L/(2*n); % Evaluate stepsize h
if (method < 4); Q=0; end; % Set Q=0 for problems 1,2,3

% Draw table to hold the results
disp(' ');
disp('      P      | Q      | Sigma_xx');
disp(' -----|-----|-----');

for P=P_range

    if (method==1); anal; end % Analytical
    if (method==2); discr; end % Discrete
    if (method==3); pic_Psin; end % Non Linearair
    if (method==4); Qsearch; end % Non Linearair with Q

    sigma_xx % Evaluate Sigma_xx
    disp( sprintf('      %4d | %1.2e | %1.2e', P, Q, s_xx ) ); % Print result
    plot_xy(x,y,'r') % Plot results
end
```

5.2 init.m

```
% Materiaal eigenschappen van de lat
E = 2e10; % Elasticiteitsmodulus
width = 0.06;
height = 0.01;
len = 0.31;
U0 = 0.0163;

R0 = (len^2+U0^2)/(2*U0); % Kromtestraal
L = 2*R0*asin(len/R0); % Booglengthe
%len = R0*cos(pi/2-L/(2*R0)) % Controle formule
I = 1/12*width*height^3; % Oppervlaktemoment
```

5.3 transf.m

```
% Transform (s,phi) -> (x,y) using the Extended Trapezoid Rule

% Transform (s,phi) -> x
x(1)=(h/2)*sin(u(1))+(h/2)*sin(u(2));
for i=2:n
    x(i,1)=x(i-1,1)+(h/2)*sin(u(i))+(h/2)*sin(u(i+1));
end

% Transform (s,phi) -> y
y(1)=(h/2)*cos(u(1))+(h/2)*cos(u(2));
for i=2:n
    y(i,1)=y(i-1,1)+(h/2)*cos(u(i))+(h/2)*cos(u(i+1));
end
```


5.4 anal.m

```
% Analytical solution of the DE:
% -EI(phi''(s)) = P/2

s=(0:h:L/2)';
phi=-P*s.^2/(4*E*I) + s/R0 + pi/2+P*L^2/(16*E*I)-L/(2*R0);

% Transform (s,phi) -> (x,y)
u=phi;
transf;
```

5.5 discr.m

```
% Solve the following DE:
% -EI(phi''(s)) = P/2

% Build matrix A
A=zeros(n,n+2);
for i=2:n+1
    A(i-1, i-1:i+1)=[1/h^2, -2/h^2, 1/h^2];
end
A(:, [1,n+2])=[];
A(1,1)=-1/h^2;
A=sparse(A);

% Build rhs vector f
f=-P/(2*E*I)*ones(n,1);
f([1,n])=[-P/(4*E*I)+1/(R0*h), -P/(2*E*I)-pi/(2*h^2)];

% Solve the system Au=f
u=A\f;

% Transform (s,phi) -> (x,y)
u=[u; pi/2];
transf;
```

5.6 pic Psin.m

```
% Solve the following DE (non linear) with Picard Iterations
% -EI(phi''(s)) = P/2*sin(phi(s))

discr % Get the matrix A and startsolution vector u from the discrete problem

u(n+1)=[]; % Remove the previously added pi/2 (dimension purpose)

k=0; % Count the number of Picard iterations and start with zero
red=1;

% Picard iterations
while (red > pic_tol) & (k < 10000)

    % Build rhs vector f
    f=-P/(2*E*I)*ones(n,1).*sin(u);
    f(1)=-P/(4*E*I)*sin(u(1))+1/(R0*h);
    f(n)=-P/(2*E*I)*sin(u(n))-pi/(2*h^2);

    % Solve the system Au=f
    u_old=u;
    u=A\f;

    k=k+1; % Count one Picard iteration

    red = norm( u - u_old , inf );
end

% Display a warning when # picard iterations exceeds limit
if k==10000; disp('WARNING: # Picard iterations exceeds limit !!!'); end

% Transform (s,phi) -> (x,y)
u=[u; pi/2];
transf;
```

5.7 Qsearch.m

```
% Find an appropriate Q (Pseudo Binary Search Method) by solving the DE:
%  $-EI(\phi''(s)) = P/2\sin(\phi(s)) - Q\cos(\phi(s))$  (Solved by pic_Qcos)
% such that the boundary condition  $x(n)=1$  is met

discr % Get the matrix A and startsolution vector u from the discrete problem

Q = 0;
if P ~= 0
    Qstep = 100; % Define starting stepsize for Binary Search Method
    while Qstep > Q_tol % Iterate until defined precision Q_tol is met
        if x(n) > len
            Q = Q + Qstep;
            pic_Qcos % Solve DE with Picard iterations for this Q
        elseif x(n) <= len
            Qstep = Qstep / 2;
            Q = Q - Qstep;
            pic_Qcos % Solve DE with Picard iterations for this Q
        end
    end
else
    pic_Qcos % Solve DE with Picard iteration for Q=0
end
```

5.8 pic Qcos.m

```
% Solve the following DE (non linear) with Picard Iterations for a given Q
%  $-EI(\phi''(s)) = P/2\sin(\phi(s)) - Q\cos(\phi(s))$ 

u(n+1)=[]; % Remove the previously added pi/2 (dimension purpose)

k=0; % Count the number of Picard iterations and start with zero
red=1;

% Picard iterations
while (red > pic_tol) & (k < 10000)

    % Build rhs vector f
    f=-P/(2*E*I)*ones(n,1).*sin(u) + Q/(E*I)*ones(n,1).*cos(u);
    f(1)=-P/(4*E*I)*sin(u(1))+Q/(2*E*I)*cos(u(1))+1/(R0*h);
    f(n)=-P/(2*E*I)*sin(u(n))+Q/(E*I)*cos(u(n))-pi/(2*h^2);

    % Solve the system Au=f
    u_old=u;
    u=A\f;

    k=k+1; % Count one Picard iteration

    red = norm( u - u_old , inf );
end

% Display a warning when # picard iterations exceeds limit
if k==10000; disp('WARNING: # Picard iterations exceeds limit !!!'); end

% Transform (s,phi) -> (x,y)
u=[u; pi/2];
transf
```

5.9 sigma_xx

```
% Evaluate Sigma_xx

% Evaluate the momentum (two methods)
% Controle: M1 = E*I*( 1/R0 - ( u(3:101)-u(1:99) ) / (2*h) );
M2 = -Q*y + P/2*x;

% Evaluate Sigma_xx (for each method)
% Controle: s_xx_1 = abs(M1) * height / (2*I);
s_xx_2 = abs(M2) * height / (2*I);

% Take largest value for Sigma_xx
s_xx = max(s_xx_2);
```

5.10 plot_xy

```
% Plot (x,y) including starting point (0,0)
function r = plot(x,y,c)

% Define Figure screen position
set(0,'Units','pixels')
scnsize = get(0,'ScreenSize');
fwidth = 660;
fheight = 300;
pos_lt = [5, scnsize(4) - fheight - 70, fwidth, fheight];

% Create Figure at defined position with certain axis
figure(1)
set( 1, 'Position', pos_lt );
%axis( 1.2*[-x(end) x(end) -y(end) y(end)] );
%axis equal
hold on;

% Plot the data x and y
plot( [ [0;x]-x(end) , -[0;x]+x(end) ], [ [0;y] , [0;y] ], c);
```

6 Symbolenlijst

<i>Symbol</i>	<i>Beschrijving</i>	<i>Eenheid</i>
E	Elasticiteitsmodulus	N/m^2
I	Oppervlaktemoment	m^4
L	Booglengte	m
ℓ	Lengte	m
U_0	Beginuitwijking	m
P	Kracht (Puntlast)	N
Q	Kracht	N
h	Stapgroote	m
φ	Hoek	rad
s	Booglengte	m
σ_{xx}	Spanningstensor	N/m^2

7 Literatuurlijst

[NMD] Numerieke methoden voor differentiaalvergelijkingen van C. Vuik

[MII] Mechanica van Continue Media II