

Optimalisatie van vakwerkbogen met 3Dtruss

***Bachelor Eindproject CTB3000
Juni 2015***

***Naam: Bart Hoogendoorn
Studienummer: 4161904
bart-hoogendoorn@hotmail.com***

Voorwoord

Geachte lezer,

Dit verslag is een Bachelor Eindproject ter afronding van de studie Civiele Techniek aan de TU Delft. Het betreft een onderzoek naar een aanbeveling voor een initiële keuze tussen twee verschillende boogconstructies met behulp van verschillende programma's. Ik heb veel plezier gehad in het leren werken met deze programma's.

Tijdens dit onderzoek heb ik veel geleerd en ik wil graag mijn begeleiders J.W. Welleman en P.C.J. Hoogenboom bedanken voor hun kostbare tijd en inzet voor dit verslag.

Bart Hoogendoorn,
Delft, Juni 2015.

Samenvatting

De dakconstructie van een hal kan op vele manieren gerealiseerd worden; bijvoorbeeld uit samengestelde liggers, uit massieve boogspanten of als ruimtevakwerk. In dit onderzoek worden twee typen dakconstructie beschouwd. Het eerste type is een ruimtevakwerk wat is gebogen in één richting en bestaat uit twee lagen gekoppeld door diagonaalstaven. Dit type wordt aangeduid als geïntegreerd vakwerk. Het tweede type bestaat vakwerkbogen die worden overspannen door liggers met een I-profiel. Deze vakwerkbogen hebben een driehoekige doorsnede. Dit type wordt aangeduid als hoofddraagsysteem.

Het doel van dit onderzoek is vaststellen voor welk type dakconstructie (geïntegreerd vakwerk of hoofddraagsysteem) het minste materiaal nodig is.

Aangenomen zijn een representatieve dakbelasting en een windbelasting. De staafkrachten zijn berekend met het elementenprogramma 3DTruss. Er zijn twee profieldoorsneden ontworpen; die voor de boogstaven en die voor de verbindingstaven. De profielen zijn gekozen op basis van vloeispanning en knik. Op basis van de profielkeuzen is het eigengewicht van de constructie berekend.

Dit ontwerpproces is geautomatiseerd met een hiertoe speciaal geschreven programma in de programmeertaal Python. Dit programma genereert een invoerbestand, start 3DTruss en leest het uitvoerbestand. Het programma herhaalt de berekening voor de volgende variabelen; hart-op-hart-afstand van twee bogen, het grootte van het cirkelsegment, de overspanning en de dikte van de constructie afhankelijk van de openingshoek. Beide constructietypen zijn afhankelijk van deze variabelen.

De belangrijkste conclusie van het onderzoek is dat, ongeacht de waarden van de variabelen, het benodigd gewicht van een hoofddraagsysteem minder is dan het benodigd gewicht van een geïntegreerd vakwerk.

Voor een optimale boogconstructie moet het cirkelsegment tussen de 1,3 en 1,9 rad zijn en is de dikte van het gebruikte vakwerk ongeveer 1/20 van de overspanning. Wanneer hiervan wordt afgeweken, kan dit leiden tot een aanzienlijke toename in het materiaalgebruik.

Inhoudsopgave

Voorwoord	ii
Samenvatting.....	iii
Inleiding	2
Inhoud	4
1. Opbouw van de constructie	4
1.1. Boogconstructie	4
1.2. Vakwerk	5
2. Te gebruiken boogvorm	6
2.1. Vorm	6
2.2. Differentiaalvergelijkingen	7
3. Boogconstructies.....	11
3.1. Boogconstructies met hoofddraagsysteem.....	11
3.2. Boogconstructies met geïntegreerd vakwerk.....	12
4. Krachten	13
4.1. Profielkeuze voor hoofddraagsysteem	13
4.2. Windkracht:	14
5. Programma's	16
5.1. 3DTruss	16
5.2. Python	18
6. Profielkeuze in boogstructuren.....	22
7. Variabelen.....	23
7.1. Hart op hart afstand	23
7.2. Boogstraal	25
7.3. Cirkelsegment	27
7.4. Dikte constructie.....	28
Conclusie	30
Aanbeveling	31
Literatuurlijst	32
Bijlagen	33
Bijlage 1: Maple bestand:	33
Bijlage 2: Technical information 3DTruss:.....	34
Bijlage 3: Python programma:	35

Inleiding

Aanleiding

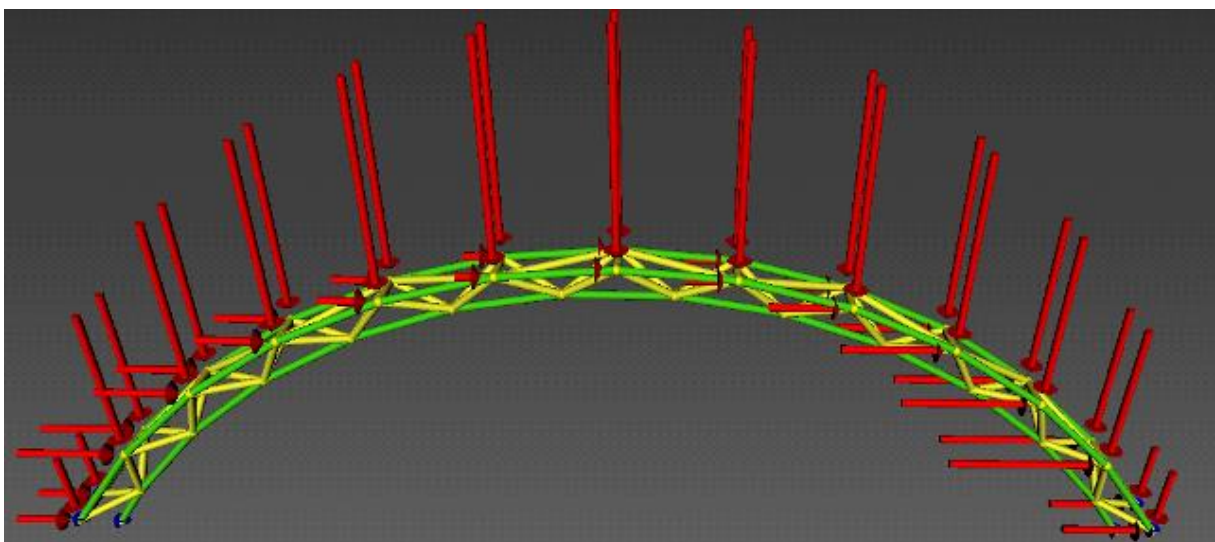
Bij het ontwerpen van constructies worden verschillende stappen gemaakt. De eerste stap is het dimensioneren op basis van vuistregels. Hoe hoog een ligger moet zijn om een bepaalde overspanning te kunnen overbruggen of hoe breed een balk moet zijn, zijn voorbeelden hiervan.

Voor liggers en kolommen is het makkelijk om deze vuistregels te vinden. Wanneer de constructie echter afwijkt van een opbouw met kolommen en liggers wordt het moeilijker om vuistregels te gebruiken.

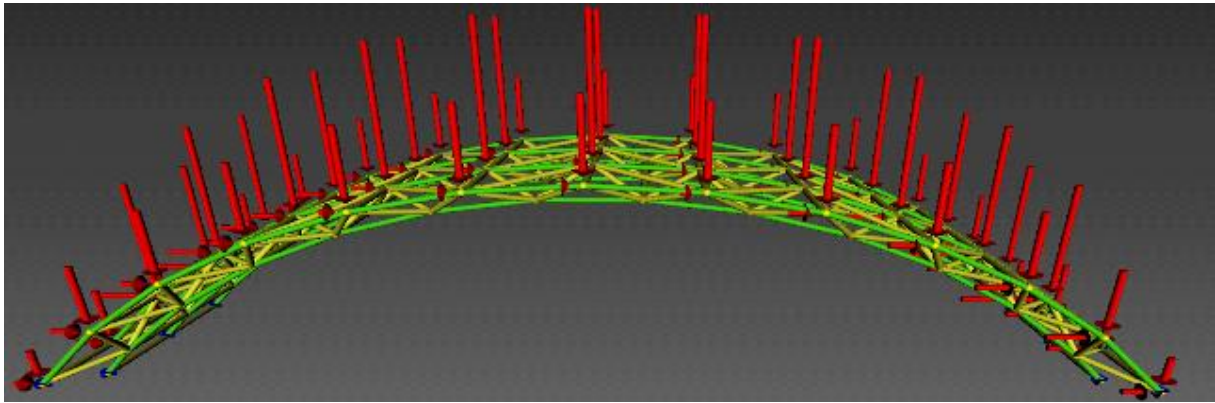
Een voorbeeld van een standaard constructie waarvoor geen vuistregels zijn, zijn boogconstructies. Boogconstructies worden al meer dan 2000 jaar gebruikt. Constructies zijn in de loop der tijd ook steeds ingewikkelder geworden door bijvoorbeeld het combineren van verschillende krachtenafdrachten. Deze veranderingen zorgen ervoor dat de constructies groter gemaakt kunnen worden en grotere krachten aan kunnen. Hoewel alle constructies tegenwoordig gemakkelijk berekend kunnen worden met behulp van computerprogramma's, die werken met de eindige elementen methode of verplaatsingenmethode, moet wel eerst de constructie bepaald worden die in de computer gezet wordt. In een voorafgaand onderzoek van dhr. Bloemers is een vakwerkboog volledig doorgerekend, een soortgelijke vakwerkboog zal gebruikt worden en vergeleken met een andere constructietype.

Doel

Het doel van dit bachelor eind project is twee verschillende boogconstructies met elkaar te vergelijken en hieruit een overgangspunt te bepalen dat aangeeft welk van de twee constructies goedkoper gedimensioneerd kan worden bij verschillende situaties. Deze constructies zijn een hoofddraagsysteem en een geïntegreerd vakwerk, weergegeven in respectievelijk figuur 1 en 2..



Figuur 1: Voorbeeld van een vakwerkboog van een hoofddraagsysteem



Figuur 2: Voorbeeld van een geïntegreerd vakwerk

Om de verschillende constructies te vergelijken wordt gebruik gemaakt van een programma genaamd 3DTruss. Dit programma gebruikt de verplaatsingen methode om staafkrachten en verplaatsingen te bepalen. Met behulp van Python wordt een bestand gecreëerd dat de eigenschappen van de constructie weergeeft. De constructies worden vergeleken met verschillende variabelen. De belangrijkste variabele is lengte loodrecht op het vlak van de boogconstructie, in Python aangegeven als *length*. Er wordt ook gekeken naar de invloed van de boogstraal, aangegeven als *R*. Het cirkelsegment wordt aangegeven als *alpha* en de vakwerkdikte op basis van afdraaghoek wordt aangegeven als *phi*.

Structuurbeschrijving

In dit rapport is de volgende structuur aangehouden. In hoofdstuk 1 en 2 wordt een beschrijving gegeven van hoe een boogconstructie werkt en wat de effecten hiervan zijn. In hoofdstuk 3 wordt bepaald welke constructietypes gebruikt worden. In hoofdstuk 4 wordt beschreven welke krachten er op de constructie aangenomen worden en de berekening van deze krachten. In hoofdstuk 5 staat een uitleg van het gemaakte Python programma en de werking van 3DTruss. In hoofdstuk 6 staat de berekening van de profielkeuze in het vakwerk. En vanaf hoofdstuk 7 staan de resultaten voor verschillende variabelen.

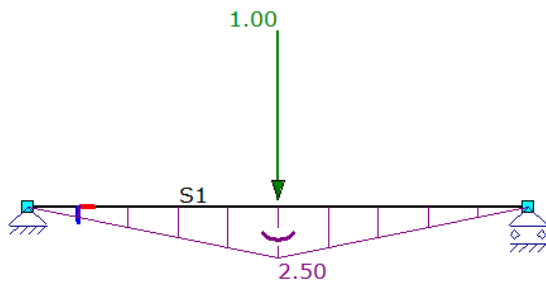
Inhoud

1. Opbouw van de constructie

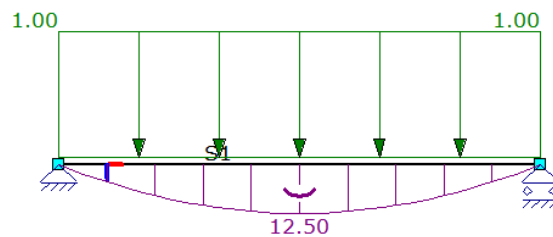
Alle constructies die behandeld worden in dit verslag bestaan uit twee complementerende constructies, een boogconstructie en een vakwerk. Om te begrijpen waarom een combinatie van deze constructies gebruikt wordt zal de krachtenafdracht van beide constructies worden uitgelegd in dit hoofdstuk. Beide constructies zijn gebaseerd op het principe om geen momenten in de constructie te creëren.

1.1. Boogconstructie

Wanneer krachten op een ligger uitgeoefend worden ontstaan er momenten. Twee verschillende momentenlijnen zijn in Figuur 3 en 4 zichtbaar.

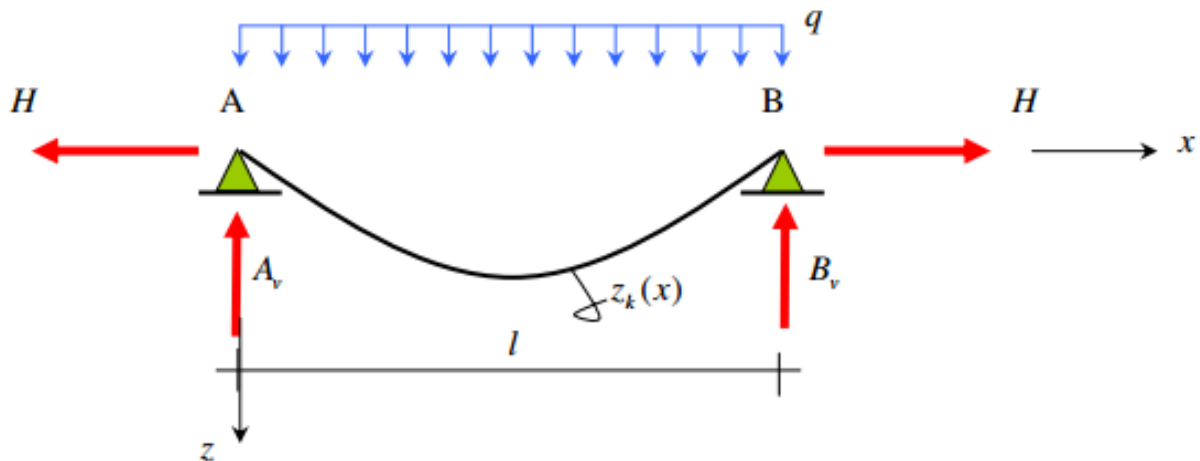


Figuur 3: Momentenlijn van een puntlast



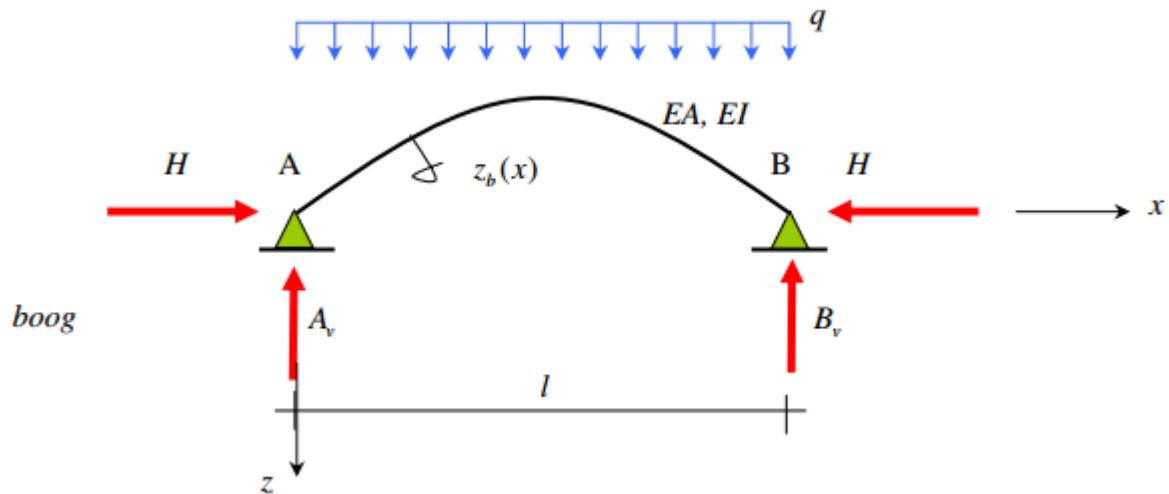
Figuur 4: Momentenlijn van een q-last

Als de ligger vervangen wordt door een kabel zal de kabel zich gaan vervormen naar de momentenlijn van de ligger. Zoals in figuur 5.



Figuur 5: Vormlijn van een kabel bij een q-last

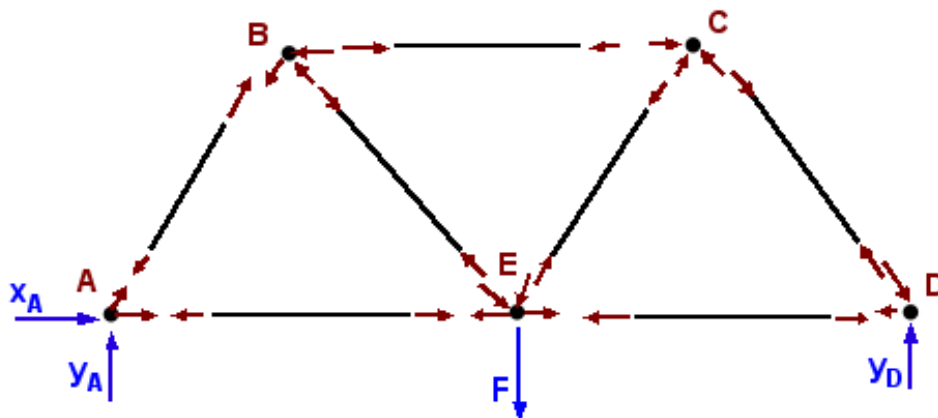
In figuur 6 is een boogconstructie zichtbaar. Dit is een starre constructie in omgekeerde vorm van de kabel in figuur 5. Doordat een boogconstructie de omgekeerde vorm van een momentenlijn heeft zullen er geen momenten ontstaan bij een gelijke belastingcombinatie. Wanneer deze vorm afwijkt, bijvoorbeeld door de vorm of de belastingssituatie te veranderen, is het mogelijk dat er momenten ontstaan.



Figuur 6: Boogvorm q-last.

1.2. Vakwerk

Een vakwerk zorgt ervoor dat er grotere krachten opgenomen kunnen worden. Het is zo opgebouwd dat de krachten alleen werken op knooppunten. Er zullen in dit geval geen momenten ontstaan en de kracht zal gesplitst worden en afgedragen worden via trek- en drukstaven. De staven hoeven dus alleen op normaalkrachten gedimensioneerd te worden. Hierdoor kunnen er grotere overspanningen overbrugd worden. Een voorbeeld van een vakwerk en de krachten in een vakwerk is in figuur 7 weergegeven. Elk knooppunt is in evenwicht met alle trek- en drukstaven en de externe krachten.



Figuur 7: Krachten in een vakwerk.

Deze twee constructietypes vullen elkaar aan zodat er geen momenten ontstaan, en de constructie met weinig materiaal gemaakt kan worden. Hoe de constructietypes in elkaar verwerkt worden zal in hoofdstuk 3. Boogconstructies beschreven worden.

2. Te gebruiken boogvorm

Zoals eerder beschreven is, zal de optimale vorm voor een boogconstructie gelijk zijn aan de momentenlijn van een ligger met dezelfde overspanning. Dit verandert echter met een andere belastingssituatie. Om voor elke situatie een nieuwe constructie te creëren is erg lastig. Aangezien er naast de boogvorm ook een vakwerk aanwezig is hoeft de constructie niet de precieze vorm te hebben van de momentenlijn. Een benadering van de momentenlijn zal voldoende zijn. Voor het opstellen van de constructie zal daarom gebruik gemaakt worden van een cirkelboog.

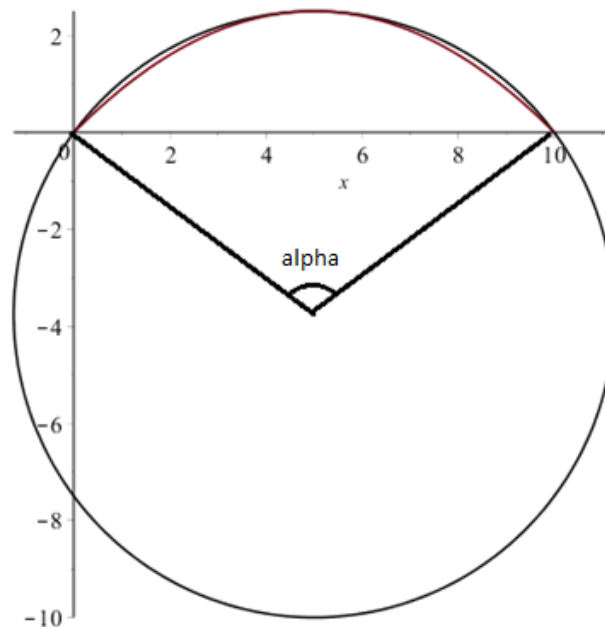
Er is voor een cirkelboog gekozen omdat het makkelijk is deze vorm te verdelen in gelijkwaardige lengtes. Hierdoor zal elk deel van het vakwerk hetzelfde zijn in dimensies.

Omdat besloten is een vorm te gebruiken die afwijkt van de momentenlijn zal gekeken worden wat het effect hiervan is.

2.1. Vorm

De eerste vergelijking is op basis van vorm en momentenlijn.

Voor de benadering van een cirkelboog met een momentenlijn is voor de belastingcombinatie een q-last gebruikt. De cirkelboog is gelijk gesteld op 3 punten: linker oplegging, rechter oplegging en top van de momentenlijn. Het resultaat is in figuur 8 weergegeven.



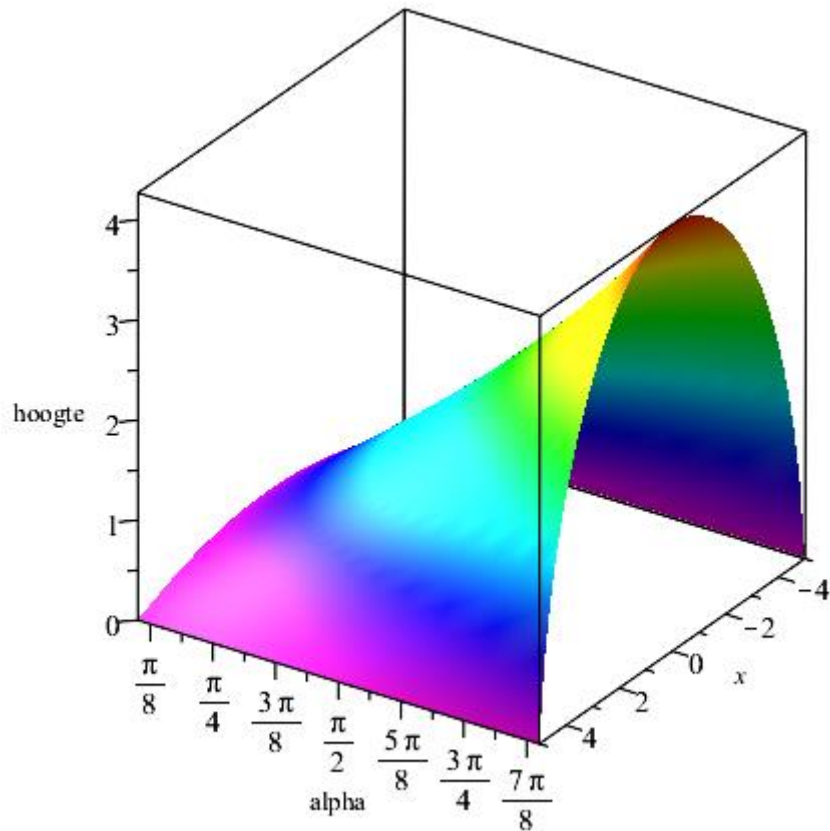
Figuur 8: Benadering van een cirkel met momentenlijn

In figuur 8 is te zien dat een cirkelvorm zeer geschikt is om een parabolische momentenlijn te benaderen. Met afwijking op ong. 15 en 85% van de overspanning. De positie van het midden van de cirkel is niet meer op grondniveau maar lager, en de cirkelboog zal geen halve cirkel meer zijn maar een kleiner gedeelte beschreven met de variabele α .

2.2. Differentiaalvergelijkingen

De tweede vergelijking is op basis van differentiaalvergelijkingen.

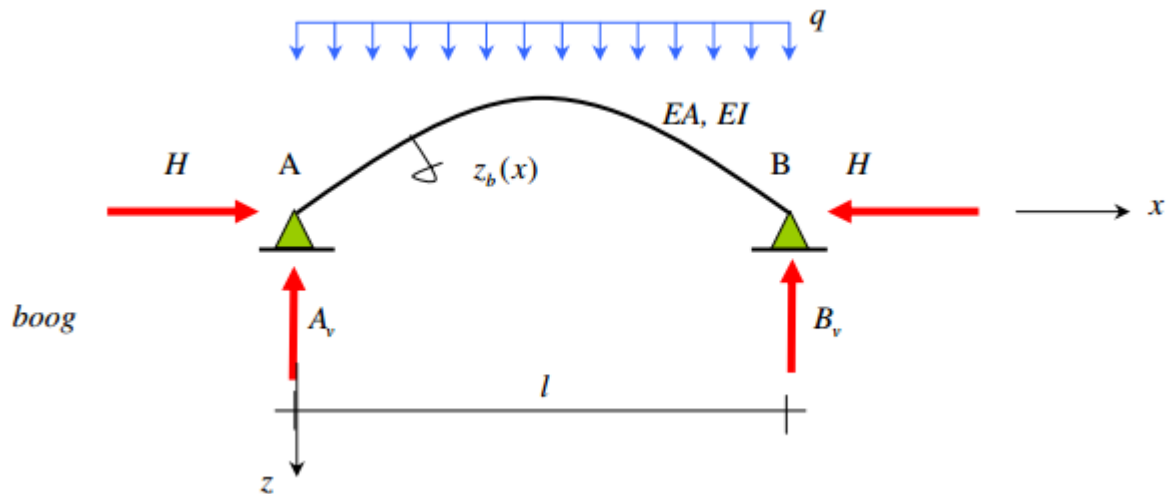
Figuur 9 representeert een 3d-plot voor een overspanninglengte van 10 meter waarbij de hoogte uitgezet is tegen de lengte en cirkelsegment (alpha).



Figuur 9: 3D plot van een boogvorm met variabele alpha

De vergelijking van deze plot zal worden uitgewerkt met differentiaalvergelijkingen in Maple. Het Maple bestand is toegevoegd in bijlage 1. De resultaten zullen per cirkelsegment gepresenteerd worden, omdat Maple dit niet algebraïsch op kan lossen voor alle variabelen. Een cirkelsegment is een gedeelte van een cirkel in rad, zoals in figuur 8 is weergegeven als alpha.

Voor deze berekeningen is aangenomen dat de overspanning 10 meter is en er een q-last op werkt. Wanneer het cirkelsegment aangepast wordt, zal alles weer geschaald worden naar een overspanning van 10 meter.



Figuur 10: Boogvorm met krachtenwerking.

Voor een vorm die afwijkt van een standaard ligger zal de differentiaalvergelijking aangepast worden naar de vorm. Voor de differentiaalvergelijkingen zijn de krachten en opleggingen aangenomen als in figuur 10. De vorm van de constructie zal afwijken en cirkelvormig zijn. De differentiaalvergelijking is dan:

$$EI \frac{d^4 w}{dx^4} + H \frac{d^2 z}{dx^2} = q$$

waarbij z de functie is van de hoogte.

$$z = \sqrt{\frac{R^2}{\sin(0.5 * angle)^2} - x^2} - \frac{R * \cos(0.5 * angle)}{\sin(0.5 * angle)}$$

Dit is een formule van een cirkel aangepast met een schaling van R naar $R/\sin(0.5 * \alpha)$, dit zorgt ervoor dat de overspanning constant blijft voor een variabele α . De $\cos(0.5 * \alpha)$ zorgt ervoor dat de formule op het begin en eindpunt op de x-as zijn.

In het programma van 3DTruss zullen de opleggingen scharnierende, niet verplaatsbare opleggingen zijn. Dit zal hier ook aangenomen worden.

Voor $x=-R$, $w=0$ en $M=0$

Voor $x=R$, $w=0$ en $M=0$

Als resultaat van het oplossen van de differentiaalvergelijking blijkt dat H lineair afhankelijk is van q. Voor q wordt een waarde aangenomen, in dit geval 10 kN/m. Deze waarde wordt verwerkt in de formules van w, M, V en N.

Met een bekende H kan de w berekend worden uit de differentiaalvergelijking, w is berekend met behulp van de solve functie.

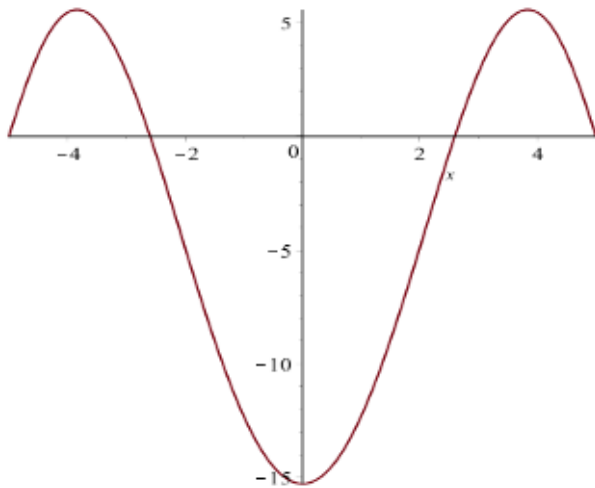
$$M = EI \frac{d^2 w}{dx^2}$$

Met standaard differentiaalvergelijkingen worden de N en V in een globaal assenstelsel bepaald, (x en z richting). Voor de werkelijke krachten moeten deze omgezet worden met behulp van de aanwezige hoek van de constructie

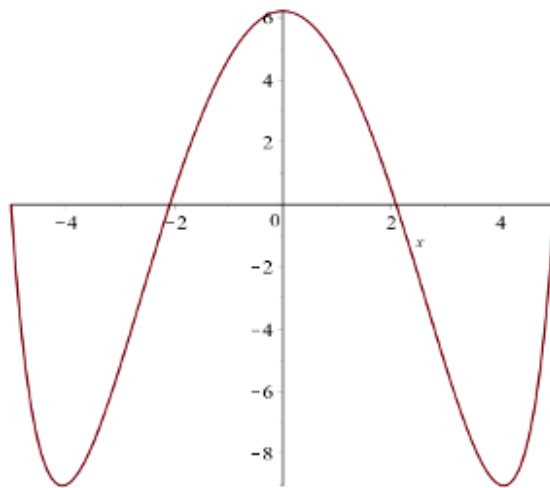
$$N = -H \cos(\beta) + \left(-EI * \frac{d^3w}{dx^3} - H \frac{dz}{dx} \right) * \sin(\beta)$$

$$\beta = \arctan\left(\frac{dz}{dx}\right)$$

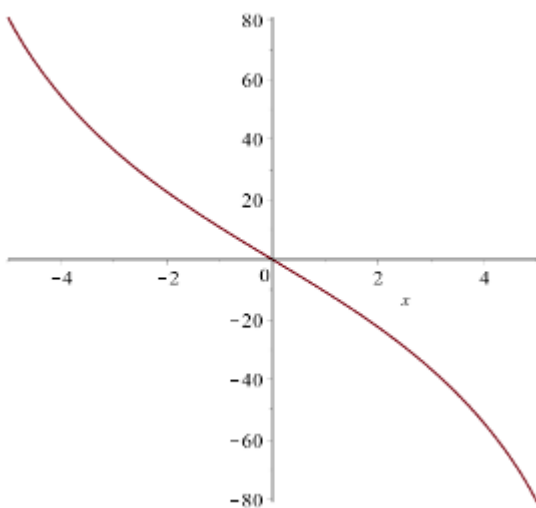
$$V = -H \sin(\beta) + \left(-EI * \frac{d^3w}{dx^3} - H \frac{dz}{dx} \right) * \cos(\beta)$$



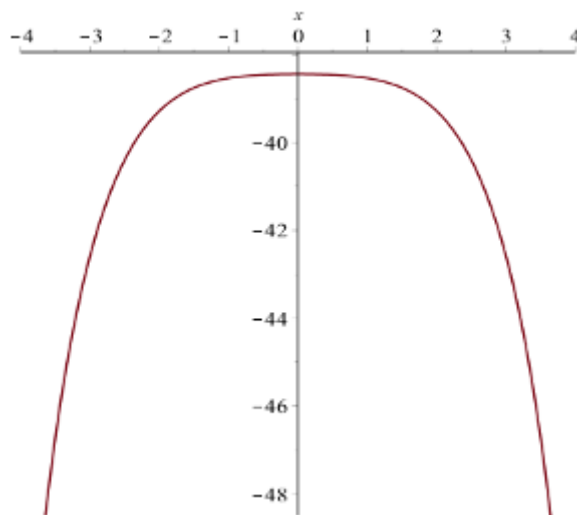
Figuur 11: Momentenlijn bij alpha is 0.5 pi.



Figuur 12: Zakkingslijn bij alpha is 0.5 pi



Figuur 13: Normaalkrachtenlijn bij alpha is 0.5 pi.



Figuur 14: Dwarskrachtenlijn bij alpha is 0.5 pi.

Figuur 11 tot en met 14 zijn berekend met een alpha van 0,5*pi en een q van 10. In figuur 12 is de maximale waarde van het moment ongeveer 1,5 keer hoger dan de waarde op x is 0.

De waarden van verschillende kritieke punten zijn in tabel 1 uiteengezet voor verschillende waarde van “alpha”.

Tabel 1: Uitkomsten bij verschillende waarden van alpha

	Z (0) m	w (0)*	M (0) kNm	V (+- R) kN	N (+- R) kN
0.1*Pi	0.39350854	0.1858	0.1104179	99.5605300	321.3724465
0.2*Pi	0.79192221	0.76449	0.4447908	98.2557251	165.7173892
0.3*Pi	1.200393795	1.80976	1.0128228	96.1260085	117.1168742
0.4*Pi	1.624598482	3.45599	1.8315179	93.2374767	96.6996390
0.5*Pi	2.071067812	5.92590	2.9266720	89.6799848	89.6799848
0.6*Pi	2.547627247	9.57403	4.3353357	85.5644820	92.8689156
0.7*Pi	3.064003941	14.96276	6.109809	81.019601	108.772349
0.8*Pi	3.632712643	22.99342	8.324210	76.187505	150.39478
0.9*Pi	4.270403428	35.1380	11.08560	71.21896	287.4839

* zakking w is afhankelijk van een onbekend aangenomen EI, daarom zijn de waarden alleen onderling relevant, niet als rekenwaarde.

Voor een balk met dezelfde overspanning zal het moment een maximum hebben van $1/8 \cdot q \cdot l^2 = 125 \text{ kNm}$. De grootste waarde op 0,9 pi zal ongeveer 16 kNm zijn, dit is minder dan 15%

Duidelijk is dat er rond 0,5 pi een optimum is van normaalkracht. Normaalkracht is het belangrijkste in vakwerken. Er zal echter met 0,6 pi verder gerekend worden omdat dit een grotere hoogte heeft zonder veel meer normaalkracht.

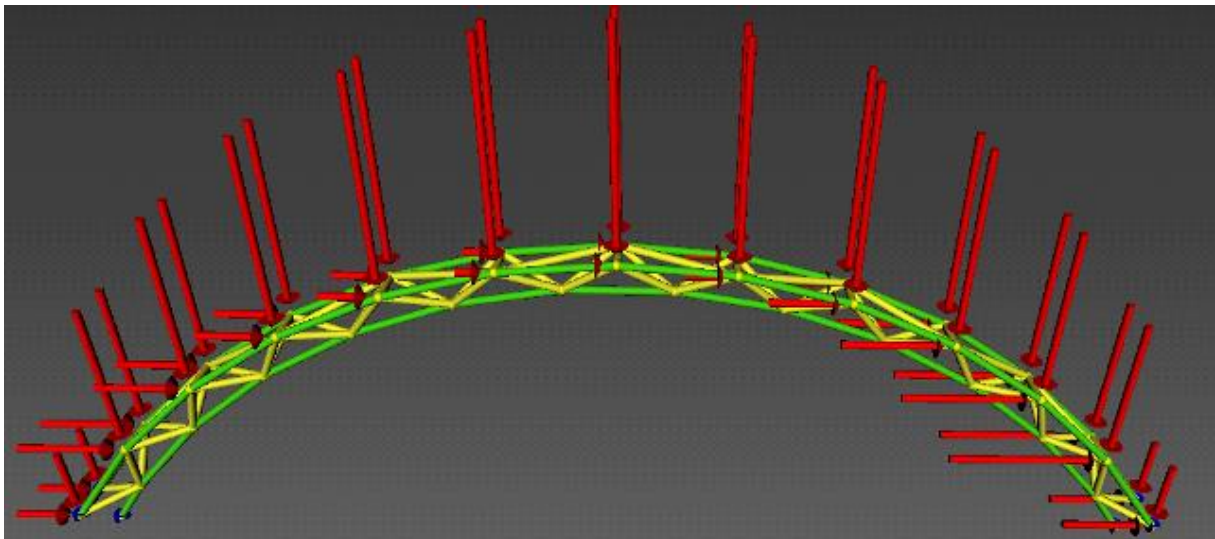
3. Boogconstructies

Twee verschillende constructies zullen met elkaar vergeleken worden. Beide constructies maken gebruik van een vakwerkboog. De eerste constructie is een vakwerkboog met hoofddraagsysteem. De tweede constructie is een geïntegreerd vakwerk. Beide systemen zijn ontstaan uit dezelfde berekeningen. Dit komt er op neer dat beide constructies gelijkvormig zijn met gelijke waarden voor alle aangenomen variabelen.

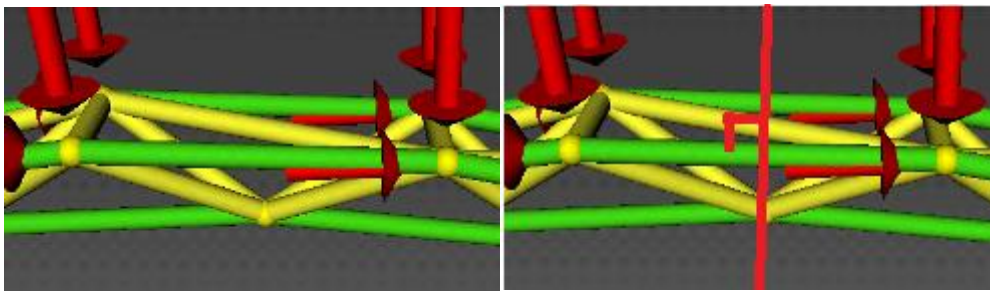
3.1. Boogconstructies met hoofddraagsysteem

Zoals vermeld bij de differentiaal vergelijkingen zijn de opleggingen plaats vaste, scharnierende verbindingen

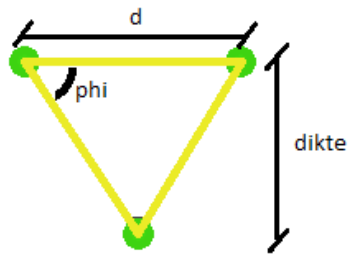
Voor hoofddraagsysteem, afgekort hds, zal er gebruik gemaakt worden van een constructie waarbij de buitenste assen een gelijke afstand hebben ten opzichte van elkaar. Deze zullen verbonden worden met elkaar door middel van staven die dan een gelijkzijdige driehoek creëren in een vlak die door het middelpunt van de cirkel gaat. De dakbedekking zal afgedragen worden door middel van een I-profiel liggers die tussen twee boogconstructies geplaatst worden. In figuur 15 is een vakwerkboog weergegeven en in figuur 16 een close up hiervan.



Figuur 15: Vakwerkboog met krachten.



Figuur 16: Segment van een vakwerkboog, zonder en met aanduiding van loodrechte doorsnede

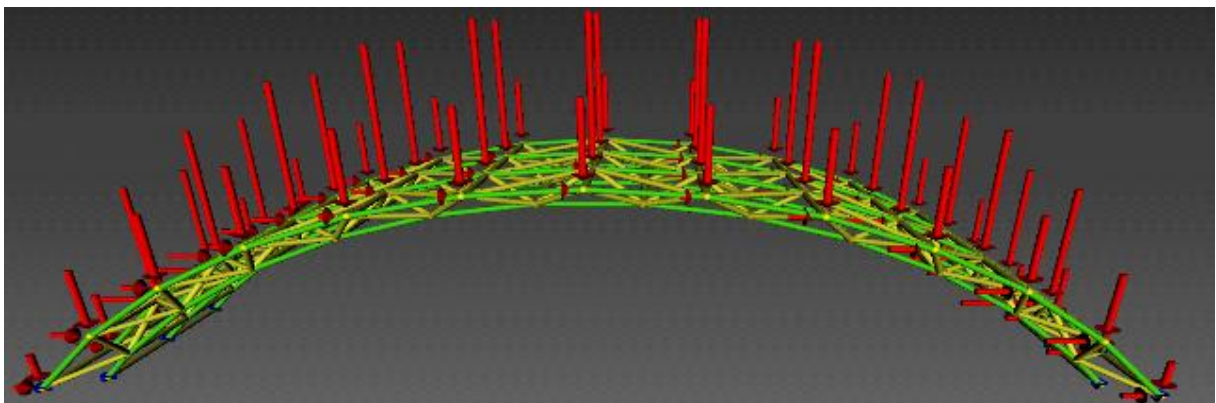


Figuur 17: Doorsnede van een vakwerkboog

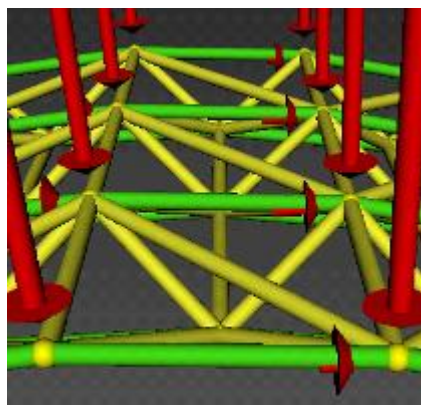
De dikte van de constructie is bepaald met behulp van de variabele ϕ . Voor een aanduiding van ϕ wordt een loodrechte doorsnede op de vakwerkboog genomen zoals aangegeven in figuur 16. ϕ is dan de hoek, in deze doorsnede (figuur 17), die de verbindingssstaaf tussen binnen- en buitenboog maakt met de verbindingssstaaf tussen de twee buitenbogen.

3.2. Boogconstructies met geïntegreerd vakwerk

Het geïntegreerd vakwerk, afgekort tot int, werkt op dezelfde manier als het hoofddraagsysteem met het verschil dat er geen gebruik gemaakt wordt van liggers, maar het vakwerk over de gehele lengte van de constructie doorgevoerd wordt. Een voorbeeld hiervan is in figuur 18 weergegeven en een close up in figuur 19.



Figuur 18: Geïntegreerd vakwerk met krachten.



Figuur 19: Segment van een geïntegreerd vakwerk.

4. Krachten

4.1. Profielkeuze voor hoofddraagstelsysteem

Bij het hoofddraagstelsysteem worden twee boogstructuren met elkaar verbonden door middel van een I-profiel ligger. Deze ligger wordt gedimensioneerd op twee eisen. Een eis op doorbuiging en een eis op sterkte. De berekeningen van beide eisen zullen uitgelegd worden in dit hoofdstuk.

De eerste check is een eis op sterkte, de volgende formule is omgeschreven naar onbekende I_{zz} en h .

$$\frac{I_{zz}}{h} = \frac{0.5 * M}{f_y}$$

Hierbij is M het maximale moment zoals berekend wordt bij een q -last op een ligger op twee steunpunten.

$$M = \frac{1}{8} q_{tot,UGT} * L^2$$

De ligger moet voor sterkte op uiterste grenstoestand getest worden. Hierbij worden de variabele belasting en de permanente belasting vermenigvuldigd met respectievelijk factoren 1,5 en 1,2.

$$q_{tot,UGT} = 1.5 * q_{var} + 1.2 * q_{per}$$

Wanneer de I_{zz}/h berekend is, wordt deze vergeleken met omgezette waarden uit tabel SD8 in quick reference. De respectievelijke eigenschappen van het profiel met ten minste deze waarde worden toegekend aan o.a. I_{zz} en eigen gewicht. Als waarden I_{zz1} en $eg1$.

De eis op doorbuiging is een bruikbaarheids grenstoestand, dit betekent dat de factoren van de variabele belasting en de permanente belasting beide 1,0 zijn dus:

$$q_{tot,BGT} = 1.0 * q_{var} + 1.0 * q_{per}$$

De eis van doorbuiging is:

$$\frac{5 * q_{tot,BGT} * L^4}{384 * EI} \leq 0.004 * L$$

omgeschreven is dit:

$$I_{zz} = \frac{5 * q_{tot,BGT} * L^3}{0.004 * 384 * E}$$

Deze waarde wordt ook omgezet naar een I-profiel. En de waarden I_{zz2} en $eg2$ worden toegekend. De grootste waarde van I_{zz1} en I_{zz2} is de bepalende factor en zal de rekenwaarde zijn voor de ligger.

4.2. Windkracht:

Voor gebouwen moet rekening gehouden worden met windkracht. In dit hoofdstuk wordt deze berekening toegelicht. Hiervoor wordt de volgende formule gebruikt:

$$F_t = c_s c_d * c_f * q_p(z_e) * A_{ref}$$

Voor gebouwen onder de 15m kan voor $c_s c_d$ 1 aangenomen worden.

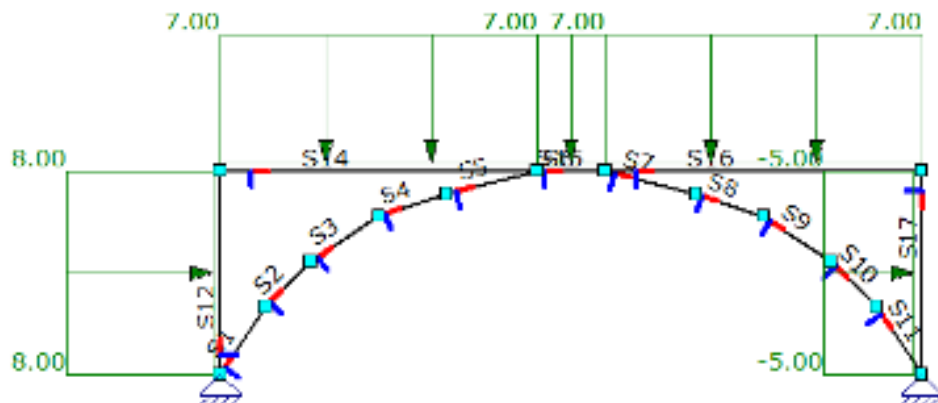
Eerst worden de spanningen bepaald. $\text{Sigma} = F/A$. Wanneer de kracht per knooppunt bepaald wordt, zal daarbij het respectievelijke oppervlak bepaald worden.

De c_f is afhankelijk van de afmetingen van het gebouw. De afmetingen waarmee gerekend wordt zijn:

- h: maximale hoogte: afhankelijk van cirkelsegment (alpha) en R. Deze waarde zal verder worden aangenomen als maximale waarde R
- d: lengte in windrichting: is gelijk aan $2 \cdot R$
- b: horizontale lengte loodrecht op windrichting: onbepaald. De waarde is equivalent aan oneindig
- e: minimum van b of $2 \cdot h$ zal gelijk zijn aan $2 \cdot h = 2 \cdot R$

De c_f is ook afhankelijk van de hoek die het gebouw maakt met de wind/grond, voor een boogvormige constructie is dit een constant veranderende waarde. De kracht zal vereenvoudigd worden en de meest ongunstige situatie zal worden gebruikt.

Omdat een boogconstructie zeer gevoelig is voor asymmetrische belastingen zal de meest ongunstige belasting zijn wanneer de constructie voor wind beschouwd wordt als een balk met dezelfde dimensies als hierboven beschreven is, weergegeven in figuur 20. Dit resulteert ook in twee type krachten: horizontaal en verticaal. Dit is zeer geschikt voor 3DTruss omdat dit programma alleen met globale richtingen werkt..



Figuur 20: Representatieve windbelasting.

De c_f zal dan een waarde krijgen van +0,8 in de wind en -0,5 uit de wind (gl 7 quick reference) voor horizontale krachten. En een waarde van -1,0/+0,7 in verticale richting. Er zijn dus twee verschillende belastingcombinaties waarvan de meest ongunstige gekozen wordt. De belastingcombinatie met -1,0 kan worden opgeheven door het eigen gewicht van de constructie. Deze waarden zijn ongeveer gelijk. Als de windkracht groter is dan het eigen gewicht zullen de staven deze kracht aankunnen.

Als laatste de q_p is een geografische eigenschap: ook hier wordt de meest ongunstige waarde gebruikt, deze bevindt zich in area 1 coastal (Noord-Holland) en is afhankelijk van de hoogte van de constructie.

Om consequent te blijven met voorgaande berekeningen zal er voor de hoogte van deze waarde ook R gebruikt worden.

Met bovenstaande gegevens zal de constructie met een constante overspanning aan gelijke windspanningen onderhevig zijn. Deze worden omgezet naar een kracht die afhankelijk is van A_{ref} .

5. Programma's

5.1. 3DTruss

Voor dit onderzoek wordt gebruik gemaakt van een programma genaamd 3DTruss. 3DTruss rekent een constructie door op basis van de verplaatsingen methode. Een beschrijving van de gebruikte methode staat vermeld in het programma zelf onder info, de Engelse versie is bijgevoegd in Bijlage 2. Dit is de technical information en gaat als volgt:

Elk element kan worden beschreven door twee punten A (a_x, a_y, a_z) en B (b_x, b_y, b_z) en heeft 6 vrijheidsgraden. Deze punten zijn ook weergegeven in figuur 21.

$$u_i^T = (u_x^A, u_y^A, u_z^A, u_x^B, u_y^B, u_z^B)$$

Een element kan aangeduid worden door middel van de volgende vector.

$$l = \frac{1}{L} \begin{pmatrix} b_x - a_x \\ b_y - a_y \\ b_z - a_z \end{pmatrix} = \begin{bmatrix} l_x \\ l_y \\ l_z \end{bmatrix}$$

Waarvan de lengte L berekend kan worden met

$$L = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2}$$

De verlenging van de balk wordt beschreven met:

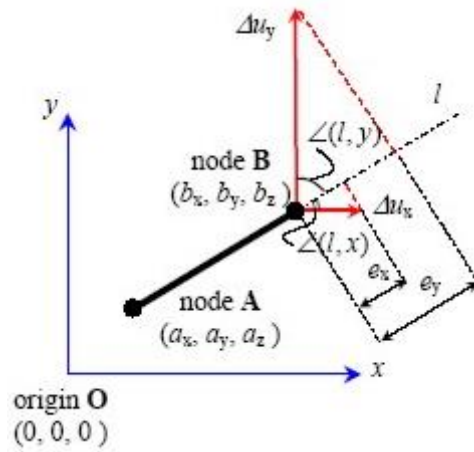
$$e_i = B_i * u_i$$

Met B is de kinematische relatie:

$$B_i = [-l_x \quad -l_y \quad -l_z \quad l_x \quad l_y \quad l_z]$$

Elk element past de stijfheid van het systeem aan volgens de volgende formule::

$$K_i = B_i^T \frac{EA_i}{L_i} B_i$$



Figuur 21: Werking verplaatsingenmethode.

$$e_i = e_x + e_y + e_z$$

met

$$\begin{aligned} e_x &= \cos \angle(l, x) * (u_x^B - u_x^A) & \cos \angle(l, x) &= l_x \\ e_y &= \cos \angle(l, y) * (u_y^B - u_y^A) & \cos \angle(l, y) &= l_y \\ e_z &= \cos \angle(l, z) * (u_z^B - u_z^A) & \cos \angle(l, z) &= l_z \end{aligned}$$

De stijfheidmatrix kan met randvoorwaarden opgelost worden en met behulp van:

$$f = K * u$$

kunnen de verplaatsingen bepaald worden.

Met deze verplaatsingen kunnen de krachten in de elementen berekend worden volgens de volgende formule:

$$N_i = B_i^T \frac{EA_i}{L_i} u_i$$

5.2. Python

Met behulp van Python worden er bestanden gemaakt die doorgerekend worden door 3DTruss. Vervolgens worden deze gegevens uitgelezen door hetzelfde Python programma en verwerkt in een overzichtelijke grafiek.

In grove lijnen is de werking van de Python programma beschreven in een stroomschema, weergegeven in figuur 22. Het bestand zelf staat in bijlage 3. Door de werking van Python staan de programma's die aangevraagd worden bovenaan en de berekeningen aan het einde.

Het programma is voor verschillende variabelen aanpasbaar, deze zijn:

```
datacreate(trusnum,trusswidth,fy,R,alpha,nm,d,phi,lengthhds,lengthint,XX+YY)
```

trusnum	Het aantal delen van de boogconstructie
trusswidth	Is voor geïntegreerd vakwerk is het aantal boogconstructies in de lengterichting
fy	is de vloeigrens van staal
R	Een halve overspanningslengte
alpha	Het cirkelsegment, gedeelte van een cirkel gebruikt voor de boogconstructie.
Nm	De krachtafdracht van hoofddraagsysteem om de zoveel knooppunten er een ligger is.
d	De breedte van een boogconstructie of bij geïntegreerd vakwerk de onderlinge afstand tussen twee boogconstructies.
Phi	De hoek die de verbindingsstaven maken, 60 graden is een gelijkzijdige driehoek bij hds
length	Gelijkwaardige variabele voor lengte voor respectievelijk hds en int.
XX+YY	Een variabele voor benoeming van bestandsnaam.

De veranderende variabelen zijn length (int,hds), R, alpha en phi. Een van deze variabelen wordt per loop veranderd en voor de rest wordt een waarde aangenomen.

Vervolgens wordt met createts een trs bestand aangemaakt. Dit bestand is als volgt opgebouwd: eerst de algemene gegevens. Hieronder vallen het maximum aantal gegevens per heading onder maxlength. Een heading is aangeduid met [...].

En staafeigenschappen zoals oppervlak van de doorsnede en elasticiteitsmodulus. Deze waarden worden aangenomen wanneer ze niet gespecificeerd zijn.

```
[settings]
maxlength=200
[properties]
E=210000000
A=1
```

Per knooppunt wordt met goniometrische formules bepaald wat de waarde van x en z zijn. Deze waarden worden in het bestand geschreven per knooppunt. Er wordt onderscheid gemaakt in de buitenste en binnenste bogen. De buitenste bogen worden geplaatst zodanig dat de knooppunten onderling een gelijke afstand hebben en de boog verdeeld is in "trusnum" delen. Deze bogen worden bepaald met behulp

van een aangepaste R die ervoor zorgt dat de overspanning constant blijft. De binnenste boog heeft een aangepaste RR, dit is gelijk aan de waarde van R – dikte van de constructie. De knooppunten van de binnenste boog zijn een halve hoek gedraaid zodat de knooppunten zich halverwege twee buitenste knooppunten bevinden. Er zijn hier begin- en eindpunten aan toegevoegd, deze zorgen ervoor dat beide bogen eenzelfde cirkelsegment bestrijken.

[coordinates]

1=11.180@0.000@-3.633

Het @ is een scheidingsteken tussen verschillende waarden en heeft verder geen betekenis, waarbij X@Y@Z coördinaten toegekend worden aan een knooppuntnummer.

Om een element te definiëren worden twee knooppunten aangegeven, een beginpunt en een eindpunt die verbonden moeten worden door dit element. Ook worden hier de staafeigenschappen aan het element toegekend.

[elements]

1=1@2@1

De waarden zijn respectievelijk beginpunt, eindpunt en barproperties, door een @ gescheiden.

Het is niet mogelijk om alle staven in een keer te definiëren. Er is voor verschillende type staven een aparte loop gemaakt. Voor de boogstaven met barproperties = 1, zit er een onderbreking in om te voorkomen dat twee punten aan verschillende kanten van de constructie verbonden worden.

De verbindingssstaven met barproperties = 2, genoemd dwarsstaven, kruisstaven en diagonaalstaven.

De dwarsstaven verbinden twee parallelle bogen. De kruisstaven verbinden de buitenste bogen met de binnenboog. En de diagonaalstaven maken een kruisverband tussen de twee buitenste bogen.

Barproperties definiëren de eigenschappen van de staven.

Er wordt gebruik gemaakt van 2 typen staven: boogstaven en verbindingssstaven.

Beide krijgen de minimale oppervlak van een rond profiel.

[barproperties]

1=210000000@0.000238

2=210000000@0.000238

Op de constructie staan twee type belastingen: een verticale dakbelasting en windbelasting. De windbelasting bestaat uit een horizontale kracht en een verticale kracht. De krachten worden per richting aan een knooppunt toegekend. Hiervoor worden de verticale krachten bij elkaar opgeteld.

[loads]

1=1@x@0.333841005224

De gegevens zijn respectievelijk knooppunt, richting en grootte van de kracht gescheiden door @.

Als laatste moet gedefinieerd worden waar de opleggingen geplaatst zijn. Deze bevinden zich op elk uiteinde van een boog. De opleggingen zijn niet verplaatsbare, scharnierende opleggingen. Voor elke richting apart moet worden gedefinieerd dat deze niet verplaatsbaar is.

[supports]

1=1@x

Dit is af te lezen als: knooppunt 1 kan niet verplaatst worden in richting x.

Nadat alle gegevens in het bestand staan wordt de batch functie van 3DTruss gebruikt om berekeningen uit te voeren. Zowel de normaalkrachten als de knooppuntverplaatsingen worden berekend en verwerkt in het bestand.

[displacements]

1=1@x@0

Dit staat voor knooppunt 1 is in x richting 0 m verplaatst

De krachten die in een element zitten staan onder:

[elementforces]

1=-4.9661

Staaft 1 ondervindt een normaalkracht met respectievelijke grootte in kN.

Als laatste zijn de oplegreacties bepaald.

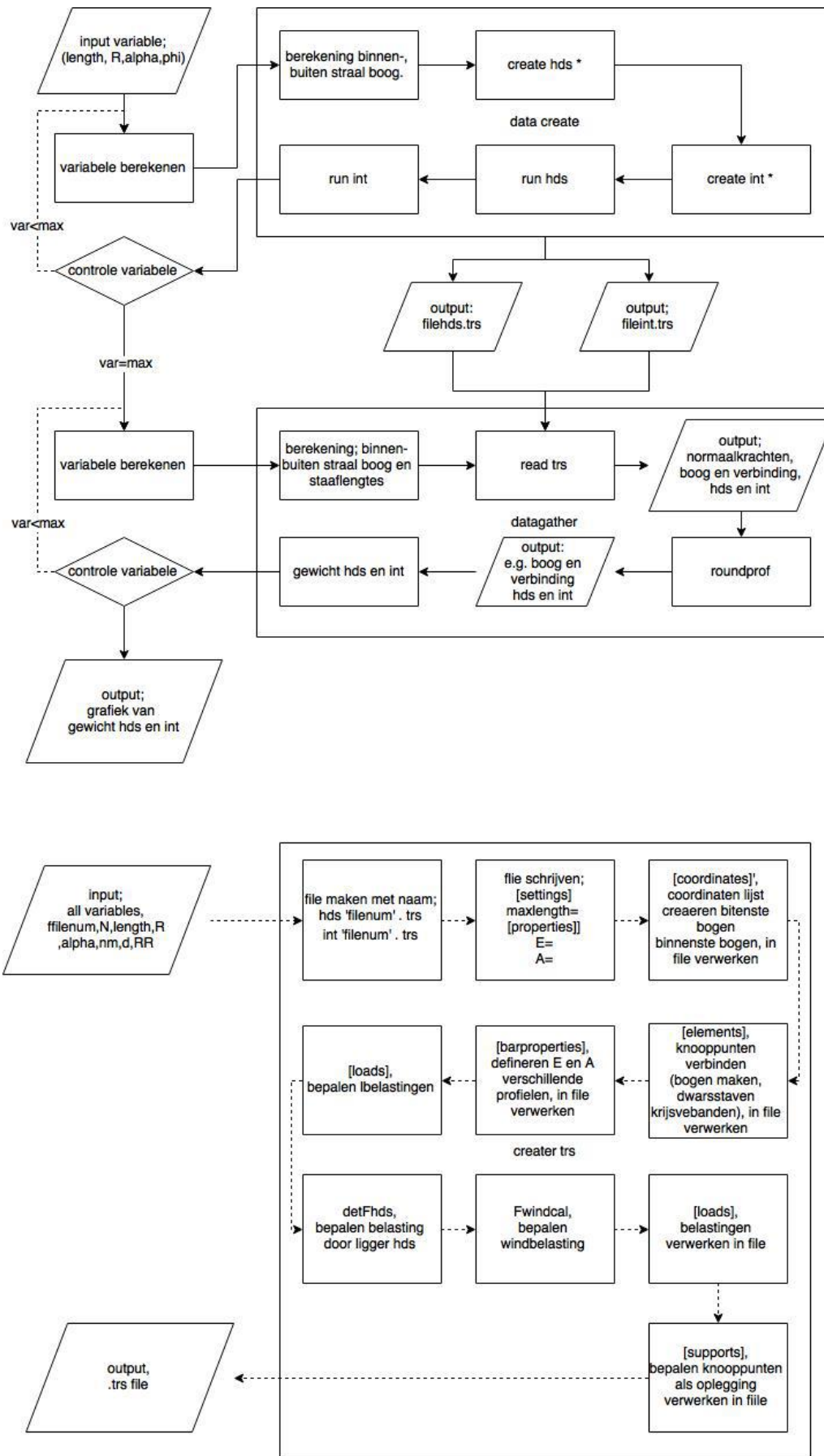
[reactions]

1=1@x@-5.7192215

De gegevens staan voor respectievelijk element, richting en grootte van kracht.

De elementforces worden uitgelezen en opgedeeld in 2 groepen: de boogstaven en de verbindingstaven. De grootste drukkrachten worden ingevoerd in de functie roundprof waarmee een profielkeuze gemaakt wordt op basis van knikkracht.

Met deze profieleigenschappen wordt de gebruikte massa berekend per lengte-eenheid die gebruikt is voor deze specifieke constructie.



Figuur 22: Flowchart van Python programma

6. Profielkeuze in boogstructuren

Zowel de boogstaven als de dwarsstaven ondervinden alleen normaalkrachten.

Voor beide typen wordt apart een profiel bepaald, afhankelijk van de grootst aanwezige normaalkracht in deze staven. Deze normaalkracht wordt vergeleken met maximale normaalkracht op basis van knik. Dit is een iteratief proces waarbij verschillende profielen in toenemende grootte worden vergeleken met de aanwezige normaalkracht totdat een profiel gevonden is die deze kracht aankan.

De berekening van knikkracht gaat volgens de onderstaande formules:

$$\lambda_1 = \pi * \sqrt{\frac{E}{f_y}}$$

$$\lambda_s = \frac{L_{cr}}{i} * \frac{1}{\lambda_1}$$

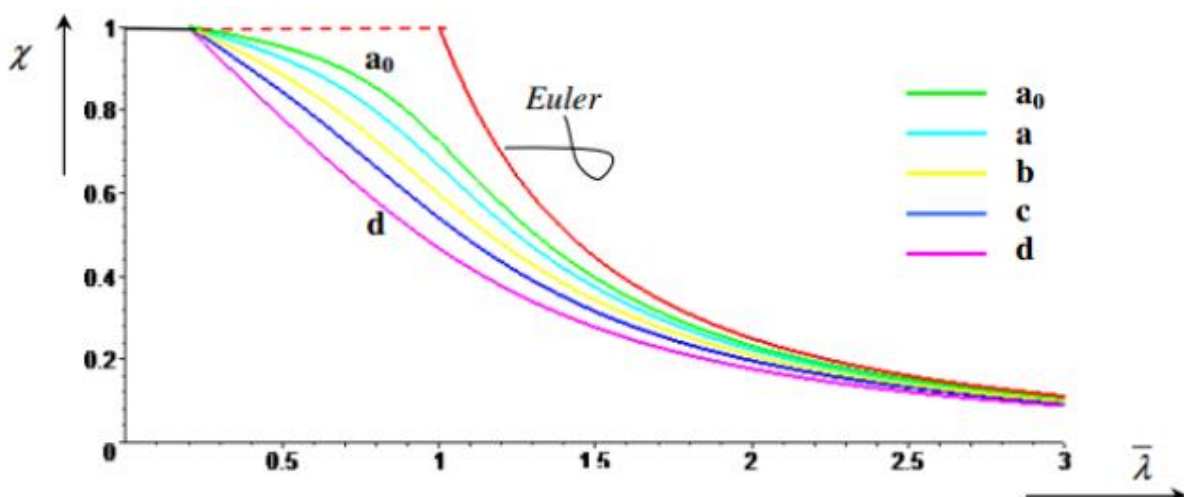
$$\varphi = 0.5 * [1 + \alpha(\lambda_s - 0.2) + \lambda_s^2]$$

$$\chi = \frac{1}{\varphi + \sqrt{\varphi^2 - \lambda_s^2}} < 1.0$$

$$N_{b,rd} = \frac{\chi * A * f_y}{\gamma_{ml}}$$

Uit tabel eurocode 3 is de waarde gehaald voor een rond, koud gewalst profiel. Hiervoor geldt de waarde $\alpha=0,21$, dit is in figuur 23 lijn a.

Voor de rest van de waarden wordt aangenomen:
 $f_y=355 \text{ N/mm}^2$ en $\gamma_{ml}=1,15$



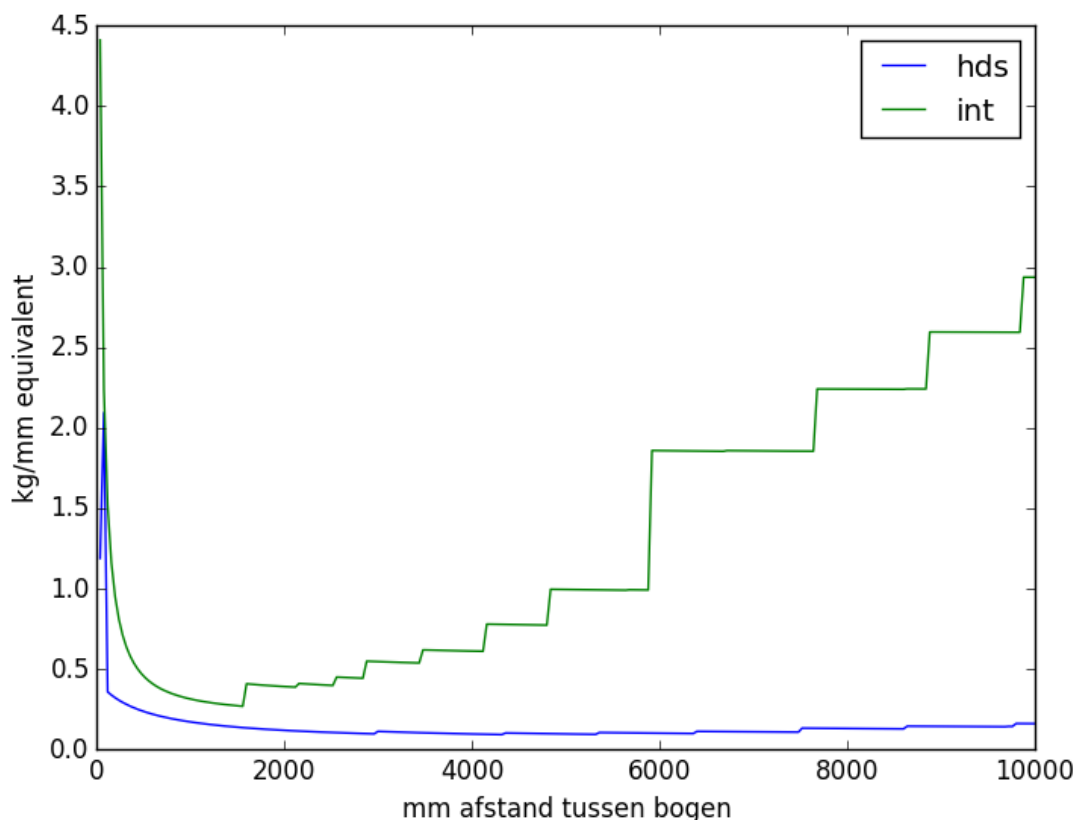
Figuur 23: Slankheid

7. Variabelen

7.1. Hart op hart afstand

Zoals eerder vermeld zijn er 4 variabelen onderzocht, length (int, hds), R, alpha en phi. De eerste variabele is lengte, de hart op hart afstand tussen twee boogconstructies. Het effect van deze variabele voor het hoofddraagsysteem is dat de ligger gedimensioneerd moet worden voor een grotere lengte en bijbehorende krachten. En voor een geïntegreerd vakwerk zullen de staven langer worden en het dakoppervlak dat afgedragen wordt per knooppunt wordt groter.

De berekeningen zijn uitgevoerd met 250 metingen



Figuur 24: Materiaalgebruik per afstand over h.o.h. afstanden.

Voor de bovenstaande grafiek zijn voor de andere variabelen waarden aangenomen van:

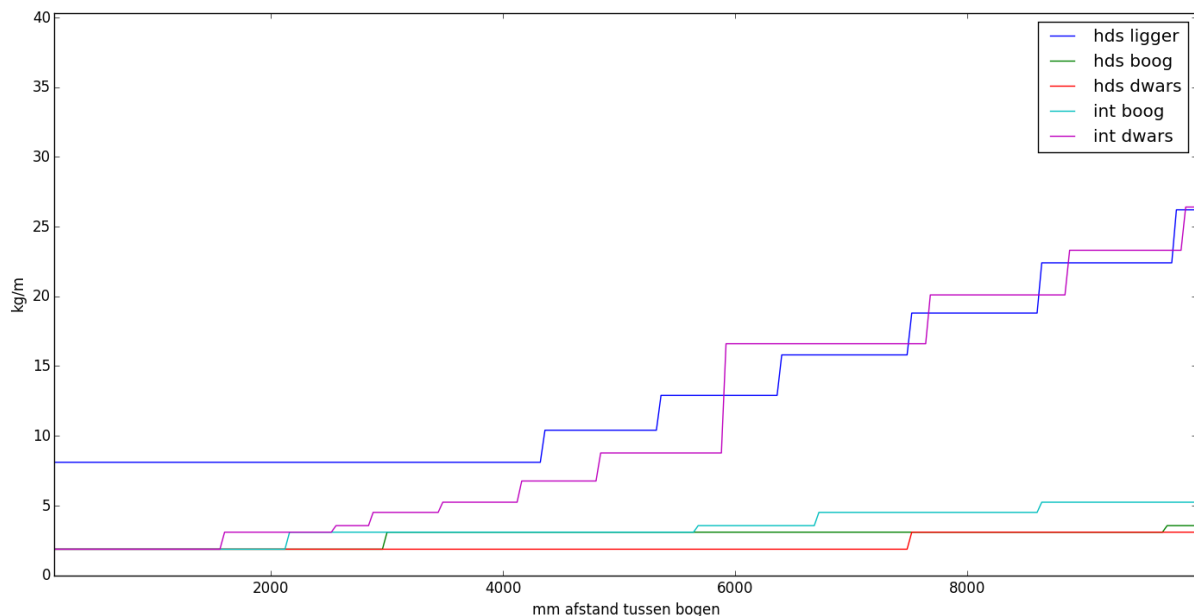
$R=5000\text{mm}$, $\alpha=0,6\text{ pi}$, $\phi=60\text{ graden}$ en $d=0,1 R$

Wat opvalt aan figuur 24 is vooral de stapsgewijze toename. Dit wordt veroorzaakt door een grotere profielkeuze om de krachten aan te kunnen. Aan het begin van de grafiek zijn de waardes erg groot doordat het profiel al minimaal is en de bogen steeds dichter op elkaar komen te staan. Bij het hoofddraagsysteem overlappen de boogconstructies.

De verwachtingen waren dat beide lijnen elkaar zouden kruisen. Echter blijkt dit niet het geval te zijn voor de aangenomen situatie. Mogelijk zijn de aangenomen waarden van andere variabelen ongunstig.

Wat wel afgelezen kan worden uit de grafiek zijn de optima voor beide constructies. Het optimum van het aangenomen geïntegreerd vakwerk is bij ongeveer 1500 mm lengte. Terwijl het optimum van het hoofddraagsysteem bij ongeveer 5500 mm is.

De hds waarden onder d zijn ongeschikt als meting omdat hierdoor een overlapping van de bogen van het hoofddraagsysteem plaatsvindt.



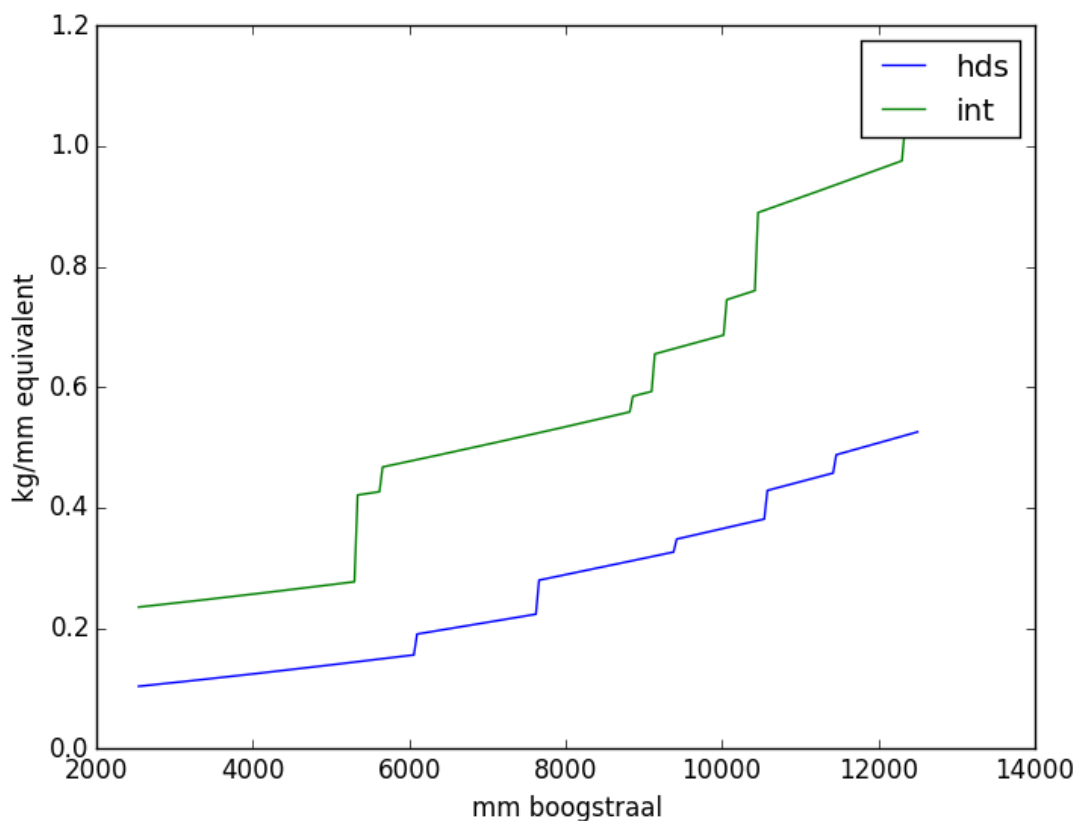
Figuur 25: Gewicht van profielen per staatype over h.o.h. afstanden

Figuur 25 laat de individuele profielkeuzes zien bij verschillende hart op hart afstanden.

Voor de volgende variabelen zal een lengte van 1500 mm aangenomen worden, dit is gekozen als geschikte waarde uit figuur 24.

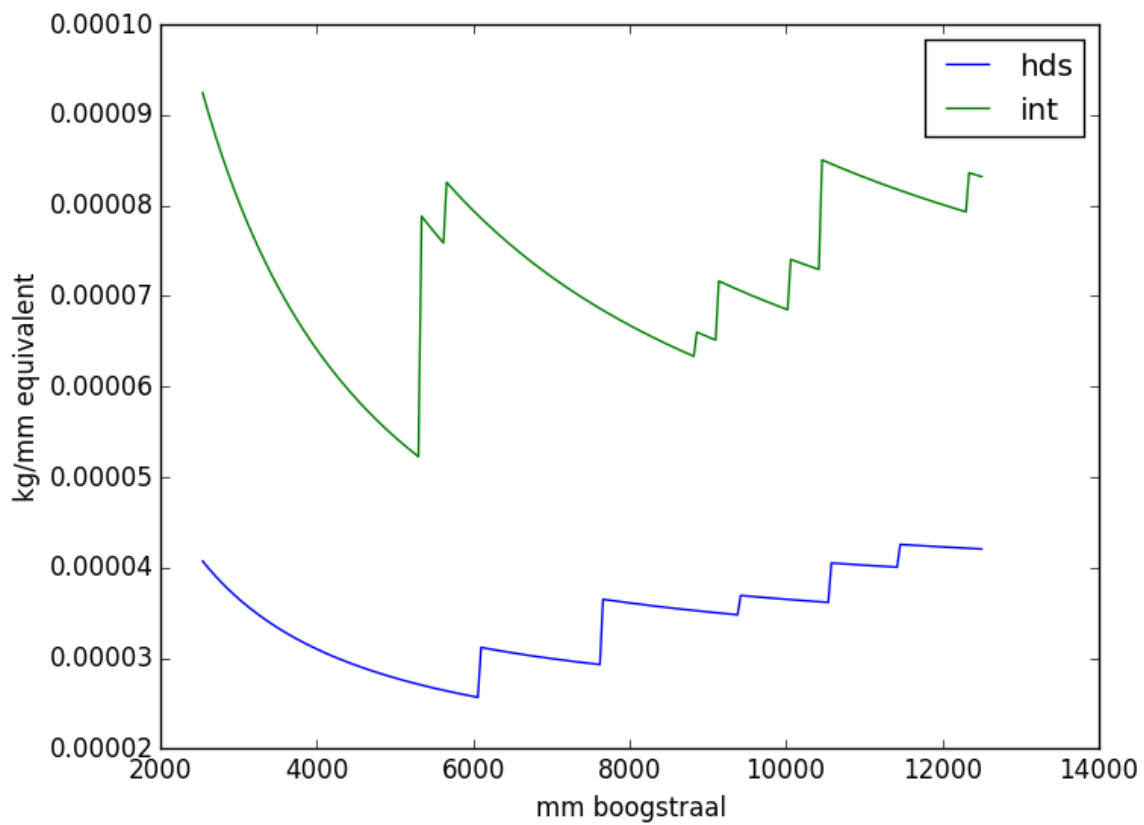
7.2. Boogstraal

De variabele R staat voor een halve overspanningslengte en komt van de straal van een cirkelboog. Wanneer geen halve cirkel gebruikt wordt, maar een gedeelte (α), wordt na het invoeren van deze variabele een nieuwe straal bepaald die geschaald is naar een constante overspanning d , zodat totale horizontale afstand tussen begin en eind gelijk is aan $2R$. R is van invloed op de overspanning en de hoogte van de constructie. De hoogte van de constructie heeft effect op de windkracht.



Figuur 26: Materiaalgebruik per afstand over boogstralen.

In figuur 26 is duidelijk te zien dat bij een grotere boogstraal het benodigd gewicht toeneemt. Wat logisch is omdat voor een grotere overspanning meer materiaal nodig is. Ook de stapsgewijze toename van de profielkeuze is weer aanwezig. Wat belangrijk is aan deze grafiek is de onderlinge locatie. Voor alle waarden van R is de waarde van het geïntegreerd vakwerk groter dan die van het hoofddraagsysteem en bij grotere waarden van R neemt dit verschil toe. Voor een verduidelijking van de gekozen waarde is de bovenstaande grafiek nogmaals berekend, alleen nu ten opzichte van kg/mm per mm boogstraal



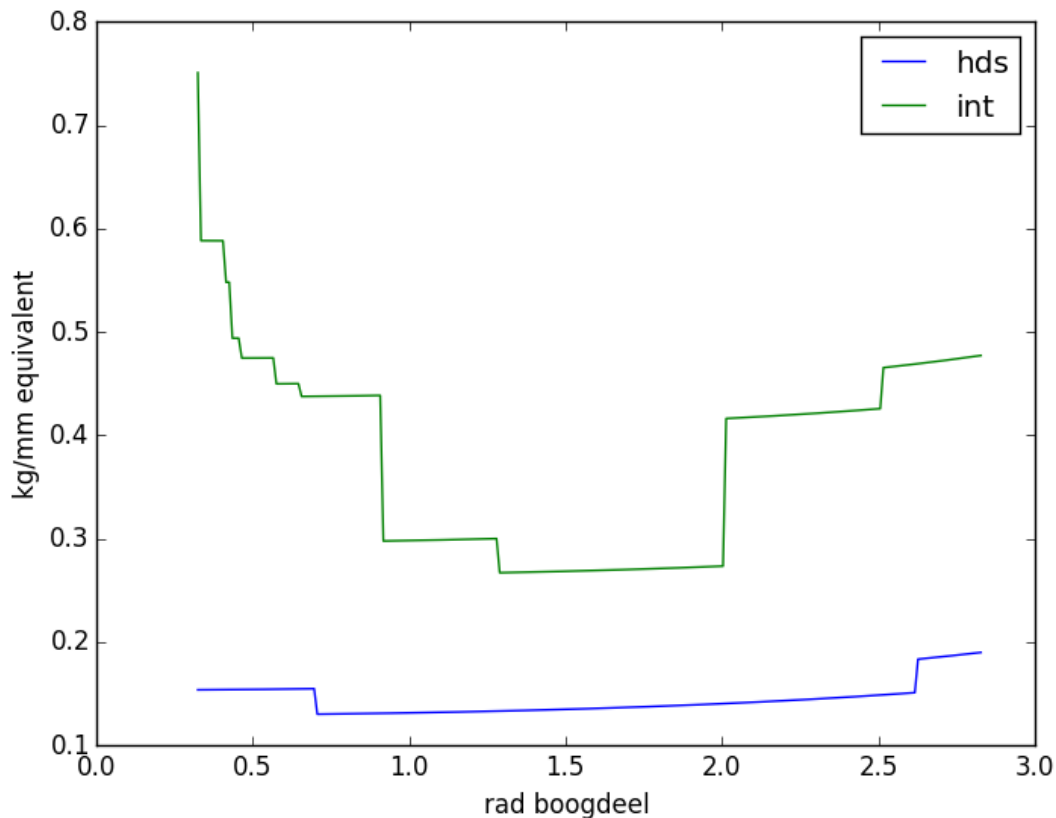
Figuur 27: Materiaalgebruik per boogdeel over boogstralen.

Figuur 27 laat zien dat voor beide constructies een boogstraal van 5000 mm redelijk optimaal is. Als de waarde voorbij 5000 mm gaat, zal het materiaalgebruik in een keer flink toenemen

7.3. Cirkelsegment

De variabele α verandert de vorm van de constructie. De α staat voor welk gedeelte van een cirkel gebruikt wordt om de constructie vorm te geven. Een α van π rad zal een perfecte halve cirkel beschrijven. De invloed van α is dat zowel de hoogte verandert als de manier van krachtenafracht.

Naar deze variabele is al gekeken in een vorig hoofdstuk en er is gekozen om de resultaten hiervan te vergelijken met die van een vakwerkboog.

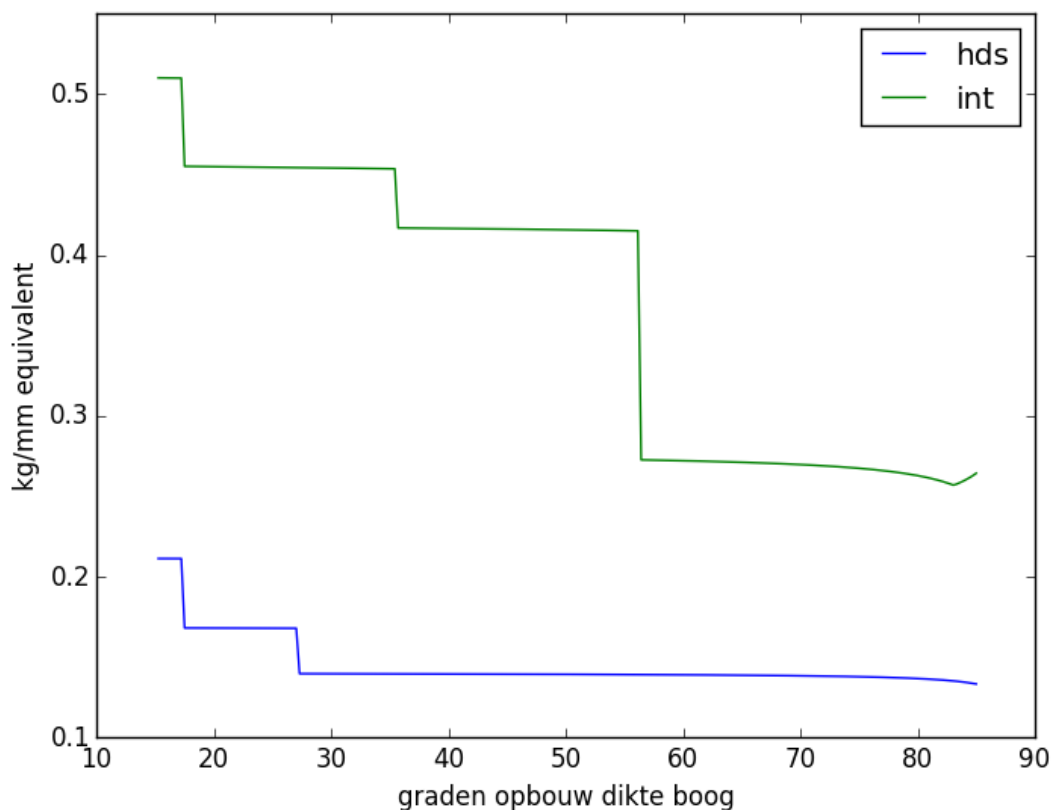


Figuur 28: Materiaalgebruik per afstand over cirkelsegment.

Vergelijkbaar met de Maple berekeningen geeft ook figuur 28 aan dat er een optimum rond de $0,5\pi$ is. Bij verandering van de krachten richting 0 en π neemt ook de profielkeuze toe. Tussen 1,4 en 2,0 rad is het benodigd materiaal minimaal. De gekozen waarde van $0,6\pi$ rad (1,88 rad) valt hiertussen en is dus een geschikte waarde.

7.4. Dikte constructie

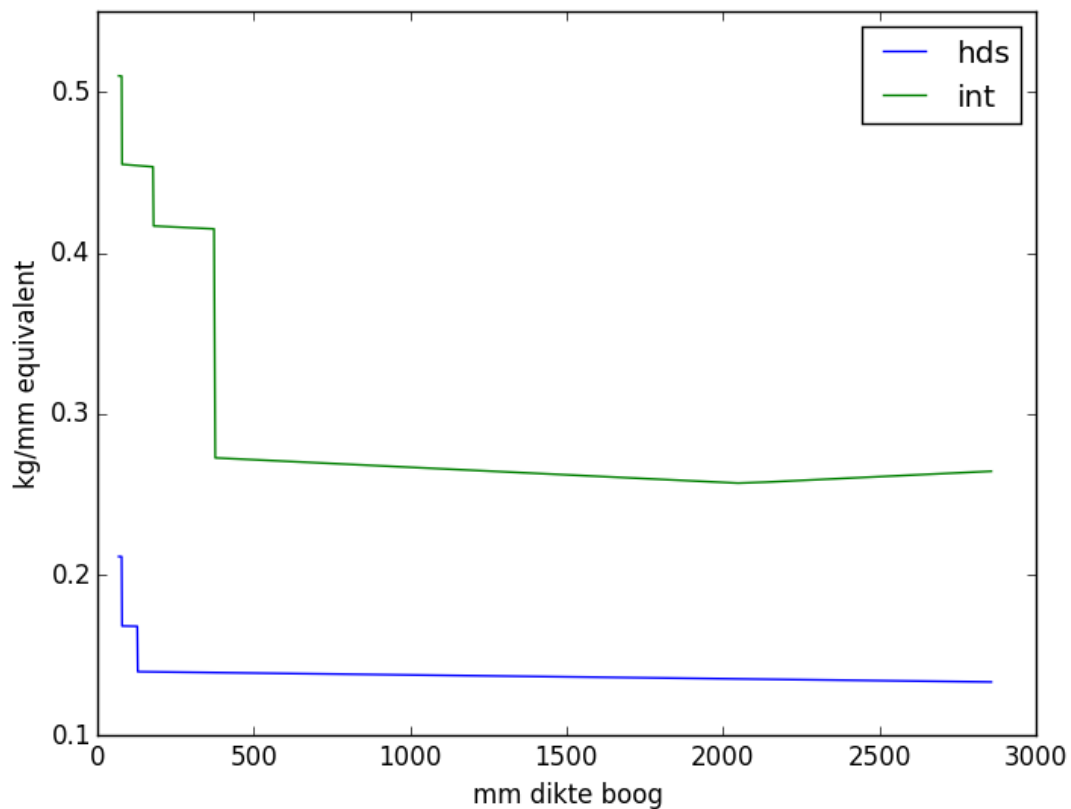
De laatste variabele is phi, phi staat voor de hoek die de verbindingstaven maken ten opzichte van de buitenste parallelle boogstaven. Een grotere phi zorgt voor een dikkere constructie. Deze variabele is gekozen om te kijken of een gelijkzijdige driehoek (phi is 60 graden) een geschikte keuze was. De phi bepaalt de dikte of verschil in boogstraal bij het hoofddraagsysteem. Deze waarden worden overgenomen door het geïntegreerd vakwerk. Dit zorgt ervoor dat beide systemen eenzelfde volume bestrijken. Als ervoor gekozen zou worden op dezelfde wijze het geïntegreerd vakwerk te bepalen, zouden nagenoeg alle waarden een onrealistische uitkomst hebben.



Figuur 29: Materiaalgebruik per afstand over phi.

Figuur 29 geeft aan dat bij toename van phi het benodigde gewicht verminderd zal worden. Vanaf ongeveer 55 graden is het gebruikte materiaal minimaal. Bij waarden richting de 90 graden zullen de constructies onrealistisch worden, het is daarom verstandig om de phi niet te hoog aan te nemen. Ook voor deze variabele is een optimale waarde gekozen.

In figuur 30 is de grafiek nogmaals weergegeven, alleen nu uitgezet ten opzichte van de dikte van de constructie. Dit is een makkelijker in te beelden variabele.



Figuur 30: Materiaalgebruik per afstand over vakwerkdikte.

Uit alle resultaten is af te leiden dat voor alle gekozen variabelen bijna optimale condities zijn gekozen voor dit onderzoek. En als resultaat is, dat voor de hier gekozen methode voor een geïntegreerd vakwerk in alle gevallen meer materiaal benodigd is dan voor een hoofddraagsysteem.

Omdat de variabelen alleen op de gekozen manier effect hebben op de constructie, is het mogelijk dat, wanneer ze op een andere manier toegepast worden, er andere resultaten uitkomen

Conclusie

Bij het ontwerpen van een boogvormige hal is het goedkoper qua hoeveelheid materiaal om van een hoofddraagsysteem uit te gaan in plaats van een geïntegreerd vakwerk.

Als er wordt aangenomen dat de gekozen vorm een cirkelboog beschrijft, is het voor beide constructies voordelig om een cirkelsegment aan te nemen tussen de 1,3 en 1,9 Rad. Een afwijking hiervan kan het voor grote toename van materiaalgebruik zorgen.

Er is geen direct verband tussen de keuze van de overspanningslengte en het benodigd gewicht voor de constructie. Kleine verschillen kunnen een groot effect hebben op het benodigd gewicht.

De dikte van het vakwerk is afhankelijk van de overspanning. Een waarde van minimaal $1/20$ van de overspanning is geschikt voor een vakwerkboog. Kleiner dan dit neemt het materiaalgebruik toe. En bij grotere waarden neemt de constructie te veel ruimte in.

Voor constructies waarvan de variabelen een optimale waarde hebben, is het benodigd gewicht optimaal voor een hart op hart afstand van 1500 mm bij een geïntegreerd vakwerk en 5500 mm bij een hoofddraagsysteem.

Aanbeveling

Voor vervolg onderzoek is het mogelijk om andere variabelen te bekijken, of de invloed van variabelen op andere variabelen. Bij grote programmeer bestanden is het mogelijk dat er delen zijn die niet werken buiten de aangenomen waarden. Er zijn alleen standaard profielen in het programma opgenomen. Wanneer er een keuze buiten deze profielen genomen moet worden, bijvoorbeeld door een te grote overspanning, moet deze optie eerst ingevoerd worden. Het is daarom verstandig om het gebruikte programma een keer volledig te controleren voordat deze aangepast wordt voor vervolg onderzoek. Het alternatief is een nieuw programma schrijven. Het gebruik van 3DTruss is geschikt, echter een vernieuwd programma kan mogelijk de berekeningen versnellen.

Literatuurlijst

Bloemers, A.J.W. 2014, Model om de staafkrachten te benaderen van een 3d vakwerkboog, Delft

Hartsuijker, C. & Welleman, J.W. Statisch onbepaalde constructies en bezwijk analyses, Academic Service, Den Haag.

Snijder, H.H. & Steenbergen, H.M.G.M. 2011, Krachtswerking. Grondslagen voor het berekenen en toetsen van staalconstructies voor gebouwen volgens Eurocode 0, 1 en 3, Bouwen met staal, Zoetermeer.

Welleman, J.W. 2013, Notities over kabels en bogen, Den Hoorn.

Bijlagen

Bijlage 1: Maple bestand:

berekening van cirkelsegment

```
restart;
with(plottools); with(plots);

nm := 9;
x := 'x'; angle := 'angle'; r := 5; EI := 1; y := sqrt(r^2-x^2);
l1 := evalf(r*cos(Pi-(Pi-angle)*(1/2))+r); l2 := evalf(r*cos((Pi-angle)*(1/2))+r); l3 :=
sqrt(-(-r+l1)^2+25);
y := sqrt((r/cos((Pi-angle)*(1/2)))^2-x^2)-r*sin((Pi-angle)*(1/2))/cos((Pi-angle)*(1/2));
plot3d(y, x = -5 .. 5, angle = .1*Pi .. .9*Pi);

angle := .1*nm*Pi;
y := evalf(y);
DV := EI*(diff(w(x), `$(x, 4)))+H*(diff(y, `$(x, 2)))-q = 0;
w := rhs(dsolve(DV, w(x)));
phi := -(diff(w, x)); M := EI*(diff(phi, x)); V := diff(M, x); qq := -(diff(V, x));
x := -r; eq1 := w = 0; eq2 := M = 0;
x := r; eq3 := w = 0; eq4 := M = 0;
sol := solve({eq1, eq2, eq3, eq4}, {_C1, _C2, _C3, _C4}); assign(sol);
x := 'x';
g := (diff(w, x))*(diff(y, x));
eq1 := int(g, x = -r .. r) = 0;
H := solve(eq1, H);

q := 10;
angle := 'angle'; r4 := evalf(r/cos((Pi-angle)*(1/2))); XX3 := r4*cos(t); YY3 := angle;
ZZ3 := r4*(sin(t)-sin((Pi-angle)*(1/2))); plot3d([angle, XX3, ZZ3], t = Pi-(Pi-angle)*(1/2)
.. (Pi-angle)*(1/2), angle = .1*Pi .. Pi);
angle := 'angle'; r4 := evalf(r/cos((Pi-angle)*(1/2))); YXY := sqrt((r/cos((Pi-
angle)*(1/2)))^2-xxx^2)-r4*sin((Pi-angle)*(1/2)); plot3d(YXY, xxx = -r .. r, angle = .1*Pi
.. Pi);

N := -H*cos(arctan(diff(y, x)))+(-EI*V(x)-H*(diff(y, x))*sin(arctan(diff(y, x))); V := -
H*sin(arctan(diff(y, x)))+(-EI*V(x)-H*(diff(y, x))*cos(arctan(diff(y, x)));
plot(y, x = -r .. r); plot(-w, x = -r .. r); plot(M, x = -r .. r); plot(V, x = -r .. r); plot(-N, x = -r
.. r);

x := -r; mat := [angle, evalf(w), evalf(M), evalf(V), evalf(N)];
x := 0; y;
x := 'x'; diff(y, x);

x := r; VV := evalf(V); NN := evalf(N); x := 0; yy := evalf(y); WW := evalf(w); MM :=
evalf(M);
mat := [angle, yy, WW, MM, VV, NN];
mdata[nm] := [mdata[nm-1], mat];
save mdata, `mat.m`;
```

Bijlage 2: Technical information 3DTruss:

3D Trusses

The element definition of a 3D truss element can be described in a discrete way. We start with a 3D-element between nodes **A** (a_x, a_y, a_z) and **B** (b_x, b_y, b_z) in a coordinate system with origin **O**.

The element i has six degrees of freedom located at its ends:

$$u_i^T = [u_x^A, u_y^A, u_z^A, u_x^B, u_y^B, u_z^B]$$

The bar element i has its local bar axis from node **A** to **B**. The vectorial direction can be denoted with the unity vector l :

$$l = \frac{1}{L} \begin{pmatrix} b_x - a_x \\ b_y - a_y \\ b_z - a_z \end{pmatrix} = \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} \quad \text{with: } |l| = 1.0$$

In which the length L of the bar can be found with:

$$L = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2}$$

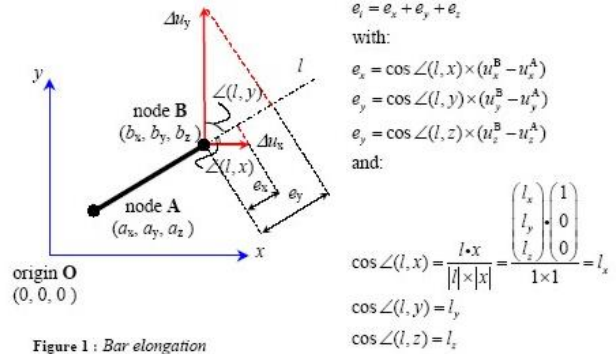
The elongation e_i of a bar can directly be expressed in terms of the six displacements u_i of the element. The relation between the elongation and the displacements u_i is given with the kinematic relation B_i (see figure 1):

$$e_i = B_i \times u_i \quad \text{with:}$$

$$B_i = [-l_x \quad -l_y \quad -l_z \quad l_x \quad l_y \quad l_z]$$

Each elements i thus adds to the system stiffness with:

$$K_i = B_i^T \times \frac{EA_i}{L_i} \times B_i$$



This stiffness should be added to the total system stiffness matrix with respect to the used degrees of freedom of the element. This process is also known as *assembling elements*.

After reducing the system with the boundary conditions, the displacements of the truss can be found by solving:

$$f = K_{system} \times u \quad \text{see footnote } ^1$$

The axial force for an individual bar element can be found with the solved displacements of the element degrees of freedom u_i :

$$N_i = B_i \times \frac{EA_i}{L_i} \times u_i$$

Close

¹ Unused degrees of freedoms (with zero stiffness) may be neglected in this system or strictly checked in case of mechanisms. The user can switch this check on or off.

Bijlage 3: Python programma:

```
from pylab import *
from numpy import *
from math import *
from mpl_toolkits.mplot3d.axes3d import
Axes3D
import time
import matplotlib.pyplot as plt
close('all')
```

```
#####
#####
####
```

```
def Fwindcal(R):
    hoogte=R
```

```
if hoogte <= 1:
    qp=0.93 #kN/m^2
if hoogte <= 2:
    qp=1.11
if hoogte <= 3:
    qp=1.22
if hoogte <= 4:
    qp=1.30
if hoogte <= 5:
    qp=1.37
if hoogte <= 6:
    qp=1.42
if hoogte <= 7:
    qp=1.47
if hoogte <= 8:
    qp=1.51
if hoogte <= 9:
    qp=1.55
if hoogte <= 10:
    qp=1.58
if hoogte <= 11:
    qp=1.71
if hoogte <= 12:
    qp=1.80
if hoogte <= 13:
    qp=1.88
if hoogte <= 14:
    qp=1.94
if hoogte <= 15:
    qp=2.00
if hoogte <= 16:
    qp=2.04
if hoogte <= 17:
    qp=2.08
if hoogte <= 18:
    qp=2.12
if hoogte <= 19:
    qp=2.16
if hoogte <= 20:
```

```
qp=2.19
```

```
cf1=0.8
cf2=-0.5
cf3=0.7
```

```
sigma1=cf1*qp
sigma2=cf2*qp
sigma3=cf3*qp
return sigma1, sigma2, sigma3
#####
#####
####
```

```
# functie kracht ligger bepalen overspanning,
qlasten, draagvlak
def detFhds(L,qvar,qper,width):
    if L<40:
        L=L*1000
```

```
E=210.*10**3 #N/mm^2
#UGT q*gamma
qtot=1.5*qvar+1.2*qper #N/mm
M=1./8.*qtot*(L)**2 #Nmm
fy=355. #N/mm^2
izz1h=M*0.5/fy #izz/h= #mm^4/mm
```

```
#BGT gamma=1.0
# 0.003*L variabel, 0.004*L totaal
```

```
E=210.*10**3 #N/mm^2
qtot=1.0*qvar+1.0*qper #N/mm
lzz21=(qvar*(L)**3*5.)/(0.003*E*384.) #mm^4
variabel
lzz22=(qtot*(L)**3*5.)/(0.004*E*384.) #mm^4
totaal
izz2=max(lzz21,lzz22) #mm^4
# bepalen van profiel en geven van
profielgegevens BGT
def forcetrs2(izz2):
```

```
if izz2<=171.*10**4:
    h=100. #mm
    izz=171.*10**4 #mm^4
    eg=8.1 #kg/m= 1000kg/mm
    elif izz2<=318.*10**4:
        h=120.
        izz=318.*10**4
        eg=10.4
    elif izz2<=541.*10**4:
        h=140.
        izz=541.*10**4
        eg=12.9
    elif izz2<=869.*10**4:
        h=160.
        izz=869.*10**4
        eg=15.8
    elif izz2<=1317.*10**4:
        h=180.
```

```

izz=1317.*10**4
eg=18.8
elif izz2<=1943.*10**4:
h=200.
izz=1943.*10**4
eg=22.4
elif izz2<=2772.*10**4:
h=220.
izz=2772.*10**4
eg=26.2
elif izz2<=3892.*10**4:
h=240.
izz=3892.*10**4
eg=30.7
elif izz2<=5790.*10**4:
h=270.
izz=5790.*10**4
eg=36.1
elif izz2<=8356.*10**4:
h=300.
izz=8356.*10**4
eg=42.2
elif izz2<=11770.*10**4:
h=330.
izz=11770.*10**4
eg=49.1
elif izz2<=16270.*10**4:
h=360.
izz=16270.*10**4
eg=57.1
elif izz2<=23130.*10**4:
h=400.
izz=23130.*10**4
eg=66.3
elif izz2<=33740.*10**4:
h=450.
izz=33740.*10**4
eg=77.6
elif izz2<=48200.*10**4:
h=500.
izz=48200.*10**4
eg=90.7
elif izz2<=67120.*10**4:
h=550.
izz=67120.*10**4
eg=105.5
else:
h=600.
izz=92080.*10**4
eg=122.4

izz2=izz

return izz2, eg

# bepalen profiel en toewijzing profielgegevens
UGT
def forcetrsh(izz1h):

if izz1h<=171.*10**4/100.: # mm^4/mm

```

```

h=100. # mm
izz=171.*10**4 # mm^4
eg=8.1 # kg/m
elif izz1h<=318.*10**4/120.:
h=120.
izz=318.*10**4
eg=10.4
elif izz1h<=541.*10**4/140.:
h=140.
izz=541.*10**4
eg=12.9
elif izz1h<=869.*10**4/160.:
h=160.
izz=869.*10**4
eg=15.8
elif izz1h<=1317.*10**4/180.:
h=180.
izz=1317.*10**4
eg=18.8
elif izz1h<=1943.*10**4/200.:
h=200.
izz=1943.*10**4
eg=22.4
elif izz1h<=2772.*10**4/220.:
h=220.
izz=2772.*10**4
eg=26.2
elif izz1h<=3892.*10**4/240.:
h=240.
izz=3892.*10**4
eg=30.7
elif izz1h<=5790.*10**4/270.:
h=270.
izz=5790.*10**4
eg=36.1
elif izz1h<=8356.*10**4/300.:
h=300.
izz=8356.*10**4
eg=42.2
elif izz1h<=11770.*10**4/330.:
h=330.
izz=11770.*10**4
eg=49.1
elif izz1h<=16270.*10**4/360.:
h=360.
izz=16270.*10**4
eg=57.1
elif izz1h<=23130.*10**4/400.:
h=400.
izz=23130.*10**4
eg=66.3
elif izz1h<=33740.*10**4/450.:
h=450.
izz=33740.*10**4
eg=77.6
elif izz1h<=48200.*10**4/500.:
h=500.
izz=48200.*10**4
eg=90.7
elif izz1h<=67120.*10**4/550.:

```

```

h=550.
izz=67120.*10**4
eg=105.5
else:
h=600.
izz=92080.*10**4
eg=122.4

izz1=izz

return izz1, eg

[izz2, eg2]=forcetrs2(izz2)
#print izz2
[izz1, eg1]=forcetrs1h(izz1h)
#print izz1
if izz2>izz1:
eg=eg2
else:
eg=eg1

# berekening kracht z-richting
F=((1.2*qper+1.5*qvar)*width/1000.+eg/1000*
9.81)*L/1000.

return F, eg

#####
#####
####

def detFint(L,qvar,qper,width,d):
if L<40:
L=L*1000

#F=((1.2*qper+1.5*qvar)*width/1000.+eg/1000
*9.81)*L/1000.
F=((1.2*qper+1.5*qvar)*width/1000.)*(0.5*L/10
00.+0.5*d*1000.)
eg=0
return F, eg

#####
#####
#

#maken van programma-file (.trs)
def
createtrshds(filenum, trusnum, length, R, alpha, n
m, d, RR):

# filenum= number van naam van file
# trusnum= aantal gelijke staaflengtes in
boogstructuur
# length = lengte tussen twee boogstructuren
# R = 0.5* overspanning boogstructuur (straal
van alpha = pi)

```

```

# alpha = boogdeel (pi = halve cirkel, 0.5*pi =
kwart cirkel)
# nm = elke nm knooppunten 1 ligger
# d = breedte boogstructuur en bepaling
hoogte boogstructuur (gelijkzijdige driehoek)

```

```

filenum = filenum
filename = 'hds'+ str(filenum)+'TRS'

```

```

g=open(filename,mode='w')
f = open(filename, mode='a+')

```

```

#Give the properties (E-modulus (E) and
cross-sectional area (A))
N=trusnum
MaxLength = 10*(N+1)
E0=210000000 #kN/m^2
A0=1 #m^2

```

```

print >>f, '[settings]'
print >>f, 'maxlength=', MaxLength
print >>f, '[properties]'
print >>f, 'E=', E0
print >>f, 'A=', A0

```

```

print >>f, '[coordinates]'
# N = amount of bars per arch, R = radius of
arch (2R is spanwidth)
N=trusnum
R=R/1000 # m
d=d/1000 # m
kk=0
x=zeros(MaxLength)
y=zeros(MaxLength)
z=zeros(MaxLength)

```

```

angle=(pi-alpha)/2+alpha

```

```

# bepaling van knooppunten op buitenste boog
for k in range(N+1):
hoek=alpha/N*k
angle=(pi-alpha)/2+hoek
kk=kk+1
x[kk-1]=R*cos(angle)+ R
y[kk-1]=0
z[kk-1]=-R*sin(angle)
s=str(kk)+'-'+str(x[kk-1])+ '@'+str(y[kk-
1])+ '@'+str(z[kk-1])
print >>f, s
for k in range(N+1):
hoek=alpha/N*k
angle=(pi-alpha)/2+hoek
kk=kk+1
x[kk-1]=R*cos(angle)+ R
y[kk-1]=d

```

```

z[kk-1]=-R*sin(angle)
s=str(kk)+'-'+str(x[kk-1])+ '@'+str(y[kk-1])+ '@'+str(z[kk-1])
print >>f, s

# bepaling van knooppunten op binnenste boog
RR=RR/1000.

kk=kk+1
s=str(kk)+'-'+str(RR*cos((pi-alpha)/2)+R)+'@'+str(0.5*d)+'@'+str(-RR*sin((pi-alpha)/2))
print >>f, s
for k in range(N):
hoek=alpha/N*k+0.5*alpha/N
angle=(pi-alpha)/2+hoek
kk=kk+1
x[kk-1]=RR*cos(angle)+ R
y[kk-1]=0.5*d
z[kk-1]=-RR*sin(angle)
s=str(kk)+'-'+str(x[kk-1])+ '@'+str(y[kk-1])+ '@'+str(z[kk-1])
print >>f, s

kk=kk+1
s=str(kk)+'-'+str(RR*cos((pi-alpha)/2+alpha)+R)+'@'+str(0.5*d)+'@'+str(-RR*sin((pi-alpha)/2+alpha))
print >>f, s

print >>f,['elements']

#boogstaven
M=N+1 #aantal punten op 1 boog
kk=0
for k2 in range(1,M):
kk=kk+1
x[kk-1]=k2
y[kk-1]=k2+1
s=str(kk)+'-'+str(int(x[kk-1]))+'@'+str(int(y[kk-1]))+'@1'
print >>f, s
for k2 in range(1,M):
kk=kk+1
x[kk-1]=M+k2
y[kk-1]=M+k2+1
s=str(kk)+'-'+str(int(x[kk-1]))+'@'+str(int(y[kk-1]))+'@1'
print >>f, s
for k2 in range(1,M+1):
kk=kk+1
x[kk-1]=2*M+k2
y[kk-1]=2*M+k2+1
s=str(kk)+'-'+str(int(x[kk-1]))+'@'+str(int(y[kk-1]))+'@1'
print >>f, s

```

```

#dwarsstaven verbindingen tussen twee boogen in y richting
for k2 in range(1,M+1):
kk=kk+1
x[kk-1]=k2
y[kk-1]=M+k2
s=str(kk)+'-'+str(int(x[kk-1]))+'@'+str(int(y[kk-1]))+'@2'
print >>f, s

```

```

#kruisstaven verbindingen tussen binnen en buitenboog
for k in range(0,2):
for k2 in range(1,M):
kk=kk+1
x[kk-1]=k2+2*M+1
y[kk-1]=k*M+k2
s=str(kk)+'-'+str(int(x[kk-1]))+'@'+str(int(y[kk-1]))+'@2'
print >>f, s
for k in range(0,2):
for k2 in range(1,M):
kk=kk+1
x[kk-1]=k2+2*M+1
y[kk-1]=k*M+k2+1
s=str(kk)+'-'+str(int(x[kk-1]))+'@'+str(int(y[kk-1]))+'@2'
print >>f, s

```

```

# diagonaalstaven verbindingen tussen buitenste bogen met volgende dwarsstaven
if (N % 2 == 0): N2=N/2
else: N2=(N+1)/2

```

```

for k2 in range(1,N2+1):
kk=kk+1
x[kk-1]=k2*2-1
y[kk-1]=M+k2*2
s=str(kk)+'-'+str(int(x[kk-1]))+'@'+str(int(y[kk-1]))+'@2'
print >>f, s
for k2 in range(1,N2):
kk=kk+1
x[kk-1]=M+k2*2
y[kk-1]=k2*2+1
s=str(kk)+'-'+str(int(x[kk-1]))+'@'+str(int(y[kk-1]))+'@2'
print >>f, s

```

```

print >>f, ['barproperties']
E1=210000000 # kN/m^2
E2=210000000 # kN/m^2
A1=0.000238 # m^2
A2=0.000238 # m^2
stringE1='1'+str(E1)+'@'+str(A1) # boogstaven

```

```

stringE2='2='+str(E2)+'@'+str(A2) #
tussenstaven
print >>f, stringE1
print >>f, stringE2

print >>f, '[loads]'

width=pi*R*nm/(trusnum)
qvar=1
qper=0.15
[F, eg]=detFhds(length,qvar,qper,width) ###
bepaling kracht op knooppunten
NM=int(nm) # elke nm knooppunten een kracht
kk=0

# windbelasting
#print Fwind

[sigma1, sigma2, sigma3]=Fwindcal(R*cos((pi-
alpha)/2.))

for k1 in range(0,2):
for k2 in range(1,M+1):
kk=kk+1
x[kk-1]=k1*M+k2
if k2 == 1:
sigma=sigma1
Fwind=(0.5*d+0.5*length/1000.)*sigma*(-
R*sin((pi-alpha)/2+(k2-
1)*alpha/trusnum)+R*sin((pi-alpha)/2+(k2-
1+0.5)*alpha/trusnum))
elif k2 == M:
sigma=sigma2
Fwind=(0.5*d+0.5*length/1000.)*sigma*(-
R*sin((pi-alpha)/2+(k2-1-
0.5)*alpha/trusnum)+R*sin((pi-alpha)/2+(k2-
1)*alpha/trusnum))
elif k2 < (M+1)/2:
sigma=sigma1
Fwind=(0.5*d+0.5*length/1000.)*sigma*(-
R*sin((pi-alpha)/2+(k2-1-
0.5)*alpha/trusnum)+R*sin((pi-alpha)/2+(k2-
1+0.5)*alpha/trusnum))
elif k2 == (M+1)/2:
sigma=(sigma1+sigma2)
Fwind=(0.5*d+0.5*length/1000.)*sigma*(-
R*sin((pi-alpha)/2+(k2-1-
0.5)*alpha/trusnum)+R*sin((pi-alpha)/2+(k2-
1)*alpha/trusnum))
elif k2 > (M+1)/2:
sigma=sigma2
Fwind=(0.5*d+0.5*length/1000.)*sigma*(-
R*sin((pi-alpha)/2+(k2-1-
0.5)*alpha/trusnum)+R*sin((pi-alpha)/2+(k2-
1+0.5)*alpha/trusnum))
y[kk-1]=Fwind
s=str(kk)+'=' +str(int(x[kk-1]))+'@x@'+str(y[kk-
1])

```

```

print >>f, s

for k1 in range(0,2):
for k2 in range(1,M+1):
kk=kk+1
sigma=sigma3
if k2 == 1:
Fwind=(0.5*d+0.5*length/1000.)*sigma*(-
R*cos((pi-alpha)/2+(k2-
1)*alpha/trusnum)+R*cos((pi-alpha)/2+(k2-
1+0.5)*alpha/trusnum))
Fligger=0.5*F
elif k2 == M:
Fwind=(0.5*d+0.5*length/1000.)*sigma*(-
R*cos((pi-alpha)/2+(k2-1-
0.5)*alpha/trusnum)+R*cos((pi-alpha)/2+(k2-
1)*alpha/trusnum))
Fligger=0.5*F
else:
Fwind=(0.5*d+0.5*length/1000.)*sigma*(-
R*cos((pi-alpha)/2+(k2-1-
0.5)*alpha/trusnum)+R*cos((pi-alpha)/2+(k2-
1+0.5)*alpha/trusnum))
Fligger=F
x[kk-1]=k1*M+k2
y[kk-1]=-Fwind+Fligger
s=str(kk)+'=' +str(int(x[kk-1]))+'@z@'+str(y[kk-
1])
print >>f, s

# elk eindpunt van bogen is een translatievaste
verbinding, in x,y en z richting
print >>f, '[supports]'
kk=0
for k2 in range(0,3):
kk=kk+1
x[kk-1]=k2*M+1
s=str(kk)+'=' +str(int(x[kk-1]))+'@x'
print >>f, s
kk=kk+1
s=str(kk)+'=' +str(int(x[kk-2]))+'@y'
print >>f, s
kk=kk+1
s=str(kk)+'=' +str(int(x[kk-3]))+'@z'
print >>f, s
for k2 in range(0,2):
kk=kk+1
x[kk-1]=(1+k2)*M
s=str(kk)+'=' +str(int(x[kk-1]))+'@x'
print >>f, s
kk=kk+1
s=str(kk)+'=' +str(int(x[kk-2]))+'@y'
print >>f, s
kk=kk+1
s=str(kk)+'=' +str(int(x[kk-3]))+'@z'
print >>f, s

kk=kk+1
x[kk-1]=3*M+1

```

```

s=str(kk)+'=' +str(int(x[kk-1]))+'@x'
print >>f, s
kk=kk+1
s=str(kk)+'=' +str(int(x[kk-2]))+'@y'
print >>f, s
kk=kk+1
s=str(kk)+'=' +str(int(x[kk-3]))+'@z'
print >>f, s
return

#####
#####
####
def
createtrussint(filenum, trusnum, trusswidth, R, alpha
, nm, d, RR):

###length in detF

# filenum= number van naam van file
# trusnum= aantal gelijke staaflengtes in
boogstructuur
# length = lengte tussen twee boogstructuren
# trusswidth = verdelingsaantal in
lengterichting (aantal boogstructuren)
# R = 0.5* overspanning boogstructuur (straal
van alpha = pi)
# alpha = boogdeel (pi = halve cirkel, 0.5*pi =
kwart cirkel)
# nm = elke nm knooppunten 1 ligger
# d = breedte boogstructuur en bepaling
hoogte boogstructuur (gelijkzijdige driehoek)

filenum = filenum
filename = 'int'+ str(filenum)+' .TRS'

g=open(filename, mode='w')
f = open(filename, mode='a+')

P=trusswidth

#Give the properties (E-modulus (E) and
cross-sectional area (A))
N=trusnum
MaxLength = 10*(N+1)*P
E0=210000000 #kN/m^2
A0=0.000238 #m^2

print >>f, '[settings]'
print >>f, 'maxlength=', MaxLength
print >>f, '[properties]'
print >>f, 'E=', E0
print >>f, 'A=', A0

print >>f, '[coordinates]'

```

```

# N = amount of bars per arch, R = radius of
arch (2R is spanwidth)
N=trusnum
R=R/1000.
d=d/1000.
kk=0
x=zeros(MaxLength)
y=zeros(MaxLength)
z=zeros(MaxLength)
angle=(pi-alpha)/2+alpha

```

```

# bepaling van knooppunten op buitenste
bogen
for k2 in range (0,1+P):
for k in range(N+1):
hoek=alpha/N*k
angle=(pi-alpha)/2+hoek
kk=kk+1
x[kk-1]=R*cos(angle)+ R
y[kk-1]=(k2)*d
z[kk-1]=-R*sin(angle)
s=str(kk)+'=' +str(x[kk-1])+'@'+str(y[kk-
1])+'@'+str(z[kk-1])
print >>f, s

```

```

# bepaling van knooppunten op binnenste
bogen
RR=RR/1000.

```

```

for k2 in range(0,P):
kk=kk+1
s=str(kk)+'=' +str(RR*cos((pi-alpha)/2)+
R)+'@'+str(0.5*d+k2*d)+'@'+str(-RR*sin((pi-
alpha)/2))
print >>f, s
for k in range(N):
hoek=alpha/N*k+0.5*alpha/N
angle=(pi-alpha)/2+hoek
kk=kk+1
x[kk-1]=RR*cos(angle)+ R
y[kk-1]=0.5*d+k2*d
z[kk-1]=-RR*sin(angle)
s=str(kk)+'=' +str(x[kk-1])+'@'+str(y[kk-
1])+'@'+str(z[kk-1])
print >>f, s

```

```

kk=kk+1
s=str(kk)+'=' +str(RR*cos((pi-alpha)/2+alpha)+
R)+'@'+str(0.5*d+k2*d)+'@'+str(-RR*sin((pi-
alpha)/2+alpha))
print >>f, s

```

```

print >>f, '[elements]'

```

```

#boogstaven
M=N+1 #aantal punten op 1 boog
kk=0

```

```

for k2 in range(1,(P+1)*(M)):

```

```

if (k2 % (M) == 0):
    continue
else :
    kk=kk+1
    x[kk-1]=k2
    y[kk-1]=k2+1
    s=str(kk)+'=' +str(int(x[kk-1]))+'@'+str(int(y[kk-1]))+'@1'
    print >>f, s
    for k2 in range(1,P*(M+1)):
        if (k2 % (M+1) == 0):
            continue
        else :
            kk=kk+1
            x[kk-1]=(P+1)*M+k2
            y[kk-1]=(P+1)*M+k2+1
            s=str(kk)+'=' +str(int(x[kk-1]))+'@'+str(int(y[kk-1]))+'@1'
            print >>f, s

```

```

#dwarsstaven verbindingen tussen twee
boogen in y richting
for k2 in range(1,P*M+1):
    kk=kk+1
    x[kk-1]=k2
    y[kk-1]=M+k2
    s=str(kk)+'=' +str(int(x[kk-1]))+'@'+str(int(y[kk-1]))+'@2'
    print >>f, s

```

```

for k2 in range(1,(P-1)*(M+1)+1):
    kk=kk+1
    x[kk-1]=(P+1)*M+k2
    y[kk-1]=(P+2)*M+k2+1
    s=str(kk)+'=' +str(int(x[kk-1]))+'@'+str(int(y[kk-1]))+'@2'
    print >>f, s

```

```

#kruisstaven verbindingen tussen binnen en
buitenboog
for k3 in range(0,P):
    for k in range(0,2):
        for k2 in range(1,M):
            kk=kk+1
            x[kk-1]=k2+(P+1)*M+1+k3*(M+1)
            y[kk-1]=k*M+k2+k3*M
            s=str(kk)+'=' +str(int(x[kk-1]))+'@'+str(int(y[kk-1]))+'@2'
            print >>f, s
            for k in range(0,2):
                for k2 in range(1,M):
                    kk=kk+1
                    x[kk-1]=k2+(P+1)*M+1+k3*(M+1)
                    y[kk-1]=k*M+k2+1+k3*M
                    s=str(kk)+'=' +str(int(x[kk-1]))+'@'+str(int(y[kk-1]))+'@2'
                    print >>f, s

```

```

# diagonaalstaven verbindingen tussen
buitenste bogen met volgende dwarsstaven
if (N % 2 == 0): N2=N/2
else: N2=(N+1)/2

```

```

for k3 in range(0,P):
    for k2 in range(1,N2+1):
        kk=kk+1
        x[kk-1]=k2*2-1+k3*(M)
        y[kk-1]=M+k2*2+k3*(M)
        s=str(kk)+'=' +str(int(x[kk-1]))+'@'+str(int(y[kk-1]))+'@2'
        print >>f, s
        for k2 in range(1,N2-1):
            kk=kk+1
            x[kk-1]=M+k2*2+k3*(M)
            y[kk-1]=k2*2+1+k3*(M)
            s=str(kk)+'=' +str(int(x[kk-1]))+'@'+str(int(y[kk-1]))+'@2'
            print >>f, s
        print >>f, '[barproperties]'
        E1=210000000 # kN/m^2
        E2=210000000 # kN/m^2
        A1=0.000238 # m^2
        A2=0.000238 # m^2
        stringE1='1='+str(E1)+'@'+str(A1) #
        boogstaven
        stringE2='2='+str(E2)+'@'+str(A2) #
        tussenstaven
        print >>f, stringE1
        print >>f, stringE2

```

```

print >>f, '[loads]'
width=pi*R*nm/(trusnum)
qvar=1
qper=0.15
[F, eg]=detFint(1000.*d,qvar,qper,width,d) ###
bepaling kracht op knooppunten

```

```

NM=int(nm) # elke nm knooppunten een kracht
kk=0

```

```

# windbelasting
#print Fwind
[sigma1, sigma2, sigma3]=Fwindcal(R*cos((pi-alpha)/2.))

```

```

for k1 in range(0,P+1):
    for k2 in range(1,M+1):
        if k1 == 0 or k1 == P:
            kk=kk+1
            x[kk-1]=k1*M+k2
            if k2 == 1:
                sigma=sigma1

```

```

Fwind=0.5*d*sigma*(-R*sin((pi-alpha)/2+(k2-1)*alpha/trusnum)+R*sin((pi-alpha)/2+(k2-1+0.5)*alpha/trusnum))
elif k2 == M:
sigma=sigma2
Fwind=0.5*d*sigma*(-R*sin((pi-alpha)/2+(k2-1-0.5)*alpha/trusnum)+R*sin((pi-alpha)/2+(k2-1)*alpha/trusnum))
elif k2 < (M+1)/2:
sigma=sigma1
Fwind=0.5*d*sigma*(-R*sin((pi-alpha)/2+(k2-1-0.5)*alpha/trusnum)+R*sin((pi-alpha)/2+(k2-1+0.5)*alpha/trusnum))
elif k2 == (M+1)/2:
sigma=(sigma1+sigma2)
Fwind=0.5*d*sigma*(-R*sin((pi-alpha)/2+(k2-1-0.5)*alpha/trusnum)+R*sin((pi-alpha)/2+(k2-1)*alpha/trusnum))
elif k2 > (M+1)/2:
sigma=sigma2
Fwind=0.5*d*sigma*(-R*sin((pi-alpha)/2+(k2-1-0.5)*alpha/trusnum)+R*sin((pi-alpha)/2+(k2-1+0.5)*alpha/trusnum))
y[kk-1]=Fwind
else :
kk=kk+1
if k2 == 1:
sigma=sigma1
Fwind=d*sigma*(-R*sin((pi-alpha)/2+(k2-1)*alpha/trusnum)+R*sin((pi-alpha)/2+(k2-1+0.5)*alpha/trusnum))
elif k2 == M:
sigma=sigma2
Fwind=d*sigma*(-R*sin((pi-alpha)/2+(k2-1-0.5)*alpha/trusnum)+R*sin((pi-alpha)/2+(k2-1)*alpha/trusnum))
elif k2 < (M+1)/2:
sigma=sigma1
Fwind=d*sigma*(-R*sin((pi-alpha)/2+(k2-1-0.5)*alpha/trusnum)+R*sin((pi-alpha)/2+(k2-1+0.5)*alpha/trusnum))
elif k2 == (M+1)/2:
sigma=(sigma1+sigma2)
Fwind=d*sigma*(-R*sin((pi-alpha)/2+(k2-1-0.5)*alpha/trusnum)+R*sin((pi-alpha)/2+(k2-1)*alpha/trusnum))
elif k2 > (M+1)/2:
sigma=sigma2
Fwind=d*sigma*(-R*sin((pi-alpha)/2+(k2-1-0.5)*alpha/trusnum)+R*sin((pi-alpha)/2+(k2-1+0.5)*alpha/trusnum))
x[kk-1]=k1*M+k2
y[kk-1]=Fwind
s=str(kk)+'-'+str(int(x[kk-1]))+'@x@'+str(y[kk-1])
print >>f, s

for k1 in range(0,P+1):
for k2 in range(1,M+1):
kk=kk+1

```

```

if k1 == 0 or k1 == P:
sigma=sigma3
if k2 == 1:
Fwind=0.5*d*sigma*(-R*cos((pi-alpha)/2+(k2-1)*alpha/trusnum)+R*cos((pi-alpha)/2+(k2-1+0.5)*alpha/trusnum))
Fligger=0.25*F
elif k2 == M:
Fwind=0.5*d*sigma*(-R*cos((pi-alpha)/2+(k2-1-0.5)*alpha/trusnum)+R*cos((pi-alpha)/2+(k2-1)*alpha/trusnum))
Fligger=0.25*F
else:
Fwind=0.5*d*sigma*(-R*cos((pi-alpha)/2+(k2-1-0.5)*alpha/trusnum)+R*cos((pi-alpha)/2+(k2-1+0.5)*alpha/trusnum))
Fligger=0.5*F
else :
sigma=sigma3
if k2 == 1:
Fwind=d*sigma*(-R*cos((pi-alpha)/2+(k2-1)*alpha/trusnum)+R*cos((pi-alpha)/2+(k2-1+0.5)*alpha/trusnum))
Fligger=0.5*F
elif k2 == M:
Fwind=d*sigma*(-R*cos((pi-alpha)/2+(k2-1-0.5)*alpha/trusnum)+R*cos((pi-alpha)/2+(k2-1)*alpha/trusnum))
Fligger=0.5*F
else:
Fwind=d*sigma*(-R*cos((pi-alpha)/2+(k2-1-0.5)*alpha/trusnum)+R*cos((pi-alpha)/2+(k2-1+0.5)*alpha/trusnum))
Fligger=F
x[kk-1]=k1*M+k2
y[kk-1]=-Fwind+Fligger
s=str(kk)+'-'+str(int(x[kk-1]))+'@z@'+str(y[kk-1])
print >>f, s
# elk eindpunt van bogen is een translatievaste verbinding, in x,y en z richting
print >>f, '[supports]'
kk=0
for k2 in range(0,P+1):
kk=kk+1
x[kk-1]=k2*M+1
s=str(kk)+'-'+str(int(x[kk-1]))+'@x'
print >>f, s
kk=kk+1
s=str(kk)+'-'+str(int(x[kk-2]))+'@y'
print >>f, s
kk=kk+1
s=str(kk)+'-'+str(int(x[kk-3]))+'@z'
print >>f, s
for k2 in range(0,P+1):
kk=kk+1
x[kk-1]=(1+k2)*M
s=str(kk)+'-'+str(int(x[kk-1]))+'@x'
print >>f, s
kk=kk+1

```

```

s=str(kk)+'=' +str(int(x[kk-2]))+'@y'
print >>f, s
kk=kk+1
s=str(kk)+'=' +str(int(x[kk-3]))+'@z'
print >>f, s

for k2 in range(0,P+1):
    kk=kk+1
    x[kk-1]=(M)*(P+k2+1)+k2
    s=str(kk)+'=' +str(int(x[kk-1]))+'@x'
    print >>f, s
    kk=kk+1
    s=str(kk)+'=' +str(int(x[kk-2]))+'@y'
    print >>f, s
    kk=kk+1
    s=str(kk)+'=' +str(int(x[kk-3]))+'@z'
    print >>f, s
    for k2 in range(0,P+1):
        kk=kk+1
        x[kk-1]=(M)*(P+k2)+k2
        s=str(kk)+'=' +str(int(x[kk-1]))+'@x'
        print >>f, s
        kk=kk+1
        s=str(kk)+'=' +str(int(x[kk-2]))+'@y'
        print >>f, s
        kk=kk+1
        s=str(kk)+'=' +str(int(x[kk-3]))+'@z'
        print >>f, s
    return

#####
####
def runtrshds(filename):
    filename = filename
    filename = 'hds'+ str(filename)+'.TRS'

import subprocess
import os

path_to_truss3d
='C:/Users/Chip/Documents/BEP/truss3D.exe'
# path to executing program 3dtruss

path_to_file =
'C:/Users/Chip/Documents/BEP/data/'+filename
e
# path to file with at least: settings, properties,
coordinates, element, Barproperties, loads,
support

p = subprocess.Popen([path_to_truss3d,'-i',
path_to_file])
# creates same name file with displacements
elementforces and reactions

return

#####
####

```

```

def runtrsint(filename):
    filename = filename
    filename = 'int'+ str(filename)+'.TRS'

import subprocess
import os

path_to_truss3d
='C:/Users/Chip/Documents/BEP/truss3D.exe'
# path to executing program 3dtruss

path_to_file =
'C:/Users/Chip/Documents/BEP/data/'+filename
e
# path to file with at least: settings, properties,
coordinates, element, Barproperties, loads,
support

p = subprocess.Popen([path_to_truss3d,'-i',
path_to_file])
# creates same name file with displacements
elementforces and reactions

return
#####
####

def readtrshds(filename, trusnum):

N=trusnum
filename = 'hds'+ str(filename)+'.TRS'

# bepalen tabel normaalkrachten in file
with open(filename) as f:
    for num, line in enumerate(f, 1):
        if '[elementforces]' in line:
            linebegin = num
            #print linebegin
        elif '[reactions]' in line:
            lineend = num - 1
            #print lineend
        f.close()
        val=zeros(lineend-linebegin)
        n=0

        f = open(filename)
        for i, line in enumerate(f):
            if (i >= linebegin and i < lineend):
                n=n+1
                string1=line
                string2=str(n) + '='
                string3=string1.strip(string2)
                value1=float(string3.strip("\n"))
                val[n-1]= value1
            #print value1
        f.close()

```

```

# bepaal grootste normaalkracht in boog,
(profiel 1)
trekmain=max(val[0:(3*(N+1)+1)])
drukmain=min(val[0:(3*(N+1)+1)])
# bepaal grootste normaalkracht in
tussenstaven (profiel 2)
treksub=max(val[(3*(N+1)+1):])
druksub=min(val[(3*(N+1)+1):])
return drukmain, druksub, trekmain, treksub

#####
####
def readtrsint(filename, trusnum, P):

N=trusnum
filename = 'int'+ str(filename)+' .TRS'

# bepalen tabel normaalkrachten in file
with open(filename) as f:
for num, line in enumerate(f, 1):
if '[elementforces]' in line:
linebegin = num
#print linebegin
elif '[reactions]' in line:
lineend = num - 1
#print lineend
f.close
val=zeros(lineend-linebegin)
n=0

f = open(filename)
for i, line in enumerate(f):
if (i >= linebegin and i < lineend):
n=n+1
string1=line
string2=str(n) + '='
string3=string1.strip(string2)
value1=float(string3.strip('\n'))
val[n-1]= value1
#print value1
f.close()

# bepaal grootste normaalkracht in boog,
(profiel 1)
trekmain=max(val[0:((2*P+1)*(N+1)+P)])
drukmain=min(val[0:((2*P+1)*(N+1)+P)])

# bepaal grootste normaalkracht in
tussenstaven (profiel 2)
treksub=max(val[((2*P+1)*(N+1)+P):])
druksub=min(val[((2*P+1)*(N+1)+P):])

return [drukmain, druksub, trekmain, treksub]

#####
####

```

```

#bepaaling profielkeuze op normaalkracht en
knik
def roundprof(druk,N,L):
Nf=druk
n=0
Nrd=0

def prof(n):
if n==1:
Atot=2.38
lzz=1.7
eg=1.87
if n==2:
Atot=3.94
lzz=7.62
eg=3.09
if n==3:
Atot=4.53
lzz=11.6
eg=3.56
if n==4:
Atot=5.74
lzz=23.5
eg=4.51
if n==5:
Atot=6.67
lzz=44.7
eg=5.24
if n==6:
Atot=8.62
lzz=79.2
eg=6.76
if n==7:
Atot=11.2
lzz=172.
eg=8.77
if n==8:
Atot=21.4
lzz=481.
eg=16.6
if n==9:
Atot=25.7
lzz=856.
eg=20.1
if n==10:
Atot=29.6
lzz=1320.
eg=23.3
if n==11:
Atot=33.6
lzz=1930.
eg=26.4
if n==12:
Atot=42.1
lzz=3780.
eg=33.0
if n==13:
Atot=62.9
lzz=7930.
eg=49.3

```

```

if n==14:
Atot=79.2
lzz=15900.
eg=62.2
if n>14:
Atot=10000
lzz=10000000
eg=100000
Atot=Atot*10**2
lzz=lzz*10**4
return Atot, lzz, eg #mm^2, mm^4, kg/m

# iteratieve check of N design groter is dan N
aanwezig. toekenning aan profiel
eigenschappen
while Nrd < Nf:
n=n+1
[A,lzz,eg]=prof(n)
alpha1=0.21 # koudgewalst rond profiel
E=210000. #N/mm2
fy=355. #N/mm2
Lcr=L #var

ivar=sqrt(lzz/A)
gammay=1.15

lambda1=pi*sqrt(E/fy)
lambdastreep=Lcr/ivar*1/lambda1
phi=0.5*(1.+alpha1*(lambdastreep-
0.2)+lambdastreep**2)
xi=1/(phi+sqrt(phi**2-lambdastreep**2))
if xi > 1.0:
xi=1.0
Nrd=xi*A*fy/gammay
return eg

#####
####

def backeghds(L, trusnum, R, nm):
width=pi*R*nm/(trusnum)
qvar=1
qper=0.15
[F, eg]=detFhds(L, qvar, qper, width)
return eg

#####
### data ###
#####

def
datacreate(trusnum, trusswidth, fy, R, alpha, nm, d
, phi, lengthhds, lengthint, XX):
R=R/cos((pi-alpha)/2.)
RRhds=R-(tan(phi/180.*pi)*0.5*d)
RRint=R-(tan(phi/180.*pi)*0.5*lengthint)
filenum=XX

```

```

egx0hds=createtrshds(filenum, trusnum, lengthh
ds, R, alpha, nm, d, RRhds)
egx0int=createtsint(filenum, trusnum, trusswidt
h, R, alpha, nm, lengthint, RRhds)
runtrshds(filenum)
runtrsint(filenum)
time.sleep(10)
return

def staaflengte(lengthb1, d):
lengthc1=sqrt((lengthb1**2+d**2)
lengthc2=sqrt((0.5*lengthb1)**2+(0.5*d)**2+(sq
rt(3./4.)*d)**2)
lengthc3=d
Lc=max(lengthc1, lengthc2, lengthc3)
return Lc, lengthc1, lengthc2, lengthc3

def
datagather(trusnum, trusswidth, fy, R, alpha, nm, d
, phi, lengthhds, lengthint, XX):
R=R/cos((pi-alpha)/2.)
RRhds=R-(tan(phi/180.*pi)*0.5*d)
RRint=R-(tan(phi/180.*pi)*0.5*lengthint)
theta1=(pi-alpha)/2
theta2=(pi-alpha)/2+alpha/trusnum
# lengte boogstaven
lengthb1=sqrt((R*cos(theta1)-
R*cos(theta2))**2+(-
R*sin(theta1)+R*sin(theta2))**2)
lengthb2hds=sqrt((RRhds*cos(theta1)-
(RRhds*cos(theta2)))**2+(-
RRhds*sin(theta1)+RRhds*sin(theta2))**2)
lengthb2int=sqrt((RRint*cos(theta1)-
(RRint*cos(theta2)))**2+(-
RRint*sin(theta1)+RRint*sin(theta2))**2)
Lbhds=max(lengthb1, lengthb2hds)
Lbint=max(lengthb1, lengthb2int)
#lengte verbindingsstaven diff voor int en hds
[Lchds, lengthc1hds, lengthc2hds, lengthc3hds]=
staaflengte(lengthb1, d)
[Lcint, lengthc1int, lengthc2int, lengthc3int]=staafl
engte(lengthb1, lengthint)

lengthbthds=(2.*lengthb1+lengthb2hds)*trusnu
m
lengthcthds=(lengthc1hds+4.*lengthc2hds)*tru
snum+lengthc3hds*(trusnum+1)
lengthbtint=(2.*lengthb1+lengthb2int)*trusnum
lengthctint=(lengthc1int+4.*lengthc2int)*trusnu
m+lengthc3int*(trusnum+1)
filenum=XX
[druk1hds, druk2hds, trek1hds,
trek2hds]=readtrshds(filenum, trusnum)
egx0hds=backeghds(lengthhds, trusnum, R, nm)
egx1hds=roundprof(abs(druk1hds)*1000, trusnu
m, Lbhds)
egx2hds=roundprof(abs(druk2hds)*1000, trusnu
m, Lchds)
[druk1int, druk2int, trek1int,
trek2int]=readtrsint(filenum, trusnum, trusswidth)

```

```

egx1int=roundprof(abs(druk1int)*1000,trusnum
,Lbint)
egx2int=roundprof(abs(druk2int)*1000,trusnum
,Lcint)
gewichtshds=((lengthbthds*egx1hds/1000.+
lengthcthds*egx2hds/1000.)*(trusswidth+1)+((t
rusnum/2-
1)*lengthhds*egx0hds/1000.)*trusswidth)/((len
gthhds+d)*trusswidth)
gewichtint=((lengthbtint*egx1int/1000.+
lengthctint*egx2int/1000.)*(trusswidth+1)+lengt
hc3int*egx2int/1000.)*(trusnum+1)*trusswidth)/
(trusswidth*lengthint)
return egx0hds, egx1hds, egx2hds, egx1int,
egx2int, gewichtshds,gewichtint

```

```

trusnum=12
fy=355.
R=5000. # mm
alpha=0.6*pi
nm=1.
d=R/10.
phi=60.
counter=1. #calc/m
maxi=250
amount=int(maxi)*int(counter)
trusswidth=3
lengthhds=5000.
lengthint=lengthhds

```

```

data0hds=zeros(amount)
data1hds=zeros(amount)
data2hds=zeros(amount)
data1int=zeros(amount)
data2int=zeros(amount)

```

```

gewichtshds=zeros(amount)
gewichtint=zeros(amount)
gewichtshds2=zeros(amount)
gewichtint2=zeros(amount)
gewichtshds3=zeros(amount)
gewichtint3=zeros(amount)
gewichtshds4=zeros(amount)
gewichtint4=zeros(amount)
gewichtshds5=zeros(amount)
gewichtint5=zeros(amount)
gewichtshds6=zeros(amount)
gewichtint6=zeros(amount)
YY=-amount

```

```

#####
####

```

```

YY=YY+amount

```

```

lengthXXhds=zeros(amount)
lengthXXint=zeros(amount)

```

```

XX=0

```

```

for var in range(1,amount+1):
XX=XX+1
####
lengthhds = 10000./amount*var #variabele hds
lengthint = 10000./amount*var #variabele int

```

```

####
datacreate(trusnum,trusswidth,fy,R,alpha,nm,d
,phi,lengthhds,lengthint,XX+YY)
time.sleep(20)

```

```

XX=0
for var in range(1,amount+1):
XX=XX+1

```

```

####
lengthhds = 10000./amount*var #variabele hds
lengthint = 10000./amount*var #variabele int
lengthXXhds[XX-1]=lengthhds
lengthXXint[XX-1]=lengthint

```

```

####
[egx0hds, egx1hds, egx2hds, egx1int, egx2int,
gewichtshdsdata1,gewichtintdata1]=datagather(
trusnum,trusswidth,fy,R,alpha,nm,d,phi,lengthh
ds,lengthint,XX+YY)
data0hds[XX-1]=egx0hds
data1hds[XX-1]=egx1hds
data2hds[XX-1]=egx2hds
data1int[XX-1]=egx1int
data2int[XX-1]=egx2int
gewichtshds[XX-1]=gewichtshdsdata1
gewichtint[XX-1]=gewichtintdata1
figure(1)
line1, =plt.plot(lengthXXhds,gewichtshds,
label="hds")
line2, =plt.plot(lengthXXint,gewichtint,
label="int")
plt.ylabel('kg/mm equivalent')
plt.xlabel('mm afstand tussen bogen')
plt.legend([line1, line2], ['hds', 'int'])

```

```

figure(7)
line1, =plt.plot(lengthXXhds,data0hds)
line2, =plt.plot(lengthXXhds,data1hds)
line3, =plt.plot(lengthXXhds,data2hds)
line4, =plt.plot(lengthXXint,data1int)
line5, =plt.plot(lengthXXint,data2int)
plt.ylabel('kg/m')
plt.xlabel('mm afstand tussen bogen')
plt.legend([line1, line2, line3, line4, line5], ['hds
ligger', 'hds boog', 'hds dwars', 'int boog', 'int
dwars'])

```

```

#####
####

```

```

YY=YY+amount

XX=0
lengthhds = 1500. #variabele hds
lengthint = 1500.
RXX=zeros(amount)

for var in range(1,amount+1):
XX=XX+1
####
R = (2.5+var*10./amount)*1000. #variabele
hds
####
datacreate(trusnum,trusswidth,fy,R,alpha,nm,d
,phi,lengthhds,lengthint,XX+YY)
time.sleep(20)

XX=0

for var in range(1,amount+1):
XX=XX+1

####
R = (2.5+var*10./amount)*1000. #variabele
hds
RXX[XX-1]=R
####
[egx0hds, egx1hds, egx2hds, egx1int, egx2int,
gewichthdsdata2,gewichtintdata2]=datagather(
trusnum,trusswidth,fy,R,alpha,nm,d,phi,lengthh
ds,lengthint,XX+YY)
data0hds[XX-1]=egx0hds
data1hds[XX-1]=egx1hds
data2hds[XX-1]=egx2hds
data1int[XX-1]=egx1int
data2int[XX-1]=egx2int
gewichthds2[XX-1]=gewichthdsdata2
gewichtint2[XX-1]=gewichtintdata2

figure(2)
line1, =plt.plot(RXX,gewichthds2/RXX,
label="hds")
line2, =plt.plot(RXX,gewichtint2/RXX,
label="int")
plt.ylabel('kg/mm equivalent')
plt.xlabel('mm boogstraal')
plt.legend([line1, line2], ['hds', 'int'])

#####
####

YY=YY+amount

XX=0
lengthhds = 1500. #variabele hds
lengthint = 1500.
R=5000.

```

```

alphaXX=zeros(amount)

for var in range(1,amount+1):
XX=XX+1

####
alpha = 0.1*pi+var*0.8*pi/amount #variabele
hds
####
datacreate(trusnum,trusswidth,fy,R,alpha,nm,d
,phi,lengthhds,lengthint,XX+YY)
time.sleep(20)

XX=0

for var in range(1,amount+1):
XX=XX+1
####
alpha = 0.1*pi+var*0.8*pi/amount #variabele
hds
alphaXX[XX-1]=alpha
####
[egx0hds, egx1hds, egx2hds, egx1int, egx2int,
gewichthdsdata3,gewichtintdata3]=datagather(
trusnum,trusswidth,fy,R,alpha,nm,d,phi,lengthh
ds,lengthint,XX+YY)
data0hds[XX-1]=egx0hds
data1hds[XX-1]=egx1hds
data2hds[XX-1]=egx2hds
data1int[XX-1]=egx1int
data2int[XX-1]=egx2int
gewichthds3[XX-1]=gewichthdsdata3
gewichtint3[XX-1]=gewichtintdata3

figure(3)
line1, =plt.plot(alphaXX,gewichthds3,
label="hds")
plt.ylabel('kg/mm equivalent')
plt.xlabel('rad boogdeel')
plt.legend([line1], ['hds'])
line2, =plt.plot(alphaXX,gewichtint3,
label="int")
plt.ylabel('kg/mm equivalent')
plt.xlabel('rad boogdeel')
plt.legend([line1, line2], ['hds', 'int'])

#####
####

YY=YY+amount

XX=0
alpha=0.6*pi
lengthhds = 1500
lengthint = 1500
R=5000.
phiXX=zeros(amount)
DDDXX=zeros(amount)

```

```

for var in range(1,amount+1):
XX=XX+1
###
phi = 15.+70./amount*var #variabele hds
###
datacreate(trusnum,trusswidth,fy,R,alpha,nm,d
,phi,lengthhds,lengthint,XX+YY)
time.sleep(20)

XX=0

for var in range(1,amount+1):
XX=XX+1

###
phi = 15.+70./amount*var #variabele hds
phiXX[XX-1]=phi
DDD=(tan(phi/180.*pi)*0.5*d)
DDDXX[XX-1]=DDD
###
[egx0hds, egx1hds, egx2hds, egx1int, egx2int,
gewichthdsdata4,gewichtintdata4]=datagather(
trusnum,trusswidth,fy,R,alpha,nm,d,phi,lengthh
ds,lengthint,XX+YY)
data0hds[XX-1]=egx0hds
data1hds[XX-1]=egx1hds
data2hds[XX-1]=egx2hds
data1int[XX-1]=egx1int
data2int[XX-1]=egx2int
gewichthds4[XX-1]=gewichthdsdata4
gewichtint4[XX-1]=gewichtintdata4

print DDDXX
print phiXX
print gewichtint4
figure(5)
line1, =plt.plot(DDDXX,gewichthds4,
label="hds")
line2, =plt.plot(DDDXX,gewichtint4, label="int")

plt.ylabel('kg/mm equivalent')
plt.xlabel('mm dikte boog')
plt.legend([line1, line2], ['hds', 'int'])

#####
####

plt.show('all'

```