

Bachelor eindwerk

Wringing in 3D-raamwerkprogramma's

Afleiding en validatie van de element-stijfheidsmatrix voor
dubbel symmetrische profielen

Door

M.J. Carton
6058175

Datum: Juni 2026
1^e begeleider: Dr. ir. P.C.J. Hoogenboom
2^e begeleider: Ir. T.R. van Woudenberg



Voorwoord

Voor u ligt het bachelor eindwerk “Wringing in 3D-raamwerkprogramma’s”, geschreven ter afronding van mijn bacheloropleiding civiele techniek aan de TU Delft. Dit rapport behandelt de analytische afleiding van de element-stijfheidsmatrix met wringing voor dubbel symmetrische profielen. Het is geschreven voor medestudenten civiele techniek, waarbij basiskennis van constructiemechanica als bekend wordt verondersteld.

Het afleiden en opbouwen van de matrixvergelijkingen en de vertaalslag naar Python vormden een uitdagende maar leerzame puzzel. Mijn dank gaat uit naar mijn begeleiders, P.C.J. Hoogenboom en T.R. van Woudenberg, voor hun sturing en feedback gedurende het project.

Bij de totstandkoming van dit eindwerk is AI als hulpmiddel ingezet voor tekstverbetering en voor de foutopsporing en optimalisatie van het Python-script.

Ik wens u veel leesplezier.

Delft, juni 2026

Martijn Carton

Samenvatting

In 3D-raamwerkprogramma's wordt wringing vaak gemodelleerd volgens de theorie van Saint-Venant, waarbij wordt uitgegaan van vrije welving. Welving is het fenomeen waarbij de dwarsdoorsnede van een profiel door torsie uit zijn vlak vervormt. In de praktijk wordt welving echter vaak verhinderd door constructieve verbindingen of opleggingen, wat leidt tot extra interne spanningen in het element. Dit rapport behandelt de afleiding en de validatie van de 3D element-stijfheidsmatrix voor dubbel symmetrische profielen, waarbij verhinderde welving correct wordt gemodelleerd volgens de theorie van Vlasov.

De resulterende 14×14 matrix is analytisch afgeleid en succesvol geïmplementeerd in een symbolisch Python-script met SymPy. Validatie met diverse standaardbelastinggevallen bevestigt dat het model het elementgedrag correct beschrijft. De resultaten tonen aan dat het meenemen van verhinderde welving leidt tot een hogere torsiestijfheid in vergelijking met de theorie van Saint-Venant. Omdat berekende vervormingen hierdoor kleiner uitvallen, kan het toepassen van dit model overdimensionering van constructies voorkomen. Het ontwikkelde script biedt bovendien een waardevol controlemiddel om de output van commerciële software onafhankelijk te kunnen verifiëren.

Voor vervolgonderzoek wordt aanbevolen de afleiding uit te breiden naar asymmetrische profielen, waarbij axiale rek, buiging en wringing niet langer onafhankelijk van elkaar zijn.

Inhoudsopgave

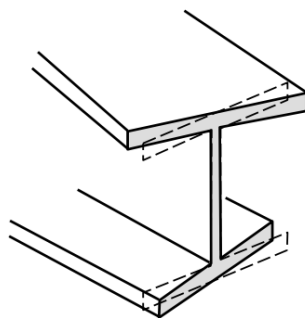
| | |
|--|-----------|
| Voorwoord | i |
| Samenvatting | ii |
| Nomenclatuur | iv |
| 1. Inleiding | 1 |
| 2. 2D-stijfheidsmatrix | 3 |
| 2.1. Definitie van het 2D-balkelement | 3 |
| 2.2. Afleiding van de stijfheidsrelaties | 4 |
| 2.2.1. Axiale stijfheid | 4 |
| 2.2.2. Buigstijfheid | 5 |
| 2.3. Implementatie in SymPy | 7 |
| 2.4. Validatie van de stijfheidsmatrix | 8 |
| 3. 3D-stijfheidsmatrix met vrije welving | 9 |
| 3.1. Definitie van het 3D-balkelement | 9 |
| 3.2. Wringing volgens de theorie van Saint-Venant | 10 |
| 3.3. Afleiding van de stijfheidsrelaties | 10 |
| 3.3.1. Buigstijfheid in het xy-vlak | 10 |
| 3.3.2. Torsiestijfheid | 11 |
| 3.4. Implementatie in SymPy | 13 |
| 3.5. Validatie van de stijfheidsmatrix | 13 |
| 4. 3D-stijfheidsmatrix met verhinderde welving | 14 |
| 4.1. Wringing volgens de theorie van Vlasov | 14 |
| 4.2. Afleiding van de stijfheidsrelaties | 15 |
| 4.3. Implementatie in SymPy | 18 |
| 4.4. Validatie van de stijfheidsmatrix | 19 |
| 5. Discussie | 20 |
| 6. Conclusies en aanbevelingen | 21 |
| 6.1. Conclusies | 21 |
| 6.2. Aanbevelingen | 22 |
| Literatuurlijst | 23 |
| Bijlagen | 24 |
| A. Volledige 2D-stijfheidsmatrix | 24 |
| B. 2D-stijfheidsmatrix Python-script | 25 |
| C. 2D-stijfheidsmatrix validatieberekeningen | 26 |
| D. Volledige 3D-stijfheidsmatrix met vrije welving | 33 |
| E. 3D-stijfheidsmatrix met vrije welving Python-script | 34 |
| F. 3D-stijfheidsmatrix met vrije welving validatieberekeningen | 36 |
| G. Volledige 3D-stijfheidsmatrix met verhinderde welving | 42 |
| H. 3D-stijfheidsmatrix met verhinderde welving Python-script | 43 |
| I. 3D-stijfheidsmatrix met verhinderde welving validatieberekeningen | 45 |

Nomenclatuur

| Symbol | Omschrijving | Eenheid |
|---------------|--------------------------------------|------------------|
| m_x | Verdeeld wringend moment | Nm/m |
| q_x | Verdeelde lijnlast in x-richting | N/m |
| q_y | Verdeelde lijnlast in y-richting | N/m |
| q_z | Verdeelde lijnlast in z-richting | N/m |
| u_x | Axiale verplaatsing | m |
| u_z | Transversale verplaatsing | m |
| x | Longitudinale coördinaat | m |
| A | Oppervlakte doorsnede | m ² |
| C_w | Welvingsconstante | m ⁶ |
| E | Elasticiteitsmodulus | N/m ² |
| G | Glijdingsmodulus | N/m ² |
| I_t | Torsie-traagheidsmoment | m ⁴ |
| I_{yy} | Oppervlakte-traagheidsmoment om z-as | m ⁴ |
| I_{zz} | Oppervlakte-traagheidsmoment om y-as | m ⁴ |
| L | Lengte van het element | m |
| M | Inwendig buigmoment | Nm |
| N | Inwendige normaalkracht | N |
| T | Inwendig wringmoment | Nm |
| V | Inwendige dwarskracht | N |
| ϕ | Hoekverdraaiing | rad |

1. Inleiding

In 3D-raamwerkprogramma's beschrijft de element-stijfheidsmatrix de relatie tussen de krachten die op een element werken en de bijbehorende vervormingen. Wringing wordt hierin gemodelleerd als een moment dat een rotatie om de lengteas van het element veroorzaakt. Naast de rotatie treedt er nog een ander vervormingseffect op, namelijk welving. Hierbij vervormt de dwarsdoorsnede van het profiel uit het vlak, waardoor verschillende delen van de doorsnede zich in de lengterichting verplaatsen en de doorsnede niet langer vlak blijft. Zoals te zien in Figuur 1.1, leidt wringing bij een I-profiel tot een vervorming waarbij de boven- en onderflens in tegengestelde richting uitbuigen.



Figuur 1.1: Welving van een I-profiel bij torsie (Hoogenboom, 2019)

Veel conventionele raamwerkprogramma's gebruiken in hun berekeningen de theorie van Saint-Venant die uitgaat van vrije welving. Dit betekent dat het profiel ongehinderd kan vervormen zonder dat er extra spanningen in het element ontstaan. Het resultaat is een 12×12 stijfheidsmatrix met 6 vrijheidsgraden per knoop (3 verplaatsingen en 3 rotaties).

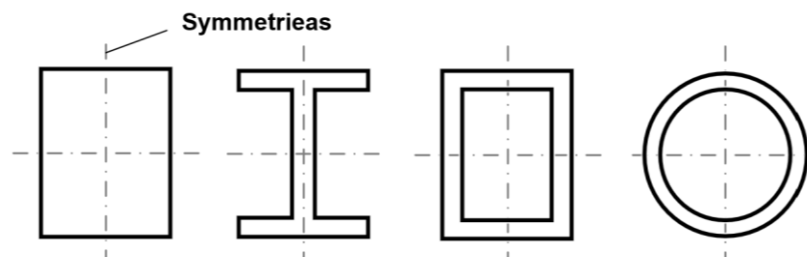
In veel constructies kan een raamwerkelement echter vaak niet vrij welven, omdat deze vervorming wordt verhinderd door de oplegging of de verbinding met aangrenzende elementen. Om deze gehinderde welving en de daarbij optredende extra spanningen in het element correct te modelleren, wordt de theorie van Vlasov toegepast. Hierbij wordt een zevende vrijheidsgraad aan de knoop toegevoegd die de welving van de dwarsdoorsnede beschrijft, wat resulteert in een 14×14 element-stijfheidsmatrix. Zonder deze extra vrijheidsgraad wordt de stijfheidsbijdrage van de gehinderde welving niet meegenomen, wat resulteert in een onnauwkeurige berekening van de vervorming van het element.

Hoewel een aantal commerciële raamwerkprogramma's (zoals SCIA Engineer en Ansys) het torsiegedrag met welving tegenwoordig wel correct modelleren, is de onderliggende stijfheidsmatrix niet openbaar. Het is van belang om deze element-stijfheidsmatrix af te leiden, zodat inzichtelijk wordt hoe de bijdrage van verhinderde welving in de stijfheidsmatrix wordt verwerkt.

Dit rapport beantwoordt de volgende hoofdvraag: *Hoe kan de element-stijfheidsmatrix worden afgeleid zodat wrijving en verhinderde welving correct worden gemodelleerd in 3D raamwerkprogramma's en op welke manier kan dit worden geïmplementeerd in een bruikbaar Python-script?*

Op basis van literatuuronderzoek wordt het constructieve gedrag bij vrije en verhinderde welving geanalyseerd, wat de basis vormt voor de afleiding van de stijfheidsmatrix. Deze afleiding wordt vervolgens uitgewerkt met behulp van SymPy, waarmee symbolische wiskundige berekeningen in Python uitgevoerd kunnen worden. De resulterende matrix wordt vervolgens gecontroleerd door verschillende testgevallen door te rekenen. Zo kan worden aangetoond dat de afleiding in het script klopt en het element zich correct gedraagt. Het onderzoek beperkt zich tot dubbel symmetrische profielen met lineair-elastisch materiaalgedrag onder statische belasting.

Een profiel is dubbel symmetrisch wanneer deze zowel een horizontale als verticale symmetrieas bezit. Hierdoor vallen het zwaartepunt en het dwarskrachtencentrum van de doorsnede samen, wat ervoor zorgt dat buiging en wrijving onafhankelijk van elkaar optreden. Voorbeelden hiervan zijn symmetrische I-profielen en kokerprofielen en ronde buisprofielen (zie Figuur 1.2).



Figuur 1.2: Dubbel symmetrische profielen (Eigen werk)

De opbouw van het rapport is als volgt. Hoofdstuk 2 beschrijft de afleiding van de 2D-stijfheidsmatrix. De uitbreiding naar 3D met vrije welving op basis van de theorie van Saint-Venant wordt gedaan in hoofdstuk 3. Vervolgens behandelt hoofdstuk 4 hoe verhinderde welving volgens de theorie van Vlasov wordt opgenomen in de 3D matrix. In deze drie hoofdstukken wordt telkens ook de implementatie in SymPy en de controle van het elementgedrag besproken. In hoofdstuk 5 volgt de discussie, waarna hoofdstuk 6 afsluit met de conclusies en aanbevelingen.

2. 2D-stijfheidsmatrix

In dit hoofdstuk wordt tweedimensionale elementstijfheidsmatrix afgeleid. De werkwijze die hier wordt toegepast, vormt de basis voor de uitgebreidere driedimensionale matrix in de latere hoofdstukken. Achtereenvolgens komen de elementdefinitie, de analytische afleiding, de symbolische implementatie in SymPy en de validatie van het model aan bod.

2.1. Definitie van het 2D-balkelement

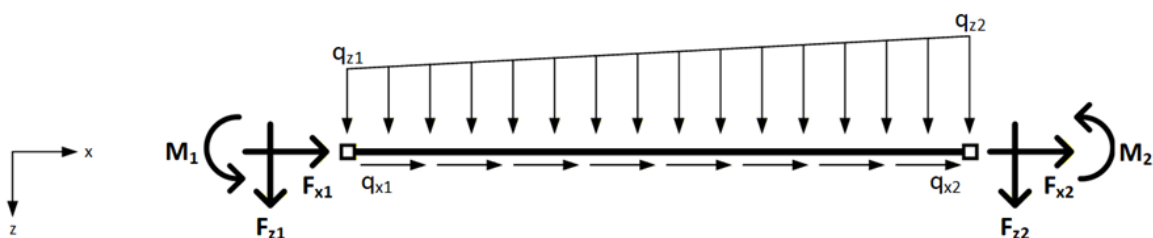
Voor de afleiding van de stijfheidsmatrix wordt het balkelement geschematiseerd als een element met lengte L . Aan beide uiteinden wordt het element begrensd door een knoop waarop de externe krachten en momenten aangrijpen.

De positie en vervorming van het element worden beschreven in een lokaal cartesiaans assenstelsel. Hierbij volgt de x-as de lengterichting van de balk, waarbij de positieve richting naar rechts is gedefinieerd. De z-as staat hier loodrecht op en is naar beneden positief gedefinieerd, conform de gangbare tekenafspraken voor de mechanica aan de TU Delft. Rotaties en momenten zijn positief gedefinieerd wanneer deze van de positieve z-as naar de positieve x-as draaien (tegen de klok in).

Elke knoop heeft een aantal vrijheidsgraden die de vervorming van het element beschrijven. In het tweedimensionale vlak beschikt elke knoop over drie vrijheidsgraden: een axiale translatie (u_x), een transversale translatie (u_z) en een rotatie (ϕ). Dit levert in totaal zes vrijheidsgraden op voor het volledige element.

Op het element kunnen enkele externe belastingen werken. Dit zijn de knooppuntenkrachten en -momenten (F_x, F_y, M) die direct op de knooppunten aangrijpen, en de lijnlasten q_x en q_z die over de lengte van het balkelement verdeeld zijn. Al deze belastingen volgen hierbij de positieve richtingen van het lokale assenstelsel.

In Figuur 2.1 zijn het lokale assenstelsel en de aangenomen richtingen van de belastingen schematische weergegeven.



Figuur 2.1: Schematisering van het 2D-balkelement inclusief tekenafspraken (Eigen werk)

2.2. Afleiding van de stijfheidsrelaties

De stijfheidsmatrix beschrijft de relatie tussen de krachten en de verplaatsingen van de knooppunten. In matrixvorm ziet dit er als volgt uit:

$$\{f\} = [K] \cdot \{u\} + \{f_q\} \quad (2.1)$$

In deze vergelijking is $[K]$ de stijfheidsmatrix, $\{f\}$ de vector met knoopkrachten, $\{u\}$ de vector met de knoopverplaatsingen en -rotaties en $\{f_q\}$ de vector met equivalente knoopkrachten. Omdat de krachten en verplaatsingen in het tweedimensionale vlak worden beschouwd, zijn er drie vrijheidsgraden per knoop. Dit resulteert in een 6×6 stijfheidsmatrix, die is opgebouwd uit de axiale stijfheid en de buigstijfheid. De analytische afleiding van deze twee componenten wordt in de volgende paragrafen stapsgewijs uitgewerkt.

2.2.1. Axiale stijfheid

Om de stijfheidsmatrix op te bouwen, wordt eerst de axiale stijfheid beschouwd. Hiervoor wordt uitgegaan van de differentiaalvergelijking voor de axiale verplaatsing, zoals beschreven in Hartsuijker en Welleman (2007, hoofdstuk 2):

$$EA \frac{d^2 u_x}{dx^2} = -q_x \quad (2.2)$$

Waarbij:

- E = Elasticiteitsmodulus $[\text{N/m}^2]$
- A = Oppervlakte van de doorsnede $[\text{m}^2]$
- u_x = Axiale verplaatsing $[\text{m}]$
- q_x = Verdeelde belasting over de lengteas $[\text{N/m}]$

De lineair variërende verdeelde belasting q_x tussen knoop 1 en knoop 2 wordt beschreven met de functie:

$$q_x = q_{x1} + \frac{q_{x2} - q_{x1}}{L} x$$

Door vergelijking 2.2 tweemaal naar x te integreren ontstaat de volgende algemene uitdrukking voor de axiale verplaatsing:

$$u_x(x) = \frac{1}{EA} \left(\frac{q_{x1} - q_{x2}}{6L} x^3 - \frac{1}{2} q_{x1} x^2 + C_1 x + C_2 \right) \quad (2.3)$$

De integratieconstanten C_1 en C_2 worden bepaald door de randvoorwaarden in te vullen. Voor een balkelement met lengte L zijn dit: $u_x(0) = u_{x1}$ en $u_x(L) = u_{x2}$. Voor het opstellen van de stijfheidsmatrix is de interne normaalkracht nodig:

$$N(x) = EA \frac{du_x}{dx}$$

Bij een verlenging van het element ($u_{x2} > u_{x1}$) ontstaat een interne trekkracht die het element terug wil trekken naar zijn oorspronkelijke lengte. Op de knooppunten werkt deze interne normaalkracht naar binnen toe (zie Figuur 2.2).



Figuur 2.2: Richting van de interne en externe normaalkracht op het balkelement (Eigen werk)

Omdat het element in statisch evenwicht is, moeten de externe knooppunten (F_{x1} en F_{x2}) evenwicht maken met de interne krachten om de knooppunten op hun positie te houden. Op basis van de richtingen als in Figuur 2.2 gedefinieerd volgt:

$$F_{x1} = -N(0) = \frac{EA}{L}(u_{x1} - u_{x2}) - \frac{1}{3}q_{x1}L - \frac{1}{6}q_{x2}L$$

$$F_{x2} = N(L) = \frac{EA}{L}(u_{x2} - u_{x1}) - \frac{1}{6}q_{x1}L - \frac{1}{3}q_{x2}L$$

In matrixvorm ziet dit er als volgt uit:

$$\begin{bmatrix} F_{x1} \\ F_{x2} \end{bmatrix} = \begin{bmatrix} \frac{EA}{L} & -\frac{EA}{L} \\ -\frac{EA}{L} & \frac{EA}{L} \end{bmatrix} \cdot \begin{bmatrix} u_{x1} \\ u_{x2} \end{bmatrix} + \begin{bmatrix} -\frac{1}{3}q_{x1}L - \frac{1}{6}q_{x2}L \\ -\frac{1}{6}q_{x1}L - \frac{1}{3}q_{x2}L \end{bmatrix} \quad (2.4)$$

2.2.2. Buigstijfheid

In de element-stijfheidsmatrix volgt de weerstand tegen buiging uit de vierde orde differentiaalvergelijking voor de doorbuiging van het element, zoals beschreven in Hartsuijker en Welleman (2007, hoofdstuk 4):

$$EI_{zz} \frac{d^4 u_z}{dx^4} = q_z \quad (2.5)$$

Waarbij:

- I_{zz} = Oppervlakte-traagheidsmoment [m^4]
- u_z = Transversale verplaatsing [m]
- q_z = Verdeelde dwarsbelasting over de lengteas [N/m]

De lineair variërende verdeelde belasting q_z is als volgt gedefinieerd:

$$q_z(x) = q_{z1} + \frac{q_{z2} - q_{z1}}{L} x$$

Door bovenstaande vergelijking te substitueren in vergelijking 2.5 en viermaal naar x te integreren, ontstaat een algemene uitdrukking voor de doorbuiging van het element:

$$u_z(x) = \frac{1}{EI} \left(\frac{q_{z2} - q_{z1}}{120L} x^5 + \frac{1}{24} q_{z1} x^4 + \frac{1}{6} C_1 x^3 + \frac{1}{2} C_2 x^2 + C_3 x + C_4 \right) \quad (2.6)$$

De vier resulterende integratieconstanten worden bepaald uit de randvoorwaarden in de knooppunten. Voor een balkelement met lengte L zijn deze:

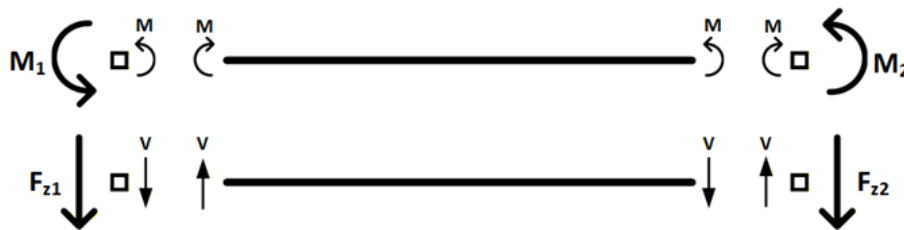
- Knooppunt 1 ($x = 0$): $u_z(0) = u_{z1}$ en $u_z'(0) = -\phi_1$
- Knooppunt 2 ($x = L$): $u_z(L) = u_{z2}$ en $u_z'(L) = -\phi_2$

Substitutie van deze randvoorwaarden in vergelijking 2.6 levert de oplossingen voor de integratieconstanten. Het interne buigend moment en de interne dwarskracht worden uitgedrukt in termen van de knoopverplaatsingen en -rotaties:

$$M(x) = -EI \frac{d^2 u_z}{dx^2}$$

$$V(x) = -EI \frac{d^3 u_z}{dx^3}$$

Bij een positief moment ontstaat onderin het balkelement trek en bovenin druk. Bij een positieve dwarskracht schuift de linkerzijde van het element omhoog ten opzichte van de rechterzijde.



Figuur 2.3: Richtingen van interne en externe moment en dwarskracht op het balkelement (Eigen werk)

Om het element in stilstand te houden, moeten de externe knooppunten en -momenten evenwicht maken met de interne krachten. Op basis van de aangenomen richtingen in Figuur 2.3 volgt:

$$F_{z1} = -V(0) = \frac{12 EI}{L^3} u_{z1} - \frac{6 EI}{L^2} \phi_1 - \frac{12 EI}{L^3} u_{z2} - \frac{6 EI}{L^2} \phi_2 - \frac{7}{20} q_{z1} L - \frac{3}{20} q_{z2} L$$

$$M_1 = -M(0) = -\frac{6 EI}{L^2} u_{z1} + \frac{4 EI}{L} \phi_1 + \frac{6 EI}{L^2} u_{z2} + \frac{2 EI}{L} \phi_2 + \frac{1}{20} q_{z1} L^2 + \frac{1}{30} q_{z2} L^2$$

$$F_{z2} = V(L) = -\frac{12 EI}{L^3} u_{z1} + \frac{6 EI}{L^2} \phi_1 + \frac{12 EI}{L^3} u_{z2} + \frac{6 EI}{L^2} \phi_2 - \frac{3}{20} q_{z1} L - \frac{7}{20} q_{z2} L$$

$$M_2 = M(L) = -\frac{6 EI}{L^2} u_{z1} + \frac{2 EI}{L} \phi_1 + \frac{6 EI}{L^2} u_{z2} + \frac{4 EI}{L} \phi_2 - \frac{1}{30} q_{z1} L^2 - \frac{1}{20} q_{z2} L^2$$

In matrixvorm ziet dit er als volgt uit:

$$\begin{bmatrix} F_{z1} \\ M_1 \\ F_{z2} \\ M_2 \end{bmatrix} = \begin{bmatrix} \frac{12 EI}{L^3} & -\frac{6 EI}{L^2} & -\frac{12 EI}{L^3} & -\frac{6 EI}{L^2} \\ -\frac{6 EI}{L^2} & \frac{4 EI}{L} & \frac{6 EI}{L^2} & \frac{2 EI}{L} \\ \frac{12 EI}{L^3} & \frac{6 EI}{L^2} & \frac{12 EI}{L^3} & \frac{6 EI}{L^2} \\ -\frac{6 EI}{L^2} & \frac{2 EI}{L} & \frac{6 EI}{L^2} & \frac{4 EI}{L} \end{bmatrix} \cdot \begin{bmatrix} u_{z1} \\ \phi_1 \\ u_{z2} \\ \phi_2 \end{bmatrix} + \begin{bmatrix} -\frac{7}{20} q_{z1} L - \frac{3}{20} q_{z2} L \\ \frac{1}{20} q_{z1} L^2 + \frac{1}{30} q_{z2} L^2 \\ -\frac{3}{20} q_{z1} L - \frac{7}{20} q_{z2} L \\ -\frac{1}{30} q_{z1} L^2 - \frac{1}{20} q_{z2} L^2 \end{bmatrix} \quad (2.7)$$

De volledige 2D element-stijfheidsmatrix combineert de axiale rek (vergelijking 2.4) en buiging (vergelijking 2.7) tot één systeem. De totale 2D-matrixvergelijking is opgenomen in bijlage A.

2.3. Implementatie in SymPy

Met behulp van de SymPy bibliotheek in Python is de 2D element-stijfheidsmatrix symbolisch afgeleid. Allereerst worden de benodigde parameters, zoals de materiaaleigenschappen, geometrie en de vrijheidsgraden gedefinieerd met de `sp.symbols()` functie. Vervolgens worden de vergelijkingen voor axiale stijfheid (2.2) en buigstijfheid (2.5) symbolisch opgebouwd met deze parameters. De `sp.solve()` functie lost de differentiaalvergelijking op waarna de randvoorwaarden worden toegepast om de integratieconstanten te bepalen en in te vullen in de vergelijking. Tot slot worden de resultaten met de `sp.Matrix()` functie omgezet naar de uiteindelijke matrixvorm. Het resultaat van de python afleiding is weergegeven in onderstaande figuur:

$$\begin{bmatrix} F_{x1} \\ F_{z1} \\ M_1 \\ F_{x2} \\ F_{z2} \\ M_2 \end{bmatrix} = \begin{bmatrix} \frac{L(-2q_{x1}-q_{x2})}{6} \\ \frac{L(-7q_{z1}-3q_{z2})}{20} \\ L^2 \left(\frac{q_{z1}}{20} + \frac{q_{z2}}{30} \right) \\ \frac{L(-q_{x1}-2q_{x2})}{6} \\ \frac{L(-3q_{z1}-7q_{z2})}{20} \\ L^2 \left(-\frac{q_{z1}}{30} - \frac{q_{z2}}{20} \right) \end{bmatrix} + \begin{bmatrix} \frac{AE}{L} & 0 & 0 & -\frac{AE}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} & 0 & \frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{AE}{L} & 0 & 0 & \frac{AE}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} & 0 & \frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{z1} \\ \phi_1 \\ u_{x2} \\ u_{z2} \\ \phi_2 \end{bmatrix}$$

Figuur 2.4: Resultaat van de symbolische afleiding van de 2D-stijfheidsmatrix met Python (Eigen werk)

Dit met de computer berekende resultaat komt exact overeen met de theoretische afleiding uit de voorgaande paragraaf. Het volledige Python-script is opgenomen in Bijlage B.

2.4. Validatie van de stijfheidsmatrix

Om te verifiëren of de symbolisch afgeleide matrix de werkelijkheid correct beschrijft, is een validatie uitgevoerd aan de hand van een aantal specifieke testgevallen. Dit is eerst met handberekeningen uitgewerkt, waarna de berekeningen in Python zijn uitgevoerd. De resultaten hiervan (zie bijlage C) tonen aan dat de stijfheidsmatrix zowel het rek- als buigingsgedrag van het element juist weergeeft.

Een andere fundamentele controle is de symmetrie van de stijfheidsmatrix langs de diagonaal. Dit volgt uit het wederkerigheidsprincipe van Maxwell, dat stelt dat in een lineair-elastisch systeem de verplaatsing van een punt i door een kracht op punt j gelijk is aan de verplaatsing van punt j door diezelfde kracht op punt i . In de stijfheidsmatrix moet daarom gelden dat $k_{ij} = k_{ji}$ (McGuire et al., 2000). Aangezien de afgeleide matrix ook hieraan voldoet, kan worden geconcludeerd dat de stijfheidsmatrix correct is afgeleid.

3. 3D-stijfheidsmatrix met vrije welving

In dit hoofdstuk wordt de driedimensionale element-stijfheidsmatrix afgeleid. De 2D matrix uit het vorige hoofdstuk vormt hiervoor het uitgangspunt en wordt uitgebreid met de extra vrijheidsgraden die nodig zijn in een 3D ruimte. Het hoofdstuk start met de definitie van het 3D-element en de theorie achter wringing met vrije welving. Daarna komen achtereenvolgens de analytische afleiding, de Python implementatie en de validatie van de stijfheidsmatrix aan bod.

3.1. Definitie van het 3D-balkelement

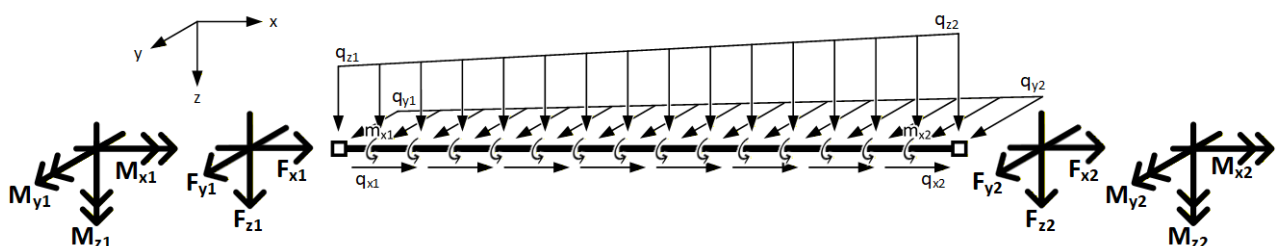
Voor de uitbreiding naar 3D wordt het cartesische assenstelsel van het balkelement aangevuld met een y-as. Deze y-as staat loodrecht op zowel de x-as als de z-as, zodanig dat een rechtsdraaiend coördinatensysteem ontstaat.

Het balkelement wordt net als in de 2D situatie, geschematiseerd met een lengte L en wordt aan beide uiteinden begrensd door een knoop waarop de externe krachten en momenten werken.

Elke knoop heeft zes vrijheidsgraden: een axiale translatie (u_x), een laterale translatie (u_y), een transversale translatie (u_z) en rotaties om de x-, y- en z-as (ϕ_x, ϕ_y, ϕ_z). Dit resulteert in twaalf vrijheidsgraden voor het volledige element.

Op het element kunnen verschillende externe belastingen werken. Dit zijn de knooppuntenkrachten (F_x, F_y, F_z) en knooppuntenmomenten (M_x, M_y, M_z) die direct op de knooppunten aangrijpen. Daarnaast er over de lengte van het element kunnen verdeelde lijnlasten (q_x, q_y en q_z) en een verdeeld wringend moment (m_x) werken. Alle belastingen volgen de positieve richtingen van het lokale assenstelsel. Voor de rotaties en momenten worden als tekenafspraken de rechterhandregel gehanteerd rondom de respectievelijke assen.

In Figuur 3.1 zijn het lokale assenstelsel en de aangenomen richtingen van de belastingen schematisch weergegeven.



Figuur 3.1: Schematisering van het 3D-balkelement inclusief tekenafspraken (Eigen werk)

3.2. Wringing volgens de theorie van Saint-Venant

In 1856 publiceerde de Franse ingenieur Adhémar Jean Claude Barré de Saint-Venant zijn theorie over wringing in het werk *Mémoire sur la torsion des prismes*. Eerdere theorieën van Coulomb en Navier veronderstelden dat een dwarsdoorsnede bij torsie altijd vlak bleef. Saint-Venant toonde echter aan dat deze aanname alleen geldig is voor cirkelvormige doorsnedes en dat niet-ronde profielen uit hun vlak vervormen, wat welving wordt genoemd (Timoshenko, 1953).

De theorie van Saint-Venant gaat uit van vrije welving. Hierbij wordt verondersteld dat het element aan de uiteinden niet wordt tegengehouden door inklemmingen of verbindingen, waardoor het welven vrij kan optreden. Dit betekent dat de dwarsdoorsnede over de gehele lengte van de balk hetzelfde vervormt, waardoor er geen longitudinale rek ontstaat. Daardoor treden er geen extra normaalspanningen op in het element, naast de schuifspanning ten gevolge van het wringend moment zelf. De inwendige weerstand tegen wringing wordt in deze theorie uitsluitend bepaald door de wringstijfheid van het element.

3.3. Afleiding van de stijfheidsrelaties

De axiale stijfheid en de buigstijfheid in het xz-vlak zijn hiervoor afgeleid in paragraaf 2.2 en blijven in deze driedimensionale situatie ongewijzigd. In het volgende deel worden de buigstijfheid in het xy-vlak en de torsiestijfheid analytisch uitgewerkt.

3.3.1. Buigstijfheid in het xy-vlak

Omdat de afleiding van de buigstijfheid in het xy-vlak sterk overeenkomt met de afleiding van de buigstijfheid in het xz-vlak uit het vorige hoofdstuk, wordt deze hier kort behandeld.

Voor de afleiding wordt dezelfde vierde orde differentiaalvergelijking voor buiging gebruikt:

$$EI_{yy} \frac{d^4 u_y}{dx^4} = q_y \quad (3.1)$$

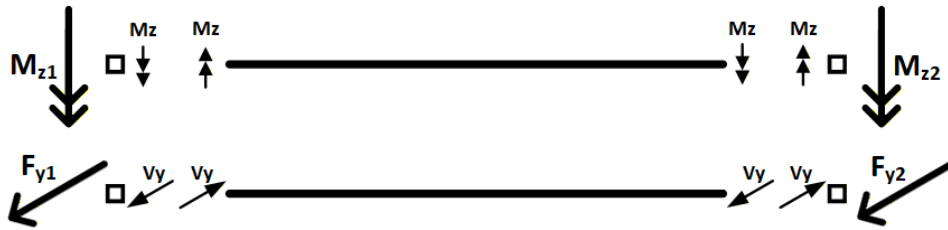
Waarbij:

- I_{yy} = Oppervlakte-traagheidsmoment om de z-as [m^4]
- u_y = Transversale verplaatsing [m]
- q_y = Verdeelde belasting over de lengteas in de y-richting [N/m]

Na viermaal integreren naar x en het invullen van de randvoorwaarden ontstaat een algemene uitdrukking voor de laterale doorbuiging van het element. Hiermee kunnen het interne buigend moment en de interne dwarskracht worden bepaald:

$$M_z(x) = EI_{yy} \frac{d^2 u_y}{dx^2} \quad V_y(x) = -EI_{yy} \frac{d^3 u_y}{dx^3}$$

Bij een positief moment ontstaat aan de achterzijde van het balkelement trek en aan de voorzijde druk. Bij een positieve dwarskracht schuift de linkerzijde van het element in de positieve y-richting ten opzichte van de rechterzijde.



Figuur 3.2: Richtingen van de interne en externe moment en dwarskracht op het balkelement (Eigen werk)

Om het element in statisch evenwicht te houden, moeten de externe knooppkrachten evenwicht maken met de interne krachten. Op basis van de richtingen zoals gedefinieerd in Figuur 3.2 volgt:

$$F_{y1} = -V_y(0), \quad M_{z1} = -M_z(0), \quad F_{y2} = V_y(L), \quad M_{z2} = M_z(L)$$

Uitgewerkt in matrixvorm ziet dit er als volgt uit:

$$\begin{bmatrix} F_{y1} \\ M_{z1} \\ F_{y2} \\ M_{z2} \end{bmatrix} = \begin{bmatrix} \frac{12 EI}{L^3} & \frac{6 EI}{L^2} & -\frac{12 EI}{L^3} & \frac{6 EI}{L^2} \\ \frac{6 EI}{L^2} & 4 EI & -\frac{6 EI}{L^2} & \frac{2 EI}{L} \\ -\frac{12 EI}{L^3} & -\frac{6 EI}{L^2} & \frac{12 EI}{L^3} & -\frac{6 EI}{L^2} \\ \frac{6 EI}{L^2} & \frac{2 EI}{L} & -\frac{6 EI}{L^2} & \frac{4 EI}{L} \end{bmatrix} \cdot \begin{bmatrix} u_{y1} \\ \phi_{z1} \\ u_{y2} \\ \phi_{z2} \end{bmatrix} + \begin{bmatrix} -\frac{7}{20} q_{y1} L - \frac{3}{20} q_{y2} L \\ -\frac{1}{20} q_{y1} L^2 - \frac{1}{30} q_{y2} L^2 \\ -\frac{3}{20} q_{y1} L - \frac{7}{20} q_{y2} L \\ \frac{1}{30} q_{y1} L^2 + \frac{1}{20} q_{y2} L^2 \end{bmatrix} \quad (3.2)$$

3.3.2. Torsiestijfheid

Voor de afleiding van de torsiestijfheid wordt uitgegaan van de differentiaalvergelijking van Saint-Venant, zoals besproken in paragraaf 3.2. Omdat naast een torsiemoment op de knooppunten ook een verdeeld wringmoment over de lengte van de balk kan werken, ontstaat de volgende vergelijking, zoals beschreven door Rajagopalan (2022, p. 98):

$$GI_t \frac{d^2 \phi}{dx^2} = -m_x \quad (3.3)$$

Waarbij:

- G = Glijdingsmodulus [N/m²]
- I_t = Torsietraagheidsmoment [m⁴]
- ϕ = Hoekverdraaiing [rad]
- m_x = Verdeelde wringbelasting over de lengteas [Nm/m]

De lineair variërende wringbelasting m_x wordt beschreven met de functie:

$$m_x = m_{x1} + \frac{m_{x2} - m_{x1}}{L} x$$

Door vergelijking 3.3 tweemaal naar x te integreren, ontstaat een algemene uitdrukking voor de axiale rotatie:

$$\phi_x(x) = \frac{1}{GI_t} \left(\frac{m_{x1} - m_{x2}}{6L} x^3 - \frac{1}{2} m_{x1} x^2 + C_1 x + C_2 \right) \quad (3.4)$$

De integratieconstanten worden bepaald door de randvoorwaarden in te vullen. Voor een balkelement met lengte L zijn dit:

- Knooppunt 1 ($x = 0$): $\phi_x(0) = \phi_{x1}$
- Knooppunt 2 ($x = L$): $\phi_x(L) = \phi_{x2}$

Substitutie van deze randvoorwaarden in vergelijking 3.4 levert een oplossing van de integratieconstanten. Hiermee kan het interne wringend moment worden uitgedrukt in termen van de knooproteraties:

$$T(x) = GI_t \frac{d\phi_x}{dx}$$

Bij een positief torsiemoment is de rechterzijde van het element met een hoek $\phi_{x2} - \phi_{x1}$ in de positieve draairichting getoerd ten opzichte van de linkerzijde.



Figuur 3.3: Richtingen interne en externe torsiemoment op het balkelement (Eigen werk)

Omdat het element in statisch evenwicht is, moeten het interne torsiemoment evenwicht maken met het externe torsiemoment. Op basis van de gedefinieerde richtingen in Figuur 3.3 volgt:

$$M_{x1} = -T(0) = \frac{GI_t}{L} (\phi_{x1} - \phi_{x2}) - \frac{1}{3} m_{x1} L - \frac{1}{6} m_{x2} L$$

$$M_{x2} = T(L) = \frac{GI_t}{L} (\phi_{x2} - \phi_{x1}) - \frac{1}{6} m_{x1} L - \frac{1}{3} m_{x2} L$$

In matrixvorm ziet dit er als volgt uit:

$$\begin{bmatrix} M_{x1} \\ M_{x2} \end{bmatrix} = \begin{bmatrix} \frac{GI_t}{L} & -\frac{GI_t}{L} \\ -\frac{GI_t}{L} & \frac{GI_t}{L} \end{bmatrix} \cdot \begin{bmatrix} \phi_{x1} \\ \phi_{x2} \end{bmatrix} + \begin{bmatrix} -\frac{1}{3} m_{x1} L - \frac{1}{6} m_{x2} L \\ -\frac{1}{6} m_{x1} L - \frac{1}{3} m_{x2} L \end{bmatrix} \quad (3.5)$$

De volledige 3D element-stijfheidsmatrix combineert vergelijking 2.4, 2.7, 3.2 en 3.5 en resulteert in de matrixvergelijking in bijlage D.

3.4. Implementatie in SymPy

Voor de symbolische afleiding van de 3D element-stijfheidsmatrix is het Python-script uit het vorige hoofdstuk verder uitgebreid. De methode komt sterk overeen met de 2D afleiding.

Allereerst worden de extra benodigde parameters en vrijheidsgraden toegevoegd met de `sp.symbols()` functie. De aanduiding van de variabelen is hierbij consistent gemaakt met de afleiding uit paragraaf 3.3. Vervolgens worden de vergelijkingen voor rek, buiging en torsie symbolisch opgebouwd met de parameters. De `sp.solve()` functie lost de differentiaalvergelijking op waarna de randvoorwaarden worden toegepast om de integratieconstanten te bepalen. Tot slot worden de resultaten met de `sp.Matrix()` functie samengevoegd tot de 12×12 matrixvorm. Het resultaat van de python afleiding is weergegeven in onderstaande figuur:

$$\begin{bmatrix} F_{x1} \\ F_{y1} \\ F_{z1} \\ M_{x1} \\ M_{y1} \\ M_{z1} \\ F_{x2} \\ F_{y2} \\ F_{z2} \\ M_{x2} \\ M_{y2} \\ M_{z2} \end{bmatrix} = \begin{bmatrix} \frac{L(-2q_{x1}-q_{x2})}{6} \\ \frac{L(-7q_{y1}-3q_{y2})}{20} \\ \frac{L(-7q_{z1}-3q_{z2})}{20} \\ \frac{L(-2m_{x1}-m_{x2})}{6} \\ L^2 \left(\frac{q_{x1}}{20} + \frac{q_{x2}}{30} \right) \\ L^2 \left(-\frac{q_{y1}}{20} - \frac{q_{y2}}{30} \right) \\ \frac{L(-q_{x1}-2q_{x2})}{6} \\ \frac{L(-3q_{y1}-7q_{y2})}{20} \\ \frac{L(-3q_{z1}-7q_{z2})}{20} \\ \frac{L(-m_{x1}-2m_{x2})}{6} \\ L^2 \left(\frac{q_{y1}}{30} + \frac{q_{y2}}{20} \right) \\ L^2 \left(\frac{q_{z1}}{30} + \frac{q_{z2}}{20} \right) \end{bmatrix} + \begin{bmatrix} \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & \frac{6EI_{yz}}{L^2} & 0 & -\frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & \frac{6EI_{yz}}{L^2} \\ 0 & 0 & \frac{12EI_{zz}}{L^3} & 0 & -\frac{6EI_{zz}}{L^2} & 0 & 0 & 0 & -\frac{12EI_{zz}}{L^3} & 0 & -\frac{6EI_{zz}}{L^2} & 0 \\ 0 & 0 & 0 & \frac{GI_t}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GI_t}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_{xz}}{L^2} & 0 & \frac{4EI_{xz}}{L} & 0 & 0 & \frac{6EI_{xz}}{L^2} & 0 & \frac{2EI_{xz}}{L} & 0 & 0 \\ 0 & \frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{4EI_{yz}}{L} & 0 & -\frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{2EI_{yz}}{L} \\ -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & -\frac{6EI_{yz}}{L^2} & 0 & \frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & -\frac{6EI_{yz}}{L^2} \\ 0 & 0 & -\frac{12EI_{zz}}{L^3} & 0 & \frac{6EI_{zz}}{L^2} & 0 & 0 & 0 & \frac{12EI_{zz}}{L^3} & 0 & \frac{6EI_{zz}}{L^2} & 0 \\ 0 & 0 & 0 & -\frac{GI_t}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GI_t}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_{xz}}{L^2} & 0 & \frac{2EI_{xz}}{L} & 0 & 0 & 0 & \frac{6EI_{xz}}{L^2} & 0 & \frac{4EI_{xz}}{L} & 0 \\ 0 & \frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{2EI_{yz}}{L} & 0 & -\frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{4EI_{yz}}{L} \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{z1} \\ \phi_{x1} \\ \phi_{y1} \\ \phi_{z1} \\ u_{x2} \\ u_{y2} \\ u_{z2} \\ \phi_{x2} \\ \phi_{y2} \\ \phi_{z2} \end{bmatrix}$$

Figuur 3.4: Resultaat van de symbolische afleiding van de 2D-stijfheidsmatrix met Python (Eigen werk)

Het resultaat komt overeen met de analytisch afgeleide matrixvergelijkingen. De volledige Python-code voor deze afleiding is opgenomen in Bijlage E.

3.5. Validatie van de stijfheidsmatrix

De 3D-stijfheidsmatrix is gecontroleerd of te verifiëren of de afleiding correct is uitgevoerd. Allereerst is conform het wederkerigheidsprincipe van Maxwell gecontroleerd of de matrix symmetrisch is ($K = K^T$), wat het geval is voor deze afgeleide stijfheidsmatrix.

Vervolgens zijn specifieke testgevallen doorgerekend, eerst met de hand en later met het Python-script, voor de termen die nieuw zijn geïntroduceerd bij de uitbreiding naar 3D (buiging in het xy-vlak en torsie). De overige termen zijn identiek aan de reeds gevalideerde 2D-matrix en vereisen geen aanvullende controle vanwege het ontkoppeld zijn van de vergelijkingen. De uitwerking van deze controleberekeningen waaruit blijkt dat de matrix het vervormingsgedrag juist weergeeft is opgenomen in bijlage F.

Ook is de matrix vergeleken met de vakliteratuur. De matrix komt exact overeen met de matrix zoals afgeleid door Hoogenboom (2019, p62). Hieruit kan geconcludeerd worden dat de 3D stijfheidsmatrix volgens de Saint-Venant theorie correct is afgeleid.

4. 3D-stijfheidsmatrix met verhinderde welving

In dit hoofdstuk wordt de driedimensionale element-stijfheidsmatrix opnieuw afgeleid, dit keer met wringing volgens de theorie van Vlasov (verhinderde welving). Hierbij worden dezelfde tekenafspraken gehanteerd als in Figuur 3.1 voor het 3D-balkelement. Het hoofdstuk start met de theoretische achtergrond van de wringingstheorie van Vlasov. Vervolgens wordt de analytische afleiding besproken en tot slot wordt de Python implementatie en validatie van de stijfheidsmatrix behandeld.

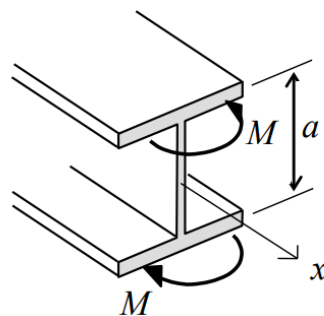
4.1. Wringing volgens de theorie van Vlasov

De theorie van Saint-Venant biedt een nauwkeurige beschrijving voor torsie bij elementen waarbij de dwarsdoorsnede vrij kan welven. In de praktijk worden elementen echter vaak aan de uiteinden gefixeerd door inklemmingen of stijve verbindingen, waardoor de welvingsvervorming wordt verhinderd. De Russische ingenieur Vasily Zacharovich Vlasov publiceerde in 1940 zijn theorie over dunwandige staven, waarin hij aantoonde dat bij verhinderde welving extra trek- en drukspanningen in de lengterichting van het element ontstaan. Om deze interne normaalspanningen te kunnen beschrijven introduceerde Vlasov het bi-moment (Hoogenboom, 2019).

Een bi-moment (B) is een koppel van twee tegengestelde buigende momenten die op een afstand van elkaar op de doorsnede werken. Het wordt berekend met de formule:

$$B = M \cdot a$$

Waarbij M het buigend moment is en a de afstand tussen het momentenkoppel, wat resulteert in de eenheid [Nm^2]. In Figuur 4.1 is de werking van het bi-moment op een I-profiel weergegeven. Hier is duidelijk te zien hoe het bi-moment is opgebouwd uit twee tegengestelde momenten in de flenzen.



Figuur 4.1: Bi-moment in een I-profiel (Hoogenboom, 2019)

Bij verhinderde welving worden specifieke delen van het profiel uitgerekt terwijl andere delen worden ingedrukt. Dit resulteert in de extra normaalspanningen in de lengterichting van het element. Het bi-moment is de kracht die deze interne spanningen beschrijft. Omdat de spanningen de verdraaiing van het element tegenwerken, neemt de torsiestijfheid van het element toe.

Als tekenafspraken wordt aangehouden dat het bi-moment positief is wanneer de welving vervorming van de doorsnede overeenkomt met de vervorming door een positief wringmoment. In onderstaande afbeelding is de richting van het positieve bi-moment weergegeven.



Figuur 4.2: Balkelement met tekenafspraken bi-moment (Eigen werk)

4.2. Afleiding van de stijfheidsrelaties

Voor wringing met verhinderde welving wordt de Vlasov differentiaalvergelijking gebruikt, zoals gedefinieerd door Rajagopalan (2022, p. 90):

$$EC_w \frac{d^4 \phi}{dx^4} - GI_t \frac{d^2 \phi}{dx^2} = m_x \quad (4.1)$$

Waarbij:

- C_w = Welvingsconstante [m^6]

De lineair variërende wringbelasting m_x wordt ook hier beschreven met de functie:

$$m_x = m_{x1} + \frac{m_{x2} - m_{x1}}{L} x$$

Omdat vergelijking 4.1 een niet homogene lineaire differentiaalvergelijking van de vierde orde is, wordt de algemene oplossing verkregen door de homogene en particuliere oplossing afzonderlijk te berekenen en deze bij elkaar op te tellen. Hierbij beschrijft de homogene oplossing de stijfheid van het balkelement en de particuliere oplossing de invloed van het verdeeld wringend moment.

Allereerst wordt de homogene oplossing afgeleid. Hiervoor wordt m_x gelijk aan 0 gesteld en wordt een aanname gedaan voor de vorm van de homogene oplossing:

$$\phi_h(x) = e^{rx}$$

Deze vorm wordt gekozen omdat de e-macht bij herhaaldelijk differentiëren dezelfde vorm behoudt. Dit is nodig omdat in de differentiaalvergelijking verschillende ordes van afgeleiden tegelijk voorkomen. Door deze functie en zijn afgeleiden in te vullen in vergelijking 4.1 ontstaat de karakteristieke vergelijking:

$$EC_w r^4 - GI_t r^2 = 0$$

Deze vergelijking oplossen voor r geeft:

$$r_1 = 0, \quad r_2 = 0, \quad r_3 = \sqrt{\frac{GI_t}{EC_w}}, \quad r_4 = -\sqrt{\frac{GI_t}{EC_w}}$$

Dit invullen in de homogene vergelijking geeft:

$$\phi_h(x) = C_1 + C_2x + C_3 \sinh\left(\sqrt{\frac{GI_t}{EC_w}}x\right) + C_4 \cosh\left(\sqrt{\frac{GI_t}{EC_w}}x\right) \quad (4.2)$$

Hierna wordt de particuliere oplossing afgeleid. Omdat de verdeelde torsiebelasting lineair varieert over de lengte van het element, wordt voor de particuliere oplossing de volgende algemene vorm aangenomen:

$$\phi_p(x) = Ax^3 + Bx^2$$

Dit invullen in de oorspronkelijke vergelijking 4.1 zorgt ervoor dat de constanten A en B kunnen worden opgelost door de termen aan beide zijden van de vergelijking aan elkaar gelijk te stellen:

$$\begin{aligned} EC_w \cdot 0 - GI_t \cdot (6Ax + 2B) &= \frac{m_{x2} - m_{x1}}{L}x + m_{x1} \\ A &= \frac{m_{x1} - m_{x2}}{6GI_tL}, \quad B = -\frac{m_{x1}}{2GI_t} \\ \phi_p(x) &= \frac{m_{x1} - m_{x2}}{6GI_tL}x^3 - \frac{m_{x1}}{2GI_t}x^2 \end{aligned} \quad (4.3)$$

De totale oplossing voor de hoekverdraaiing ϕ is de som van de homogene en particuliere oplossing uit vergelijking 4.2 en 4.3:

$$\phi_x(x) = C_1 + C_2x + C_3 \sinh\left(\sqrt{\frac{GI_t}{EC_w}}x\right) + C_4 \cosh\left(\sqrt{\frac{GI_t}{EC_w}}x\right) - \frac{m_{x1}}{2GI_t}x^2 + \frac{m_{x1} - m_{x2}}{6GI_tL}x^3 \quad (4.4)$$

De integratieconstanten worden opgelost door de randvoorwaarden in de knooppunten in te vullen in bovenstaande vergelijking. Voor een balkelement met lengte L zijn dit:

- Knooppunt 1 ($x = 0$): $\phi_x(0) = \phi_{x1}$ en $\phi'_x(0) = -w_1$
- Knooppunt 2 ($x = L$): $\phi_x(L) = \phi_{x2}$ en $\phi'_x(L) = -w_2$

Hiermee kunnen het interne torsiemoment en het interne bi-moment worden uitgedrukt in termen van de knooproductie en welving:

$$B = -EC_w \frac{d^2\phi}{dx^2} = \frac{EC_w(m_{x1} - m_{x2})}{GI_tL}x + \frac{EC_w m_{x1}}{GI_t} - C_3GI_t \cosh(\lambda x) - C_4GI_t \sinh(\lambda x)$$

$$T = GI_t \frac{d\phi}{dx} - EC_w \frac{d^3\phi}{dx^3} = \frac{m_{x1} - m_{x2}}{2L}x^2 - m_{x1}x - \frac{EC_w(m_{x1} - m_{x2})}{GI_tL} + w_1GI_t +$$

$$C_3(\lambda GI_t \sinh(\lambda x) - \lambda^3 EC_w \sinh(\lambda x)) - C_4(\lambda GI_t(1 + \cosh(\lambda x)) + \lambda^3 EC_w \cosh(\lambda x))$$

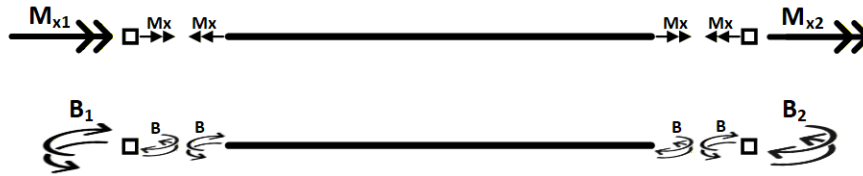
Waarbij:

$$C_3 = \frac{\left((\phi_{x2} - \phi_{x1} - w_1 L + \frac{m_{x1}}{2GI_t} L^2 + \frac{m_{x2} - m_{x1}}{6GI_t} L^2) \cdot (\lambda \cosh(\lambda L) - \lambda) - \left(w_2 - w_1 + \frac{m_{x1}}{GI_t} L + \frac{m_{x2} - m_{x1}}{2GI_t} L \right) \cdot (\sinh(\lambda L) - \lambda L) \right)}{\lambda (\cosh(\lambda L) - 1)^2 - (\lambda \sinh(\lambda L)) \cdot (\sinh(\lambda L) - \lambda L)}$$

$$C_4 = \frac{\left((\phi_{x2} - \phi_{x1} - w_1 L + \frac{m_{x1}}{2GI_t} L^2 + \frac{m_{x2} - m_{x1}}{6GI_t} L^2) \cdot (\lambda \sinh(\lambda L)) - \left(w_2 - w_1 + \frac{m_{x1}}{GI_t} L + \frac{m_{x2} - m_{x1}}{2GI_t} L \right) \cdot (\cosh(\lambda L) - 1) \right)}{\lambda (\cosh(\lambda L) - 1)^2 - (\lambda \sinh(\lambda L)) \cdot (\sinh(\lambda L) - \lambda L)}$$

$$\lambda = \sqrt{\frac{GI_t}{EC_w}}$$

Omdat het element in statisch evenwicht is, moeten de externe knooppunten (M_{x1}, M_{x2}, B_1, B_2) evenwicht maken met de interne krachten om de knooppunten op hun positie te houden.



Figuur 4.3: Richtingen interne en externe torsie en bi-moment op het balkelement (Eigen werk)

Hieruit volgt:

$$M_{x1} = -T(0), \quad B_1 = -B(0), \quad M_{x2} = T(L), \quad B_2 = B(L)$$

Gezien de grote breuken en de aanzienlijke kans op algebraïsche fouten bij handmatige uitwerking, is ervoor gekozen om meteen SymPy te gebruiken voor het afleiden van de element-stijfheidsmatrix. De matrix ziet er als volgt uit:

$$\begin{bmatrix} M_{x1} \\ B_1 \\ M_{x2} \\ B_2 \end{bmatrix} = \begin{bmatrix} \frac{C_w E \lambda^3 (e^{L\lambda} + 1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & \frac{C_w E \lambda^2 (e^{L\lambda} - 1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & -\frac{C_w E \lambda^3 (e^{L\lambda} + 1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & \frac{C_w E \lambda^2 (e^{L\lambda} - 1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} \\ \frac{C_w E \lambda^2 (e^{L\lambda} - 1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & \frac{C_w E \lambda (L\lambda e^{2L\lambda} + L\lambda - e^{2L\lambda} + 1)}{L\lambda e^{2L\lambda} - L\lambda - 2e^{2L\lambda} + 4e^{L\lambda} - 2} & \frac{C_w E \lambda^2 \cdot (1 - e^{L\lambda})}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & \frac{C_w E \lambda (-2L\lambda e^{L\lambda} + e^{2L\lambda} - 1)}{L\lambda e^{2L\lambda} - L\lambda - 2e^{2L\lambda} + 4e^{L\lambda} - 2} \\ -\frac{C_w E \lambda^3 (e^{L\lambda} + 1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & \frac{C_w E \lambda^2 \cdot (1 - e^{L\lambda})}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & \frac{C_w E \lambda^3 (e^{L\lambda} + 1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & \frac{C_w E \lambda^2 \cdot (1 - e^{L\lambda})}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} \\ \frac{C_w E \lambda^2 (e^{L\lambda} - 1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & \frac{C_w E \lambda (-2L\lambda e^{L\lambda} + e^{2L\lambda} - 1)}{L\lambda e^{2L\lambda} - L\lambda - 2e^{2L\lambda} + 4e^{L\lambda} - 2} & \frac{C_w E \lambda^2 \cdot (1 - e^{L\lambda})}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & \frac{C_w E \lambda (L\lambda e^{2L\lambda} + L\lambda - e^{2L\lambda} + 1)}{L\lambda e^{2L\lambda} - L\lambda - 2e^{2L\lambda} + 4e^{L\lambda} - 2} \end{bmatrix} \cdot \begin{bmatrix} \Phi_{x1} \\ w_1 \\ \Phi_{x2} \\ w_2 \end{bmatrix} + f_q \quad (4.5)$$

$$\text{Met } f_q = \begin{bmatrix} \frac{m_{x1} (-2L^3 \lambda^3 - 3L^2 \lambda^2 + 6L\lambda + (-2L^3 \lambda^3 + 3L^2 \lambda^2 + 6L\lambda - 12) e^{L\lambda} + 12) + m_{x2} (-L^3 \lambda^3 - 3L^2 \lambda^2 - 6L\lambda + (-L^3 \lambda^3 + 3L^2 \lambda^2 - 6L\lambda + 12) e^{L\lambda} - 1)}{6L\lambda^2 (L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2)} \\ \frac{m_{x1} (-2L^2 \lambda^2 - 9L\lambda + (-2L^2 \lambda^2 + 24) e^{L\lambda} + (-2L^2 \lambda^2 + 9L\lambda - 12) e^{2L\lambda} - 12) + m_{x2} (-4L^2 \lambda^2 e^{L\lambda} - L^2 \lambda^2 - 3L\lambda + (-L^2 \lambda^2 + 3L\lambda) e^{2L\lambda})}{6\lambda^2 (e^{L\lambda} - 1) (L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2)} \\ \frac{m_{x1} (-L^3 \lambda^3 - 3L^2 \lambda^2 - 6L\lambda + (-L^3 \lambda^3 + 3L^2 \lambda^2 - 6L\lambda + 12) e^{L\lambda} - 12) + m_{x2} (-2L^3 \lambda^3 - 3L^2 \lambda^2 + 6L\lambda + (-2L^3 \lambda^3 + 3L^2 \lambda^2 + 6L\lambda - 12) e^{L\lambda} + 12)}{6L\lambda^2 (L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2)} \\ \frac{m_{x1} (4L^2 \lambda^2 e^{L\lambda} + L^2 \lambda^2 + 3L\lambda + (L^2 \lambda^2 - 3L\lambda) e^{2L\lambda}) + m_{x2} (2L^2 \lambda^2 + 9L\lambda + (2L^2 \lambda^2 - 24) e^{L\lambda} + (2L^2 \lambda^2 - 9L\lambda + 12) e^{2L\lambda} + 12)}{6\lambda^2 (e^{L\lambda} - 1) (L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2)} \end{bmatrix}$$

De volledige 14×14 matrix volgt uit het samenvoegen van vergelijkingen 2.4, 2.7, 3.2 en 4.5 en is in zijn geheel te vinden in bijlage G.

4.3. Implementatie in SymPy

De element-stijfheidsmatrix met verhinderde welving is symbolisch afgeleid in Python, waarbij is voortgebouwd op de bestaande SymPy-code uit bijlage B en E.

Eerst worden de parameters met de `sp.symbols()` functie gedefinieerd. Voor de Vlasov torsie zijn hierbij de welvingsconstante C_w en de extra vrijheidsgraden voor welving (w_1 en w_2) aan het script toegevoegd. Vervolgens worden de vergelijkingen voor axiale rek, buiging en torsie met verhinderde welving symbolisch opgebouwd. De differentiaalvergelijking voor torsie is opgelost door deze expliciet op te splitsen in een homogene en particuliere oplossing. Om het rekenwerk te vereenvoudigen is een parameter λ geïntroduceerd, waarbij $\lambda = \sqrt{GI_t/EC_w}$. Dit resulteert in een compactere matrixnotatie en een efficiënter rekenproces, waardoor de computer de matrix sneller kan oplossen. Na het toepassen van de randvoorwaarden waarmee de integratieconstanten worden bepaald, worden de resultaten met de `sp.Matrix()` functie omgeschreven naar de matrixvorm. Het resultaat van de python afleiding hieronder weergegeven:

$$\begin{bmatrix} F_{z1} \\ F_{y1} \\ F_{x1} \\ M_{z1} \\ M_{y1} \\ M_{x1} \\ B_1 \\ F_{z2} \\ F_{y2} \\ F_{x2} \\ M_{z2} \\ M_{y2} \\ M_{x2} \\ B_2 \end{bmatrix} = \begin{bmatrix} -\frac{Lq_{z1}}{3} - \frac{Lq_{y1}}{6} \\ -\frac{7Lq_{z1}}{20} - \frac{3Lq_{y1}}{20} \\ -\frac{7Lq_{z1}}{20} - \frac{3Lq_{y1}}{20} \\ m_{z1}(-2L^2\lambda^2 - 3L^2\lambda^2 + 6L\lambda + (-2L^2\lambda^2 + 3L^2\lambda^2 + 6L\lambda - 12)e^{\lambda L} + 12) + m_{y1}(-L^2\lambda^2 - 3L^2\lambda^2 - 6L\lambda + (-L^2\lambda^2 + 3L^2\lambda^2 - 6L\lambda + 12)e^{\lambda L} - 12) \\ \frac{L^2q_{z1}}{30} + \frac{L^2q_{y1}}{30} \\ -\frac{L^2q_{z1}}{20} - \frac{L^2q_{y1}}{30} \\ m_{z1}(2L^2\lambda^2 + 9L\lambda + (2L^2\lambda^2 - 24)e^{\lambda L} + (2L^2\lambda^2 - 9L\lambda + 12)e^{2\lambda L} + 12) + m_{y1}(4L^2\lambda^2 e^{\lambda L} + L^2\lambda^2 + 3L\lambda + (L^2\lambda^2 - 3L\lambda)e^{2\lambda L}) \\ \frac{6\lambda^3(e^{\lambda L} - 1)(L\lambda e^{\lambda L} + L\lambda - 2e^{2\lambda L} + 2)}{6\lambda^3(e^{\lambda L} - 1)(L\lambda e^{\lambda L} + L\lambda - 2e^{2\lambda L} + 2)} \\ -\frac{Lq_{z1}}{6} - \frac{Lq_{y1}}{3} \\ -\frac{3Lq_{z1}}{20} - \frac{7Lq_{y1}}{20} \\ -\frac{3Lq_{z1}}{20} - \frac{7Lq_{y1}}{20} \\ m_{z1}(-L^2\lambda^2 - 3L^2\lambda^2 - 6L\lambda + (-L^2\lambda^2 + 3L^2\lambda^2 - 6L\lambda + 12)e^{\lambda L} - 12) + m_{y1}(-2L^2\lambda^2 - 3L^2\lambda^2 + 6L\lambda + (-2L^2\lambda^2 + 3L^2\lambda^2 + 6L\lambda - 12)e^{\lambda L} + 12) \\ \frac{L^2q_{z1}}{30} + \frac{L^2q_{y1}}{30} \\ -\frac{L^2q_{z1}}{20} - \frac{L^2q_{y1}}{20} \\ m_{z1}(-4L^2\lambda^2 e^{\lambda L} - L^2\lambda^2 - 3L\lambda + (-L^2\lambda^2 + 3L\lambda)e^{2\lambda L}) + m_{y1}(-2L^2\lambda^2 - 9L\lambda + (-2L^2\lambda^2 + 24)e^{\lambda L} + (-2L^2\lambda^2 + 9L\lambda - 12)e^{2\lambda L} - 12) \\ \frac{6\lambda^3(e^{\lambda L} - 1)(L\lambda e^{\lambda L} + L\lambda - 2e^{2\lambda L} + 2)}{6\lambda^3(e^{\lambda L} - 1)(L\lambda e^{\lambda L} + L\lambda - 2e^{2\lambda L} + 2)} \end{bmatrix} + \begin{bmatrix} -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & \frac{6EI_{yy}}{L^2} & 0 & 0 & -\frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & \frac{6EI_{yy}}{L^2} & 0 & 0 \\ 0 & 0 & \frac{12EI_{zz}}{L^3} & 0 & -\frac{6EI_{zz}}{L^2} & 0 & 0 & 0 & 0 & 0 & -\frac{12EI_{zz}}{L^3} & 0 & -\frac{6EI_{zz}}{L^2} & 0 & 0 \\ 0 & 0 & 0 & \frac{C_w EA^2 (e^{\lambda L} + 1)}{L\lambda e^{\lambda L} + L\lambda - 2e^{2\lambda L} + 2} & 0 & 0 & \frac{C_w EA^2 (1 - e^{\lambda L})}{L\lambda e^{\lambda L} + L\lambda - 2e^{2\lambda L} + 2} & 0 & 0 & 0 & -\frac{C_w EA^2 (e^{\lambda L} + 1)}{L\lambda e^{\lambda L} + L\lambda - 2e^{2\lambda L} + 2} & 0 & 0 & \frac{C_w EA^2 (1 - e^{\lambda L})}{L\lambda e^{\lambda L} + L\lambda - 2e^{2\lambda L} + 2} & 0 \\ 0 & 0 & -\frac{6EI_{zz}}{L^2} & 0 & \frac{4EI_{zz}}{L} & 0 & 0 & 0 & 0 & 0 & \frac{6EI_{zz}}{L^2} & 0 & \frac{2EI_{zz}}{L} & 0 & 0 \\ 0 & \frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{4EI_{yy}}{L} & 0 & 0 & -\frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{2EI_{yy}}{L} & 0 & 0 \\ 0 & 0 & 0 & \frac{C_w EA^2 (1 - e^{\lambda L})}{L\lambda e^{\lambda L} + L\lambda - 2e^{2\lambda L} + 2} & 0 & 0 & \frac{C_w EA (L\lambda e^{2\lambda L} + L\lambda - e^{2\lambda L} + 1)}{L\lambda e^{2\lambda L} - L\lambda - 2e^{2\lambda L} + 4e^{\lambda L} - 2} & 0 & 0 & 0 & \frac{C_w EA^2 (e^{\lambda L} - 1)}{L\lambda e^{\lambda L} + L\lambda - 2e^{2\lambda L} + 2} & 0 & 0 & \frac{C_w EA (-2L\lambda e^{\lambda L} + e^{2\lambda L} - 1)}{L\lambda e^{2\lambda L} - L\lambda - 2e^{2\lambda L} + 4e^{\lambda L} - 2} & 0 \\ -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & -\frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & 0 & \frac{12EI_{yy}}{L^3} & 0 & 0 & -\frac{6EI_{yy}}{L^2} & 0 \\ 0 & 0 & -\frac{12EI_{zz}}{L^3} & 0 & \frac{6EI_{zz}}{L^2} & 0 & 0 & 0 & 0 & 0 & -\frac{12EI_{zz}}{L^3} & 0 & \frac{6EI_{zz}}{L^2} & 0 & 0 \\ 0 & 0 & 0 & -\frac{C_w EA^2 (e^{\lambda L} + 1)}{L\lambda e^{\lambda L} + L\lambda - 2e^{2\lambda L} + 2} & 0 & 0 & \frac{C_w EA^2 (e^{\lambda L} - 1)}{L\lambda e^{\lambda L} + L\lambda - 2e^{2\lambda L} + 2} & 0 & 0 & 0 & \frac{C_w EA^2 (e^{\lambda L} + 1)}{L\lambda e^{\lambda L} + L\lambda - 2e^{2\lambda L} + 2} & 0 & 0 & \frac{C_w EA^2 (e^{\lambda L} - 1)}{L\lambda e^{\lambda L} + L\lambda - 2e^{2\lambda L} + 2} & 0 \\ 0 & 0 & -\frac{6EI_{zz}}{L^2} & 0 & \frac{2EI_{zz}}{L} & 0 & 0 & 0 & 0 & 0 & \frac{6EI_{zz}}{L^2} & 0 & \frac{4EI_{zz}}{L} & 0 & 0 \\ 0 & \frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{2EI_{yy}}{L} & 0 & 0 & -\frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{4EI_{yy}}{L} & 0 & 0 \\ 0 & 0 & 0 & \frac{C_w EA^2 (1 - e^{\lambda L})}{L\lambda e^{\lambda L} + L\lambda - 2e^{2\lambda L} + 2} & 0 & 0 & \frac{C_w EA (-2L\lambda e^{\lambda L} + e^{2\lambda L} - 1)}{L\lambda e^{2\lambda L} - L\lambda - 2e^{2\lambda L} + 4e^{\lambda L} - 2} & 0 & 0 & 0 & \frac{C_w EA^2 (e^{\lambda L} - 1)}{L\lambda e^{\lambda L} + L\lambda - 2e^{2\lambda L} + 2} & 0 & 0 & \frac{C_w EA (L\lambda e^{2\lambda L} + L\lambda - e^{2\lambda L} + 1)}{L\lambda e^{2\lambda L} - L\lambda - 2e^{2\lambda L} + 4e^{\lambda L} - 2} & 0 \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{z1} \\ \phi_{z1} \\ \phi_{y1} \\ \phi_{x1} \\ w_1 \\ u_{z2} \\ u_{y2} \\ u_{x2} \\ \phi_{z2} \\ \phi_{y2} \\ \phi_{x2} \\ w_2 \end{bmatrix}$$

Figuur 4.4: Resultaat van de symbolische afleiding van de 2D-stijfheidsmatrix met Python (Eigen werk)

De volledige Python code voor deze afleiding is opgenomen in Bijlage H.

4.4. Validatie van de stijfheidsmatrix

De validatie van de verkregen stijfheidsmatrix is een belangrijke laatste stap om de correctheid van het SymPy-script te verifiëren. Allereerst is gecontroleerd of de matrix symmetrisch is langs de diagonaal, de matrix voldoet aan deze eis.

Vervolgens zijn met twee specifieke testgevallen de nieuw afgeleide matrixtermen voor de Vlasov wringing gecontroleerd. Omdat de matrixtermen voor axiale rek en buiging in de voorgaande hoofdstukken reeds gevalideerd zijn, vereisen deze aanvullende controle. Hiervoor zijn de specifieke testsituaties met de hand afgeleid en vergeleken met de uitkomst van het Python-script. Uit deze controleberekeningen in bijlage I blijkt dat de stijfheidsmatrix het juiste mechanische gedrag vertoont bij verhinderde welving.

Tot slot komen de afgeleide stijfheidsmatrix en de belastingvector overeen met de oplossing van Hoogenboom (2006, p.12). Het tekenverschil bij een aantal termen is het gevolg van een tegengestelde aanname voor de positieve richting van de welving en het bi-moment.

Op basis van deze controles kan worden geconcludeerd dat de 3D stijfheidsmatrix met wringing volgens de theorie van Vlasov correct is afgeleid.

5. Discussie

In dit onderzoek is de 3D element-stijfheidsmatrix afgeleid voor balkelementen met zowel vrije als verhinderde welving. Wanneer de opbouw van de verkregen matrices wordt geanalyseerd, valt op dat de matrix bestaat uit de samenvoeging van losse, onafhankelijke vergelijkingen. De axiale rek, buiging om de y-as, buiging om de z-as en torsie zijn volledig van elkaar ontkoppeld. Dit betekent dat de axiale stijfheid geen invloed heeft op buig- en wringstijfheid van het element en omgekeerd. In de stijfheidsmatrix is deze ont koppeling goed zichtbaar aan de grote hoeveelheid termen die gelijk zijn aan nul. Dit komt doordat het normaalkrachten centrum en het dwarskrachten centrum van de doorsnede exact in hetzelfde punt vallen. Dit laat meteen een beperking van de afleiding zien: de matrix is namelijk niet bruikbaar voor asymmetrische profielen (zoals U- of L-profielen) waarbij het zwaartepunt niet samenvalt met het dwarskrachten centrum. In dat geval zou een excentriciteit in de vergelijkingen moeten worden meegenomen, waardoor een dwarskracht op het element bijvoorbeeld ook direct wringing veroorzaakt.

Wanneer de resultaten van de controleberekeningen worden vergeleken, wordt het verschil in torsiestijfheid tussen de theorie van Saint-Venant en de theorie van Vlasov goed zichtbaar. Voor een uitkragende ligger met een wringend moment M op het uiteinde resulteert de uitkomst met vrije welving in een hoekverdraaiing van: $\phi_{x2} = \frac{ML}{GI_t}$. Bij de uitkomst met verhinderde welving is dit: $\phi_{x2} = \frac{ML}{GI_t} \left(1 - \frac{\tanh(\lambda L)}{\lambda L} \right)$ waarbij $\lambda = \sqrt{GI_t/EC_w}$. De basis is bij beide uitkomsten gelijk, maar bij verhinderde welving gaat er nog een term $\frac{\tanh(\lambda L)}{\lambda L}$ van af. Omdat λ en L beide positief zijn is deze term altijd positief, waardoor de hoekverdraaiing bij de theorie van Vlasov kleiner is dan bij de theorie van Saint-Venant. Het element vervormt dus minder onder dezelfde torsiebelasting, wat de toegenomen torsiestijfheid bij verhinderde welving bevestigt. Om de grootte van deze toegenomen stijfheid inzichtelijk te maken, is een specifiek testgeval doorgerekend: een HEA160 profiel met een lengte van 3 meter met een wringend moment van 500 Nm op het uiteinde. Vergelijkbaar met een verkeersbord boven de weg waar windbelasting op werkt. Hieruit blijkt dat de hoekverdraaiing door de verhinderde welving bij Vlasov 27.2% kleiner uitvalt dan bij de theorie van Saint-Venant (zie bijlage J).

Een kritische kanttekening moet worden geplaatst bij de validatie van de afgeleide stijfheidsmatrices. In dit onderzoek zijn de matrices uitsluitend gecontroleerd met simpele en opzichzelfstaande testgevallen, zoals een ligger met enkel buiging of een wringend moment. Bij deze validatie zijn echter geen complexere belastingsituaties getest, zoals een trapeziumvormige verdeelde lijnlast. Hoewel de theoretische afleiding wel is ontworpen op deze lineair variërende belasting, is de werking hiervan niet met controleberekeningen getest. Om goed te kunnen vaststellen of de stijfheidsmatrix ook in deze complexe situaties correct functioneert, hadden er uitgebreidere controles uitgevoerd moeten worden.

6. Conclusies en aanbevelingen

In dit hoofdstuk worden de belangrijkste conclusies uit het onderzoek gepresenteerd. Daarnaast worden er concrete aanbevelingen gedaan voor de verdere ontwikkeling en validatie van de afgeleide element-stijfheidsmatrix.

6.1. Conclusies

Het hoofddoel van dit onderzoek was de analytische afleiding en Python implementatie van een 3D element-stijfheidsmatrix waarin wringing en verhinderde welving correct zijn gemodelleerd. Door de theorie van Vlasov toe te passen, worden de extra interne trek- en drukspanningen in de lengterichting van het element, die ontstaan wanneer de welvingsvervorming wordt verhinderd door de randvoorwaarden, correct in de stijfheid meegenomen. Dit resulteert in een hogere torsiestijfheid van het element in vergelijking met Saint-Venant theorie, die uitgaat van vrije welving.

De verificatie met testgevallen laat zien dat de symbolische afleiding en de daaropvolgende implementatie in Python het gedrag van het balkelement onder zuivere belastingen correct beschrijven. Een belangrijke randvoorwaarde hierbij is dat de afleiding alleen toepasbaar is op dubbel symmetrische profielen, aangezien axiale rek, buiging en wringing volledig van elkaar ontkoppeld zijn in de stijfheidsmatrix. Bij asymmetrische profielen valt het normaalkrachten- en dwarskrachtencentrum van de doorsnede niet samen, waardoor deze ontkoppeling niet meer geldt.

Desondanks is met dit onderzoek een volledig open en controleerbare afleiding in Python gerealiseerd waarmee wringing en verhinderde welving gemodelleerd kunnen worden. Deze Python implementatie biedt een concrete basis die door derden gebruikt kan worden voor het schrijven van hun eigen 3D-raamwerkprogramma. Daarnaast kan het gebruikt worden om de uitkomsten van commerciële raamwerkprogramma's te controleren, omdat de onderliggende stijfheidsmatrices daarvan vaak niet openbaar zijn.

Tot slot benadrukt dit onderzoek het voordeel van de Vlasov theorie ten opzichte van de theorie van Saint-Venant. Hoewel de theorie van Saint-Venant met vrije welving door de minder complexe vergelijkingen snellere berekeningen mogelijk maakt, is deze methode minder nauwkeurig. De Saint-Venant stijfheidsmatrix resulteert namelijk in grotere vervormingen in vergelijking met de theorie van Vlasov. In een specifiek testgeval tot wel 27% grotere vervormingen. Omdat de theorie van Saint-Venant de vervormingen overschat, zal een constructeur sneller een zwaarder profiel moeten kiezen om aan de vervormingseisen te voldoen, wat leidt tot overgedimensioneerde profielen. Omdat de theorie van Vlasov de werkelijkheid nauwkeuriger benadert, kan deze overdimensionering worden voorkomen. Hierdoor kan materiaal worden bespaard, terwijl de faalkans binnen de geldende veiligheidsnormen blijft.

6.2. Aanbevelingen

In dit onderzoek is een afleiding van de element-stijfheidsmatrix opgesteld voor dubbel symmetrische profielen. Om deze methode in de toekomst verder te ontwikkelen en breder inzetbaar te maken worden de volgende aanbevelingen gedaan:

- **Uitbreiding naar asymmetrische profielen**
De huidige matrix gaat uit van een volledige ontkoppeling tussen axiale rek, buiging en wringing. Bij asymmetrische profielen (zoals U- en L-profielen) valt het normaalkrachtencentrum en het dwarskrachtencentrum niet samen, waardoor een excentriciteit aan de vergelijkingen moet worden toegevoegd. Deze excentriciteit zorgt voor de koppeling tussen de vergelijkingen, wat resulteert in een stijfheidsmatrix die in veel meer situaties toepasbaar wordt.
- **Offset toevoegen**
Het balkelement kan ook worden verschoven ten opzichte van de knopen. De knoopbelastingen grijpen dan op een afstand van het zwaartepunt van de doorsnede aan, wat resulteert in extra buig- en wringspanningen. Het toevoegen van een offset aan de stijfheidsmatrix maakt het mogelijk om deze verschuivingen correct te kunnen modelleren.
- **Controle van commerciële raamwerkprogramma's**
Omdat de stijfheidsmatrices van commerciële 3D raamwerkprogramma's vaak niet openbaar zijn, kan het python-script uit dit rapport ingezet worden om de resultaten te verifiëren. Dit biedt een transparante manier om te controleren of deze software wringing en verhinderde welving daadwerkelijk correct berekent.

Literatuurlijst

- Hartsuijker, C., & Welleman, J. W. (2007). *Engineering mechanics: Volume 2: Stresses, strains, displacements*. Springer Dordrecht. <https://doi.org/10.1007/978-1-4020-5763-2>
- Hoogenboom, P. C. J. (2006). *Vlasov torsion theory: Restrained Warping* (Chapter 7). Technische Universiteit Delft. https://phoogenboom.nl/b16_chap7.pdf
- Hoogenboom, P. C. J. (2019). *Aantekeningen over wringing* (5e dr.). Technische Universiteit Delft. <https://phoogenboom.nl/dictaatwringing.pdf>
- McGuire, W., Gallagher, R. H., & Ziemian, R. D. (2000). *Matrix structural analysis*. (2nd ed). Wiley.
- Rajagopalan, K. (2022). *Torsion of thin walled structures*. Springer Singapore. <https://doi.org/10.1007/978-981-16-7458-7>
- Timoshenko, S. P. (1953). *History of strength of materials: With a brief account of the history of theory of elasticity and theory of structures*. Dover Publications.

Bijlagen

A. Volledige 2D-stijfheidsmatrix

$$\begin{bmatrix} F_{x1} \\ F_{z1} \\ M_1 \\ F_{x2} \\ F_{z2} \\ M_2 \end{bmatrix} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} & 0 & \frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} & 0 & \frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \cdot \begin{bmatrix} u_{x1} \\ u_{z1} \\ \phi_1 \\ u_{x2} \\ u_{z2} \\ \phi_2 \end{bmatrix} + \{f_q\}$$

Hierin is $\{f_q\}$ de vector die de bijdrage van de verdeelde belastingen q_x en q_z aan de knooppunten bevat:

$$\{f_q\} = \begin{bmatrix} -\frac{1}{3}q_{x1}L - \frac{1}{6}q_{x2}L \\ \frac{7}{20}q_{z1}L - \frac{3}{20}q_{z2}L \\ \frac{1}{20}q_{z1}L^2 + \frac{1}{30}q_{z2}L^2 \\ -\frac{1}{6}q_{x1}L - \frac{1}{3}q_{x2}L \\ -\frac{3}{20}q_{z1}L - \frac{7}{20}q_{z2}L \\ -\frac{1}{30}q_{z1}L^2 - \frac{1}{20}q_{z2}L^2 \end{bmatrix}$$

B. 2D-stijfheidsmatrix Python-script

```
import sympy as sp

# Definieren parameters
x, L, E, A, I = sp.symbols('x L E A I')
ux1, uz1, phi1, ux2, uz2, phi2 = sp.symbols('u_x1 u_z1 phi_1 u_x2 u_z2 phi_2')
qx1, qx2, qz1, qz2 = sp.symbols('q_x1 q_x2 q_z1 q_z2')

# Rek
qx = qx1 * (1 - x / L) + qx2 * x / L
u_func = sp.Function('u')(x)
diff_rek = sp.Eq(E * A * u_func.diff(x, 2), -qx)
u = sp.dsolve(diff_rek, u_func, ics={
    u_func.subs(x, 0): ux1,
    u_func.subs(x, L): ux2
}).rhs
N = E * A * u.diff(x)

# Buiging
qz = qz1 * (1 - x / L) + qz2 * x / L
w_func = sp.Function('w')(x)
diff_buiging = sp.Eq(E * I * w_func.diff(x, 4), qz)
w = sp.dsolve(diff_buiging, w_func, ics={
    w_func.subs(x, 0): uz1,
    w_func.subs(x, L): uz2,
    w_func.diff(x).subs(x, 0): -phi1,
    w_func.diff(x).subs(x, L): -phi2
}).rhs
M = - E * I * w.diff(x, 2)
V = - E * I * w.diff(x, 3)

# Opstellen stijfheidsmatrix
Fx1 = -N.subs(x, 0)
Fz1 = -V.subs(x, 0)
M1 = -M.subs(x, 0)
Fx2 = N.subs(x, L)
Fz2 = V.subs(x, L)
M2 = M.subs(x, L)

F = [Fx1, Fz1, M1, Fx2, Fz2, M2]
u = [ux1, uz1, phi1, ux2, uz2, phi2]
K = sp.Matrix([[sp.diff(i, j) for j in u] for i in F])
Fq = sp.Matrix(F) - (K * sp.Matrix(u))

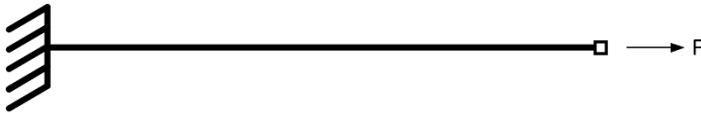
sp.Eq(sp.Matrix(sp.symbols('F_{x1} F_{z1} M_1 F_{x2} F_{z2} M_2')), sp.Ma-
tAdd(sp.MatMul(sp.simplify(K), sp.Matrix(u)), sp.simplify(Fq), evaluate=False))
```

C. 2D-stijfheidsmatrix validatieberekeningen

Voor de validatie van de 2D element-stijfheidsmatrix zijn 4 standaard belastinggevallen bekeken die verschillende onderdelen van de stijfheidsmatrix en het bijbehorende elementgedrag controleren.

Situatie 1: Uitkragende ligger met trek

De eerste situatie die is bekeken is een uitkragende ligger onder zuivere trek.



Figuur 0.1: Uitkragende ligger met trekkraft

Het element is aan de linkzijde (knoop 1) ingeklemd waardoor verplaatsing en rotatie verhinderd zijn. Aan de rechterzijde (knoop 2) is het element belast met een horizontale puntlast F . Dit geeft de volgende randvoorwaarden:

$$u_{x1} = 0, u_{z1} = 0, \phi_1 = 0$$

$$F_{x2} = F, F_{z2} = 0, M_2 = 0$$

$$q_{x1} = 0, q_{x2} = 0, q_{z1} = 0, q_{z2} = 0$$

Het substitueren van deze randvoorwaarden in de matrixvergelijking geeft:

$$\begin{bmatrix} F_{x1} \\ F_{z1} \\ M_1 \\ F \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} & 0 & \frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} & 0 & \frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ u_{x2} \\ u_{z2} \\ \phi_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Uitwerking met Python:

```
Fx1_ext, Fz1_ext, M1_ext = sp.symbols('Fx1_ext Fz1_ext M1_ext')
Fx2_ext, Fz2_ext, M2_ext = sp.symbols('Fx2_ext Fz2_ext M2_ext')
F_ext = sp.Matrix([Fx1_ext, Fz1_ext, M1_ext, Fx2_ext, Fz2_ext, M2_ext])
F, q, M = sp.symbols('F q M')
u = sp.Matrix(u)
vgl = K * u + Fq - F_ext
```

```

# Randvoorwaarden
randvoorwaarden = {
    qx1: 0, qx2: 0, qz1: 0, qz2: 0,
    ux1: 0, uz1: 0, phi1: 0,
    Fx2_ext: F, Fz2_ext: 0, M2_ext: 0
}
vgl_subs = vgl.subs(randvoorwaarden)

# oplossen
onbekenden = [ux2, uz2, phi2, Fx1_ext, Fz1_ext, M1_ext]
oplossing = sp.solve(vgl_subs, onbekenden)

for i, j in oplossing.items():
    print(f'{i} = {j}')

```

Geeft de volgende oplossing:

$$\begin{aligned}
 u_{x2} &= \frac{FL}{EA} & F_{x1} &= -F \\
 u_{z2} &= 0 & F_{z1} &= 0 \\
 \phi_2 &= 0 & M_1 &= 0
 \end{aligned}$$

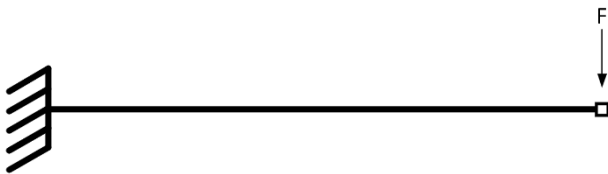
Het resultaat voor u_{x2} komt overeen met de afleiding van zuivere rek volgens Hooke:

$$\varepsilon = \frac{\Delta L}{L} = \frac{\sigma}{E} = \frac{F}{EA} \Rightarrow \Delta L = \frac{FL}{EA}$$

De kracht F_{x1} volgt logisch uit het statisch evenwicht van het element in de horizontale richting: $\sum F_x = 0$. De overige verplaatsingen en reactiekrachten zijn gelijk aan 0, zoals verwacht bij zuivere rek ontstaan alleen horizontale verplaatsingen en horizontale reactiekrachten. Dit betekent dat het element zich in deze situatie correct gedraagt.

Situatie 2: Uitkragende ligger met buiging

De volgende situatie die wordt bekeken is een uitkragende ligger met zuivere buiging door een puntlast op het uiteinde van het balkelement.



Figuur 0.2: Uitkragende ligger met buiging

Het element is aan de linkerkzijde (knoop 1) volledig ingeklemd waardoor verplaatsing en rotatie verhinderd zijn. Op het rechter uiteinde (knoop 2) is het element belast met een verticale puntlast F . Dit resulteert in de randvoorwaarden:

$$u_{x1} = 0, u_{z1} = 0, \phi_1 = 0$$

$$F_{x2} = 0, F_{z2} = F, M_2 = 0$$

$$q_{x1} = 0, q_{x2} = 0, q_{z1} = 0, q_{z2} = 0$$

Substitueren in de matrixvergelijking:

$$\begin{bmatrix} F_{x1} \\ F_{z1} \\ M_1 \\ 0 \\ F \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} & 0 & \frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} & 0 & \frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ u_{x2} \\ u_{z2} \\ \phi_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Uitwerking met Python:

```

Fx1_ext, Fz1_ext, M1_ext = sp.symbols('Fx1_ext Fz1_ext M1_ext')
Fx2_ext, Fz2_ext, M2_ext = sp.symbols('Fx2_ext Fz2_ext M2_ext')
F_ext = sp.Matrix([Fx1_ext, Fz1_ext, M1_ext, Fx2_ext, Fz2_ext, M2_ext])
F, q, M = sp.symbols('F q M')
u = sp.Matrix(u)
vgl = K * u + Fq - F_ext

```

```

# Randvoorwaarden
randvoorwaarden = {
    qx1: 0, qx2: 0, qz1: 0, qz2: 0,
    ux1: 0, uz1: 0, phi1: 0,
    Fx2_ext: 0, Fz2_ext: F, M2_ext: 0
}
vgl_subs = vgl.subs(randvoorwaarden)

# oplossen
onbekenden = [ux2, uz2, phi2, Fx1_ext, Fz1_ext, M1_ext]
oplossing = sp.solve(vgl_subs, onbekenden)

for i, j in oplossing.items():
    print(f'{i} = {j}')

```

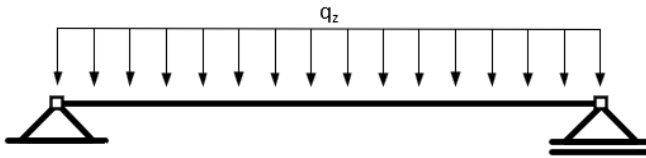
Geeft de volgende oplossing:

$$\begin{aligned}
 u_{x2} &= 0 & F_{x1} &= 0 \\
 u_{z2} &= \frac{FL^3}{3EI} & F_{z1} &= -F \\
 \phi_2 &= -\frac{FL^2}{2EI} & M_1 &= FL
 \end{aligned}$$

Het resultaat van de zakkingen en rotaties komt exact overeen met de formules in de ver-geet-mij-nietjes (Hartsuijker en Welleman, 2007). Daarnaast kloppen ook de reactie-krachten met het krachten- en momenten evenwicht van het element. Hiermee kan ge-concludeerd worden dat de stijfheidsmatrix ook zuivere buiging correct beschrijft.

Situatie 3: Ligger op twee steunpunten met verdeelde belasting

Vervolgens is gekeken naar een ligger op twee steunpunten met een verdeelde verticale belasting over de hele lengte van het element.



Figuur 0.3: Ligger op twee steunpunten met verdeelde verticale belasting

Het element is aan de linkerkzijde (knoop 1) scharnierend opgelegd waardoor verplaatsing verhinderd is en rotatie vrij is. Het rechter uiteinde (knoop 2) is een rolsteunpunt waardoor de verticale verplaatsing verhinderd is, horizontale verplaatsing en rotatie zijn in dit punt vrij. Dit geeft de randvoorwaarden:

$$u_{x1} = 0, u_{z1} = 0, u_{z2} = 0$$

$$M_1 = 0, F_{x2} = 0, M_2 = 0$$

$$q_{x1} = 0, q_{x2} = 0, q_{z1} = q, q_{z2} = q$$

Dit substitueren in de matrixvergelijking geeft:

$$\begin{bmatrix} F_{x1} \\ F_{z1} \\ 0 \\ 0 \\ F_{z2} \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} & 0 & \frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} & 0 & \frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ \phi_1 \\ u_{x2} \\ 0 \\ \phi_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{2}qL \\ \frac{1}{12}qL^2 \\ 0 \\ -\frac{1}{2}qL \\ -\frac{1}{12}qL^2 \end{bmatrix}$$

Uitwerking met Python:

```
Fx1_ext, Fz1_ext, M1_ext = sp.symbols('Fx1_ext Fz1_ext M1_ext')
Fx2_ext, Fz2_ext, M2_ext = sp.symbols('Fx2_ext Fz2_ext M2_ext')
F_ext = sp.Matrix([Fx1_ext, Fz1_ext, M1_ext, Fx2_ext, Fz2_ext, M2_ext])
F, q, M = sp.symbols('F q M')
u = sp.Matrix(u)
vgl = K * u + Fq - F_ext
```

```

# Randvoorwaarden
randvoorwaarden = {
    qx1: 0, qx2: 0, qz1: q, qz2: q,
    ux1: 0, uz1: 0, uz2: 0,
    M1_ext: 0, Fx2_ext: 0, M2_ext: 0
}
vgl_subs = vgl.subs(randvoorwaarden)

# oplossen
onbekenden = [ux2, phi1, phi2, Fx1_ext, Fz1_ext, Fz2_ext]
oplossing = sp.solve(vgl_subs, onbekenden)

for i, j in oplossing.items():
    print(f'{i} = {j}')

```

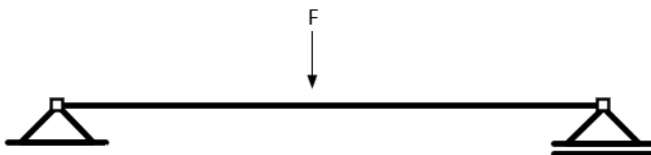
Geeft de oplossing:

$$\begin{aligned}
 u_{x2} &= 0 & F_{x1} &= 0 \\
 \phi_1 &= -\frac{qL^3}{24EI} & F_{z1} &= -\frac{1}{2}qL \\
 \phi_2 &= \frac{qL^3}{24EI} & F_{z2} &= -\frac{1}{2}qL
 \end{aligned}$$

Ook hier komt het resultaat van de zakkingen en rotaties exact overeen met de oplossing zoals beschreven in de vergeet-mij-nietjes (Hartsuijker en Welleman, 2007). De verdeelde belasting zorgt voor een verticale reactiekracht in de steunpunten die elk de helft van de belasting dragen. Hieruit blijkt dat het element voor deze situatie correct functioneert.

Situatie 4: Ligger op twee steunpunten met puntlast in het midden

Tot slot is de situatie bekeken van een ligger op twee steunpunten waarbij een puntlast in het midden van het balkelement is aangebracht.



Figuur 0.4: Ligger op twee steunpunten met puntlast in het midden

Omdat een puntlast alleen op een knoop kan aangrijpen, is het element opgedeeld in twee losse elementen die met elkaar verbonden zijn met een centrale knoop waarop de belasting F werkt. Het element is aan de linkerkzijde (knoop 1) scharnierend opgelegd en aan de rechterzijde (knoop 3) scharnierend op een rolsteunpunt. In het midden (knoop 2) moet de verplaatsing en rotatie van de twee elementen gelijk zijn.

Dit geeft de volgende randvoorwaarden:

$$\begin{aligned}
 u_{x1} &= 0, u_{z1} = 0, u_{z3} = 0 \\
 M_1 &= 0, F_{z2} = F, M_2 = 0, F_{x3} = 0, M_3 = 0 \\
 q_{x1} &= 0, q_{x2} = 0, q_{x3} = 0, q_{z1} = 0, q_{z2} = 0, q_{z3} = 0
 \end{aligned}$$

Het oplossen van de matrixvergelijking gebeurt door de twee losse stijfheidsmatrices te assembleren tot een globale stijfheidsmatrix voor de hele constructie. Na het substitueren van de randvoorwaarden ziet de matrix er als volgt uit:

$$\begin{bmatrix} F_{x1} \\ F_{z1} \\ 0 \\ 0 \\ F \\ 0 \\ 0 \\ F_{z3} \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{EA}{L/2} & 0 & 0 & -\frac{EA}{L/2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI}{L/2^3} & -\frac{6EI}{L/2^2} & 0 & -\frac{12EI}{L/2^3} & -\frac{6EI}{L/2^2} & 0 & 0 & 0 \\ 0 & -\frac{6EI}{L/2^2} & \frac{4EI}{L/2} & 0 & \frac{6EI}{L/2^2} & \frac{2EI}{L/2} & 0 & 0 & 0 \\ -\frac{EA}{L/2} & 0 & 0 & \frac{2EA}{L/2} & 0 & 0 & -\frac{EA}{L/2} & 0 & 0 \\ 0 & -\frac{12EI}{L/2^3} & \frac{6EI}{L/2^2} & 0 & \frac{24EI}{L/2^3} & 0 & 0 & -\frac{12EI}{L/2^3} & -\frac{6EI}{L/2^2} \\ 0 & -\frac{6EI}{L/2^2} & \frac{2EI}{L/2} & 0 & 0 & \frac{8EI}{L/2} & 0 & \frac{6EI}{L/2^2} & \frac{2EI}{L/2} \\ 0 & 0 & 0 & -\frac{EA}{L/2} & 0 & 0 & \frac{EA}{L/2} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{12EI}{L/2^3} & \frac{6EI}{L/2^2} & 0 & \frac{12EI}{L/2^3} & \frac{6EI}{L/2^2} \\ 0 & 0 & 0 & 0 & -\frac{6EI}{L/2^2} & \frac{2EI}{L/2} & 0 & \frac{6EI}{L/2^2} & \frac{4EI}{L/2} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ \phi_1 \\ u_{x2} \\ u_{z2} \\ \phi_2 \\ u_{x3} \\ 0 \\ \phi_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Uitwerking met Python:

```

K_half = K.subs(L, L/2)
Fq_half = Fq.subs(L, L/2)

ux11, uz11, phi11, ux21, uz21, phi21, ux12, uz12, phi12, ux22, uz22, phi22 = sp.symbols(
    'u_x11 u_z11 phi_11 u_x21 u_z21 phi_21 u_x12 u_z12 phi_12 u_x22 u_z22 phi_22')
Fx11, Fz11, M11, Fx21, Fz21, M21, Fx12, Fz12, M12, Fx22, Fz22, M22 = sp.symbols(
    'F_x11 F_z11 M_11 F_x21 F_z21 M_21 F_x12 F_z12 M_12 F_x22 F_z22 M_22')

u1 = sp.Matrix([ux11, uz11, phi11, ux21, uz21, phi21])
F_ext1 = sp.Matrix([Fx11, Fz11, M11, Fx21, Fz21, M21])
vgl1 = K_half * u1 + Fq_half - F_ext1

u2 = sp.Matrix([ux12, uz12, phi12, ux22, uz22, phi22])
F_ext2 = sp.Matrix([Fx12, Fz12, M12, Fx22, Fz22, M22])
vgl2 = K_half * u2 + Fq_half - F_ext2

vgl_totaal = list(vgl1) + list(vgl2)

```

```

# Randvoorwaarden
randvoorwaarden = [
    sp.Eq(qx1, 0), sp.Eq(qx2, 0), sp.Eq(qz1, 0), sp.Eq(qz2, 0),
    sp.Eq(ux11, 0), sp.Eq(uz11, 0), sp.Eq(M11, 0),
    sp.Eq(uz22, 0), sp.Eq(Fx22, 0), sp.Eq(M22, 0),
    sp.Eq(ux21, ux12), sp.Eq(uz21, uz12), sp.Eq(phi21, phi12),
    sp.Eq(Fx21 + Fx12, 0), sp.Eq(Fz21 + Fz12, F), sp.Eq(M21 + M12, 0)
]
vgl_samen = vgl_totaal + randvoorwaarden

# Oplossen
onbekenden = [
    ux11, uz11, phi11, ux21, uz21, phi21,
    ux12, uz12, phi12, ux22, uz22, phi22,
    Fx11, Fz11, M11, Fx21, Fz21, M21,
    Fx12, Fz12, M12, Fx22, Fz22, M22,
    qx1, qx2, qz1, qz2
]
nodig = [ux12, ux22, phi11, phi12, phi22, uz12, Fx11, Fz11, Fz22, M12]
oplossing = sp.solve(vgl_samen, onbekenden)

for i in nodig:
    print(f"{i} = {oplossing[i]}")

```

Dit geeft het volgende resultaat:

$$\begin{array}{lll}
 u_{x2} = 0 & \phi_1 = -\frac{FL^2}{16EI} & F_{x1} = 0 \\
 u_{z2} = \frac{FL^3}{48EI} & \phi_2 = 0 & F_{z1} = -\frac{1}{2}F \\
 u_{x3} = 0 & \phi_3 = \frac{FL^2}{16EI} & F_{z3} = -\frac{1}{2}F
 \end{array}$$

De hierboven gegeven verplaatsingen en rotaties komen overeen met de resultaten van de vergeet-mij-nietjes (Hartsuijker en Welleman, 2007). De reactiekrachten in de steunpunten en het interne moment zijn in overeenstemming met het krachtenevenwicht in de knopen ($\sum F_x = 0$, $\sum F_y = 0$). Ook in deze laatste situatie wordt het elementgedrag correct beschreven door de stijfheidsmatrix.

Op basis van deze vier situaties kan worden geconcludeerd dat de 2D element-stijfheidsmatrix het elementgedrag voor zowel trek als buiging correct beschrijft, waarmee is aangetoond dat de afleiding van de matrix juist is uitgevoerd.

E. 3D-stijfheidsmatrix met vrije welving Python-script

```
import sympy as sp

# Definieren parameters
x, L, E, A, G, Iyy, Izz, It = sp.symbols('x L E A G I_yy I_zz I_t')
ux1, uy1, uz1, phix1, phiy1, phiz1, ux2, uy2, uz2, phix2, phiy2, phiz2 =
sp.symbols('u_x1 u_y1 u_z1 phi_x1 phi_y1 phi_z1 u_x2 u_y2 u_z2 phi_x2 phi_y2
phi_z2')
qx1, qx2, qy1, qy2, qz1, qz2, mx1, mx2 = sp.symbols('q_x1 q_x2 q_y1 q_y2 q_z1
q_z2 m_x1 m_x2')

# Rek
qx = qx1 * (1 - x / L) + qx2 * x / L
u_func = sp.Function('u')(x)
diff_rek = sp.Eq(E * A * u_func.diff(x, 2), -qx)
u = sp.dsolve(diff_rek, u_func, ics={
    u_func.subs(x, 0): ux1,
    u_func.subs(x, L): ux2
}).rhs
N = E * A * u.diff(x)

# Buiging z-richting
qz = qz1 * (1 - x / L) + qz2 * x / L
wz_func = sp.Function('wz')(x)
diff_buigingz = sp.Eq(E * Izz * wz_func.diff(x, 4), qz)
wz = sp.dsolve(diff_buigingz, wz_func, ics={
    wz_func.subs(x, 0): uz1,
    wz_func.subs(x, L): uz2,
    wz_func.diff(x).subs(x, 0): -phiy1,
    wz_func.diff(x).subs(x, L): -phiy2
}).rhs
Mz = - E * Izz * wz.diff(x, 2)
Vz = - E * Izz * wz.diff(x, 3)

# Buiging y-richting
qy = qy1 * (1 - x / L) + qy2 * x / L
wy_func = sp.Function('wy')(x)
diff_buigingy = sp.Eq(E * Iyy * wy_func.diff(x, 4), qy)
wy = sp.dsolve(diff_buigingy, wy_func, ics={
    wy_func.subs(x, 0): uy1,
    wy_func.subs(x, L): uy2,
    wy_func.diff(x).subs(x, 0): phiz1,
    wy_func.diff(x).subs(x, L): phiz2
}).rhs

My = E * Iyy * wy.diff(x, 2)
Vy = - E * Iyy * wy.diff(x, 3)
```

```

# Torsie
mx = mx1 * (1 - x / L) + mx2 * x / L
phix_func = sp.Function('phix')(x)
diff_SaintVenant = sp.Eq(G * It * phix_func.diff(x, 2), -mx)
phix = sp.dsolve(diff_SaintVenant, phix_func, ics={
    phix_func.subs(x, 0): phix1,
    phix_func.subs(x, L): phix2
}).rhs
T = G * It * phix.diff(x, 1)

# Opstellen stijfheidsmatrix
Fx1 = -N.subs(x, 0)
Fy1 = -Vy.subs(x, 0)
Fz1 = -Vz.subs(x, 0)
Mx1 = -T.subs(x, 0)
My1 = -Mz.subs(x, 0)
Mz1 = -My.subs(x, 0)
Fx2 = N.subs(x, L)
Fy2 = Vy.subs(x, L)
Fz2 = Vz.subs(x, L)
Mx2 = T.subs(x, L)
My2 = Mz.subs(x, L)
Mz2 = My.subs(x, L)

F = [Fx1, Fy1, Fz1, Mx1, My1, Mz1, Fx2, Fy2, Fz2, Mx2, My2, Mz2]
u = [ux1, uy1, uz1, phix1, phiy1, phiz1, ux2, uy2, uz2, phix2, phiy2, phiz2]
K = sp.Matrix([[sp.diff(i, j) for j in u] for i in F])
Fq = sp.Matrix(F) - (K * sp.Matrix(u))

sp.Eq(sp.Matrix(sp.symbols('F_{x1} F_{y1} F_{z1} M_{x1} M_{y1} M_{z1} F_{x2}
F_{y2} F_{z2} M_{x2} M_{y2} M_{z2}')), sp.MatAdd(sp.MatMul(sp.simplify(K),
sp.Matrix(u)), sp.simplify(Fq), evaluate=False))

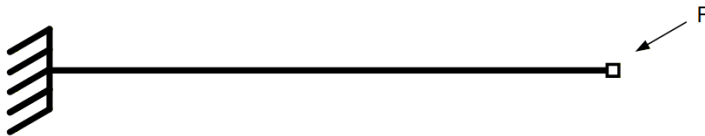
```

F. 3D-stijfheidsmatrix met vrije welving validatieberekeningen

Voor de validatie van de 3D element-stijfheidsmatrix zijn 3 standaard belastinggevallen bekeken die het elementgedrag controleren.

Situatie 1: Uitkragende ligger met buiging in het xy-vlak

De eerste situatie die is bekeken is een uitkragende ligger met zuivere buiging door een puntlast op het uiteinde van het balkelement



Figuur 0.5: Uitkragende ligger met buiging in het xy-vlak

Het element is aan de linkerkzijde (knoop 1) volledig ingeklemd waardoor verplaatsing en rotatie verhinderd zijn. Op het rechter uiteinde (knoop 2) is het element belast met een puntlast F in de y -richting. Dit resulteert in de randvoorwaarden:

$$u_{x1} = 0, u_{y1} = 0, u_{z1} = 0, \phi_{x1} = 0, \phi_{y1} = 0, \phi_{z1} = 0$$

$$F_{x2} = 0, F_{y2} = F, F_{z2} = 0, M_{x2} = 0, M_{y2} = 0, M_{z2} = 0$$

$$q_{x1} = 0, q_{x2} = 0, q_{y1} = 0, q_{y2} = 0, q_{z1} = 0, q_{z2} = 0, m_{x1} = 0, m_{x2} = 0$$

Het substitueren van deze randvoorwaarden in de matrixvergelijking geeft:

$$\begin{bmatrix} F_{x1} \\ F_{y1} \\ F_{z1} \\ M_{x1} \\ M_{y1} \\ M_{z1} \\ 0 \\ F \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & \frac{6EI_{yy}}{L^2} & 0 & -\frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & \frac{6EI_{yy}}{L^2} \\ 0 & 0 & \frac{12EI_{zz}}{L^3} & 0 & -\frac{6EI_{zz}}{L^2} & 0 & 0 & 0 & -\frac{12EI_{zz}}{L^3} & 0 & -\frac{6EI_{zz}}{L^2} & 0 \\ 0 & 0 & 0 & \frac{GI_t}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GI_t}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_{zz}}{L^2} & 0 & \frac{4EI_{zz}}{L} & 0 & 0 & 0 & \frac{6EI_{zz}}{L^2} & 0 & \frac{2EI_{zz}}{L} & 0 \\ 0 & \frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{4EI_{yy}}{L} & 0 & -\frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{2EI_{yy}}{L} \\ -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & -\frac{6EI_{yy}}{L^2} & 0 & \frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & -\frac{6EI_{yy}}{L^2} \\ 0 & 0 & -\frac{12EI_{zz}}{L^3} & 0 & \frac{6EI_{zz}}{L^2} & 0 & 0 & 0 & \frac{12EI_{zz}}{L^3} & 0 & \frac{6EI_{zz}}{L^2} & 0 \\ 0 & 0 & 0 & -\frac{GI_t}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GI_t}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_{zz}}{L^2} & 0 & \frac{2EI_{zz}}{L} & 0 & 0 & 0 & \frac{6EI_{zz}}{L^2} & 0 & \frac{4EI_{zz}}{L} & 0 \\ 0 & \frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{2EI_{yy}}{L} & 0 & -\frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{4EI_{yy}}{L} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ u_{x2} \\ u_{y2} \\ u_{z2} \\ \phi_{x2} \\ \phi_{y2} \\ \phi_{z2} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Uitwerking met Python:

```
Fx1_ext, Fy1_ext, Fz1_ext, Mx1_ext, My1_ext, Mz1_ext = sp.symbols('Fx1_ext Fy1_ext Fz1_ext Mx1_ext My1_ext Mz1_ext')
Fx2_ext, Fy2_ext, Fz2_ext, Mx2_ext, My2_ext, Mz2_ext = sp.symbols('Fx2_ext Fy2_ext Fz2_ext Mx2_ext My2_ext Mz2_ext')
F_ext = sp.Matrix([Fx1_ext, Fy1_ext, Fz1_ext, Mx1_ext, My1_ext, Mz1_ext, Fx2_ext, Fy2_ext, Fz2_ext, Mx2_ext, My2_ext, Mz2_ext])
F, q, mx, M = sp.symbols('F q mx M')
u = sp.Matrix(u)
vg1 = K * u + Fq - F_ext
```

```
# Randvoorwaarden
randvoorwaarden = {
    qx1: 0, qx2: 0, qy1: 0, qy2: 0, qz1: 0, qz2: 0, mx1: 0, mx2: 0,
    ux1: 0, uy1: 0, uz1: 0, phix1: 0, phiy1: 0, phiz1: 0,
    Fx2_ext: 0, Fy2_ext: F, Fz2_ext: 0, Mx2_ext: 0, My2_ext: 0, Mz2_ext: 0
}
vg1_subs = vg1.subs(randvoorwaarden)

# oplossen
onbekenden = [ux2, uy2, uz2, phix2, phiy2, phiz2, Fx1_ext, Fy1_ext, Fz1_ext, Mx1_ext, My1_ext, Mz1_ext]
oplossing = sp.solve(vg1_subs, onbekenden)

for i, j in oplossing.items():
    print(f'{i} = {j}')
```

Geeft de volgende oplossing:

$$\begin{array}{llll} u_{x2} = 0 & \phi_{x2} = 0 & F_{x1} = 0 & M_{x1} = 0 \\ u_{y2} = \frac{FL^3}{3EI_{yy}} & \phi_{y2} = 0 & F_{y1} = -F & M_{y1} = 0 \\ u_{z2} = 0 & \phi_{z2} = \frac{FL^2}{2EI_{yy}} & F_{z1} = 0 & M_{z1} = -FL \end{array}$$

Het resultaat wordt gecontroleerd door de Euler-Bernoulli vergelijking voor buiging voor deze situatie handmatig af te leiden.

$$M_z(x) = EI_{yy} \frac{d^2 u_y}{dx^2}$$

$$M_z(x) = F_y (L - x)$$

$$EI_{yy} \frac{d^2 u_y}{dx^2} = F_y (L - x)$$

$$\frac{du_y}{dx} = \frac{F_y}{EI_{yy}} \left(Lx - \frac{1}{2} x^2 \right) + C_1$$

$$u_y(x) = \frac{F_y}{EI_{yy}} \left(\frac{1}{2} Lx^2 - \frac{1}{6} x^3 \right) + C_1 x + C_2$$

Invullen randvoorwaarden geeft:

$$u_y(0) = u_{y1} = 0 \Rightarrow C_2 = 0$$

$$\frac{du_y(0)}{dx} = \phi_{z1} = 0 \Rightarrow C_1 = 0$$

$$u_y(x) = \frac{F_y}{EI_{yy}} \left(\frac{1}{2} Lx^2 - \frac{1}{6} x^3 \right)$$

Oplossing:

$$u_y(x=L) = \frac{F_y}{EI_{yy}} \left(\frac{L^3}{2} - \frac{L^3}{6} \right) = \frac{F_y L^3}{3EI_{yy}} \quad (\text{komt overeen met sympy oplossing})$$

$$\phi_{z2}(x=L) = \frac{F_y}{EI_{yy}} \left(L^2 - \frac{1}{2} L^2 \right) = \frac{F_y L^2}{2EI_{yy}} \quad (\text{komt overeen met sympy oplossing})$$

De krachten bij knoop 1 volgen uit de evenwichtsvergelijkingen:

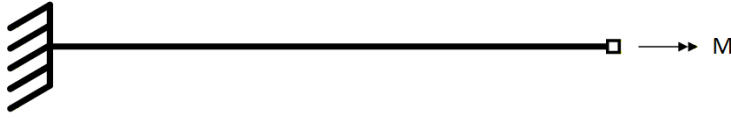
$$\sum F_y = 0 \Rightarrow F_{y1} + F = 0 \Rightarrow F_{y1} = -F \quad (\text{komt overeen met sympy oplossing})$$

$$\sum M_z = 0 \Rightarrow M_{z1} + F \cdot L = 0 \Rightarrow M_{z1} = -FL \quad (\text{komt overeen met sympy oplossing})$$

De stijfheidsmatrix beschrijft deze situatie voor zuivere buiging in het xy-vlak dus correct

Situatie 2: Uitkragende ligger met wringing

De tweede situatie die wordt bekeken is een uitkragende ligger met wringing door een wringmoment op het uiteinde van het balkelement.



Figuur 0.6: Uitkragende ligger met wringing

Het element is aan de linkerzijde (knoop 1) ingeklemd waardoor verplaatsing en rotatie verhinderd zijn. Aan de rechterzijde (knoop 2) is het element belast met een wringend moment M . Dit geeft de volgende randvoorwaarden:

$$u_{x1} = 0, u_{y1} = 0, u_{z1} = 0, \phi_{x1} = 0, \phi_{y1} = 0, \phi_{z1} = 0$$

$$F_{x2} = 0, F_{y2} = 0, F_{z2} = 0, M_{x2} = M, M_{y2} = 0, M_{z2} = 0$$

$$q_{x1} = 0, q_{x2} = 0, q_{y1} = 0, q_{y2} = 0, q_{z1} = 0, q_{z2} = 0, m_{x1} = 0, m_{x2} = 0$$

Het substitueren van deze randvoorwaarden in de matrixvergelijking geeft:

$$\begin{bmatrix} F_{x1} \\ F_{y1} \\ F_{z1} \\ M_{x1} \\ M_{y1} \\ M_{z1} \\ 0 \\ 0 \\ 0 \\ M \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & \frac{6EI_{yy}}{L^2} & 0 & -\frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & \frac{6EI_{yy}}{L^2} \\ 0 & 0 & \frac{12EI_{zz}}{L^3} & 0 & -\frac{6EI_{zz}}{L^2} & 0 & 0 & 0 & -\frac{12EI_{zz}}{L^3} & 0 & -\frac{6EI_{zz}}{L^2} & 0 \\ 0 & 0 & 0 & \frac{GI_t}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GI_t}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_{zz}}{L^2} & 0 & \frac{4EI_{zz}}{L} & 0 & 0 & 0 & \frac{6EI_{zz}}{L^2} & 0 & \frac{2EI_{zz}}{L} & 0 \\ 0 & \frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{4EI_{yy}}{L} & 0 & -\frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{2EI_{yy}}{L} \\ -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & -\frac{6EI_{yy}}{L^2} & 0 & \frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & -\frac{6EI_{yy}}{L^2} \\ 0 & 0 & -\frac{12EI_{zz}}{L^3} & 0 & \frac{6EI_{zz}}{L^2} & 0 & 0 & 0 & \frac{12EI_{zz}}{L^3} & 0 & \frac{6EI_{zz}}{L^2} & 0 \\ 0 & 0 & 0 & -\frac{GI_t}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GI_t}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_{zz}}{L^2} & 0 & \frac{2EI_{zz}}{L} & 0 & 0 & 0 & \frac{6EI_{zz}}{L^2} & 0 & \frac{4EI_{zz}}{L} & 0 \\ 0 & \frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{2EI_{yy}}{L} & 0 & -\frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{4EI_{yy}}{L} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ u_{x2} \\ u_{y2} \\ u_{z2} \\ \phi_{x2} \\ \phi_{y2} \\ \phi_{z2} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Uitwerking met Python:

```
Fx1_ext, Fy1_ext, Fz1_ext, Mx1_ext, My1_ext, Mz1_ext = sp.symbols('Fx1_ext Fy1_ext Fz1_ext Mx1_ext My1_ext Mz1_ext')
Fx2_ext, Fy2_ext, Fz2_ext, Mx2_ext, My2_ext, Mz2_ext = sp.symbols('Fx2_ext Fy2_ext Fz2_ext Mx2_ext My2_ext Mz2_ext')
F_ext = sp.Matrix([Fx1_ext, Fy1_ext, Fz1_ext, Mx1_ext, My1_ext, Mz1_ext, Fx2_ext, Fy2_ext, Fz2_ext, Mx2_ext, My2_ext, Mz2_ext])
F, q, mx, M = sp.symbols('F q mx M')
u = sp.Matrix(u)
vg1 = K * u + Fq - F_ext
```

```

# Randvoorwaarden
randvoorwaarden = {
    qx1: 0, qx2: 0, qy1: 0, qy2: 0, qz1: 0, qz2: 0, mx1: 0, mx2: 0,
    ux1: 0, uy1: 0, uz1: 0, phix1: 0, phiy1: 0, phiz1: 0,
    Fx2_ext: 0, Fy2_ext: 0, Fz2_ext: 0, Mx2_ext: M, My2_ext: 0, Mz2_ext: 0
}
vg1_subs = vg1.subs(randvoorwaarden)

# oplossen
onbekenden = [ux2, uy2, uz2, phix2, phiy2, phiz2, Fx1_ext, Fy1_ext, Fz1_ext, Mx1_ext, My1_ext, Mz1_ext]
oplossing = sp.solve(vg1_subs, onbekenden)

for i, j in oplossing.items():
    print(f'{i} = {j}')

```

Geeft de volgende oplossing:

$$\begin{array}{llll}
 u_{x2} = 0 & \phi_{x2} = \frac{LM}{GI_t} & F_{x1} = 0 & M_{x1} = -M \\
 u_{y2} = 0 & \phi_{y2} = 0 & F_{y1} = 0 & M_{y1} = 0 \\
 u_{z2} = 0 & \phi_{z2} = 0 & F_{z1} = 0 & M_{z1} = 0
 \end{array}$$

Het resultaat wordt gecontroleerd door de Saint-Venant vergelijking voor torsie voor deze situatie handmatig af te leiden.

$$\begin{aligned}
 M_x(x) &= GI_t \frac{d\phi}{dx} \\
 M_x(x) &= M \\
 GI_t \frac{d\phi}{dx} &= M \\
 \phi(x) &= \frac{M}{GI_t} x + C_1
 \end{aligned}$$

Invullen randvoorwaarde geeft:

$$\begin{aligned}
 \phi_x(0) = \phi_{x1} = 0 &\Rightarrow C_1 = 0 \\
 \phi(x) &= \frac{M}{GI_t} x
 \end{aligned}$$

Oplossing:

$$\phi(x=L) = \frac{ML}{GI_t} \quad (\text{komt overeen met sympy oplossing})$$

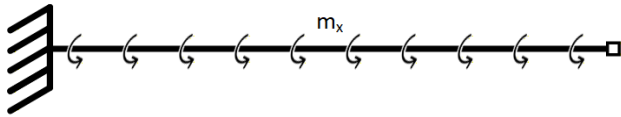
De krachten bij knoop 1 volgen uit de evenwichtsvergelijkingen:

$$\sum M_x = 0 \Rightarrow M_{x1} + M = 0 \Rightarrow M_{x1} = -M \quad (\text{komt overeen met sympy oplossing})$$

De stijfheidsmatrix beschrijft deze situatie voor wringing dus ook correct

Situatie 3: Uitkragende ligger met verdeeld wringend moment

De derde situatie die wordt bekeken is een uitkragende ligger met een verdeeld wringmoment over de lengte van het balkelement.



Figuur 0.7: Uitkragende ligger met verdeeld wringend moment

Het element is aan de linkerzijde (knoop 1) ingeklemd waardoor verplaatsing en rotatie verhinderd zijn. Aan de rechterzijde (knoop 2) kan het element vrij verplaatsen en roteren. Dit geeft de volgende randvoorwaarden:

$$u_{x1} = 0, u_{y1} = 0, u_{z1} = 0, \phi_{x1} = 0, \phi_{y1} = 0, \phi_{z1} = 0$$

$$F_{x2} = 0, F_{y2} = 0, F_{z2} = 0, M_{x2} = M, M_{y2} = 0, M_{z2} = 0$$

$$q_{x1} = 0, q_{x2} = 0, q_{y1} = 0, q_{y2} = 0, q_{z1} = 0, q_{z2} = 0, m_{x1} = m_x, m_{x2} = m_x$$

Het substitueren van deze randvoorwaarden in de matrixvergelijking geeft:

$$\begin{bmatrix} F_{x1} \\ F_{y1} \\ F_{z1} \\ M_{x1} \\ M_{y1} \\ M_{z1} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & \frac{6EI_{yy}}{L^2} & 0 & -\frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & \frac{6EI_{yy}}{L^2} \\ 0 & 0 & \frac{12EI_{zz}}{L^3} & 0 & -\frac{6EI_{zz}}{L^2} & 0 & 0 & 0 & -\frac{12EI_{zz}}{L^3} & 0 & -\frac{6EI_{zz}}{L^2} & 0 \\ 0 & 0 & 0 & \frac{GI_t}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GI_t}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_{zz}}{L^2} & 0 & \frac{4EI_{zz}}{L} & 0 & 0 & 0 & \frac{6EI_{zz}}{L^2} & 0 & \frac{2EI_{zz}}{L} & 0 \\ 0 & \frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{4EI_{yy}}{L} & 0 & -\frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{2EI_{yy}}{L} \\ -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & -\frac{6EI_{yy}}{L^2} & 0 & \frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & -\frac{6EI_{yy}}{L^2} \\ 0 & 0 & -\frac{12EI_{zz}}{L^3} & 0 & \frac{6EI_{zz}}{L^2} & 0 & 0 & 0 & \frac{12EI_{zz}}{L^3} & 0 & \frac{6EI_{zz}}{L^2} & 0 \\ 0 & 0 & 0 & -\frac{GI_t}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GI_t}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_{zz}}{L^2} & 0 & \frac{2EI_{zz}}{L} & 0 & 0 & 0 & \frac{6EI_{zz}}{L^2} & 0 & \frac{4EI_{zz}}{L} & 0 \\ 0 & \frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{2EI_{yy}}{L} & 0 & -\frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{4EI_{yy}}{L} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ u_{x2} \\ u_{y2} \\ u_{z2} \\ \phi_{x2} \\ \phi_{y2} \\ \phi_{z2} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{1}{2}m_x L \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -\frac{1}{2}m_x L \\ 0 \\ 0 \end{bmatrix}$$

Uitwerking met Python:

```
Fx1_ext, Fy1_ext, Fz1_ext, Mx1_ext, My1_ext, Mz1_ext = sp.symbols('Fx1_ext Fy1_ext Fz1_ext Mx1_ext My1_ext Mz1_ext')
Fx2_ext, Fy2_ext, Fz2_ext, Mx2_ext, My2_ext, Mz2_ext = sp.symbols('Fx2_ext Fy2_ext Fz2_ext Mx2_ext My2_ext Mz2_ext')
F_ext = sp.Matrix([Fx1_ext, Fy1_ext, Fz1_ext, Mx1_ext, My1_ext, Mz1_ext, Fx2_ext, Fy2_ext, Fz2_ext, Mx2_ext, My2_ext, Mz2_ext])
F, q, mx, M = sp.symbols('F q mx M')
u = sp.Matrix(u)
vg1 = K * u + Fq - F_ext
```

```

# Randvoorwaarden
randvoorwaarden = {
    qx1: 0, qx2: 0, qy1: 0, qy2: 0, qz1: 0, qz2: 0, mx1: mx, mx2: mx,
    ux1: 0, uy1: 0, uz1: 0, phix1: 0, phiy1: 0, phiz1: 0,
    Fx2_ext: 0, Fy2_ext: 0, Fz2_ext: 0, Mx2_ext: 0, My2_ext: 0, Mz2_ext: 0
}
vgl_subs = vgl.subs(randvoorwaarden)

# oplossen
onbekenden = [ux2, uy2, uz2, phix2, phiy2, phiz2, Fx1_ext, Fy1_ext, Fz1_ext, Mx1_ext, My1_ext, Mz1_ext]
oplossing = sp.solve(vgl_subs, onbekenden)

for i, j in oplossing.items():
    print(f'{i} = {j}')

```

Geeft de volgende oplossing:

$$\begin{array}{llll}
 u_{x2} = 0 & \phi_{x2} = \frac{m_x L^2}{2G I_t} & F_{x1} = 0 & M_{x1} = -m_x L \\
 u_{y2} = 0 & \phi_{y2} = 0 & F_{y1} = 0 & M_{y1} = 0 \\
 u_{z2} = 0 & \phi_{z2} = 0 & F_{z1} = 0 & M_{z1} = 0
 \end{array}$$

Het resultaat wordt gecontroleerd door de Saint-Venant vergelijking voor torsie voor deze situatie handmatig af te leiden.

$$M_x(x) = G I_t \frac{d\phi}{dx}$$

$$\frac{dM_x}{dx} = -m_x$$

$$G I_t \frac{d^2\phi}{dx^2} = -m_x$$

$$\frac{d\phi}{dx} = -\frac{m_x}{G I_t} x + C_1$$

$$\phi(x) = -\frac{m_x}{2G I_t} x^2 + C_1 x + C_2$$

Invullen randvoorwaarde geeft:

$$\phi_x(0) = \varphi_{x1} = 0 \Rightarrow C_2 = 0$$

$$M_x(L) = M_{x2} = 0 \Rightarrow G I_t \left(-\frac{m_x}{G I_t} L + C_1 \right) = 0 \Rightarrow C_1 = \frac{m_x L}{G I_t}$$

$$\phi(x) = -\frac{m_x}{2G I_t} x^2 + \frac{m_x L}{G I_t} x$$

Oplossing:

$$\phi(x = L) = -\frac{1}{2} \frac{m_x L^2}{G I_t} + \frac{m_x L^2}{G I_t} = \frac{m_x L^2}{2G I_t} \quad (\text{komt overeen met sympy oplossing})$$

De krachten bij knoop 1 volgen uit de evenwichtsvergelijkingen:

$$\sum M_x = 0 \Rightarrow M_{x1} + m_x L = 0 \Rightarrow M_{x1} = -m_x L \quad (\text{komt overeen met sympy oplossing})$$

De stijfheidsmatrix beschrijft deze situatie voor een verdeeld wringend moment dus ook correct. Hiermee kan worden geconcludeerd dat de 3D element-stijfheidsmatrix het elementgedrag voor buiging en wringing ook correct beschrijft, waarmee is aangetoond dat de afleiding van de matrix juist is uitgevoerd.

G. Volledige 3D-stijfheidsmatrix met verhinderde welving

$$\begin{bmatrix} F_{x1} \\ F_{y1} \\ F_{z1} \\ M_{x1} \\ M_{y1} \\ M_{z1} \\ B_1 \\ F_{x2} \\ F_{y2} \\ F_{z2} \\ M_{x2} \\ M_{y2} \\ M_{z2} \\ B_2 \end{bmatrix} = \begin{bmatrix} \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{12EI_{zz}}{L^3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{C_{wE}\lambda^3(e^{L\lambda}+1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{4EI_{yy}}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & \frac{4EI_{yy}}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{C_{wE}\lambda^2(e^{L\lambda}-1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & 0 & 0 & \frac{C_{wE}\lambda(L\lambda e^{2L\lambda} + L\lambda - e^{2L\lambda} + 1)}{L\lambda e^{2L\lambda} - L\lambda - 2e^{2L\lambda} + 4e^{L\lambda} - 2} & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{12EI_{yy}}{L^3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{6EI_{zz}}{L^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2EI_{yy}}{L} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2EI_{yy}}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{C_{wE}\lambda^2(1-e^{L\lambda})}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{C_{wE}\lambda(-2L\lambda e^{L\lambda} + e^{2L\lambda} - 1)}{L\lambda e^{2L\lambda} - L\lambda - 2e^{2L\lambda} + 4e^{L\lambda} - 2} \end{bmatrix} \cdot \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{z1} \\ \phi_{x1} \\ \phi_{y1} \\ \phi_{z1} \\ w_1 \\ u_{x2} \\ u_{y2} \\ u_{z2} \\ \phi_{x2} \\ \phi_{y2} \\ \phi_{z2} \\ w_2 \end{bmatrix} +$$

42

$$\begin{bmatrix} m_{x1}(-2L^2\lambda^3 - 3L^2\lambda^2 + 6L\lambda + (-2L^2\lambda^2 + 3L^2\lambda^2 + 6L\lambda - 12)e^{L\lambda} + 12) + m_{x2}(2L^2\lambda^2 + 9L\lambda + (2L^2\lambda^2 - 24)e^{L\lambda} + (2L^2\lambda^2 - 9L\lambda + 12)e^{2L\lambda} + 12) \\ m_{y1}(-2L^2\lambda^2 - 9L\lambda + (-2L^2\lambda^2 + 24)e^{L\lambda} + (-2L^2\lambda^2 + 9L\lambda - 12)e^{2L\lambda} - 12) + m_{y2}((-4L^2\lambda^2 e^{L\lambda} - L^2\lambda^2 - 3L\lambda + (-L^2\lambda^2 + 3L\lambda)e^{2L\lambda}) \\ m_{z1}(-L^3\lambda^3 - 3L^2\lambda^2 - 6L\lambda + (-L^3\lambda^3 + 3L^2\lambda^2 - 6L\lambda + 12)e^{L\lambda} + (-2L^2\lambda^2 - 6L\lambda + 12)e^{2L\lambda} + 12) \\ m_{x1}(4L^2\lambda^2 e^{L\lambda} + L^2\lambda^2 + 3L\lambda + (L^2\lambda^2 - 3L\lambda)e^{2L\lambda}) + m_{x2}(2L^2\lambda^2 + 9L\lambda + (2L^2\lambda^2 - 24)e^{L\lambda} + (2L^2\lambda^2 - 9L\lambda + 12)e^{2L\lambda} + 12) \\ m_{y1}(4L^2\lambda^2 e^{L\lambda} + L^2\lambda^2 + 3L\lambda + (L^2\lambda^2 - 3L\lambda)e^{2L\lambda}) + m_{y2}(2L^2\lambda^2 + 9L\lambda + (2L^2\lambda^2 - 24)e^{L\lambda} + (2L^2\lambda^2 - 9L\lambda + 12)e^{2L\lambda} + 12) \\ m_{z1}(4L^2\lambda^2 e^{L\lambda} + L^2\lambda^2 + 3L\lambda + (L^2\lambda^2 - 3L\lambda)e^{2L\lambda}) + m_{z2}(2L^2\lambda^2 + 9L\lambda + (2L^2\lambda^2 - 24)e^{L\lambda} + (2L^2\lambda^2 - 9L\lambda + 12)e^{2L\lambda} + 12) \end{bmatrix}$$

$$\text{met } \lambda = \sqrt{\frac{Gt}{EC_w}}$$

H. 3D-stijfheidsmatrix met verhinderde welving Python-script

```
import sympy as sp

# Definieren parameters
x, L, E, A, G, Iyy, Izz, It, Cw, lam = sp.symbols('x L E A G I_yy I_zz I_t C_w
lambda', positive=True, real=True)
ux1, uy1, uz1, phix1, phiy1, phiz1, w1, ux2, uy2, uz2, phix2, phiy2, phiz2, w2 =
sp.symbols('u_x1 u_y1 u_z1 phi_x1 phi_y1 phi_z1 w_1 u_x2 u_y2 u_z2 phi_x2 phi_y2
phi_z2 w_2')
qx1, qx2, qy1, qy2, qz1, qz2, mx1, mx2 = sp.symbols('q_x1 q_x2 q_y1 q_y2 q_z1
q_z2 m_x1 m_x2')

# Rek
qx = qx1 * (1 - x / L) + qx2 * x / L
u_func = sp.Function('u')(x)
diff_rek = sp.Eq(E * A * u_func.diff(x, 2), -qx)
u = sp.dsolve(diff_rek, u_func, ics={
    u_func.subs(x, 0): ux1,
    u_func.subs(x, L): ux2
}).rhs
N = E * A * u.diff(x)

# Buiging z
qz = qz1 * (1 - x / L) + qz2 * x / L
wz_func = sp.Function('wz')(x)
diff_buigingz = sp.Eq(E * Izz * wz_func.diff(x, 4), qz)
wz = sp.dsolve(diff_buigingz, wz_func, ics={
    wz_func.subs(x, 0): uz1,
    wz_func.subs(x, L): uz2,
    wz_func.diff(x).subs(x, 0): -phiy1,
    wz_func.diff(x).subs(x, L): -phiy2
}).rhs
Mz = - E * Izz * wz.diff(x, 2)
Vz = - E * Izz * wz.diff(x, 3)

# Buiging y
qy = qy1 * (1 - x / L) + qy2 * x / L
wy_func = sp.Function('wy')(x)
diff_buigingy = sp.Eq(E * Iyy * wy_func.diff(x, 4), qy)
wy = sp.dsolve(diff_buigingy, wy_func, ics={
    wy_func.subs(x, 0): uy1,
    wy_func.subs(x, L): uy2,
    wy_func.diff(x).subs(x, 0): phiz1,
    wy_func.diff(x).subs(x, L): phiz2
}).rhs
My = E * Iyy * wy.diff(x, 2)
Vy = - E * Iyy * wy.diff(x, 3)
```

```

# Torsie
mx = mx1 * (1 - x / L) + mx2 * x / L
phix_func = sp.Function('phix')(x)

diff_Vlasov_hom = sp.Eq(phix_func.diff(x, 4) - lam**2 * phix_func.diff(x, 2), 0)
phix_hom = sp.dsolve(diff_Vlasov_hom, phix_func, ics={
    phix_func.subs(x, 0): phix1,
    phix_func.diff(x).subs(x, 0): -w1,
    phix_func.subs(x, L): phix2,
    phix_func.diff(x).subs(x, L): -w2
}).rhs

diff_Vlasov_par = sp.Eq(phix_func.diff(x, 4) - lam**2 * phix_func.diff(x, 2), mx
/ (E * Cw))
phix_par = sp.dsolve(diff_Vlasov_par, phix_func, ics={
    phix_func.subs(x, 0): 0,
    phix_func.diff(x).subs(x, 0): 0,
    phix_func.subs(x, L): 0,
    phix_func.diff(x).subs(x, L): 0
}).rhs
phix = phix_hom + phix_par
B = - E * Cw * phix.diff(x, 2)
Mw = G * It * phix.diff(x, 1) - E * Cw * phix.diff(x,3)

# Opstellen stijfheidsmatrix
F = [-N.subs(x, 0), -Vy.subs(x, 0), -Vz.subs(x, 0), -Mw.subs(x, 0), -Mz.subs(x,
0), -My.subs(x, 0), -B.subs(x, 0), N.subs(x, L), Vy.subs(x, L), Vz.subs(x, L),
Mw.subs(x, L), Mz.subs(x, L), My.subs(x, L), B.subs(x, L)]
u = [ux1, uy1, uz1, phix1, phiy1, phiz1, w1, ux2, uy2, uz2, phix2, phiy2, phiz2,
w2]
K = sp.Matrix([[sp.diff(i, j) for j in u] for i in F]).subs(G * It, lam**2 * E *
Cw).applyfunc(sp.simplify)
Fq = sp.Matrix(F).subs({i: 0 for i in u}).subs(G * It, lam**2 * E * Cw).apply-
func(lambda expr: sp.collect(sp.fraction(sp.cancel(expr))[0], (mx1, mx2), lambda
c: sp.collect(c, sp.exp(L*lam))) / sp.factor(sp.fraction(sp.cancel(expr))[1]))

F_symbols = sp.Matrix(sp.symbols('F_{x1} F_{y1} F_{z1} M_{x1} M_{y1} M_{z1} B_1
F_{x2} F_{y2} F_{z2} M_{x2} M_{y2} M_{z2} B_2'))
sp.Eq(F_symbols, sp.MatAdd(sp.MatMul(K, sp.Matrix(u)), Fq, evaluate=False))

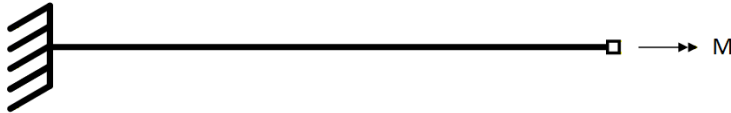
```

I. 3D-stijfheidsmatrix met verhinderde welving validatieberekeningen

Voor de validatie van de 3D element-stijfheidsmatrix met verhinderde welving zijn 3 standaard belastinggevallen bekeken die het elementgedrag controleren.

Situatie 1: Uitkragende ligger met wringing

De eerste situatie die wordt bekeken is een uitkragende ligger met een wringmoment op het uiteinde van het balkelement.



Figuur 0.8: Uitkragende ligger met wringing

Het element is aan de linkerzijde (knoop 1) ingeklemd waardoor verplaatsing en rotatie verhinderd zijn. Aan de rechterzijde (knoop 2) is het element belast met een wringend moment M . Dit geeft de volgende randvoorwaarden:

$$u_{x1} = 0, u_{y1} = 0, u_{z1} = 0, \phi_{x1} = 0, \phi_{y1} = 0, \phi_{z1} = 0, w_1 = 0$$

$$F_{x2} = 0, F_{y2} = 0, F_{z2} = 0, M_{x2} = M, M_{y2} = 0, M_{z2} = 0, B_2 = 0$$

$$q_{x1} = 0, q_{x2} = 0, q_{y1} = 0, q_{y2} = 0, q_{z1} = 0, q_{z2} = 0, m_{x1} = 0, m_{x2} = 0$$

Het substitueren van deze randvoorwaarden in de matrixvergelijking geeft:

$$\begin{bmatrix} F_{x1} \\ F_{y1} \\ F_{z1} \\ M_{x1} \\ M_{y1} \\ M_{z1} \\ B_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ M \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12E}{l^3} I_{yy} & 0 & 0 & 0 & \frac{6E}{l^2} I_{yy} & 0 & 0 & -\frac{12E}{l^3} I_{yy} & 0 & 0 & 0 & \frac{6E}{l^2} I_{yy} & 0 & 0 & 0 \\ 0 & 0 & \frac{12E}{l^3} I_{zz} & 0 & -\frac{6E}{l^2} I_{zz} & 0 & 0 & 0 & 0 & -\frac{12E}{l^3} I_{zz} & 0 & 0 & -\frac{6E}{l^2} I_{zz} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{C_{wE}\lambda^2(e^{L\lambda}-1)}{L\lambda e^{L\lambda}+L\lambda-2e^{L\lambda}+2} & 0 & 0 & \frac{C_{wE}\lambda^2(e^{L\lambda}-1)}{L\lambda e^{L\lambda}+L\lambda-2e^{L\lambda}+2} & 0 & 0 & 0 & -\frac{C_{wE}\lambda^2(e^{L\lambda}+1)}{L\lambda e^{L\lambda}+L\lambda-2e^{L\lambda}+2} & 0 & 0 & \frac{C_{wE}\lambda^2(e^{L\lambda}-1)}{L\lambda e^{L\lambda}+L\lambda-2e^{L\lambda}+2} & 0 & 0 \\ 0 & 0 & -\frac{6E}{l^2} I_{zz} & 0 & \frac{4E}{l} I_{zz} & 0 & 0 & 0 & 0 & \frac{6E}{l^2} I_{zz} & 0 & 0 & \frac{2E}{l} I_{zz} & 0 & 0 & 0 \\ 0 & \frac{6E}{l^2} I_{yy} & 0 & 0 & 0 & \frac{4E}{l} I_{yy} & 0 & 0 & -\frac{6E}{l^2} I_{yy} & 0 & 0 & 0 & 0 & \frac{2E}{l} I_{yy} & 0 & 0 \\ 0 & 0 & 0 & \frac{C_{wE}\lambda^2(e^{L\lambda}-1)}{L\lambda e^{L\lambda}+L\lambda-2e^{L\lambda}+2} & 0 & 0 & \frac{C_{wE}\lambda(L\lambda e^{2L\lambda}+L\lambda-e^{2L\lambda}+1)}{L\lambda e^{2L\lambda}-L\lambda-2e^{2L\lambda}+4e^{L\lambda}-2} & 0 & 0 & 0 & \frac{C_{wE}\lambda^2(1-e^{L\lambda})}{L\lambda e^{L\lambda}+L\lambda-2e^{L\lambda}+2} & 0 & 0 & \frac{C_{wE}\lambda(-2L\lambda e^{L\lambda}+e^{2L\lambda}-1)}{L\lambda e^{2L\lambda}-L\lambda-2e^{2L\lambda}+4e^{L\lambda}-2} & 0 & 0 \\ -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12E}{l^3} I_{yy} & 0 & 0 & 0 & -\frac{6E}{l^2} I_{yy} & 0 & 0 & \frac{12E}{l^3} I_{yy} & 0 & 0 & 0 & 0 & -\frac{6E}{l^2} I_{yy} & 0 & 0 & 0 \\ 0 & 0 & -\frac{12E}{l^3} I_{zz} & 0 & \frac{6E}{l^2} I_{zz} & 0 & 0 & 0 & 0 & -\frac{12E}{l^3} I_{zz} & 0 & 0 & \frac{6E}{l^2} I_{zz} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{C_{wE}\lambda^2(e^{L\lambda}+1)}{L\lambda e^{L\lambda}+L\lambda-2e^{L\lambda}+2} & 0 & 0 & \frac{C_{wE}\lambda^2(1-e^{L\lambda})}{L\lambda e^{L\lambda}+L\lambda-2e^{L\lambda}+2} & 0 & 0 & 0 & \frac{C_{wE}\lambda^2(e^{L\lambda}+1)}{L\lambda e^{L\lambda}+L\lambda-2e^{L\lambda}+2} & 0 & 0 & \frac{C_{wE}\lambda^2(1-e^{L\lambda})}{L\lambda e^{L\lambda}+L\lambda-2e^{L\lambda}+2} & 0 & 0 \\ 0 & 0 & 0 & -\frac{6E}{l^2} I_{zz} & 0 & \frac{2E}{l} I_{zz} & 0 & 0 & 0 & 0 & \frac{6E}{l^2} I_{zz} & 0 & 0 & \frac{4E}{l} I_{zz} & 0 & 0 & 0 \\ 0 & 0 & \frac{6E}{l^2} I_{yy} & 0 & 0 & \frac{2E}{l} I_{yy} & 0 & 0 & -\frac{6E}{l^2} I_{yy} & 0 & 0 & 0 & 0 & \frac{4E}{l} I_{yy} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{C_{wE}\lambda^2(e^{L\lambda}-1)}{L\lambda e^{L\lambda}+L\lambda-2e^{L\lambda}+2} & 0 & 0 & \frac{C_{wE}\lambda(-2L\lambda e^{L\lambda}+e^{2L\lambda}-1)}{L\lambda e^{2L\lambda}-L\lambda-2e^{2L\lambda}+4e^{L\lambda}-2} & 0 & 0 & 0 & \frac{C_{wE}\lambda^2(1-e^{L\lambda})}{L\lambda e^{L\lambda}+L\lambda-2e^{L\lambda}+2} & 0 & 0 & \frac{C_{wE}\lambda(L\lambda e^{2L\lambda}+L\lambda-e^{2L\lambda}+1)}{L\lambda e^{2L\lambda}-L\lambda-2e^{2L\lambda}+4e^{L\lambda}-2} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{x2} \\ u_{y2} \\ u_{z2} \\ \phi_{x2} \\ \phi_{y2} \\ \phi_{z2} \\ w_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Uitwerking met Python:

```
Fx1_ext, Fy1_ext, Fz1_ext, Mx1_ext, My1_ext, Mz1_ext, B1_ext = sp.symbols('Fx1_ext Fy1_ext Fz1_ext Mx1_ext My1_ext Mz1_ext B1_ext')
Fx2_ext, Fy2_ext, Fz2_ext, Mx2_ext, My2_ext, Mz2_ext, B2_ext = sp.symbols('Fx2_ext Fy2_ext Fz2_ext Mx2_ext My2_ext Mz2_ext B2_ext')
F_ext = sp.Matrix([Fx1_ext, Fy1_ext, Fz1_ext, Mx1_ext, My1_ext, Mz1_ext, B1_ext, Fx2_ext, Fy2_ext, Fz2_ext, Mx2_ext, My2_ext, Mz2_ext, B2_ext])
F, q, mx, M = sp.symbols('F q mx M')
u = sp.Matrix(u)
vg1 = K * u + Fq - F_ext
```

```

# Randvoorwaarden
randvoorwaarden = {
    qx1: 0, qx2: 0, qy1: 0, qy2: 0, qz1: 0, qz2: 0, mx1: 0, mx2: 0,
    ux1: 0, uy1: 0, uz1: 0, phix1: 0, phiy1: 0, phiz1: 0, w1: 0,
    Fx2_ext: 0, Fy2_ext: 0, Fz2_ext: 0, Mx2_ext: M, My2_ext: 0, Mz2_ext: 0, B2_ext: 0
}
vgl_subs = vgl.subs(randvoorwaarden)

# oplossen
onbekenden = [ux2, uy2, uz2, phix2, phiy2, phiz2, w2, Fx1_ext, Fy1_ext, Fz1_ext, Mx1_ext, My1_ext, Mz1_ext, B1_ext]
oplossing = sp.solve(vgl_subs, onbekenden)

for i, j in oplossing.items():
    print(f'{i} = {j}')

```

Geeft de volgende oplossing:

$$M_{x1} = -M$$

$$B_1 = \frac{M}{\lambda} \left(\frac{e^{2\lambda L} - 1}{e^{2\lambda L} + 1} \right) = \frac{M}{\lambda} \tanh(\lambda L)$$

$$\phi_{x2} = \frac{ML}{G I_t} \left(1 - \frac{e^{2L\lambda} - 1}{\lambda L (e^{2L\lambda} + 1)} \right) = \frac{ML}{G I_t} \left(1 - \frac{\tanh(\lambda L)}{\lambda L} \right)$$

$$w_2 = -\frac{M}{G I_t} \left(1 - \frac{2e^{L\lambda}}{e^{2L\lambda} + 1} \right) = -\frac{M}{G I_t} \left(1 - \frac{1}{\cosh(\lambda L)} \right)$$

Het resultaat wordt gecontroleerd door de torsie vergelijking van Vlasov voor deze situatie handmatig af te leiden.

$$M_x(x) = G I_t \frac{d\phi}{dx} - E C_w \frac{d^3\phi}{dx^3}$$

$$M_x(x) = M$$

$$G I_t \frac{d\phi}{dx} - E C_w \frac{d^3\phi}{dx^3} = M$$

$$\phi(x) = \frac{M}{G I_t} x + C_1 + C_2 \cosh\left(\sqrt{\frac{G I_t}{E C_w}} x\right) + C_3 \sinh\left(\sqrt{\frac{G I_t}{E C_w}} x\right)$$

Invullen randvoorwaarde geeft:

$$\phi_x(0) = \phi_{x1} = 0, \frac{d\phi_x}{dx}(0) = -w_1 = 0, B(L) = B_2 = -E C_w \frac{d^2\phi_x}{dx^2} = 0$$

$$\phi(x) = \frac{M}{G I_t} \left(x - \frac{1}{\lambda} (\tanh(\lambda L) (1 - \cosh(\lambda x)) + \sinh(\lambda x)) \right)$$

$$\phi'(x) = \frac{M}{G I_t} (1 + \tanh(\lambda L) \sinh(\lambda x) - \cosh(\lambda x))$$

Oplossing:

$$\phi_{x2} = \phi_x(x = L) = \frac{ML}{G I_t} \left(1 - \frac{\tanh(\lambda L)}{\lambda L} \right) \quad (\text{komt overeen met sympy oplossing})$$

$$\frac{d\phi_x}{dx}(x = L) = \frac{M}{G I_t} \left(1 - \frac{1}{\cosh(\lambda L)} \right)$$

$$w_2 = -\frac{d\phi_x}{dx}(x = L) = -\frac{M}{G I_t} \left(1 - \frac{1}{\cosh(\lambda L)} \right) \quad (\text{komt overeen met sympy oplossing})$$

$$B_1 = -B(x = 0) = E C_w \frac{d^2\phi_x}{dx^2} = \frac{M}{\lambda} \tanh(\lambda L) \quad (\text{komt overeen met sympy oplossing})$$

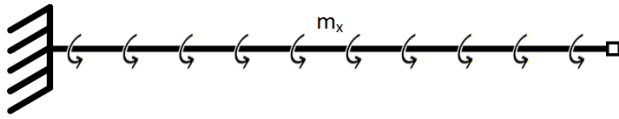
De krachten bij knoop 1 volgen uit de evenwichtsvergelijkingen:

$$\sum M_x = 0 \Rightarrow M_{x1} + M = 0 \Rightarrow M_{x1} = -M \quad (\text{komt overeen met sympy oplossing})$$

De stijfheidsmatrix beschrijft deze situatie voor wringing dus correct

Situatie 2: Uitkragende ligger met verdeeld wringend moment

De derde situatie die wordt bekeken is een uitkragende ligger met een verdeeld wringend moment over de lengte van het balkelement.



Figuur 0.9: Uitkragende ligger met verdeeld wringend moment

Het element is aan de linkerzijde (knoop 1) ingeklemd waardoor verplaatsing en rotatie verhinderd zijn. Aan de rechterzijde (knoop 2) kan het element vrij verplaatsen en roteren. Dit geeft de volgende randvoorwaarden:

$$u_{x1} = 0, u_{y1} = 0, u_{z1} = 0, \phi_{x1} = 0, \phi_{y1} = 0, \phi_{z1} = 0, w_1 = 0$$

$$F_{x2} = 0, F_{y2} = 0, F_{z2} = 0, M_{x2} = M, M_{y2} = 0, M_{z2} = 0, B_2 = 0$$

$$q_{x1} = 0, q_{x2} = 0, q_{y1} = 0, q_{y2} = 0, q_{z1} = 0, q_{z2} = 0, m_{x1} = m_x, m_{x2} = m_x$$

Het substitueren van deze randvoorwaarden in de matrixvergelijking geeft:

$$\begin{bmatrix} F_{x1} \\ F_{y1} \\ F_{z1} \\ M_{x1} \\ M_{y1} \\ M_{z1} \\ B_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12E}{l^3} I_{yy} & 0 & 0 & 0 & \frac{6E}{l^2} I_{yy} & 0 & 0 & -\frac{12E}{l^3} I_{yy} & 0 & 0 & 0 & \frac{6E}{l^2} I_{yy} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{12E}{l^3} I_{zz} & 0 & 0 & -\frac{6E}{l^2} I_{zz} & 0 & 0 & 0 & -\frac{12E}{l^3} I_{zz} & 0 & 0 & -\frac{6E}{l^2} I_{zz} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{C_{wE}\lambda^2 (e^{L\lambda} + 1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & 0 & 0 & \frac{C_{wE}\lambda^2 (e^{L\lambda} - 1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & 0 & 0 & 0 & -\frac{C_{wE}\lambda^2 (e^{L\lambda} + 1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & 0 & 0 & \frac{C_{wE}\lambda^2 (e^{L\lambda} - 1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{6E}{l^2} I_{zz} & 0 & 0 & \frac{4E}{l} I_{zz} & 0 & 0 & 0 & \frac{6E}{l^2} I_{zz} & 0 & 0 & 0 & \frac{2E}{l} I_{zz} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{6E}{l^2} I_{yy} & 0 & 0 & 0 & 0 & \frac{4E}{l} I_{yy} & 0 & 0 & -\frac{6E}{l^2} I_{yy} & 0 & 0 & 0 & \frac{2E}{l} I_{yy} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{C_{wE}\lambda^2 (e^{L\lambda} - 1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & 0 & 0 & \frac{C_{wE}\lambda (L\lambda e^{2L\lambda} + L\lambda - e^{2L\lambda} + 1)}{L\lambda e^{2L\lambda} - L\lambda - 2e^{2L\lambda} + 4e^{L\lambda} - 2} & 0 & 0 & 0 & \frac{C_{wE}\lambda^2 (1 - e^{L\lambda})}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & 0 & 0 & \frac{C_{wE}\lambda (-2L\lambda e^{L\lambda} + e^{2L\lambda} - 1)}{L\lambda e^{2L\lambda} - L\lambda - 2e^{2L\lambda} + 4e^{L\lambda} - 2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12E}{l^3} I_{yy} & 0 & 0 & 0 & -\frac{6E}{l^2} I_{yy} & 0 & 0 & \frac{12E}{l^3} I_{yy} & 0 & 0 & 0 & -\frac{6E}{l^2} I_{yy} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{12E}{l^3} I_{zz} & 0 & 0 & \frac{6E}{l^2} I_{zz} & 0 & 0 & 0 & -\frac{12E}{l^3} I_{zz} & 0 & 0 & \frac{6E}{l^2} I_{zz} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{C_{wE}\lambda^2 (e^{L\lambda} + 1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & 0 & 0 & \frac{C_{wE}\lambda^2 (1 - e^{L\lambda})}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & 0 & 0 & 0 & \frac{C_{wE}\lambda^3 (e^{L\lambda} + 1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & 0 & 0 & \frac{C_{wE}\lambda^2 (1 - e^{L\lambda})}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{6E}{l^2} I_{zz} & 0 & \frac{2E}{l} I_{zz} & 0 & 0 & 0 & \frac{6E}{l^2} I_{zz} & 0 & 0 & 0 & \frac{4E}{l} I_{zz} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{6E}{l^2} I_{yy} & 0 & 0 & 0 & 0 & \frac{2E}{l} I_{yy} & 0 & 0 & -\frac{6E}{l^2} I_{yy} & 0 & 0 & 0 & \frac{4E}{l} I_{yy} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{C_{wE}\lambda^2 (e^{L\lambda} - 1)}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & 0 & 0 & \frac{C_{wE}\lambda (-2L\lambda e^{L\lambda} + e^{2L\lambda} - 1)}{L\lambda e^{2L\lambda} - L\lambda - 2e^{2L\lambda} + 4e^{L\lambda} - 2} & 0 & 0 & 0 & \frac{C_{wE}\lambda^2 (1 - e^{L\lambda})}{L\lambda e^{L\lambda} + L\lambda - 2e^{L\lambda} + 2} & 0 & 0 & \frac{C_{wE}\lambda (L\lambda e^{2L\lambda} + L\lambda - e^{2L\lambda} + 1)}{L\lambda e^{2L\lambda} - L\lambda - 2e^{2L\lambda} + 4e^{L\lambda} - 2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{x2} \\ u_{y2} \\ u_{z2} \\ \phi_{x2} \\ \phi_{y2} \\ \phi_{z2} \\ w_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{1}{2} m_x L \\ 0 \\ 0 \\ 0 \\ \frac{Lm_x + 2m_x + (L\lambda m_x - 2m_x)e^{L\lambda}}{2\lambda^2 (e^{L\lambda} - 1)} \\ 0 \\ 0 \\ 0 \\ -\frac{1}{2} m_x L \\ 0 \\ 0 \\ -\frac{Lm_x - 2m_x - (L\lambda m_x - 2m_x)e^{L\lambda}}{2\lambda^2 (e^{L\lambda} - 1)} \end{bmatrix}$$

Uitwerking met Python:

```
Fx1_ext, Fy1_ext, Fz1_ext, Mx1_ext, My1_ext, Mz1_ext, B1_ext = sp.symbols('Fx1_ext Fy1_ext Fz1_ext Mx1_ext My1_ext Mz1_ext B1_ext')
Fx2_ext, Fy2_ext, Fz2_ext, Mx2_ext, My2_ext, Mz2_ext, B2_ext = sp.symbols('Fx2_ext Fy2_ext Fz2_ext Mx2_ext My2_ext Mz2_ext B2_ext')
F_ext = sp.Matrix([Fx1_ext, Fy1_ext, Fz1_ext, Mx1_ext, My1_ext, Mz1_ext, B1_ext, Fx2_ext, Fy2_ext, Fz2_ext, Mx2_ext, My2_ext, Mz2_ext, B2_ext])
F, q, mx, M = sp.symbols('F q mx M')
u = sp.Matrix(u)
vgl = K * u + Fq - F_ext

# Randvoorwaarden
randvoorwaarden = {
    qx1: 0, qx2: 0, qy1: 0, qy2: 0, qz1: 0, qz2: 0, mx1: mx, mx2: mx,
    ux1: 0, uy1: 0, uz1: 0, phix1: 0, phiy1: 0, phiz1: 0, w1: 0,
    Fx2_ext: 0, Fy2_ext: 0, Fz2_ext: 0, Mx2_ext: 0, My2_ext: 0, Mz2_ext: 0, B2_ext: 0
}
vgl_subs = vgl.subs(randvoorwaarden)

# oplossen
onbekenden = [ux2, uy2, uz2, phix2, phiy2, phiz2, w2, Fx1_ext, Fy1_ext, Fz1_ext, Mx1_ext, My1_ext, Mz1_ext, B1_ext]
oplossing = sp.solve(vgl_subs, onbekenden)

for i, j in oplossing.items():
    print(f'{i} = {j}')
```

Geeft de volgende oplossing:

$$M_{x1} = -m_x L$$

$$B_1 = \frac{m_x(L\lambda e^{2\lambda L} - L\lambda - e^{2L\lambda} + 2e^{L\lambda} - 1)}{\lambda^2(e^{2L\lambda} + 1)} = \frac{m_x}{\lambda^2} \left(\frac{1 + L\lambda \sinh(\lambda L) - \cosh(\lambda L)}{\cosh(\lambda L)} \right)$$

$$\phi_{x2} = \frac{m_x(L^2\lambda^2 e^{2L\lambda} + L^2\lambda^2 - 2L\lambda e^{2L\lambda} + 2L\lambda + 2e^{2L\lambda} - 4e^{L\lambda} + 2)}{2EC_w\lambda^4(e^{2L\lambda} + 1)} = \frac{m_x}{GI_t\lambda^2} \left(1 - \frac{1}{\cosh(\lambda L)} - \lambda L \tanh(\lambda L) + \frac{\lambda^2 L^2}{2} \right)$$

$$w_2 = \frac{m_x(2L\lambda e^{L\lambda} - e^{2L\lambda} + 1)}{EC_w\lambda^3(e^{2L\lambda} + 1)} = \frac{M_x}{GI_t\lambda} \left(\frac{\lambda L - \sinh(\lambda L)}{\cosh(\lambda L)} \right)$$

Het resultaat wordt gecontroleerd door de torsie vergelijking van Vlasov voor deze situatie handmatig af te leiden.

$$M_x(x) = GI_t \frac{d\phi}{dx} - EC_w \frac{d^3\phi}{dx^3}$$

$$\frac{dM_x}{dx} = -m_x$$

$$EC_w \frac{d^4\phi}{dx^4} - GI_t \frac{d^2\phi}{dx^2} = m_x$$

$$\phi(x) = C_1 + C_2 x + C_3 \sinh\left(\sqrt{\frac{GI_t}{EC_w}} x\right) + C_4 \cosh\left(\sqrt{\frac{GI_t}{EC_w}} x\right) - \frac{m_x}{2GI_t} x^2$$

Invullen randvoorwaarde geeft:

$$\phi_x(0) = \phi_{x1} = 0, \frac{d\phi_x}{dx}(0) = -w_1 = 0, M_x(L) = M_{x2} = 0, B(L) = B_2 = -EC_w \frac{d^2\phi_x}{dx^2} = 0$$

$$\phi(x) = \frac{m_x}{GI_t} \left(-\frac{1}{\lambda^2} \left(\frac{1 + \lambda L \sinh(\lambda L)}{\cosh(\lambda L)} \right) + Lx - \frac{L}{\lambda} \sinh(\lambda x) + \frac{1}{\lambda^2} \left(\frac{1 + \lambda L \sinh(\lambda L)}{\cosh(\lambda L)} \right) \cosh(\lambda x) - \frac{1}{2} x^2 \right)$$

$$\phi'(x) = \frac{m_x}{GI_t} \left(L - L \cosh(\lambda x) + \frac{1}{\lambda} \left(\frac{1 + \lambda L \sinh(\lambda L)}{\cosh(\lambda L)} \right) \sinh(\lambda x) - x \right)$$

Oplossing:

$$\phi_{x2} = \phi_x(L) = \frac{m_x}{GI_t\lambda^2} \left(1 - \frac{1}{\cosh(\lambda L)} - \lambda L \tanh(\lambda L) + \frac{\lambda^2 L^2}{2} \right) \quad (\text{komt overeen})$$

$$w_2 = -\frac{d\phi_x}{dx}(L) = \frac{m_x}{GI_t} \left(L \cosh(\lambda L) - \frac{1}{\lambda} \left(\frac{1 + \lambda L \sinh(\lambda L)}{\cosh(\lambda L)} \right) \sinh(\lambda L) \right) \quad (\text{komt overeen})$$

$$B_1 = -B(0) = EC_w \frac{d^2\phi_x}{dx^2} = \frac{m_x}{\lambda^2} \left(\frac{1 + \lambda L \sinh(\lambda L) - \cosh(\lambda L)}{\cosh(\lambda L)} \right) \quad (\text{komt overeen})$$

Deze 3 onbekenden komen exact overeen met de python oplossing, de krachten bij knoop 1 volgen uit de evenwichtsvergelijkingen:

$$\sum M_x = 0 \Rightarrow M_{x1} + m_x L = 0 \Rightarrow M_{x1} = -m_x L \quad (\text{komt overeen met sympy oplossing})$$

De stijfheidsmatrix beschrijft deze situatie voor een verdeeld wringend moment dus ook correct met de Vlasov theorie. Hiermee kan worden geconcludeerd dat de 3D element-stijfheidsmatrix het elementgedrag correct beschrijft, waarmee is aangetoond dat de afleiding van de matrix juist is uitgevoerd.

J. Uitwerking testberekening

Om het verschil in torsiestijfheid tussen de Saint-Venant en Vlasov afleiding inzichtelijk te maken, is in deze bijlage een praktijksituatie doorgerekend. De geschematiseerde situatie betreft een verkeersbord boven de weg dat is gemonteerd op een uitkragende ligger waarop een windbelasting werkt wat een wringend moment veroorzaakt in de balk.

Voor de berekening is uitgegaan van een 3.0 meter lange uitkragend HEA160 profiel dat aan de ene zijde is ingeklemd, waarbij welving volledig is verhinderd, en aan het vrije uiteinde wordt belast met een wringend moment van 500 Nm. De volgende gegevens voor deze situatie zijn van toepassing:

- $M = 500 \text{ Nm}$
- $L = 3.0 \text{ m}$
- $E = 210 \cdot 10^9 \text{ N/m}^2$
- $G = 81 \cdot 10^9 \text{ N/m}^2$
- $I_t = 122 \cdot 10^9 \text{ m}^4$
- $C_w = 31.4 \cdot 10^{-9} \text{ m}^6$

Saint-Venant

$$\phi_{x2} = \frac{ML}{GI_t} = \frac{500 \cdot 3}{81 \cdot 10^9 \cdot 122 \cdot 10^{-9}} = 0.15179 \text{ rad} = 8.7^\circ$$

Vlasov

$$\lambda = \sqrt{\frac{GI_t}{EC_w}} = \sqrt{\frac{81 \cdot 10^9 \cdot 122 \cdot 10^{-9}}{210 \cdot 10^9 \cdot 31.4 \cdot 10^{-9}}} = 1.224$$

$$\phi_{x2} = \frac{ML}{GI_t} \left(1 - \frac{\tanh(\lambda L)}{\lambda L}\right) = \frac{500 \cdot 3}{81 \cdot 10^9 \cdot 122 \cdot 10^{-9}} \left(1 - \frac{\tanh(1.224 \cdot 3)}{1.224 \cdot 3}\right) = 0.1105 \text{ rad} = 6.3^\circ$$

Vershil

$$\frac{0.1105 - 0.1518}{0.1518} = -0.272$$

Dus de Vlasov theorie resulteert leidt tot een 27.2% kleinere hoekverdraaiing ten opzichte van de Saint-Venant theorie.