MASTER

Onderzoek naar het geometrisch niet-lineair gedrag van constructies

Beijers, A.P.A.M.

Award date:
1971

Afstudeeropdracht

Onderzoek naar het geometrisch niet-lineair gedrag van constructies

A.P.A.H. BEIJERS.

WE 71-8

Inhoud

Literatuurlijst

I Het in rekening van particuliere oplossingen bij ANA-elementen.
D.H. van Campen en C.M. Hencken.
PRGL-ANA-R70-1 KR-273

II Uit het „IUTAM Symposium on High Speed Computing of elastic Structures - Liège - Belgium
25 August 1970"
de volgende delen:

1) Geometrically nonlinear static and dynamic analysis of shells of revolution
James A. Stricklin

2) Effective use of the incremental stiffness matrices in nonlinear geometric analysis.
G.A. Dupuis, D.D. Pfaffinger, P.V. Marcal

3) Geometrically nonlinear structural analysis by the direct stiffness method
J.A. Stricklin, W.E. Haisler, W.A. von Riesemann

III Stabiliteitsonderzoek A.P.A.H. Beijers
WE 71-5

IV Collegedictaat Technische Mechanika III
Prof. ir. W.L. Esmeijer

V Numerieke Methode in de Mechanika dr.ir. J.D. Janssen

VI Recent advances in matrix methods of structural analysis J.H. Argyris.
pergamon press N.Y. 1964

VII Numeriek spannings en trillingsonderzoek
9-13 sept. 1968
Leergang te Delft.

VIII Matrix methods in structural mechanics
J.S. Przemieniecki, R.M. Bader, W.F. Bozich,
J.R. Johnson, W.J. Mykytow.

IX Journal of the structural division
„Numerical formulation of nonlinear elasticity
problems" by T. Oden (June 1967)

X International journal for numerical methods in
engineering vol. 2, 229-241 (1970)
„Finite element solution to an elastica
problem of beams."
J. Tada, G.C. Lee.

XI Collegedictaat Elastische elementen Prof. dr. ir. J.A. Haringx.

Inleiding

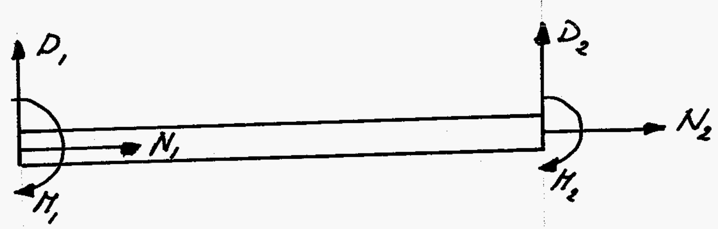
Als afstudeeropdracht heb ik in de groep Technische Mechanica een onderzoek ingesteld naar het geometrisch niet-lineaire gedrag van constructies. Deze opdracht is uitgemond in een computerprogramma voor geometrisch niet-lineaire balkconstructies. Het verslag behandelt verder de afleiding van het lineaire en het geometrisch niet-lineaire balkelement. Tijdens het onderzoek bleek, dat het probleem zich hoofdzakelijk rond de oplossingsmethodieken voor stelsels niet-lineaire vergelijkingen toespitste, vandaar dat hieraan een hoofdstuk gewijd is, met daaropvolgend een hoofdstuk over de toepassing van deze methodieken in de methode der eindige elementen. In de berekende voorbeelden zijn een aantal reeds in de litteratuur bekende problemen geverifieerd. Tenslotte wordt nog een hoofdstuk gewijd aan de stabiliteit van balkconstructies in het lineaire gebied. Het verslag wordt besloten met de afleiding van het lineaire rotatorisch symmetrische element TRIAX3 en een afleiding van het geometrisch niet-lineaire TRIAX 3 element.

I Beschrijving van het balkelement

We beschouwen een balk met konstante dwarsdoorsnede, waarvan de hoofdtraagheidsas met het kleinste oppervlaktetraagheidsmoment loodrecht op het vlak van tekening staat. Verder maken we gebruik van de klassieke balkentheorie, zodat de hypothese van Bernouilli geldig blijft. De belastingen en de verplaatsingen treden alleen in het vlak van tekening op. Uit voorgaande aannames volgt, dat de verplaatsingen van de uiteinden van het balkelement, de knooppunten, klein moeten zijn. We definiëren de verplaatsingsvektor u als volgt:

$$\vec{u} = [u_1 \quad w_1 \quad \varphi_1 \quad u_2 \quad w_2 \quad \varphi_2]$$

Verder definiëren we de belastingsvektor van het element als volgt:



$$\vec{f} = [N_1 \quad D_1 \quad M_1 \quad N_2 \quad D_2 \quad M_2]$$

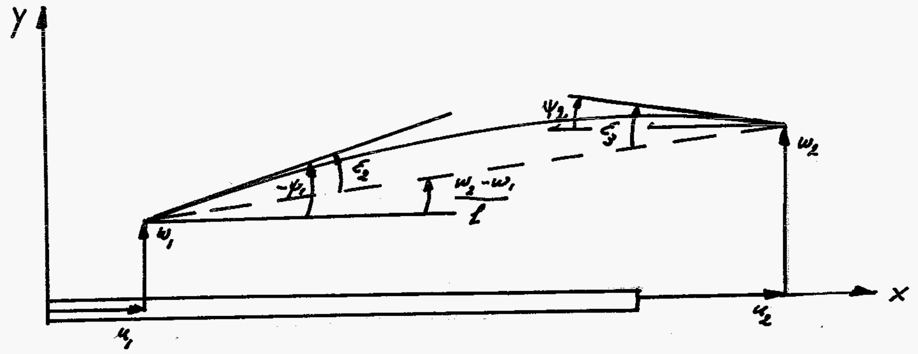
Uit deze definitie voor de belastingsvektor blijkt, dat we het element uitsluitend met krachtgrootheden in de knooppunten kunnen belasten. Willen we bijvoorbeeld een gelijkmatige belasting in rekening brengen, dan moeten we deze door equivalente knooppuntskrachten vervangen. (zie litt. I). Deze knooppuntskrachten moeten dan eenzelfde hoeveelheid arbeid verrichten dan de gelijkmatige belasting.

Omdat we een twee-dimensionaal balkelement bestuderen zijn we in staat om door middel van 3 rek- en 3-spanningsgrootheden het geheel te beschrijven, de zogenaamde gegeneraliseerde spannings- en rekgrootheden.
De gegeneraliseerde rekgrootheden worden als volgt gedefinieerd:

$$\varepsilon_1 = u_2 - u_1$$

$$\varepsilon_2 = -\psi_1 - \frac{w_2 - w_1}{l} \qquad (I1)$$

$$\varepsilon_3 = \psi_2 + \frac{w_2 - w_1}{l}$$



De rekvektor heeft de volgende gedaante:

$$\vec{\varepsilon} = [\varepsilon_1 \ \varepsilon_2 \ \varepsilon_3]$$

We gaan nu op het balkelement het principe van minimale potentiële energie toepassen. Volgens de klassieke theorie kunnen we de inwendige elastische energie ten gevolge van normaalkracht en buigend moment als volgt opschrijven:

$$U = \int_{x=0}^{l} \left[\frac{EI}{2} \left(\frac{d^2 w}{dx^2} \right)^2 + \frac{EF}{2} \left(\frac{du_a}{dx} \right)^2 \right] dx$$

w = het verplaatsingsveld in y-richting van het element
u_a = het verplaatsingsveld in x-richting van het element

Voor de potentiaal voor de uitwendige krachten kunnen we, met voorgaande belastings- en verplaatsingsvektor, schrijven als het produkt van:

$$\vec{f}\,.\,u$$

Totaal verkrijgen we nu voor de potentiële energie de volgende uitdrukking:

$$V = \int_{x=0}^{x=l} \left[\frac{EI}{2} \left(\frac{d^2 w}{dx^2} \right)^2 + \frac{EF}{2} \left(\frac{du_a}{dx} \right)^2 \right] dx - \vec{f}\, u \qquad (I\,2)$$

We nemen nu voor de verplaatsingsvelden u_a en w de volgende polynomen aan:

$$u_a = a_1 + a_2\, x$$

$$w = a_3 + a_4\, x + a_5\, x^2 + a_6\, x^3$$

Deze verplaatsingsvelden moeten aan de volgende randvoorwaarden voldoen:

$x = 0$ $\quad u_a = u_1$, $\quad w = w_1$, $\quad \frac{dw}{dx} = -\varphi_1$

$x = l$ $\quad u_a = u_2$, $\quad w = w_2$, $\quad \frac{dw}{dx} = -\varphi_2$

We kunnen nu de verplaatsingsvelden als volgt schrijven

$$w = w_1 \left\{ 1 - \frac{x}{l} \right\} + w_2 \frac{x}{l} + \varepsilon_2\, x \left\{ 1 - \frac{x}{l} \right\}^2 + \varepsilon_3 \frac{x^2}{l} \left\{ 1 - \frac{x}{l} \right\}$$

$$u_a = u_1 + \varepsilon_1 \frac{x}{l}$$

Met behulp van de uit de klassieke theorie bekende relaties

$$N = EF \left(\frac{du_a}{dx} \right)$$

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

kunnen we de gegeneraliseerde

spanningsvektor als volgt definieren:

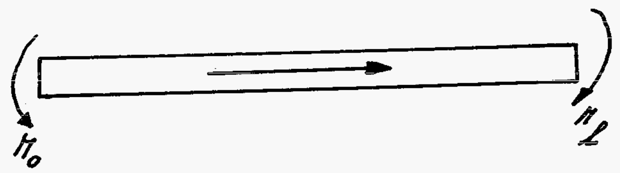
$$N = EF\,\frac{\varepsilon_1}{l}$$

$$M_{x=0} = M_0 = -EI\left\{-\frac{4\varepsilon_2}{l} + \frac{2\varepsilon_3}{l}\right\} \qquad (I\,3)$$

$$M_{x=l} = M_l = -EI\left\{\frac{2\varepsilon_2}{l} - \frac{4\varepsilon_3}{l}\right\}$$

de spanningsvektor: $\vec{\sigma} = [N \;\; M_0 \;\; M_l]$

De tekenafspraken zijn als volgt:

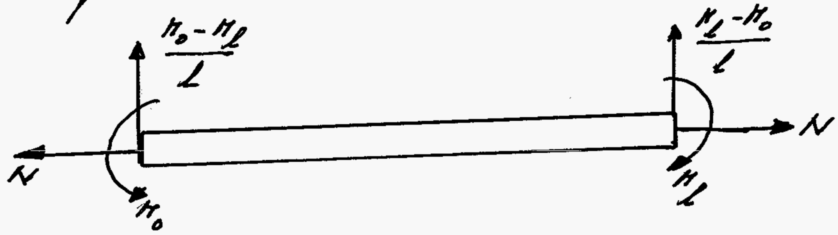


De berekening van de dwarskracht in het element geschiedt met de klassieke relatie

$$\frac{dM}{dx} = D$$

$$D = \frac{M_l - M_0}{l}$$

De knooppuntskrachten zijn nu als volgt uit de spanningsvektor σ te berekenen:



We kunnen relatie (I3) ook in matrixvorm schrijven:

$$\sigma = S\,\varepsilon \qquad (I4)$$

$$S = \begin{bmatrix} \frac{EF}{l} & 0 & 0 \\ 0 & \frac{4EI}{l} & -\frac{2EI}{l} \\ 0 & -\frac{2EI}{l} & \frac{4EI}{l} \end{bmatrix}$$

Met behulp van formule (I 4) zijn we in staat om de spanningen te berekenen, als de rekken bekend zijn. De rekken kunnen we uit de verplaatsingsvektor u met (I 1) berekenen. De verplaatsingsvektor u berekenen we uit de relatie

$$Q_e\, u = f \qquad (I\,5)$$

waarin Q_e de stijfheidsmatrix van het element is, welke we hierna uit het principe van minimale potentiële energie zullen afleiden.

$$\delta V = 0$$

$$\int_{x=0}^{x=l}\left[EI\,\frac{d^2w}{dx^2}\,\delta\left(\frac{d^2w}{dx^2}\right) + EF\,\frac{du_a}{dx}\,\delta\left(\frac{du_a}{dx}\right)\right]dx - f^T\delta u = 0$$

In eerste instantie gaan we nu $\delta\left(\frac{d^2w}{dx^2}\right)$ en $\delta\left(\frac{du_a}{dx}\right)$ nader bekijken:

$$\delta\left(\frac{d^2w}{dx^2}\right) = \frac{d}{d\varepsilon}\left(\frac{d^2w}{dx^2}\right)\delta\varepsilon = \frac{d}{d\varepsilon}\left(\frac{d^2w}{dx^2}\right)\frac{\partial\varepsilon}{\partial u}\,\delta u \qquad (I\,6)$$

$$\delta\left(\frac{du_a}{dx}\right) = \frac{d}{d\varepsilon}\left(\frac{du_a}{dx}\right)\delta\varepsilon = \frac{d}{d\varepsilon}\left(\frac{du_a}{dx}\right)\frac{\partial\varepsilon}{\partial u}\,\delta u$$

We zijn in staat om formule (I 6) in matrixnotatie als volgt te schrijven:

$$\delta\left(\frac{d^2w}{dx^2}\right) = \left[\begin{matrix} 0 & \frac{6x}{l^2} - \frac{4}{l} & -\frac{6x}{l} + \frac{2}{l}\end{matrix}\right]\frac{\partial\varepsilon}{\partial u}\,\delta u$$

$$\delta\left(\frac{du_a}{dx}\right) = \left[\begin{matrix} \frac{1}{l} & 0 & 0\end{matrix}\right]\frac{\partial\varepsilon}{\partial u}\,\delta u$$

met $\frac{\partial\varepsilon}{\partial u}$ de volgende matrix:

$$\frac{\partial\varepsilon}{\partial u} = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{1}{l} & -1 & 0 & -\frac{1}{l} & 0 \\ 0 & -\frac{1}{l} & 0 & 0 & \frac{1}{l} & 1 \end{bmatrix} \qquad (I\,7)$$

Substitutie in formule van minimale potentiële energie levert de volgende betrekking:

$$\int_{x=0}^{l}\left\{ EI\,\check{\varepsilon}\begin{bmatrix} 0 \\ \frac{6x}{l^2}-\frac{4}{l} \\ -\frac{6x}{l^2}+\frac{2}{l}\end{bmatrix}\begin{bmatrix} 0 & \frac{6x}{l^2}-\frac{4}{l} & -\frac{6x}{l^2}+\frac{2}{l}\end{bmatrix} + EF\begin{bmatrix}\frac{1}{l}\\ 0 \\ 0\end{bmatrix}\begin{bmatrix}\frac{1}{l} & 0 & 0\end{bmatrix}\right\}\frac{\partial\varepsilon}{\partial u}\,\delta u = \check{f}\,\delta u$$

oftewel:

$$\check{\varepsilon}\, S\, \frac{\partial\varepsilon}{\partial u}\,\delta u = \check{f}\,\delta u$$

$$\check{\varepsilon}\, S\, \frac{\partial\varepsilon}{\partial u} = \check{f} \qquad (I\ 8)$$

Tenslotte zijn we in staat om het verband tussen de rekken en de verplaatsingen te leggen met relatie (I 1)

$$\varepsilon = C\,u \qquad (I\ 9)$$

De matrix C heeft precies dezelfde gedaante als de matrix $\frac{\partial\varepsilon}{\partial u}$, hetgeen te verklaren is uit het lineair blijven van de relatie (I 1).
Relatie (I 8) kunnen we nu als volgt schrijven:

$$\check{u}\,\check{C}\,S\,C = \check{f}$$

$$\check{C}\,S\,C\,u = f \qquad (I\ 10)$$

Relatie (I 10) is de gezochte relatie tussen belastings- en verplaatsingsvektor. De stijfheidsmatrix Q_e is nu gedefinieerd als:

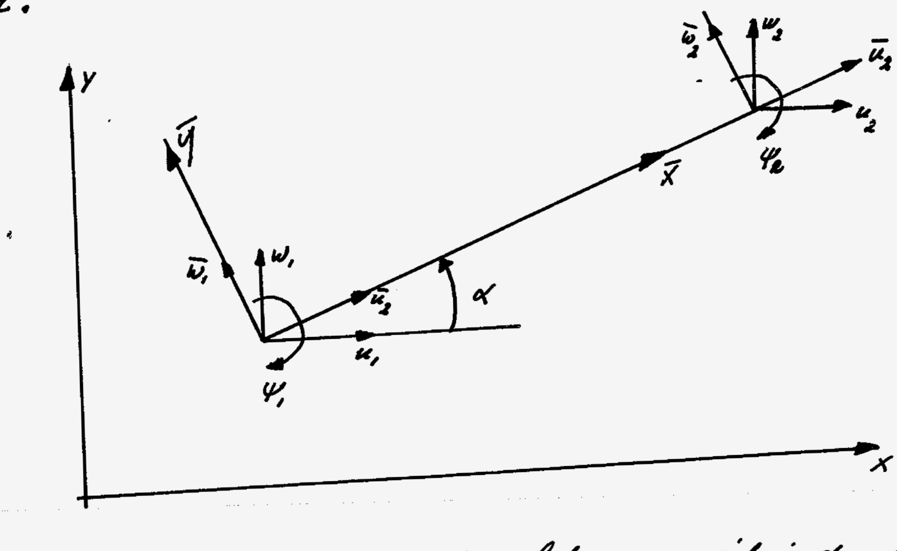
$$Q_e = \check{C}\,S\,C$$

die alleen geldt voor een balkelement, waarvan de staafas samenvalt met de x-as

$$Q_e = \begin{bmatrix} S_1 & 0 & 0 & -S_1 & 0 & 0 \\ & 12S_2 & -6S_2 l & 0 & -12S_2 & -6S_2 l \\ & & 4S_2 l^2 & 0 & 6S_2 l & 2S_2 l^2 \\ & & & S_1 & 0 & 0 \\ & & & & 12S_2 & 6S_2 l \\ & & & & & 4S_2 l^2 \end{bmatrix}$$

$$S_1 = \frac{EF}{l} \qquad S_2 = \frac{EI}{l^3}$$

In het algemeen zal de staafas van het balkelement niet met de x-as van het globale assenstelsel samenvallen. We voeren dan een transformatie uit. We nemen aan, dat het element een hoek α met de positieve globale x-as maakt.



Stel nu dat de verplaatsvektor u, dit is de verplaatsingsvektor t.o.v. het globale assenstelsel, bekend is. Willen we om de rekvektor te bepalen relatie (I 6) hanteren, dan moeten we de verplaatsingsvektor u transformeren naar het lokale $\bar{x}$-$\bar{y}$-assenstelsel.

$$\bar{u} = T\,u \qquad (I\ 11)$$

$$T = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 & 0 & 0 & 0 \\ -\sin\alpha & \cos\alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & 0 & 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Passen we relatie (I9) op het lokale assenstelsel toe dan verkrijgen we:

$$\bar{\varepsilon} = C\,\bar{u}$$

$$\bar{\varepsilon} = C\,T\,u$$

zodat we de stijfheidsmatrix t.o.v. globale assenstelsel als volgt kunnen schrijven:

$$\overset{\blacktriangledown}{T}\,\overset{\blacktriangledown}{C}\,S\,C\,T\;u = f.$$

of: $\quad Q\,u = f. \qquad\qquad$ (I 12)

(I12) is de relatie tussen belastings- en verplaatsingsvektor t.o.v. globale assenstelsel.

II Niet-Lineaire Theorie

Evenwicht van uitwendige belastingen en inwendige spanningen wordt gekarakteriseerd door gelijkheid van uitwendige arbeid en inwendige arbeid bij willekeurige kinematisch toelaatbare randvoorwaarden.

$$\int_V \sigma_{ij}\,\delta\varepsilon_{ij}\,dV = \int_V f_i\,\delta u_i\,dV + \int_A p_i\,\delta u_i\,dA$$

Passen we deze virtuele arbeidsvergelijking uit de continuummechanika toe op het discrete model van de constructie, die we uit een aantal elementen samengesteld denken, dan verkrijgen we:

$$\sum_{k=1}^{N} \sigma_{ij}^{k}\,\delta\varepsilon_{ij}^{k} = f_k\,\delta u_k \qquad (\text{II } 1)$$

In dit hoofdstuk zullen we ons beperken tot het beschrijven van één element:

$$\sigma_{ij}\,\delta\varepsilon_{ij} = f_i\,\delta u_i$$

Stel nu, dat we de relatie tussen verplaatsingsvektor en rekvektor kennen, dan kunnen we de volgende relatie afleiden:

$$\varepsilon_i = \varepsilon(u_p) \qquad p = 1 \ldots 6 \qquad i = 1 \ldots 3$$

$$\delta\varepsilon_i = \frac{\partial\varepsilon_i}{\partial u_p}\,\delta u_p$$

$$\sigma_i\,\frac{\partial\varepsilon_i}{\partial u_p}\,\delta u_p = f_p\,\delta u_p \qquad (\text{II } 2)$$

Formule (II 2) is een algemene relatie, die we in principe voor allerlei niet-lineaire problemen kunnen toepassen.

Bestuderen we het lineaire elastische element, dan kunnen we de volgende relaties opschrijven voor één element:

1) Het verband tussen de rekvektor en de verplaatsingsvektor is lineair en wordt gekarakteriseerd door een „ overdrachtsmatrix" D

$$\varepsilon_i = D_{ij} u_j \qquad j = 1 \ldots 6$$

2) Bovendien weten we, dat het verband tussen de gegeneraliseerde rekken en spanningen ook lineair is:

$$\sigma_i = S_{ij} \varepsilon_j \qquad i = 1 \ldots 3 \quad j = 1 \ldots 3$$

Substitueren we voorgaande relaties in (II 2), de algemene virtuele arbeidsvergelijking, dan verkrijgen we de reeds bekende betrekking (I 8)

$$\varepsilon_i S_{ij} \frac{d\varepsilon_j}{du_p} = f_p \qquad i = 1 \ldots 3 \quad j = 1 \ldots 3 \quad p = 1 \ldots 6$$

Om het niet-lineaire gedrag van de constructie te bestuderen maken we onderscheid tussen 2 vormen van niet-lineair gedrag:

ⓐ <u>fysische niet-lineariteit.</u>
Dit kan zijn, niet lineair gedrag bij kleine vervormingen, zoals kruip, plasticiteit, en niet-lineair gedrag bij grote vervormingen, zoals rubberelasticiteit, grote plastische deformatie.

ⓑ <u>Geometrische niet-lineariteit</u>
Bij geometrische niet-lineariteit gaan kleine vervormingen gepaard met grote verplaatsingen, terwijl de lineaire relatie tussen de gegeneraliseerde rek- en spanningsgrootheden gehandhaafd blijft.

Wij beperken ons tot geometrische niet-lineariteit. Zoals gezegd blijft de relatie $\sigma_i = S_{ij} \varepsilon_j$ geldig.

Het verband tussen rekken en verplaatsingen mag niet meer gelineariseerd worden zodat geldt:

$$\delta \varepsilon_i = \frac{\partial \varepsilon_i}{\partial u_p} \delta u_p$$

Substitutie in (II 2) levert:

$$\varepsilon_i S_{ij} \frac{\partial \varepsilon_j}{\partial u_p} = f_p \qquad (\text{II } 3)$$

Formule (II 3) is een stelsel niet-lineaire vergelijkingen voor één element, dat het geometrische niet-lineair gedrag van dat element beschrijft.
Voor de gehele constructie geldt:

$$\sum_{k=1}^{N} \varepsilon_i S_{ij} \frac{\partial \varepsilon_j}{\partial u_p} = f_k \qquad (\text{II } 4)$$

$i = 1 \dots 3$
$j = 1 \dots 3$
$p = 1 \dots 6$
$k = 1 \dots$ aantal vrijheidsgraden van de constructie.

III Oplossingsmethoden voor een stelsel niet-lineaire vergelijkingen.

In litteratuur (II) worden verschillende methodes aangegeven hoe men een niet-lineair stelsel vergelijkingen kan oplossen. We zullen in dit hoofdstuk de verschillende methodes behandelen.

a) De Newton-Raphson methode.

Stel het stelsel niet-lineaire vergelijkingen heeft de volgende gedaante.

$$\begin{aligned} F_1 \; (x_1, \ldots \ldots x_n) &= b_1 \\ F_2 \; (x_1, \ldots \ldots x_n) &= b_2 \\ &\vdots \\ F_n \; (x_1, \ldots \ldots x_n) &= b_n \end{aligned} \qquad (\text{III } 1)$$

$$A_{NL} \, (x_p) = b_\alpha \qquad (\text{III } 2)$$

A_{NL} = niet-lineaire matrix
b_α = voorgeschreven rechterlid
x_p = de onbekende vektor.

We schrijven nu voor het stelsel (III 2)

$$E_\alpha = A_{NL} \, (x_p) - b_\alpha \qquad (\text{III } 3)$$

Formule (III 3) zal dan slechts gelijk aan nul zijn, als we voor de vektor x_p de exakte oplossing hebben gevonden. Stel dat we deze kennen, dan willen we de oplossing kennen voor een kleine stap verder in de voorgeschreven vektor b_α. Hiervoor gaan we formule (III 3) in een Taylorreeks ontwikkelen.

$$E_\alpha + \Delta E_\alpha = E_\alpha + \frac{\partial E_\alpha}{\partial x_q} \, \Delta x_q + \ldots\ldots$$

$$= A_{NL} \, (x_p) - b_\alpha + \frac{\partial A_{NL}(x_p)}{\partial x_q} \, \Delta x_q + \ldots\ldots \qquad (\text{III } 4)$$

Vergelijking (III 4) moet voor de exakte oplossing voor x_p gelijk aan nul zijn.

$$\frac{\partial A_{NL}(x_p)}{\partial x_q} \Delta x_q = b_\alpha - A_{NL}(x_p) \qquad (III\ 5)$$

Met formule (III 5) zijn we nu in staat om het niet-lineaire stelsel voor een bepaalde vektor b op te lossen. De werkwijze gaat dan als volgt:

1) Kies een startwaarde x_0
2) Los (III 5) op voor deze waarde x_0
3) Los stelsel (III 5) dan weer op voor $x_1 = x_0 + \Delta x$.
Δx is de correctie op x_0 gevonden met
$$\Delta x = \left[\{A_{NL}(x_p)\}_{x_0} - b_\alpha \right] * \frac{-1}{\frac{\partial A_{NL}(x_p)}{\partial x_q}}$$
4) etc.

Het zal duidelijk zijn, dat het hiervoor beschreven proces een iteratief proces is, waarbij bij denselfde iteratiestap de term $\frac{\partial A_{NL}}{\partial x_q}$ als konstant wordt beschouwd.

b) „Incremental approach"

Hierbij lossen we het stelsel (III 2) op, door de vektor b_α stapsgewijs op te bouwen met een vektor Δb_α en de bij deze incrementvektor horende Δx_q te bepalen.
We gaan weer uit van het stelsel (III 2)

$$A_{NL}(x_p) = b_\alpha$$

$$A_{NL}(x_p + \Delta x_p) = b_\alpha + \Delta b_\alpha$$

$$A_{NL}(x_p) + \frac{\partial A_{NL}(x_p)}{\partial x_q} \Delta x_q = b_\alpha + \Delta b_\alpha$$

$$A_{NL}(x_p) - b_\alpha = 0$$

$$\frac{\partial A_{NL}(x_p)}{\partial x_q} \Delta x_q = \Delta b_\alpha$$

Stel nu dat we n stappen gedaan hebben, dan verkrijgen we:

$$\left\{\frac{\partial A_{NL}(x_p)}{\partial x_q}\right\}_{x_p = x_p^n} \Delta x_q = \Delta b_\alpha \qquad (\text{III } 6)$$

Het increment Δb_α hoeft in principe niet constant te blijven. De oplossing wordt:

$$x_p^{n+1} = x_p^n + \Delta x_q$$

c) <u>De gemodificeerde Newton-Raphson methode</u>

Deze methode is een combinatie tussen de „Newton-Raphson" en de „incremental approach". We schrijven nu de oorspronkelijke relatie in een zodanige vorm, dat bij elk increment een slag Newton-Raphson wordt uitgevoerd, oftewel bij elk increment één iteratieslap.

$$A_{NL}(x_p) = b_\alpha$$

$$A_{NL}(x_p) + \frac{\partial A_{NL}(x_p)}{\partial x_q} \Delta x_q = b_\alpha + \Delta b_\alpha$$

$$\frac{\partial A_{NL}(x_p)}{\partial x_q} \Delta x_q = \Delta b_\alpha + b_\alpha - A_{NL}(x_p)$$

We schrijven voorgaande formule weer na n stappen op:

$$\left\{\frac{\partial A_{NL}(x_p)}{\partial x_q}\right\}_{x_p = x_p^n} \Delta x_q = \Delta b_\alpha + \{b_\alpha\}_n - \{A_{NL}(x_p)\}_{x_p = x_p^n} \qquad (\text{III } 7)$$

De oplossing na n stappen is nu:

$$b_n = \sum_{i=1}^{n} \Delta b_i$$

$$x_n = \sum_{i=1}^{n} \Delta x_i$$

Het zal duidelijk zijn, dat na n stappen de relatie $A_{NL}(x_p) - b_\alpha$ niet gelijk aan nul is, omdat het een benaderingsmethodiek is. Als we nu dit verschil berekenen en als correctie bij het volgende increment in de voorgeschreven vektor b_α toevoegen, dan is dit de slag Newton-Raphson.

IV Toepassing van de oplossingsmethodieken voor niet-lineaire stelsels in de elementenmethode

In hoofdstuk II is gesteld, dat we ons zullen beperken tot geometrisch niet-lineaire problemen. We vonden daar het volgende stelsel, genoteerd met indexnotatie:

$$\sum_{k=1}^{N} \varepsilon_i \, S_{ij} \frac{\partial \varepsilon_j}{\partial u_p} = f_k \qquad (\text{IV } 1)$$

We zullen nu de oplossingsmethodieken zoals beschreven in hoofdstuk III toepassen op dit stelsel. Voor elke methode is een computerprogramma gemaakt, aan de hand waarvan de beslissing is genomen welke methode voldoend nauwkeurige resultaten geeft.

a) „Incremental approach"

Het niet-lineaire stelsel heeft nu de volgende gedaante

$$\left\{ \sum_{k=1}^{N} \varepsilon_i \, S_{ij} \frac{\partial \varepsilon_j}{\partial u_p} \right\} = f_k$$

De incrementele vergelijking wordt nu:

$$\left\{ \sum_{k=1}^{N} \frac{\partial \varepsilon_i}{\partial u_q} S_{ij} \frac{\partial \varepsilon_j}{\partial u_p} + \varepsilon_i S_{ij} \frac{\partial^2 \varepsilon_j}{\partial u_q \partial u_p} \right\}_{u_k = u_k^n} \Delta u_k = \Delta f_k$$

$$\left\{ \sum_{k=1}^{N} \frac{\partial \varepsilon_i}{\partial u_q} S_{ij} \frac{\partial \varepsilon_j}{\partial u_p} + \sigma_j \frac{\partial^2 \varepsilon_j}{\partial u_q \partial u_p} \right\}_{u_k = u_k^n} \Delta u_k = \Delta f_k \qquad (\text{IV } 2)$$

$i = 1 \dots 3$ $j = 1 \dots 3$
$p = 1 \dots 6$ $q = 1 \dots 6$
$k = 1 \dots$ vrijheidsgraden

Aan de verschillende componenten in deze vergelijking zal in hoofdstuk V aandacht worden besteed.

b) De „gemodificeerde Newton-Raphson" methode.

Het stelsel (IV 1) kunnen we nu met behulp van formule (III 7) als volgt schrijven:

$$\left\{\sum_{k=1}^{N}\frac{\partial\varepsilon_i}{\partial u_q}\,S_{ij}\,\frac{\partial\varepsilon_j}{\partial u_p}+\sigma_j\,\frac{\partial^2\varepsilon_j}{\partial u_q\partial u_p}\right\}_{u_k=u_k^n}\Delta u_k=\Delta f_k+f_k^n-\left\{\sum_{k=1}^{N}\varepsilon_i\,S_{ij}\,\frac{\partial\varepsilon_j}{\partial u_p}\right\}_{u_k=u_k^n}\quad(\text{IV}\,3)$$

In deze formule is f_k^n de belastingvektor na n stappen.
Ook hier zal in hoofdstuk V aandacht besteed worden aan de componenten.

c) De „Newton - Raphson" methode

~~In deze laatste methode is in principe het stapsgewijs~~ proces verdwenen. De uitwendige belastingsvektor wordt in één keer aangebracht en we itereren naar de verplaatsingvektor u_k met de volgende formule:

$$\left\{\sum_{k=1}^{N}\frac{\partial\varepsilon_i}{\partial u_q}\,S_{ij}\,\frac{\partial\varepsilon_j}{\partial u_p}+\sigma_j\,\frac{\partial^2\varepsilon_j}{\partial u_q\partial u_p}\right\}_{u=u_k^t}\Delta u_k=f_k-\left\{\sum_{k=1}^{N}\varepsilon_i\,S_{ij}\,\frac{\partial\varepsilon_j}{\partial u_p}\right\}_{u=u_k^t}\quad(\text{IV}\,4)$$

u_k^t = verplaatsingsvektor van de constructie na t itteratiestappen.

Deze Newton- Raphson methode, waarbij de uitwendige belasting in één keer wordt opgebracht, levert in het algemeen moeilijkheden op, omdat het itteratieproces divergeert. Om dit te voorkomen, brengen we de uitwendige belasting nu ook in een aantal stappen aan, waarbij ná élke stap geïtereerd wordt naar de exakte oplossing. Is de benadering voldoende nauwkeurig, dat wil zeggen, dat het verschil

$$f_k^n-\left\{\sum_{k=1}^{N}\varepsilon_i\,S_{ij}\,\frac{\partial\varepsilon_j}{\partial u_p}\right\}_{u=u_k^{n,t}}$$

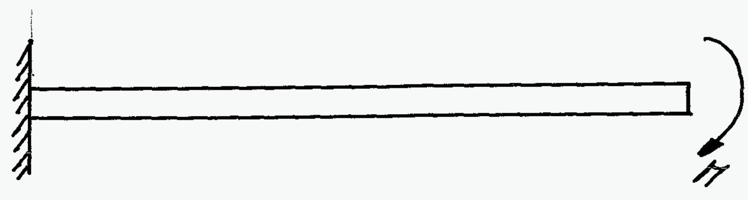
voldoende dicht bij nul ligt, dan eindigt het itteratieproces.

$u_k^{n,t}$ = verplaatsingsvektor van de constructie na n stappen in de belastingvektor en t itteratiestappen na de laatste belastingstap

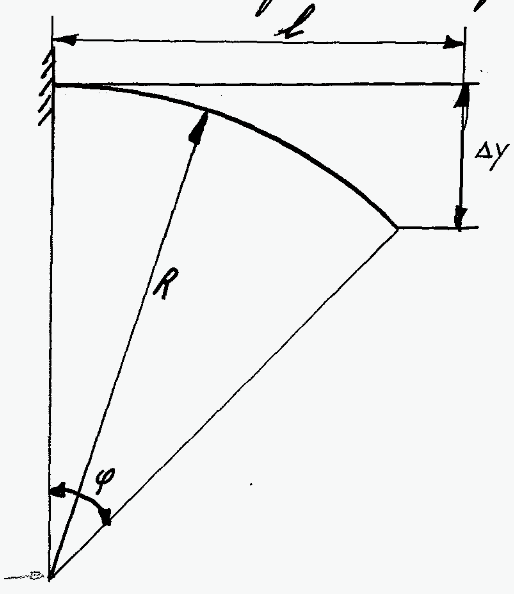
De toelaatbare afwijkingen kan men zelf bepalen. Als aan de hiervoor gestelde criteria wordt voldaan, dan wordt de uitwendige belasting met een stap verhoogd.

$$f_k^{n+1} = f_k^{n} + \Delta f_k$$

Volledigheidshalve moet nog opgemerkt worden, dat de incrementele belastingsvektor niet voor elke stap dezelfde grootte behoeft te hebben. In het computerprogramma, dat in hoofdstuk VIII beschreven zal worden, is de grootte van het increment afhankelijk gesteld van het aantal iteratiestappen dat na elke stap uitgevoerd moet worden. In appendex I GRAF. no I zijn de resultaten van de hiervoor beschreven oplossingsmethodes vergeleken aan de hand van een voorbeeld, wat exakt berekend kan worden.



Het voorbeeld bestond uit een eenzijdig ingeklemde balk, welke door een uitwendig moment wordt belast. We weten dat de balk t.g.v. het buigend moment volgens een cirkelboog uitbuigt.



$$\Delta y = R - R\cos\varphi$$

$$\varphi = \frac{l}{R}$$

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

$$\Delta y = -\frac{EI}{M}(R - R\cos\varphi)$$

Wij hebben nu 2 contrôle-mogelijkheden:
1) De exakte Δy vergelijken met de benaderde Δy uit de verschillende oplossingsmethodes.
2) Verder kunnen we contrôleren of de knooppunten van de elementen allemaal op één cirkel liggen.

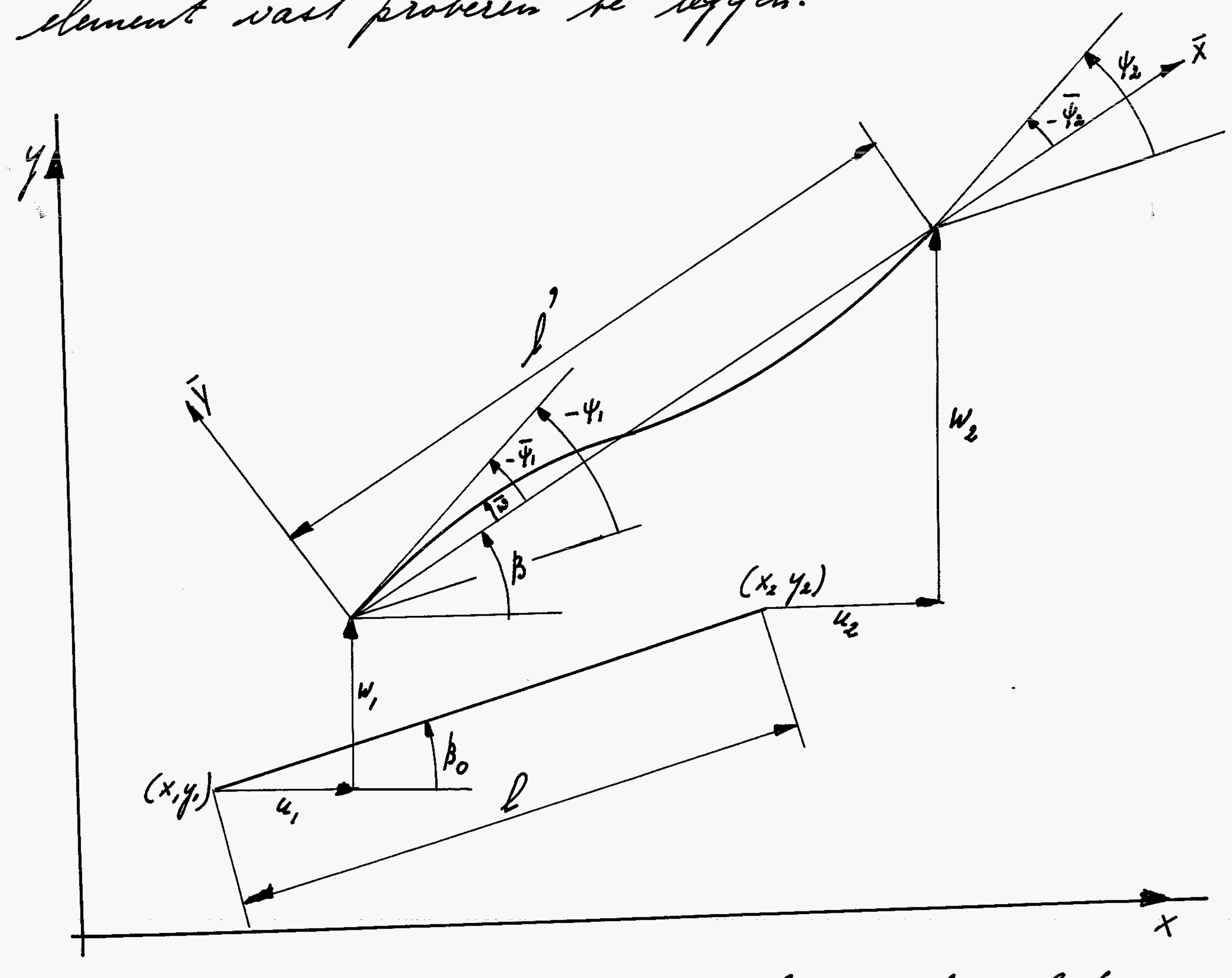
Deze contrôles zijn gedaan voor de verschillende oplossingsmethodes en het bleek dat de Newton-Raphson methode inderdaad de exakte oplossing benaderde, terwijl de andere 2 methodes steeds verder van de exakte waarde gingen afwijken. Op grond hiervan is besloten om het Newton Raphson proces toe te passen.

V Het balkelement bij geometrisch niet-lineariteit met de afleiding van $\frac{\partial \varepsilon_i}{\partial u_p}$ en $\frac{\partial^2 \varepsilon_i}{\partial u_q \partial u_p}$.

Bij geometrische niet-lineariteit blijven de rek-spanningsrelaties

$$\sigma = S\,\varepsilon$$

lineair, terwijl de rekken geen lineaire funkties meer van de verplaatsingen zijn. We moeten dus in eerste instantie de niet-lineaire relaties tussen rekken en verplaatsingen van het element vast proberen te leggen.



We definiëren nu de rekgrootheden als volgt:

$$\varepsilon_1 = \left[(x_2 - x_1 + u_2 - u_1)^2 + (y_2 - y_1 + w_2 - w_1)^2\right]^{1/2} + \frac{1}{2}\int_{\bar{x}=0}^{l'} \left(\frac{d\bar{w}}{d\bar{x}}\right)^2 d\bar{x} - l$$

$$\varepsilon_2 = \left[-\psi_1 + \operatorname{arctan}\frac{y_2 - y_1}{x_2 - x_1} - \operatorname{arctan}\frac{y_2 - y_1 + w_2 - w_1}{x_2 - x_1 + u_2 - u_1}\right] \qquad (\text{V } 1)$$

$$\varepsilon_3 = \left[\psi_2 - \arctan \frac{y_2 - y_1}{x_2 - x_1} + \arctan \frac{y_2 - y_1 + w_2 - w_1}{x_2 - x_1 + u_2 - u_1} \right]$$

$\bar{w}$ is het verplaatsingsveld van het balkelement in de $\bar{y}$-richting van het $\bar{x}, \bar{y}$ assenkruis.
Wij zijn in staat om $\int_{\bar{x}=0} \left(\frac{d\bar{w}}{d\bar{x}}\right)^2 d\bar{x}$ in de componenten van de verplaatsingsvektor u van het element uit te drukken, zie appendix II.

$$\frac{1}{2} \int_{\bar{x}=0}^{l'} \left(\frac{d\bar{w}}{d\bar{x}}\right)^2 = \frac{l'}{30} \left[2\bar{\psi}_1^{\,2} - \bar{\psi}_1 \bar{\psi}_2 + 2\bar{\psi}_2^{\,2} \right]. \qquad (\text{V}\,2)$$

Substitueren we relatie (V 2) in (V 1) dan verkrijgen we het volgende stelsel vergelijkingen voor de rekken:

$$\varepsilon_1 = \left[(x_2 - x_1 + u_2 - u_1)^2 + (y_2 - y_1 + w_2 - w_1)^2 \right]^{1/2} \left[1 + \frac{1}{30} \left(2\bar{\psi}_1^{\,2} - \bar{\psi}_1 \bar{\psi}_2 + 2\bar{\psi}_2^{\,2} \right) \right] - l$$

$$\varepsilon_2 = \left[-\psi_1 + \arctan \frac{y_2 - y_1}{x_2 - x_1} - \arctan \frac{y_2 - y_1 + w_2 - w_1}{x_2 - x_1 + u_2 - u_1} \right] \qquad (\text{V}\,3)$$

$$\varepsilon_3 = \left[\psi_2 - \arctan \frac{y_2 - y_1}{x_2 - x_1} + \arctan \frac{y_2 - y_1 + w_2 - w_1}{x_2 - x_1 + u_2 - u_1} \right]$$

Zoals in hoofdstuk IV gesteld is, kunnen we nu vrij eenvoudig uit (V 3) de relaties

$$\frac{\partial \varepsilon_i}{\partial u_p} \quad \text{en} \quad \frac{\partial^2 \varepsilon_i}{\partial u_q \partial u_p} \quad \text{afleiden.}$$

$$\frac{\partial \varepsilon_1}{\partial u_p} = \begin{bmatrix} -\cos\beta - \frac{\cos\beta}{30}\left[2\bar\psi_1^2 - \bar\psi_1\bar\psi_2 + 2\bar\psi_2^2\right] + \frac{\sin\beta}{10}\left[\bar\psi_1 + \bar\psi_2\right] \\ -\sin\beta - \frac{\sin\beta}{30}\left[2\bar\psi_1^2 - \bar\psi_1\bar\psi_2 + 2\bar\psi_2^2\right] - \frac{\cos\beta}{10}\left[\bar\psi_1 + \bar\psi_2\right] \\ \frac{l'}{30}\left[4\bar\psi_1 - \bar\psi_2\right] \\ \cos\beta + \frac{\cos\beta}{30}\left[2\bar\psi_1^2 - \bar\psi_1\bar\psi_2 + 2\bar\psi_2^2\right] - \frac{\sin\beta}{10}\left[\bar\psi_1 + \bar\psi_2\right] \\ \sin\beta + \frac{\sin\beta}{30}\left[2\bar\psi_1^2 - \bar\psi_1\bar\psi_2 + 2\bar\psi_2^2\right] + \frac{\cos\beta}{10}\left[\bar\psi_1 + \bar\psi_2\right] \\ \frac{l'}{30}\left[-\bar\psi_1 + 4\bar\psi_2\right] \end{bmatrix}$$

$$\frac{\partial \varepsilon_2}{\partial u_p} = \left[-\frac{\sin\beta}{l'} \quad \frac{\cos\beta}{l'} \quad -1 \quad \frac{\sin\beta}{l'} \quad -\frac{\cos\beta}{l'} \quad 0 \right]$$

$$\frac{\partial \varepsilon_3}{\partial u_p} = \left[\frac{\sin\beta}{l'} \quad -\frac{\cos\beta}{l'} \quad 0 \quad -\frac{\sin\beta}{l'} \quad \frac{\cos\beta}{l'} \quad 1 \right]$$

We gaan nu de verschillende termen afschatten:

$$\bar\psi_1 \ll 1$$
$$\bar\psi_2 \ll 1$$
$$\frac{\Delta l}{l} \ll 1 \qquad \Delta l = l' - l$$

Bovenstaande ongelijkheden blijven geldig, omdat de rek-spanningsrelaties lineair blijven, bij geometrische niet-lineariteit.
We schrijven nu $\frac{\partial \varepsilon_i}{\partial u_p}$ in de vorm van de som van 2 matrices:

$$\frac{\partial \varepsilon_i^o}{\partial u_p} = Q_1 + O(\varepsilon) \qquad (\text{IV } 4)$$

$$Q_1 = \begin{bmatrix} -\cos\beta & -\sin\beta & 0 & \cos\beta & \sin\beta & 0 \\ -\dfrac{\sin\beta}{l^0} & \dfrac{\cos\beta}{l^0} & -1 & \dfrac{\sin\beta}{l^0} & -\dfrac{\cos\beta}{l^0} & 0 \\ \dfrac{\sin\beta}{l^0} & -\dfrac{\cos\beta}{l^0} & 0 & -\dfrac{\sin\beta}{l^0} & \dfrac{\cos\beta}{l^0} & 1 \end{bmatrix}$$

$$O(\varepsilon) = \begin{bmatrix} \dfrac{\sin\beta}{10}(-\varepsilon_2+\varepsilon_3) & -\dfrac{\cos\beta}{10}(-\varepsilon_2+\varepsilon_3) & \dfrac{l^0}{30}(-4\varepsilon_2-\varepsilon_3) & -\dfrac{\sin\beta}{10}(-\varepsilon_2+\varepsilon_3) & \dfrac{\cos\beta}{10}(-\varepsilon_2+\varepsilon_3) & \dfrac{l^0}{30}(\varepsilon_2+4\varepsilon_3) \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Om de termen van $\dfrac{\partial^2\varepsilon_i}{\partial u_q \partial u_p}$ te verkrijgen, moeten we de termen van $\dfrac{\partial\varepsilon_i}{\partial u_p}$ naar de componenten van de verplaatsings-vektor differentiëren. Om nu de orde grootte te bepalen, welke we in deze termen mee willen nemen, moeten we terug naar de oorspronkelijke vergelijking (IV 4).

$$\left\{\sum_{k=1}^{N} \frac{\partial\varepsilon_i}{\partial u_q} S_{ij} \frac{\partial\varepsilon_j}{\partial u_p} + \sigma_j \frac{\partial^2\varepsilon_j}{\partial u_q \partial u_p}\right\}_{u=u_k^t} \Delta u_k = f_k - \left\{\sum_{k=1}^{N} \varepsilon_i S_{ij} \frac{\partial\varepsilon_j}{\partial u_p}\right\}_{u=u_k^t}$$

In de termen $\dfrac{\partial\varepsilon_i}{\partial u_p}$ hebben we de $O(\varepsilon)$ meegenomen. Daar $\sigma = S\varepsilon$ zelf van de orde epsilon is behoeven we in de termen van $\dfrac{\partial^2\varepsilon_j}{\partial u_p \partial u_q}$ slechts de $O(1)$ mee te nemen. We beschouwen daarom eerst de term $\dfrac{\partial^2\varepsilon_1}{\partial u_1 \partial u_1}$.

$$\frac{\partial^2\varepsilon_1}{\partial u_1 \partial u_1} = \frac{\sin^2\beta}{l} + \frac{2\sin^2\beta}{10\,l} - \frac{\Delta l}{l^2}\sin^2\beta - \frac{\Delta l}{10\,l^2}\,2\sin^2\beta +$$

$$+ \frac{\sin\beta\cos\beta}{10\,l}\left[\bar\Psi_1 + \bar\Psi_2\right] -$$

$$- \frac{\cos\beta}{30}\left[\frac{3\bar\Psi_1\sin\beta}{l} - \frac{3\Delta l}{l^2}\bar\Psi_1\sin\beta + 3\bar\Psi_2\frac{\sin\beta}{l} - 3\bar\Psi_2\frac{\sin\beta\,\Delta l}{l^2}\right] +$$

$$+\frac{\sin^2\beta}{30\,l}\left[2\bar{\psi}_1^2-\bar{\psi}_1\bar{\psi}_2+2\bar{\psi}_2^2\right]-\frac{\Delta l}{l^3}\sin^2\beta\left[2\bar{\psi}_1^2-\bar{\psi}_1\bar{\psi}_2+2\bar{\psi}_2^2\right]-$$

$$-\frac{\Delta l}{l^2}\sin\beta\cos\beta\left[\bar{\psi}_1+\bar{\psi}_2\right]$$

Zoals reeds gesteld is, is alleen $O(1)$ belangrijk, zodat we kunnen schrijven:

$$\frac{\partial^2\varepsilon_1}{\partial u_1\partial u_1}=\frac{36}{30}\frac{\sin^2\beta}{l}$$

De bepaling van de andere termen verloopt geheel analoog, zodat we tenslotte de volgende resultaten vinden:

$$\frac{\partial^2\varepsilon_1}{\partial u_q\partial u_p}=\frac{1}{30}\begin{bmatrix}\frac{36\sin^2\beta}{l} & -\frac{36\sin\beta\cos\beta}{l} & 3\sin\beta & -\frac{36\sin^2\beta}{l} & \frac{36\sin\beta\cos\beta}{l} & 3\sin\beta\\ & \frac{36\cos^2\beta}{l} & -3\cos\beta & \frac{36\sin\beta\cos\beta}{l} & -\frac{36\cos^2\beta}{l} & -3\cos\beta\\ & & 4l & -3\sin\beta & 3\cos\beta & -l\\ & \text{sym.} & & \frac{36\sin^2\beta}{l} & -\frac{36\sin\beta\cos\beta}{l} & -3\sin\beta\\ & & & & \frac{36\cos^2\beta}{l} & 3\cos\beta\\ & & & & & 4l\end{bmatrix}$$

$$\frac{\partial^2\varepsilon_2}{\partial u_q\partial u_p}=-\frac{1}{l^2}\begin{bmatrix}-\sin 2\beta & \cos 2\beta & 0 & \sin 2\beta & -\cos 2\beta & 0\\ & \sin 2\beta & 0 & -\cos 2\beta & -\sin 2\beta & 0\\ & & 0 & 0 & 0 & 0\\ & \text{sym.} & & -\sin 2\beta & \cos 2\beta & 0\\ & & & & \sin 2\beta & 0\\ & & & & & 0\end{bmatrix}$$

$$\frac{\partial^2\varepsilon_3}{\partial u_q\partial u_p}=-\frac{\partial^2\varepsilon_2}{\partial u_q\partial u_p}$$

Definieren we nu $\frac{\partial^2 \varepsilon_1}{\partial u_q \partial u_p} = G_1$, $\frac{\partial^2 \varepsilon_2}{\partial u_q \partial u_p} = G_2$ en $\frac{\partial^2 \varepsilon_3}{\partial u_q \partial u_p} = G_3$, dan kunnen we relatie (IV 4) als volgt schrijven in matrixvorm:

$$\sum_{k=1}^{N}\left[\left[Q_1+O(\varepsilon)\right]^T S\left[Q_1+O(\varepsilon)\right] + N G_1 + M_0 G_2 + M_l G_3\right]_{u=u_{n,t}} \Delta u = \qquad (V\,5)$$

$$\Delta f + f_n - \sigma^T\left[Q_1+O(\varepsilon)\right]_{u=u_{n,t}}$$

f_n = uitwendige belastingsvektor na n stappen.

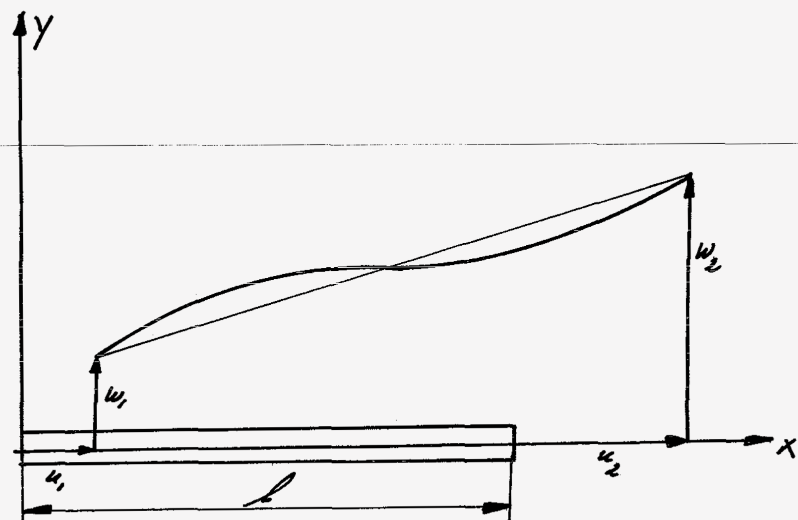
$u_{n,t}$ = verplaatsingsvektor na n stappen en t iteratiestappen.

Op formule (V 5) is nu het computerprogramma 05064608 „Berekening veerkarakteristiek met behulp van de methode der eindige elementen, waarbij de stapgrootte in de belasting afhankelijk is van het aantal iteratiestappen" gebaseerd.

Δu en Δf: zijn de incrementen van de respectievelijke verplaatsings- en belastingsvektor van de gehele constructie.

VI Fysische interpretatie van de gebruikte matrices van het balkelement.

Om een uitspraak te doen over de fysische betekenis van de matrices $O(\varepsilon)$, G_1, G_2 en G_3 zoals afgeleid in hoofdstuk V, gaan we uit van een balkelement, dat onvervormd is en waarvan de staafas samenvalt met de x-as.



Ten gevolge van een uitwendige belasting ontstaat de verplaatsingsvektor

$$\vec{u} = [u_1 \; w_1 \; \varphi_1 \; u_2 \; w_2 \; \varphi_2]$$

De niet-lineaire rekformules gaan we nu voor dit balkelement in een reeks ontwikkelen, waarin we termen van hogere orde verwaarlozen: De rekformules hebben voor het hiervoor beschreven element de volgende gedaante:

$$\varepsilon_1 = \left[(l+u_2-u_1)^2+(w_2-w_1)^2\right]^{1/2}\left[1+\frac{1}{30}\left(2\bar{\varphi}_1^2-\bar{\varphi}_1\bar{\varphi}_2+2\bar{\varphi}_2^2\right)\right]-l$$

$$\varepsilon_2 = \left[-\varphi_1 - \arctan\frac{w_2-w_1}{l+u_2-u_1}\right] \qquad (VI\,1)$$

$$\varepsilon_3 = \left[\varphi_2 + \arctan\frac{w_2-w_1}{l+u_2-u_1}\right]$$

We gebruiken de volgende benaderingen:

$$\left[(l+u_2-u_1)^2+(w_2-w_1)^2\right]^{1/2} = l+u_2-u_1+\frac{(w_2-w_1)^2}{2l}+\ldots \qquad (VI\,2)$$

$$\operatorname{arctan} \frac{w_2-w_1}{l+u_2-u_1} = \frac{w_2-w_1}{l} - \frac{(w_2-w_1)(u_2-u_1)}{l^2} + \dots \qquad (VI\ 3)$$

In (VI 2) en (VI 3) zijn termen van $\left(\frac{w_2-w_1}{l}\right)^n$ en $\frac{(w_2-w_1)(u_2-u_1)^{n-1}}{l^n}$

met $n > 2$ verwaarloosd.
Substitutie van (VI 2) en (VI 3) in (VI 1) leveren de volgende formules op, waarvan het eerste gedeelte de lineaire en het tweede gedeelte het niet-lineaire gedeelte omvat.

$$\varepsilon_1 = u_2 - u_1 + \frac{3}{5}\frac{(w_2-w_1)^2}{l} + \frac{1}{15}\varphi_1^2 l + \frac{1}{15}\varphi_2^2 l - \frac{l}{30}\varphi_1\varphi_2 + \frac{1}{10}\varphi_1\frac{w_2-w_1}{l} + \frac{1}{10}\varphi_2\frac{w_2-w_1}{l}$$

$$\varepsilon_2 = -\varphi_1 - \frac{w_2-w_1}{l} + \frac{(w_2-w_1)(u_2-u_1)}{l^2}$$

$$\varepsilon_3 = \varphi_2 + \frac{w_2-w_1}{l} - \frac{(w_2-w_1)(u_2-u_1)}{l^2} \qquad (VI\ 4)$$

lineair — niet lineair

Voor de inwendige energie van het balkelement geldt:

$$U = \frac{N}{2}\varepsilon_1 + \frac{M_0}{2}\varepsilon_2 + \frac{M_l}{2}\varepsilon_3$$

$$U = U_{lineair} + U_{niet\ lineair}.$$

$U_{lineair}$ hebben we in hoofdstuk I in matrixvorm afgeleid.

$$U_{lineair} = \frac{1}{2} u^T Q u.$$

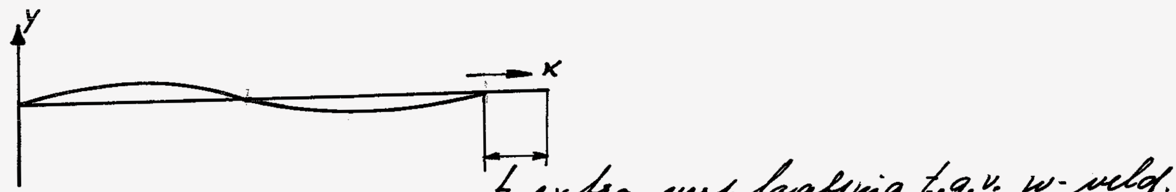
Voor de niet lineaire bijdragen kunnen we schrijven:

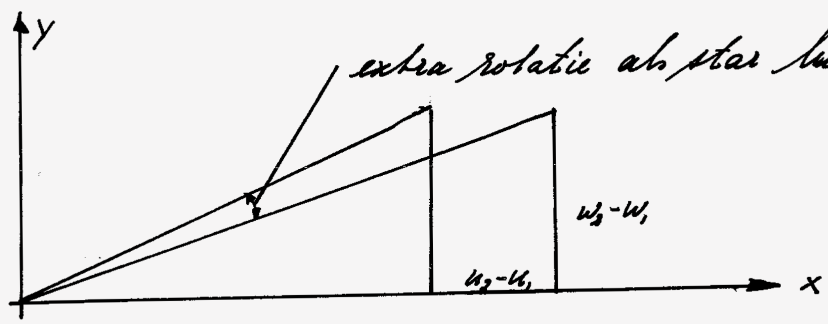
$$U_{niet\ lineair} = \frac{N}{2} u^T G_{01} u + \frac{M_0}{2} u^T G_{02} u + \frac{M_l}{2} u^T G_{03} u$$

$$G_{01} = \frac{1}{30}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ & \frac{36}{l} & -3 & 0 & -\frac{36}{l} & -3 \\ & & 4l & 0 & 3 & -l \\ & & & 0 & 0 & 0 \\ & \text{sym} & & & \frac{36}{l} & 3 \\ & & & & & 4l \end{bmatrix}$$

$$G_{02} = -G_{03} = \frac{1}{l^2} \begin{bmatrix} 0 & 1 & 0 & 0 & -1 & 0 \\ & 0 & 0 & -1 & 0 & 0 \\ & & 0 & 0 & 0 & 0 \\ & \text{sym.} & & 0 & 1 & 0 \\ & & & & 0 & 0 \\ & & & & & 0 \end{bmatrix}$$

De term $\frac{N}{2} \overset{\blacktriangledown}{u} G_{01} u$ geeft de extra arbeid aan, die de normaalkracht verricht t.g.v. relatieve knooppuntsverplaatsingen van het element, veroorzaakt door het w-veld. De term $\frac{M_0}{2} \overset{\blacktriangledown}{u} G_{02} u$ geeft de extra arbeid aan, dat het moment verricht door een extra rotatie als star lichaam t.g.v. u-veld. Voor $\frac{Ml}{2} \overset{\blacktriangledown}{u} G_{03} u$ geldt hetzelfde.





Willen we deze voorgaande afleidingen toepassen op een element, dat een hoek α met de positieve x-as maakt, dan ontstaan de volgende relaties. (α linksom positief)

$$G_1 = \frac{\partial^2 \varepsilon_1}{\partial u_q \partial u_p} = \overset{\blacktriangledown}{T} G_{01} T$$

$$G_2 = \frac{\partial^2 \varepsilon_2}{\partial u_q \partial u_p} = \overset{\blacktriangledown}{T} G_{02} T \qquad (VI\ 5)$$

$$G_3 = \frac{\partial^2 \varepsilon_3}{\partial u_q \partial u_p} = \overset{\blacktriangledown}{T} G_{03} T$$

T is de transformatiematrix en bestaat uit de hierna volgende componenten:

$$T = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 & 0 & 0 & 0 \\ -\sin\alpha & \cos\alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & 0 & 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Tenslotte rest nog de fysische interpretatie van:

$$\frac{\partial \varepsilon_i}{\partial u_p} = Q_1 + O(\varepsilon)$$

We bekijken in eerste instantie de lineaire rekrelaties voor een balkelement, waarvan de staafas samenvalt met de x-as. We vinden dan de volgende relatie:

$$\varepsilon = D u$$

$$D = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{1}{l} & -1 & 0 & -\frac{1}{l} & 0 \\ 0 & -\frac{1}{l} & 0 & 0 & \frac{1}{l} & 1 \end{bmatrix}$$

Valt de staafas niet meer met de x-as samen, maar maakt een hoek α, dan kunnen we volgende relatie opschrijven:

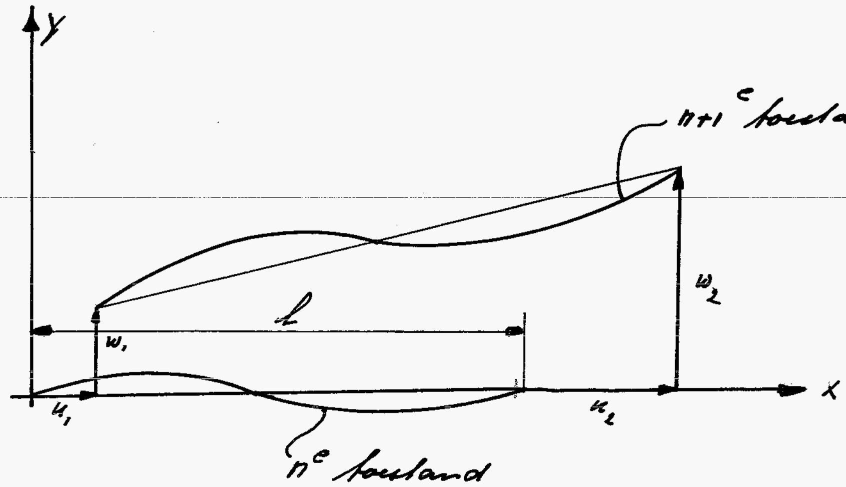
$$u_{lokaal} = T\, u_{globaal}.$$

$$\varepsilon = D\, T\, u_{globaal}$$

Het blijkt nu dat : $Q_1 = D\,T$ (VI 6)

Met andere woorden Q_1 is niets anders dan een soort transformatiematrix van globaal naar lokaal-assenstelsel. In de niet-lineaire theorie is D niet zo'n nette matrix meer. De matrix $O(\varepsilon)$ vormt nu een correctie op de Q_1 matrix van het element, omdat we de stijfheidsmatrix in vervormde toestand willen berekenen. Als we vergelijking (V 5) uitwerken, dan zien we on-

middellijk, dat de termen $\dot{O}(\varepsilon)\, 5\, Q_1$, $\dot{Q}_1\, 5\, O(\varepsilon)$ en $\dot{O}(\varepsilon)\, 5\, O(\varepsilon)$ correcties zijn op de lineaire stijfheidsmatrix.
We kunnen de $O(\varepsilon)$ als volgt bepalen:
We beschouwen een element, waarvan de verbindingslijn van de knooppunten in vervormde toestand samenvalt met de positieve x-as.



We definiëren nu de rekken als volgt:

$$\varepsilon_{1(n+1)} = \varepsilon_{1(n)} + \Delta\varepsilon_1$$

$$\varepsilon_{2(n+1)} = \varepsilon_{2(n)} + \Delta\varepsilon_2$$

$$\varepsilon_{3(n+1)} = \varepsilon_{3(n)} + \Delta\varepsilon_3$$

Hanteren we nu de rekformules zoals gedefinieerd voor geometrisch niet-lineaire problemen, dan verkrijgen we de volgende relaties:

$$\varepsilon_{1(n+1)} = \varepsilon_{1(n)} + \left[(l+u_2-u_1)^2+(w_2-w_1)^2\right]^{1/2}\left[1+\frac{1}{30}\left(2\varepsilon_{2(n+1)}^2+\varepsilon_{2(n+1)}\varepsilon_{3(n+1)}+2\varepsilon_{3(n+1)}^2\right)\right]$$

$$-\,l\left[1+\frac{1}{30}\left(2\varepsilon_{2(n)}^2+\varepsilon_{2(n)}\varepsilon_{3(n)}+2\varepsilon_{3(n)}^2\right)\right]$$

$$\varepsilon_{2(n+1)} = \varepsilon_{2(n)} - \psi_1 - \arctan\frac{w_2-w_1}{l+u_2-u_1} \qquad (\mathrm{VI}\ 7)$$

$$\varepsilon_{3(n+1)} = \varepsilon_{3(n)} + \psi_2 + \arctan\frac{w_2-w_1}{l+u_2-u_1}$$

We gaan formule (VI 7) weer in een reeks ontwikkelen:

$$\varepsilon_{1(n+1)} = \varepsilon_{1(n)} + u_2 - u_1 + \frac{l}{30}\Big[3\varepsilon_{2(n)}\frac{w_1}{l} - 3\varepsilon_{3(n)}\frac{w_1}{l} - 4\varepsilon_{2(n)} - \varepsilon_{3(n)} - 3\varepsilon_{2(n)}\frac{w_2}{l} + 3\varepsilon_{3(n)}\frac{w_2}{l} +$$

$$4\varepsilon_{3(n)} + \varepsilon_{2(n)}\Big] + \frac{3}{5}\frac{(w_2-w_1)^2}{l} + \frac{1}{15}\psi_1^2 l + \frac{1}{15}\psi_2^2 l - \frac{l}{30}\psi_1\psi_2$$

$$+ \frac{1}{10}\psi_1\left(\frac{w_2-w_1}{l}\right) + \frac{\psi_2}{10}\left(\frac{w_2-w_1}{l}\right)$$

$$\varepsilon_{2(n+1)} = \varepsilon_{2(n)} - \psi_1 - \frac{w_2-w_1}{l} + \frac{(w_2-w_1)(u_2-u_1)}{l^2} \qquad \text{(VI 8)}$$

$$\varepsilon_{3(n+1)} = \varepsilon_{3(n)} + \psi_2 + \frac{w_2-w_1}{l} - \frac{(w_2-w_1)(u_2-u_1)}{l^2}$$

Zoals uit formule (VI 8) blijkt, geeft alleen $\varepsilon_{1(n+1)}$ een bijdrage in de $O(\varepsilon)$ en wel met de volgende bijdrage:

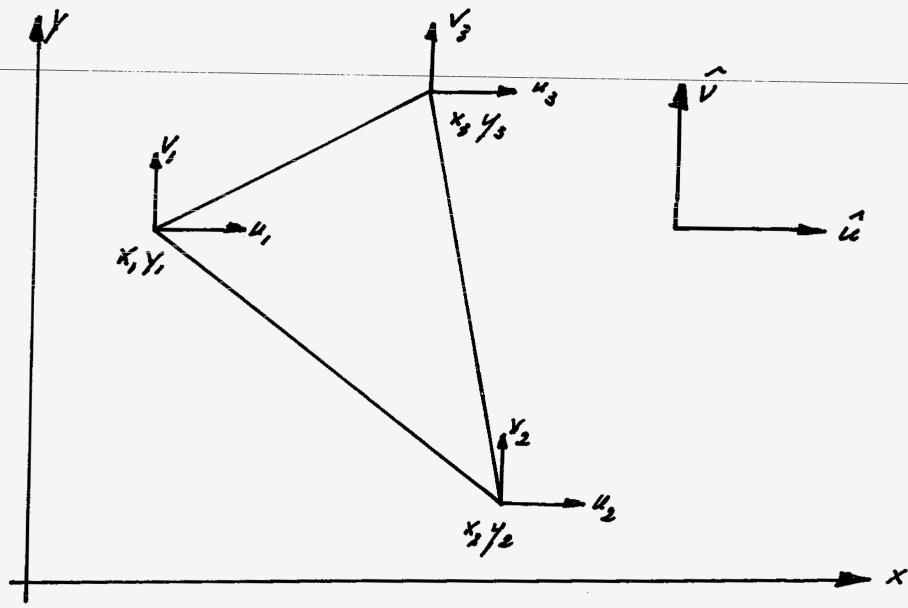
$$O_0(\varepsilon_1) = \frac{l}{30}\Big[3\varepsilon_{2(n)}\frac{w_1}{l} - 3\varepsilon_{3(n)}\frac{w_1}{l} - 4\varepsilon_{2(n)} - \varepsilon_{3(n)} - 3\varepsilon_{2(n)}\frac{w_2}{l} + 3\varepsilon_{3(n)}\frac{w_2}{l} + 4\varepsilon_{3(n)} + \varepsilon_{2(n)}\Big]$$

In matrixnotatie levert dit juist de ons reeds bekende matrix op namelijk $O_0(\varepsilon)$ voor $\beta = 0$. Staat het element onder een bepaalde hoek β met de x-as dan heeft de matrix $O(\varepsilon)$ de volgende gedaante:

$$O(\varepsilon) = O_0(\varepsilon)\, T \qquad \text{(VI 9)}$$

VII Beschrijving van het element TRIAX 3

We beschouwen een rotatorisch symmetrisch driehoekig element. We nemen aan, dat de verplaatsingen van de 3 knooppunten zodanig klein blijven, dat de klassieke formules voor rotatorisch symmetrische constructies geldig blijven.



We nemen de y-as als rotatorische symmetrie-as aan. De verplaatsingsvektor per element definiëren we als:

$$\vec{u} = [u_1\ v_1\ u_2\ v_2\ u_3\ v_3]$$

Voor de verplaatsingsvelden $\hat{u}$ en $\hat{v}$ nemen we de volgende polynomen aan:

$$\hat{u} = \alpha_1 + \alpha_2 x + \alpha_3 y$$

$$\hat{v} = \alpha_4 + \alpha_5 x + \alpha_6 y \qquad (\text{VII},1)$$

Uit deze aannames volgt direkt, dat de zijden van het element TRIAX 3 na vervorming recht blijven, met als randvoorwaarden:

$$x = x_1 \text{ en } y = y_1 \quad : \quad \hat{u} = u_1 \quad \hat{v} = v_1$$

$$x = x_2 \text{ en } y = y_2 \quad : \quad \hat{u} = u_2 \quad \hat{v} = v_2 \qquad (\overline{VII}\ 2)$$

$$x = x_3 \text{ en } y = y_3 \quad : \quad \hat{u} = u_3 \quad \hat{v} = v_3$$

Verder definiëren we de vektor α als:

$$\vec{\alpha} = [\alpha_1 \ \alpha_2 \ \alpha_3 \ \alpha_4 \ \alpha_5 \ \alpha_6]. \qquad (\overline{VII}\ 3)$$

We kunnen nu met de relaties $(\overline{VII}\ 1)$, $(\overline{VII}\ 2)$ en $(\overline{VII}\ 3)$ het verband tussen de vektoren α en u bepalen.

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{bmatrix} = \frac{1}{A} \begin{bmatrix} x_2y_3 - x_3y_2 & x_3y_1 - x_1y_3 & x_1y_3 - x_2y_1 & 0 & 0 & 0 \\ y_2 - y_3 & y_3 - y_1 & y_1 - y_2 & 0 & 0 & 0 \\ x_3 - x_2 & x_1 - x_3 & x_2 - x_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2y_3 - x_3y_2 & x_3y_1 - x_1y_3 & x_1y_2 - x_2y_1 \\ 0 & 0 & 0 & y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ 0 & 0 & 0 & x_3 - x_2 & x_1 - x_3 & x_2 - x_1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

$(\overline{VII}\ 3a)$

In deze formule is A gelijk aan:

$$A = x_2y_3 - x_3y_2 + x_3y_1 - x_1y_3 + x_1y_2 - x_2y_1$$

Uit de klassieke theorie weten we, hoe de rekken voor rotatorisch symmetrische constructies, aan het verplaatsingsveld zijn gekoppeld.

$$\varepsilon_r = \frac{\partial u}{\partial x} = \alpha_2$$

$$\varepsilon_y = \frac{\partial v}{\partial y} = \alpha_6 \qquad (\overline{VII}\ 4)$$

$$\varepsilon_t = \frac{u}{x} = \frac{u_{zw}}{x_{zw}} = \frac{\alpha_1}{x_{zw}} + \alpha_2 + \frac{y_{zw}}{x_{zw}}\,\alpha_3$$

$$\gamma_{ry} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = \alpha_3 + \alpha_5$$

We merken op, dat de tangentiële rek niet precies afgeleid is, maar benaderd is als $\frac{u_{zw}}{x_{zw}}$.

De gegeneraliseerde rekvektor heeft de volgende gedaante:

$$\vec{\varepsilon} = \left[\varepsilon_z \quad \varepsilon_y \quad \varepsilon_t \quad \gamma_{zy}\right] \qquad \text{(VII 5)}$$

Met behulp van de relaties (VII 4) en (VII 3a) zijn we in staat om het verband tussen de rekken en de verplaatsingsvektor te leggen.

$$\varepsilon = \frac{1}{A}\, C\, u \qquad \text{(VII 6)}$$

$$C = \begin{bmatrix} y_2 - y_3 & 0 & y_3 - y_1 & 0 & y_1 - y_2 & 0 \\ 0 & x_3 - x_2 & 0 & x_1 - x_3 & 0 & x_2 - x_1 \\ a_1 & 0 & a_2 & 0 & a_3 & 0 \\ x_3 - x_2 & y_2 - y_3 & x_1 - x_3 & y_3 - y_1 & x_2 - x_1 & y_1 - y_2 \end{bmatrix}$$

met: $$a_1 = \frac{x_2 y_3 - x_3 y_2}{x_{zw}} + (y_2 - y_3) + (x_3 - x_2)\frac{y_{zw}}{x_{zw}}$$

$$a_2 = \frac{x_3 y_1 - x_1 y_3}{x_{zw}} + (y_3 - y_1) + (x_1 - x_3)\frac{y_{zw}}{x_{zw}}$$

$$a_3 = \frac{x_1 y_2 - x_2 y_1}{x_{zw}} + (y_1 - y_2) + (x_2 - x_1)\frac{y_{zw}}{x_{zw}}$$

Het verband tussen de gegeneraliseerde rekvektor en de gegeneraliseerde spanningsvektor, kunnen we leggen met behulp van de wet van Hooke:

$$\sigma_z = E\,\frac{(1-\nu)\varepsilon_z + \nu\varepsilon_t + \nu\varepsilon_y}{(1+\nu)(1-2\nu)}$$

$$\sigma_y = E\,\frac{(1-\nu)\varepsilon_y + \nu\varepsilon_z + \nu\varepsilon_t}{(1+\nu)(1-2\nu)}$$

$$\sigma_t = E\,\frac{(1-\nu)\varepsilon_t + \nu\varepsilon_z + \nu\varepsilon_y}{(1+\nu)(1-2\nu)}$$

$$\tau_{ry} = \frac{E}{2(1+\nu)}\gamma_{ry}$$

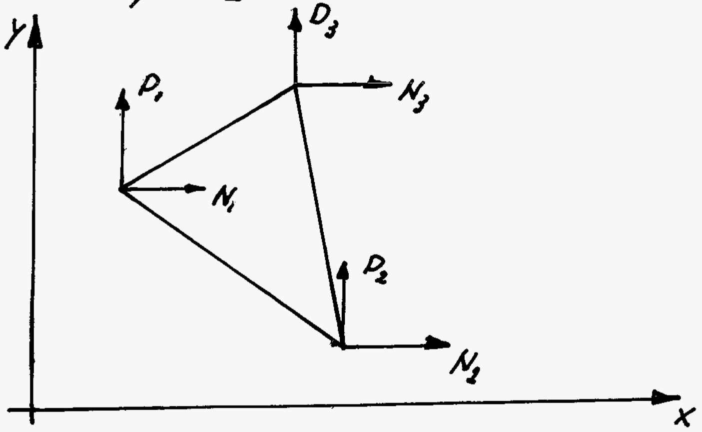
zodat we in vektornotatie de volgende vergelijking krijgen:

$$\sigma = S\,\epsilon \qquad (\text{VII } 7)$$

$$S = \frac{E}{1+\nu}\begin{bmatrix} \frac{1-\nu}{1-2\nu} & \frac{\nu}{1-2\nu} & \frac{\nu}{1-2\nu} & 0 \\ \frac{\nu}{1-2\nu} & \frac{1-\nu}{1-2\nu} & \frac{\nu}{1-2\nu} & 0 \\ \frac{\nu}{1-2\nu} & \frac{\nu}{1-2\nu} & \frac{1-\nu}{1-2\nu} & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}$$

Om de stijfheidsmatrix van het element TRIAX 3 te bepalen, gaan we uit van het principe van minimale potentiële energie. De belastingsvektor f definiëren we als:

$$\vec{f} = [N_1 \quad D_1 \quad N_2 \quad D_2 \quad N_3 \quad D_3] \qquad (\text{VII } 8)$$



We kunnen nu de potentiële energie opschrijven:

$$V = \int_V \left(\tfrac{1}{2}\sigma_r\varepsilon_r + \tfrac{1}{2}\sigma_t\varepsilon_t + \tfrac{1}{2}\sigma_y\varepsilon_y + \tfrac{1}{2}\tau_{ry}\gamma_{ry}\right) dV - \vec{f}\,\vec{u} \qquad (\text{VII } 9)$$

Substitutie in (VII 9) van de lineaire Hooke-relaties geeft:

$$V = \text{Volume} * \frac{E}{2(1+\nu)(1-2\nu)}\left\{(1-\nu)(\varepsilon_r^2+\varepsilon_t^2+\varepsilon_y^2) + 2\nu(\varepsilon_r\varepsilon_t+\varepsilon_r\varepsilon_y+\varepsilon_t\varepsilon_y) + \frac{E}{4(1+\nu)}\gamma_{ry}^2\right\} - \vec{f}\,\vec{u}$$

$$V = \frac{1}{2}\, O \cdot 2\pi\, x_{zw}\, \overset{\blacktriangledown}{\varepsilon} S \varepsilon - \overset{\blacktriangledown}{f} u$$

$$O = \frac{|A|}{2} = \text{oppervlakte TRIAX3 element}$$

$$V = \frac{1}{2} \cdot \frac{|A|}{2}\, 2\pi\, x_{zw}\, \overset{\blacktriangledown}{\varepsilon} S \varepsilon - \overset{\blacktriangledown}{f} u \qquad (\text{VII } 10)$$

Substitueren we relatie (VII 6) nu in (VII 10) dan ontstaat de volgende uitdrukking voor de potentiële energie:

$$V = \frac{1}{2}\, \frac{\pi\, x_{zw}}{|A|}\, \overset{\blacktriangledown}{u}\, \overset{\blacktriangledown}{C} S C u - \overset{\blacktriangledown}{f} u$$

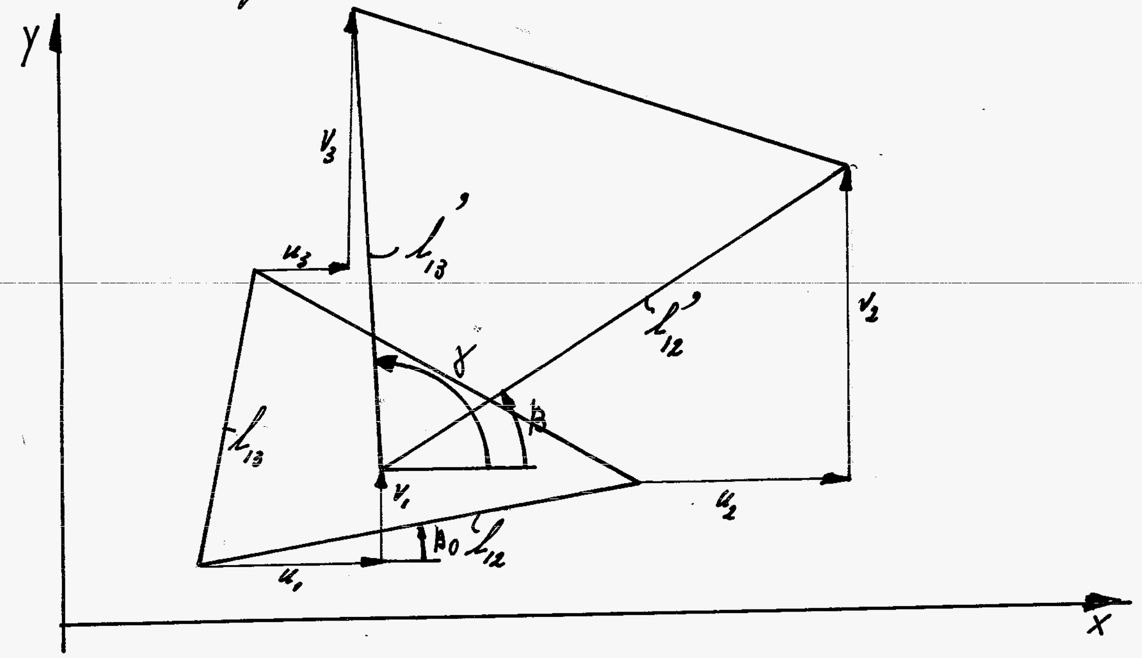
Uit het principe van minimale potentiële energie kunnen we afleiden, dat:

$$Q u = f.$$

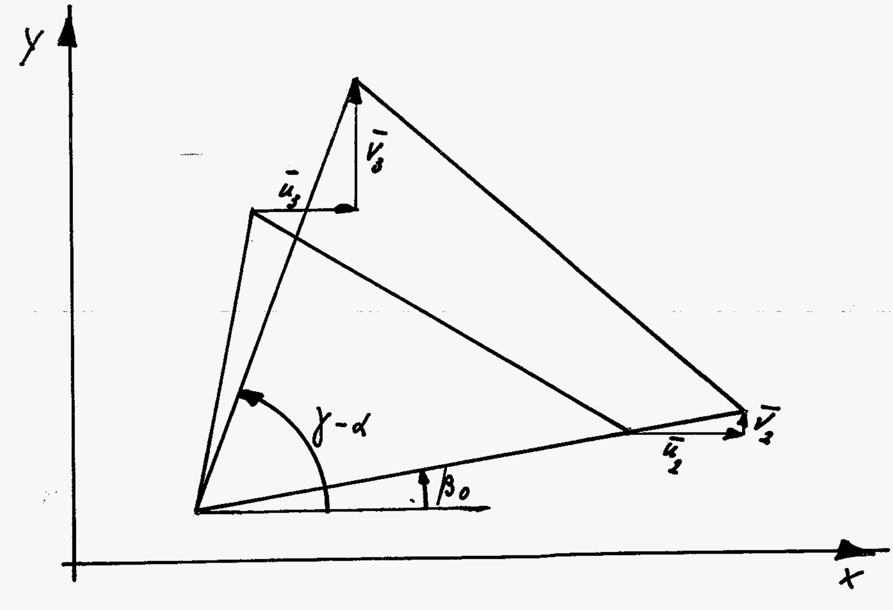
met $Q = \dfrac{\pi\, x_{zw}}{|A|}\, \overset{\blacktriangledown}{C} S C$ = stijfheidsmatrix element TRIAX 3. (VIII 11)

VIII Het element TRIAX3 bij geometrische niet-lineariteit

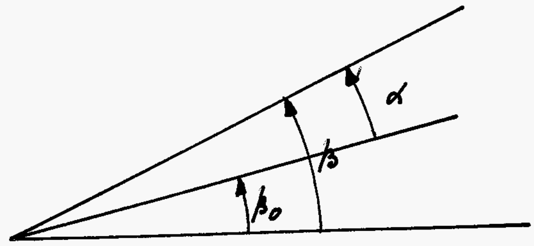
De gedachtengang verloopt geheel analoog als bij het balkelement, namelijk dat het niet-lineaire gedrag tot uitdrukking komt in de rek-verplaatsings relaties.



Om nu de vervormingen van het element te bestuderen, verplaatsen we het element naar zijn uitgangspositie, en dat doen we zodanig, dat één knooppunt samenvalt met het oorspronkelijke.



Bovendien is een bepaalde rotatie van het element noodzakelijk, om één zijde samen te laten vallen met de oorspronkelijke toestand, we kiezen hiervoor zijde l_{12}



De rotatiehoek α is nu gelijk aan:

$$\alpha = \beta - \beta_0$$

$$\alpha = \arctan\left(\frac{y_2-y_1+v_2-v_1}{x_2-x_1+u_2-u_1}\right) - \arctan\left(\frac{y_2-y_1}{x_2-x_1}\right) \qquad (\text{VIII } 1)$$

We definieren de relatieve verplaatsingsvektor $\bar{u}$ als:

$$\bar{u} = \left[\bar{u}_1 \;\; \bar{v}_1 \;\; \bar{u}_2 \;\; \bar{v}_2 \;\; \bar{u}_3 \;\; \bar{v}_3\right] \qquad (\text{VIII } 2)$$

We kunnen uit voorgaande figuren de volgende relaties voor de componenten van $\bar{u}$ afleiden:

$$\bar{u}_1 = \bar{v}_1 = 0$$

$$\bar{u}_2 = \left[\left[(x_2-x_1+u_2-u_1)^2+(y_2-y_1+v_2-v_1)^2\right]^{1/2} - \left[(x_2-x_1)^2+(y_2-y_1)^2\right]^{1/2}\right]\cos\beta_0$$

$$\bar{v}_2 = \left[\left[(x_2-x_1+u_2-u_1)^2+(y_2-y_1+v_2-v_1)^2\right]^{1/2} - \left[(x_2-x_1)^2+(y_2-y_1)^2\right]^{1/2}\right]\sin\beta_0$$

$$\begin{aligned}\bar{u}_3 = {} & \left[(x_3-x_1+u_3-u_1)^2+(y_3-y_1+v_3-v_1)^2\right]^{1/2}\cos\left\{\arctan\left(\frac{y_3-y_1+v_3-v_1}{x_3-x_1+u_3-u_1}\right)-\alpha\right\} \\ & -\left[(x_3-x_1)^2+(y_3-y_1)^2\right]^{1/2}\cos\left\{\arctan\frac{y_3-y_1}{x_3-x_1}\right\}\end{aligned} \qquad (\text{VIII } 3)$$

$$\begin{aligned}\bar{v}_3 = {} & \left[(x_3-x_1+u_3-u_1)^2+(y_3-y_1+v_3-v_1)^2\right]^{1/2}\sin\left\{\arctan\left(\frac{y_3-y_1+v_3-v_1}{x_3-x_1+u_3-u_1}\right)-\alpha\right\} \\ & -\left[(x_3-x_1)^2+(y_3-y_1)^2\right]^{1/2}\sin\left\{\arctan\left(\frac{y_3-y_1}{x_3-x_1}\right)\right\}\end{aligned}$$

We moeten hierbij echter onmiddellijk opmerken, dat de relatieve verplaatsingen $\bar{u}_2$, $\bar{v}_2$, $\bar{u}_3$ en $\bar{v}_3$ alleen gelden voor de vervormingen in het vlak van tekening. Dit houdt in, dat we voor de tangentiële rek de echte verplaatsingsvektor u moeten gebruiken.

$$\varepsilon_t = \frac{u_{zw}}{x_{zw}} = \frac{\alpha_1}{x_{zw}} + \alpha_2 + \frac{y_{zw}}{x_{zw}}\alpha_3$$

Wij zijn nu in staat om met behulp van de relatie (VII 6) de gegeneraliseerde rekvektor te berekenen:

$$\varepsilon = \frac{1}{A} C u \quad (\text{voor de } \varepsilon_t)$$

$$\varepsilon = \frac{1}{A} C \bar{u} \quad (\text{voor } \varepsilon_x, \varepsilon_y, \gamma_{xy})$$

De vektor $\bar{u}$ heeft de volgende gedaante,

$$\bar{u} = [0 \quad 0 \quad \bar{u}_2 \quad \bar{v}_2 \quad \bar{u}_3 \quad \bar{v}_3]$$

zodat we voor de gegeneraliseerde rekgrootheden de volgende niet-lineaire relaties vinden in de knooppuntsverplaatsingen.

$$\varepsilon_x = \frac{1}{A}\left[(y_3 - y_1)\bar{u}_2 + (y_1 - y_2)\bar{u}_3\right]$$

$$\varepsilon_y = \frac{1}{A}\left[(x_1 - x_3)\bar{v}_2 + (x_2 - x_1)\bar{v}_3\right]$$

$$\begin{aligned}\varepsilon_t = {} & \frac{3}{A(x_1 + x_2 + x_3)}\left[(x_2 y_3 - x_3 y_2)u_1 + (x_3 y_1 - x_1 y_3)u_2 + (x_1 y_2 - x_2 y_1)u_3\right] + \\ & + \frac{1}{A}\left[(y_2 - y_3)u_1 + (y_3 - y_1)u_2 + (y_1 - y_2)u_3\right] + \\ & + \frac{1}{A}\left[\frac{y_1 + y_2 + y_3}{x_1 + x_2 + x_3}\left[(x_3 - x_2)u_1 + (x_1 - x_3)u_2 + (x_2 - x_1)u_3\right]\right]\end{aligned} \qquad (\text{VIII } 4)$$

$$\gamma_{xy} = \frac{1}{A}\left[(x_1 - x_3)\bar{u}_2 + (x_2 - x_1)\bar{u}_3 + (y_3 - y_1)\bar{v}_2 + (y_1 - y_2)\bar{v}_3\right]$$

Bestuderen we nu weer de oorspronkelijke vergelijking voor geometrisch niet-lineaire problemen (IV 4):

$$\left\{\sum_{k=1}^{N} \frac{\partial \varepsilon_i}{\partial u_q} S_{ij} \frac{\partial \varepsilon_j}{\partial u_p} + \sigma_j \frac{\partial^2 \varepsilon_j}{\partial u_q \partial u_p}\right\}_{u=u_n^t} \Delta u_k = f_k - \left\{\sum_{k=1}^{N} \varepsilon_i S_{ij} \frac{\partial \varepsilon_j}{\partial u_p}\right\}_{u=u_n^t}$$

S is de matrix, die het lineaire verband tussen de rekken en spanningen legt, zie formule (VII 7).
In appendix no. V zijn de afleidingen gegeven van de benodigde matrices. We volstaan hier met de vermelding dat geldt:

$$\frac{\partial \varepsilon_i}{\partial u_q} = Q_1$$

$$\frac{\partial^2 \varepsilon_2}{\partial u_q \partial u_p} = G_1 \qquad \frac{\partial^2 \varepsilon_y}{\partial u_q \partial u_p} = G_2 \qquad \text{(VIII 5)}$$

$$\frac{\partial^2 \varepsilon_t}{\partial u_q \partial u_p} = [0] \qquad \frac{\partial^2 \gamma_{zy}}{\partial u_q \partial u_p} = G_4$$

Substitueren wij nu (VIII 5) in de niet-lineaire vergelijking (IV 4), dan ontstaat tenslotte de vergelijking, die het geometrisch niet-lineaire gedrag beschrijft van rotatorisch symmetrische constructies in matrixnotatie:

$$\left\{\sum_{k=1}^{N} Q_1^{\mathrm{T}} S Q_1 + \sigma_z G_1 + \sigma_y G_2 + \tau_{zy} G_4\right\}_{u=u_n^t} \Delta u = \Delta f + f_n - \left\{\sigma^{\mathrm{T}} [Q_1]\right\}_{u=u_n^t}$$

f_n = uitwendige belastingsvektor na n stappen.

u_n^t = verplaatsingsvektor na n stappen en t iteraties.

Δu en Δf zijn de incrementen van de respectievelijke verplaatsings- en belastingsvektor van de gehele constructie.

IV <u>Beschrijving van het programma 05064608</u>

„ Berekening veerkarakteristiek met behulp van de methode der eindige elementen, waarbij de stapgrootte in de belasting afhankelijk is van het aantal iteratiestappen"

Allereerst wil ik beginnen met de beschrijving van de gebruikte procedures, die in het programma gebruikt worden. Hierna zal ik een beschrijving van het hoofdprogramma geven aan de hand van het gemaakte schema.

<u>procedure</u> STYFEL (E, I, A, l, S).

In deze procedure wordt de matrix S, die het verband tussen spanningen en rekken legt, gevuld.

<u>real.</u> E = elasticiteitsmodulus.

I = oppervlaktetraagheidsmoment van de dwarsdoorsnede

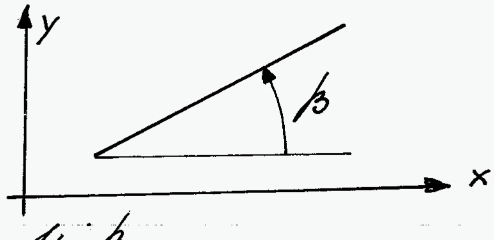
A = oppervlakte van de dwarsdoorsnede.

l = lengte van het element, oorspronkelijk.

<u>array</u> S[1:3, 1:3] = voornoemde matrix.

<u>procedure</u> HULP (a, b, c, d, e, f, LEN, st, en, le, KN).

In deze procedure worden een aantal hulpgrootheden berekend van een bepaald balkelement.



<u>real</u> a = $\sin\beta$

b = $\cos\beta$

c = $\sin^2\beta$

d = $\cos^2\beta$

e = $\sin 2\beta$

f = $\cos 2\beta$

<u>integer</u> st = 1 : bij de eerste stap in de belastingsvektor

= 2 : bij alle volgende stappen.

en = teller voor het elementnummer.

<u>array</u> LEN[1:2, 1:Ne]: De eerste kolom is gevuld met de oorspronkelijke lengtes van het balkelement, terwijl de tweede kolom gevuld is met de nieuwe afstand tussen de knooppunten van het element.

le[1:2, 1:Ne] = de le-vektor

KN[1:2, 1:N] = array waarin de nieuwe knooppunts-coördinaten opgeborgen zijn.

<u>procedure</u> BELASTINGSVEKTOR(T, lnp, f).

In deze procedure wordt de totale uitwendige belastingsvektor f gevuld met de invoergegevens.

<u>integer</u> T = totaal aantal vrijheidsgraden van de constructie.

<u>array</u> lnp[1:3, 1:N] = lnp-vektor

f[1:T] = belastingsvektor.

<u>procedure</u> ONDERSTEUNING(N, T, lnp).

In deze procedure wordt de lnp-vektor gevuld, en het aantal vrijheidsgraden T van de constructie bepaald.

<u>integer</u> N = aantal knooppunten

T = aantal vrijheidsgraden

<u>array</u> lnp[1:3, 1:N] = lnp-vektor van de constructie.

<u>procedure</u> INLE(Ne, le).

In deze procedure wordt de le-vektor gevuld.

<u>integer</u> Ne = aantal elementen

<u>array</u> le[1:2, 1:Ne] = de le-vektor.

<u>procedure</u> GOH(l, sigma1, sigma2, sigma3, G1, G2, G3, G, a, b, c, d, e, f).

In deze procedure worden de matrices $N * \frac{\partial^2 \varepsilon_1}{\partial u_q \partial u_p}$,

$M_0 * \frac{\partial^2 \varepsilon_2}{\partial u_q \partial u_p}$ en $M_l * \frac{\partial^2 \varepsilon_3}{\partial u_q \partial u_p}$ gevuld. Hierna worden de de overeenkomstige termen in deze 3 matrices opgeteld en in de totale geometrische matrix van het element geplaatst.

__real__ l = nieuwe lengte van het element, de verbindingslijn tussen de knooppunten.
sigma1 = normaalkracht N
sigma2 = moment M_0
sigma3 = moment M_l
a, b, c, d, e, f zijn de hulpgrootheden uit de procedure HULP, te weten $\sin\beta$, $\cos\beta$, $\sin^2\beta$, $\cos^2\beta$, $\sin 2\beta$, $\cos 2\beta$

__array__ $G1[1:6, 1:6] = N * \frac{\partial^2 \varepsilon_1}{\partial u_q \partial u_p}$.

$G2[1:6, 1:6] = M_0 * \frac{\partial^2 \varepsilon_2}{\partial u_q \partial u_p}$

$G3[1:6, 1:6] = M_l * \frac{\partial^2 \varepsilon_3}{\partial u_q \partial u_p}$

$G[1:6, 1:6]$ = som van $G1 + G2 + G3$

__procedure__ ORDEEPS (a, b, l, eps1, eps2, eps3, Oeps)
In deze procedure wordt de matrix $O(\varepsilon)$ gevuld.
__real__ $a = \sin\beta$
$b = \cos\beta$
l = nieuwe lengte van het element.
eps1 = ε_1
eps2 = ε_2
eps3 = ε_3
__array__ Oeps$[1:6, 1:6] = O(\varepsilon)$

__procedure__ DEPSDU (a, b, l, Q1, Q1T)
In deze procedure worden matrices Q1 en de getransformeerde van Q1 bepaald.
__real__ $a = \sin\beta$
$b = \cos\beta$
l = nieuwe lengte van het element
__array__ Q1[1:3, 1:6] = de matrix Q1
Q1T[1:6, 1:3] = de getransformeerde van Q1

__procedure__ VERMENIGVULDIGING (A, B, C, D, DT).
In deze procedure wordt de vermenigvuldiging van 3 matrices uitgevoerd, te weten $D = ABC$. Bovendien

wordt de getransformeerde van de matrix D bepaald.
<u>array</u> A[1:6, 1:3] - B[1:3, 1:3] - C[1:3, 1:6] - D[1:6, 1:6] - DT[1:6, 1:6].

<u>procedure</u> OPTELLING (A, B, C, D, Qtot).
In deze procedure kunnen we 4 matrices optellen en het resultaat op de overeenkomstige plaats in Qtot opbergen.
<u>array</u> A, B, C, D, Qtot[1:6, 1:6].

<u>procedure</u> QKONSTRUKTIE (en, Qtot, le, lup, Qkon)
In deze procedure wordt de totale stijfheidsmatrix van het element op de gebruikelijke manier in de stijfheidsmatrix van de gehele constructie opgeborgen.
<u>integer</u> en = teller voor het elementnummer.
<u>array</u> Qtot[1:6, 1:6] = totale stijfheidsmatrix van het element
le[1:2, 1:Ne] = le-vektor
lup[1:3, 1:N] = lup-vektor.
Qkon[1:T, 0:bb] = stijfheidsmatrix voor de gehele constructie in de zogenaamde bandstructuur.

<u>procedure</u> CHOLBD (n, m, dec, a, b, fail).
In deze procedure lossen we het interessante stelsel Qkon u = f op. De procedure wordt in appendix n° III beschreven.

<u>procedure</u> UITVOERFFTOT (N, fftot, lup).
In deze procedure voeren we de belastingsvektor uit en wel op de volgende manier:
„knooppunt = i a b c"
a = belastingscomponent in knooppunt i in x-richting
b = belastingscomponent in knooppunt i in y-richting
c = het moment in knooppunt i.
<u>integer</u> N = aantal knooppunten.
<u>array</u> fftot[1:T] = totale uitwendige belastingsvektor.
lup[1:3, 1:N] = lup-vektor.

<u>procedure</u> UITVOERKNSIR (KN, SI, R, Ne, k, st, N).
In deze procedure worden de knooppuntscoördinaten de spanningen en de rekken van de elementen na elke stap in de uitwendige belasting uitgevoerd.

<u>integer</u> Ne = aantal elementen
k = teller in procedure UITVOER
st = teller voor het aantal uitgevoerde stappen.
N = aantal knooppunten.

<u>array</u> KN[1:2, 1:N] = knooppuntscoördinaten.
SI[1:3, 1:Ne] = elke rij vertegenwoordigt de spanningsvektor voor een bepaald element.
R[1:3, 1:Ne] = elke rij vertegenwoordigt hier de rekvektor voor een bepaald element.

<u>procedure</u> UITGANGSHOEK(en, KN, le, lup, HOEK).
In deze procedure wordt het array HOEK gevuld met de hoeken welke de elementen met de x-as maken in uitgangspositie.

<u>integer</u> en = teller voor het elementnummer.

<u>array</u> KN[1:2, 1:N] = knooppuntscoördinaten.
le[1:2, 1:Ne] = le-vektor
lup[1:3, 1:N] = lup-vektor.
HOEK[1:Ne] = HOEK[i] is de hoek welke element i met de x-as in uitgangspositie maakt.

<u>procedure</u> REKSPANITTERATIE (en, ue, R, KN, KNH, LEN, le, lup, u, SI, S, E, I, A, PSI, PSIH, HOEK).
In deze procedure worden na elke stap of itteratie de knooppuntscoördinaten aangepast, de verbindingslijn tussen de knooppunten of de nieuwe lengte van het element berekend, de spanningen in het element berekend en de hoek bepaald welke het element met de positieve x-as maakt. De reden waarom deze hoek telkens berekend en aangepast moet worden, wordt in appendix nō IV behandeld.

<u>integer</u> en = teller elementnummer
<u>real</u> E = elasticiteitsmodulus
I = oppervlaktetraagheid van de dwarsdoorsnede
A = oppervlakte dwarsdoorsnede
<u>array</u> ue[1:6] = verplaatsingsvektor per element
R[1:3, 1:Ne] = rekken van de elementen.
KN[1:2, 1:N] = knooppuntscoördinaten.
KNH[1:2, 1:N] = hulparray voor de knooppuntscoördinaten.
LEN[1:2, 1:Ne] = lengtes van de elementen: $l_{oud} - l_{nieuw}$.
le[1:2, 1:Ne] = le-vektor
lnp[1:3, 1:N] = lnp-vektor
u[1:T] = verplaatsingsvektor van de konstruktie.
SI[1:3, 1:Ne] = spanningen in de elementen.
S[1:3, 1:3] = matrix voor de relatie tussen rekken en spanningen.
PSI[1:2, 1:Ne] = absolute hoekverdraaiing t.o.v. uitgangspositie van knooppunt 1 en 2 van element i.
PSIH[1:Ne] = hoeken, die de verbindingslijn tussen de knooppunten van element i maakt met de positieve x-as.
HOEK[1:Ne] = hoeken die de elementen in uitgangspositie met de x-as maken.

<u>procedure</u> UITVOER (Ne, k, M).
In deze procedure wordt de matrix M op een speciale overzichtelijke manier uitgevoerd.
<u>integer</u> Ne = aantal rijen van de matrix.
k = aantal kolommen van de matrix
<u>array</u> M[1:Ne, 1:k].

<u>procedure</u> VERMVEKTOR (SI, DEDU, vektor, en)
In deze procedure wordt een vektor met een matrix vermenigvuldigd en opgeborgen in „vektor"
<u>integer</u> en = teller voor elementnummer
<u>array</u> SI[1:3, 1:Ne] = spanningen in de elementen
DEDU[1:3, 1:6] = $Q_1 + O(\epsilon)$
vektor[1:6].

procedure RECHTERLID (vektor, rechter, en, le, lnp)
In deze procedure wordt de uitwendige belastings-vektor berekend, door de krachtvektoren, getransformeerd naar het globaal assenstelsel, per element over de gehele constructie bij elkaar op te tellen.
Het resultaat wordt opgeborgen in de vektor rechter.
integer en = teller voor elementnummer
array vektor[1:6] = krachtvektor per element getransformeerd naar het globaal assenstelsel.
rechter[1:T] = wordt de benaderde uitwendige belastings-vektor in opgeborgen.
le[1:2, 1:Ne] = le-vektor
lnp[1:3, 1:N] = lnp-vektor.

procedure BANDBREEDTE (ae, ake, avk, lnp, le2, bb).
In deze procedure wordt de bandbreedte van de stijfheidsmatrix berekend.
integer ae = aantal elementen
ake = aantal knooppunten per element
avk = aantal vrijheidsgraden per knooppunt
bb = bandbreedte
array lnp[1:3, 1:N] = lnp-vektor
le2[1:2, 1:Ne] = le-vektor.

Zoals in het begin van dit hoofdstuk gesteld is, volgt nu een korte beschrijving van het hoofdprogramma (05064608) aan de hand van het gemaakte blokschema:

Allereerst worden de nodige invoergegevens ingelezen, waarna met de procedure UITGANGSHOEK de hoeken van de elementen met de positieve x-as in uitgangspositie worden vastgelegd. De procedure BANDBREEDTE bepaalt vervolgens de te verwachten bandbreedte van de totale stijfheidsmatrix. Nadat de totale uitwendige belastingsvektor is ingelezen, wordt de belastingsstap bepaald door de componenten van de totale belastingsvektor te delen door het aantal geschatte stappen in de belasting. Hierna wordt met behulp van de procedures HULP, STYFEL, GOM, ORDEEPS, DEPSDU, VERMENIGVULDIGING, OPTELLING en QKONSTRUKTIE de totale stijfheidsmatrix van de constructie bepaald. Met de procedures VERMVEKTOR en RECHTERLID wordt de belastingsvektor berekend door alle snedegrootheden in de knooppunten bij elkaar op te tellen en op de juiste wijze in de vektor rechter op te bergen. De vektor rechterlid wordt nu opgevuld met het verschil van de werkelijke belastingsvektor en de zojuist gevulde vektor rechter. Het zal duidelijk zijn, dat voor de exakte oplossing dit verschil gelijk aan nul zal moeten zijn. Dit zal echter meestal niet het geval zijn. Daarom hebben we een aantal criteria ingelezen waaraan de vektor rechterlid zal moeten voldoen om tot uitvoering van de volgende stap over te gaan. Deze criteria zijn als volgt vastgelegd: een procentuele afwijking van „procent" % van de werkelijke belastingsvektor, indien de component ongelijk aan nul is en een absolute afwijking ter grootte van „absol" indien de component gelijk aan nul is. We hebben nu duidelijk 2 mogelijkheden:

(a) Aan de gestelde criteria wordt voldaan.
Eerst wordt nu gekeken of de werkelijke belastingsvektor groter of gelijk is aan de totale belastingsvektor, zo ja, dan wordt het programma beëindigd, zo neen, dan wordt de volgende belastingsstap uitgevoerd, nadat ook eerst weer gekeken is of we de belastingsstap moeten vergroten of verkleinen, wat afhankelijk gesteld is van het aantal iteratiestappen uit de voorgaande stap. Vervolgens wordt de werkelijke belastingsvektor en de vektor rechterlid gecorrigeerd met deze nieuwe belastingsstap. Tenslotte worden alle gegevens van de voorgaande stap uitgevoerd: te weten, knooppuntscoördinaten, rekken en spanningen.

(b) Aan de gestelde criteria wordt niet voldaan.
Nu wordt eerst gecontroleerd of het aantal reeds uitgevoerde iteratiestappen beneden een bepaald in te lezen getal ligt. Is dit het geval dan wordt wordt de volgende iteratiestap uitgevoerd. Is dit echter niet het geval, dan wordt opnieuw met het iteratieproces gestart, en wel met halve stapgrootte in de belasting. De vektoren rechterlid en werkelijke belastingsvektor moeten uiteraard gecorrigeerd worden voordat met het iteratieproces gestart kan worden.

Nu wordt het stelsel $Q\Delta u$ = rechterlid opgelost met behulp van de procedure CHOLBD. Moet de matrix Q niet positief definiet zijn, dan wordt opnieuw gestart bij het einde van de voorgaande stap, maar dan met halve stapgrootte in de belasting. De vektor rechterlid en werkelijke belastingsvektor moeten weer aangepast worden. Het niet positief definiet zijn van de stijfheidsmatrix Q kan maximaal 5× achter elkaar gebeuren, omdat bij een aantal groter dan 5 het programma beëindigd wordt (het getal 5 kan vrij eenvoudig in het programma gewijzigd worden). Is de stijfheidsmatrix Q wel positief definiet, dan wordt het stelsel met CHOLBD opgelost en met behulp van de procedure REKSPANITTERATIE de knooppuntskoördinaten aangepast, spanningen en rekken berekend etc. Hierna wordt weer opnieuw met de opbouw van de stijfheidsmatrix en rechterlid begonnen, waarna het hiervoor beschreven proces weer helemaal herhaald wordt.

X Lineaire Stabiliteit.

In hoofdstuk IV is de volgende relatie voor geometrisch niet-lineaire problemen afgeleid:

$$\left\{\sum_{k=1}^{N} \frac{\partial \varepsilon_i}{\partial u_q} S_{ij} \frac{\partial \varepsilon_j}{\partial u_p} + \sigma_j \frac{\partial^2 \varepsilon_j}{\partial u_q \partial u_p}\right\}_{u=u_k^t} = f_k - \left\{\sum_{k=1}^{N} \varepsilon_i S_{ij} \frac{\partial \varepsilon_j}{\partial u_p}\right\}_{u=u_k^t} \quad (X1)$$

Bij lineaire stabiliteit of knik, wordt aangenomen, dat de lineaire relaties tussen de verplaatsingen en rekken gehandhaafd blijven, zodat we de $O(\varepsilon)$ bijdrage rustig kunnen verwaarlozen.

$$\frac{\partial \varepsilon_i}{\partial u_p} = Q_1 \quad (X2)$$

Substitutie van (X 2) in (X1) geeft nu de volgende vergelijking in matrix-notatie (zie formules (VIII 5)).

$$\left\{\sum_{k=1}^{N} Q_1^{\mathsf{T}} S Q_1 + N * G_1 + M_0 G_2 + M_l G_3\right\} \Delta u_k = f_k - \left\{\sum_{k=1}^{N} \sigma^{\mathsf{T}} Q_1\right\} \quad (X3)$$

Het stapsgewijze proces van (X3) is niet meer nodig, vanwege het lineair blijven, zodat vergelijking (X 3) zich reduceert tot

$$\left\{\sum_{k=1}^{N} Q_1^{\mathsf{T}} S Q_1 + N G_1 + M_0 G_2 + M_l G_3\right\} u_k = f_k \quad (X4)$$

De constructie is dan en slechts dan instabiel, zodra de relatie tussen belasting en verplaatsing niet meer gedefinieerd is. Dit betekent, dat het stelsel (X4) afhankelijk wordt, wat we als volgt kunnen formuleren:

$$\mathrm{Det}\left\{Q_1^{\mathsf{T}} S Q_1 + N G_1 + M_0 G_2 + M_l G_3\right\} = 0 \quad (X5)$$

De waarden van de spanningsvektoren, want deze zijn immers nog de enige variabelen in (X 5), waarbij (X 5) afhankelijk wordt, zal de laagste eigenwaarde van het stelsel zijn. De hierbij behorende spanningsvektoren kunnen we berekenen, zodat we ook de uitwendige belastingsvektor

kunnen bepalen, waarbij de constructie juist instabiel is.
Deze hiervoor omschreven gedachtengang ligt aan het computerprogramma 0506435 5,, berekening van de knikkracht van een vakwerk, met behulp van de elementenmethode, waarin tevens de verplaatsingen en de krachten in de knooppunten bepaald kunnen worden." aan ten grondslag. We zullen hier een korte beschrijving van het programma geven aan de hand van het bijgaande blokschema:
We beginnen met het inlezen van de nodige invoergegevens waarbij we met behulp van de procedure PROFIELKEUZE in staat zijn om in principe een constructie met 5 soorten balkelementen te berekenen. Vervolgens wordt met de procedures HULP, STYFEL, DEPSDU, VERMENIGVULDIGING en QKONSTRUKTIE de totale stijfheidsmatrix van de constructie opgebouwd. Met de bibliotheekprocedures CHOLESKIDECOMPOSITION en CHOLESKISOLUTION wordt het lineaire stelsel

$$Qu = f \qquad (X6)$$

opgelost. Hierna wordt voor elk element de verplaatsingsvektor t.o.v. het globale assenstelsel getransformeerd naar het lokale assenstelsel, zodat we met behulp van formule (I1) de lokale spanningen in het element kunnen berekenen. Deze spanningen zijn nu ook juist de grootheden, die we in de procedure GOM nodig hebben om de geometrische matrices te vullen, zodat we de totale Geometrische matrix van de constructie kunnen opbouwen. Met behulp van de bibliotheekprocedure GENERAL EIGEN VALUE PROBLEM lossen we nu het volgende stelsel op:

$$\det\{Q - G\} = 0 \qquad (X7)$$

Deze procedure berekent de eigenwaarden en eigenvektoren van de constructie. De kleinste eigenwaarde nu is een maat voor de knikbelasting, het is namelijk een vermenigvuldigingsfactor, zodat we voor de knikbelasting het volgende kunnen schrijven:

$$F_{knik} = F * PK$$

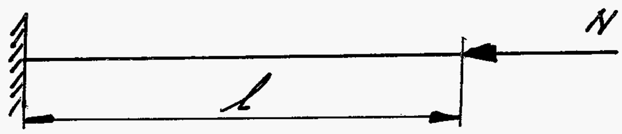
F_{knik} = knikbelasting

F = uitwendige belastingsvektor

PK = laagste eigenwaarde, de vermenigvuldigingsfactor

De bij de laagste eigenwaarde horende eigenvektor geeft tenslotte een beeld hoe de stand van de konstruktie na uitknikken is.

We zullen nu het volgende voorbeeld doorrekenen:



$E = 1 \; N/m^2$

$l = 1 \; m$

$I = 1 \; m^4$

We vinden het volgende stelsel:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 12 & 6 \\ 0 & 6 & 4 \end{bmatrix} \begin{bmatrix} u_2 \\ w_2 \\ \varphi_2 \end{bmatrix} = \begin{bmatrix} -N \\ 0 \\ 0 \end{bmatrix}$$

$u_2 = -N$

$w_2 = 0$

$\varphi_2 = 0$

$\varepsilon_1 = -N$

$\varepsilon_2 = 0$

$\varepsilon_3 = 0$

$$\vec{\sigma} = [-N \;\; 0 \;\; 0]$$

Vergelijking (X7) krijgt nu de volgende gedaante:

$$\mathrm{Det}\left[\begin{bmatrix}1 & 0 & 0\\ 0 & 12 & 6\\ 0 & 6 & 4\end{bmatrix} + \frac{N}{30}\begin{bmatrix}0 & 0 & 0\\ 0 & 36 & 3\\ 0 & 3 & 4\end{bmatrix}\right] = 0$$

Dit kunnen we herleiden tot:

$$\mathrm{Det}\left[\frac{180}{N}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix} + \begin{bmatrix}0 & 0 & 0\\ 0 & 63 & -90\\ 0 & -6 & 15\end{bmatrix}\right] = 0$$

Stellen we nu $\lambda = \frac{180}{N}$ dan moeten we het volgende eigenwaardenprobleem oplossen:

$$\mathrm{Det}\begin{bmatrix}-\lambda & 0 & 0\\ 0 & 63-\lambda & -90\\ 0 & -6 & 15-\lambda\end{bmatrix} = 0$$

De karakteristieke vergelijking is nu:

$-\lambda\{(63-\lambda)(15-\lambda) - 540\} = 0$ met als eigenwaarden

$$\lambda_1 = 0$$
$$\lambda_2 = 72 \qquad \text{of} \quad \frac{180}{N} = 72$$
$$\lambda_3 = 6$$

De kniklasting is nu: $P_{kN} = \frac{180}{72} = 2,5\ N.$

oftewel voor willekeurige E, I en l:

$$P_{kN} = 2,5\ \frac{EI}{l^2}\ N$$

De exakte oplossing is:

$$P_{euler} = 2{,}485 \; \frac{EI}{l^2} \; N.$$

In litteratuur (III) heb ik de stabiliteit behandeld aan het effect van de normaalkracht. Tijdens de bestudering van de geometrische niet-lineariteit bleek echter dat de geometrische matrices G_2 en G_3 (zie hoofdstuk II) ook nog een rol spelen in het eigenwaardeprobleem. In litteratuur (IV) wordt echter duidelijk gesteld, dat het moment geen invloed heeft op de stabiliteitswaarde. Daarom is het programma 05064355 geschreven om eventueel verschillen in de knikbelasting aan te tonen. De conclusie moet zijn, dat er, althans voor de berekende voorbeelden, geen enkel verschil bestaat voor de knikbelasting. Voor hogere eigenwaardes trad een duidelijk verschil op.

VI Resultaten en opmerkingen

Met het in hoofdstuk IV beschreven computerprogramma 05064608 hebben we een aantal voorbeelden doorgerekend, waarvan de gegevens verzameld zijn in appendix I. Op de eerste plaats is de eenzijdig ingeklemde bladveer belast met een uitwendig buigend moment berekend om het programma te testen. We vonden met verschillende aantallen elementen een zeer goede benadering van de exakte kromme (grafiek II). Grafiek III geeft de vervormde toestand van de bladveer weer bij $M = 46{,}752$ Nm en bij $M = 42$ Nm.
Grafiek IV geeft de veerkarakteristiek van een bladveer belast met een dwarskracht. Verificatie vindt van deze resultaten in grafiek VI plaats. In deze grafiek is de getrokken lijn berekend op de klassieke methode te weten met de elleptische integralen, zie litteratuur (I + II). In litteratuur (I) wordt bovendien de beschrijving van een ansa-element gegeven voor grote verplaatsingen. De streep-stip lijn in grafiek VI is de berekende kromme met het in dit verslag beschreven geometrisch niet lineaire balkelement. De conclusie moet zijn, dat het resultaat uitstekend klopt met de klassieke oplossing. Grafiek V geeft een indruk van de vervormde toestand van een bladveer onder een dwarskracht van 150 N.
Grafiek VII geeft de veerkarakteristiek t.g.v. een dwarskracht en buigend moment van een pendel.
Grafiek VIII geeft weer een indruk van de vervormde toestand bij een moment van 1,45 Nm en een dwarskracht van 2,9 N.
Aan de hand van grafiek VII kunnen we duidelijk een beperking van het programma demonstreren. In het programma kunnen we alleen maar een positieve respectievelijk negatieve belastingsstap toedienen. Dit wil zeggen, zodra we een maximum respectievelijk minimum van de veerkarakteristiek naderen, de

relatie tussen belasting en verplaatsing steeds slechter gedefinieerd zal zijn, oftewel de totale stijfheidsmatrix zal wel zo best meer geconditioneerd zijn. Dit betekent dat in het programma steeds meer iteratiestappen plaatsvinden bij steeds kleiner wordende belastingsstappen. De vraag rijst nu, hoe kunnen we nu over een dergelijk maximum respectievelijk minimum heenkomen. Het vermoeden bestond dat dit vrij gemakkelijk zou kunnen met een voorgeschreven verplaatsingsvektor in plaats van een belastingsvektor. We hebben het programma aangepast, maar het bleek op grote moeilijkheden te stuiten en wel de volgende:
De verplaatsingsvektor van de voorgaande stap wordt gebruikt voor de volgende stap. Het stelsel $Q.\Delta u = \Delta f$ wordt uitgerekend. We vinden nu een Δf. Deze Δf wordt nu gebruikt voor het formeren van de nieuwe uitwendige belastingsvektor. Het criterium voor het al of niet itereren hebben we ongewijzigd gelaten, zodat berekend werd voor eventuele iteratiestappen het verschil

$$\dot{\varepsilon}\, 5\, \frac{\partial \varepsilon}{\partial u} - f\ .$$

Bij de berekeningen bleek, dat het iteratieproces bleef divergeren ook nog na een aantal malen halveren van de voorgeschreven verplaatsingsvektor. Kennelijk is de voorgeschreven verplaatsingsvektor Δu zo'n slechte schatting voor de werkelijke verplaatsingsvektor, dat de bij deze Δu behorende Δf zo slecht is (vaak veel te grote stappengrootte voor Δf) dat bij het itereren het proces divergeert. We zullen de voorgeschreven verplaatsingsvektor daarom preciezer moeten berekenen, bijv. door eerst k maal een belastingsstap te doen, de bijbehorende verplaatsingsvektoren onthouden, om na deze k stappen door numerieke extrapolatie een betere schatting voor

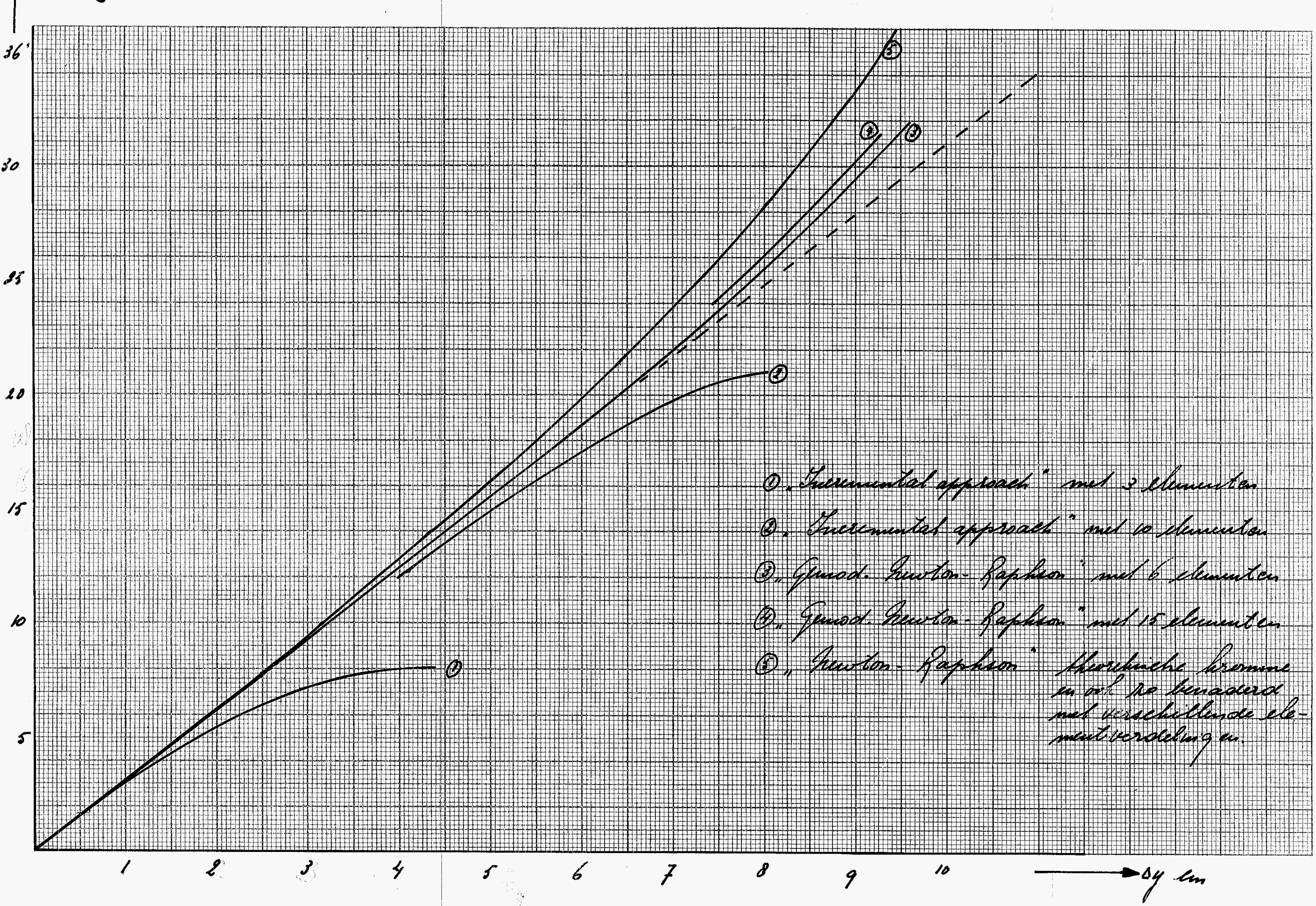
de werkelijke su te krijgen.
Grafiek IV geeft de veerkarakteristiek van een gekromde bladveer, terwijl grafiek XII een indruk geeft van de vervormde toestand. Aan de hand van dit voorbeeld zullen we een merkwaardig verschijnsel bespreken, wat ook bij de andere voorbeelden is geconstateerd. In grafiek IV hebben we de gekromde bladveer in 16 elementen verdeeld van ongeveer gelijke lengte. We constateren dat de veerkarakteristiek na 9 belastingsstappen op een grootte van 82 N is gekomen, het oppervlakte-traagheidsmoment is .5 10^{-11} m^4. We laten het programma opnieuw rekenen met meer rekentijd en constateren dat na het bereiken van deze 82 N de stapgrootte heel snel afneemt en het aantal iteratiestappen sterk toeneemt. Dit laatste duidt erop dat het iteratieproces slecht convergeert. In 29 stappen bereiken we punt B in grafiek IV met nog een stapgrootte van 0,00125 N.
Grafiek V geeft de veerkarakteristiek van dezelfde gekromde bladveer met dit verschil, dat de veer nu in 8 elementen van ongeveer gelijke lengte verdeeld is. We merken onmiddellijk op, dat het proces hier netjes naar het maximum van de veerkarakteristiek loopt met redelijke stapgroottes van 2 à 4 N. De laatste stap was 0,5 N.
Het merkwaardige is nu, dat we met minder elementen kennelijk betere resultaten verkrijgen, waarvan mij de oorzaak op dit moment nog niet geheel duidelijk is. Een mogelijke verklaring is de volgende:
Als we een fijne elementverdeling hebben, dan bestaat de mogelijkheid, dat tijdens een belastingsstap een bepaald element weinig vervormd. Dat wil zeggen dat de relaties tussen spanningen en verplaatsingen op dat moment slecht gedefinieerd zijn, omdat de kromme van de spanning - verplaatsing op een maximum respectievelijk minimum zit. Dit kan dan zijn effect hebben in de totale stijfheidsmatrix, die dan slecht geconditi-

tioneerd kan zijn, tenminste in die betekenis, dat ik de stapgrootte steeds kleiner zal moeten maken om het iteratieproces voldoende snel te laten convergeren.
Met minder elementen zal de vervorming per element groter zijn, zodat de spannings-verplaatsingsrelaties beter gedefinieerd zijn.
Uit het voorgaande kan misschien geconcludeerd worden, dat we de elementverdeling telkens moeten aanpassen en wel zodanig, dat een fijne elementverdeling aangebracht wordt in gebieden met grote spanningsgradiënten en een grove elementverdeling in gebieden waar de spanning weinig meer verandert.
Ik dacht dat het gebruikte convergentiecriterium goed moet zijn, omdat toch altijd aan evenwicht voldaan zal moeten worden.
In grafiek XIII is de moment-verplaatsingskromme van een gekromde bladveer weergegeven met tot slot in grafiek XIV de vervormde toestand bij een moment van 50 Nm.

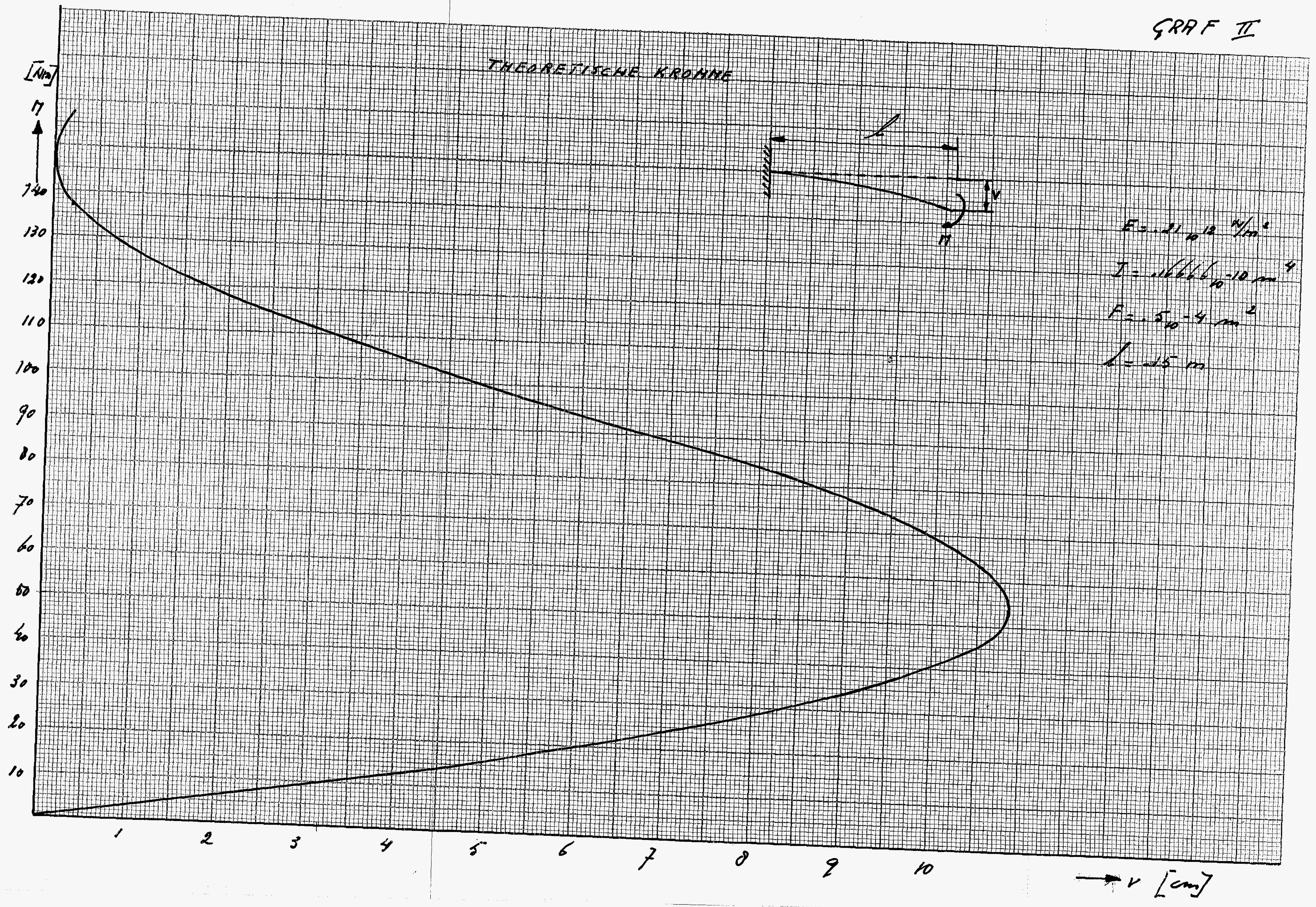
APPENDIX I

GRAFIEKEN

GRAF. I


M [Nm]
36
30
25
20
15
10
5
1
2
3
4
5
6
7
8
9
10
Δy cm
① „Incremental approach" met 3 elementen
② „Incremental approach" met 10 elementen
③ „Gemod. Newton-Raphson" met 6 elementen
④ „Gemod. Newton-Raphson" met 15 elementen
⑤ „Newton-Raphson" theoretische kromme en ook zo benaderd met verschillende ele-ment verdelingen.

GRAF II


THEORETISCHE KROMME
M
[Nm]
v [cm]
l
v
M
E = .21 10 12 N/m²
I = .16666 10 -10 m 4
F = .5 10 -4 m 2
l = .15 m
10
20
30
40
50
60
70
80
90
100
110
120
130
140
1
2
3
4
5
6
7
8
9
10

GRAF. III

VERVORMDE TOESTAND

t.g.v. UITWENDIG MOMENT

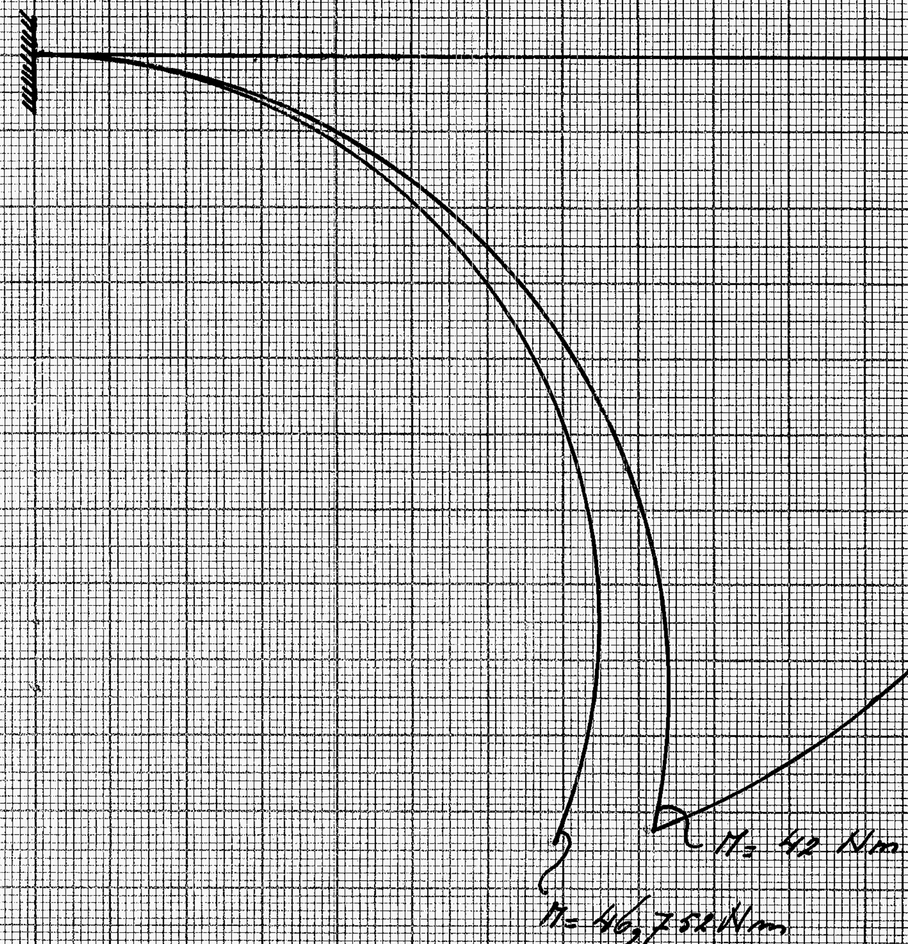
$E = .21_{10}12 \; N/m^2$

$I = .16666_{10}{-10} \; m^4$

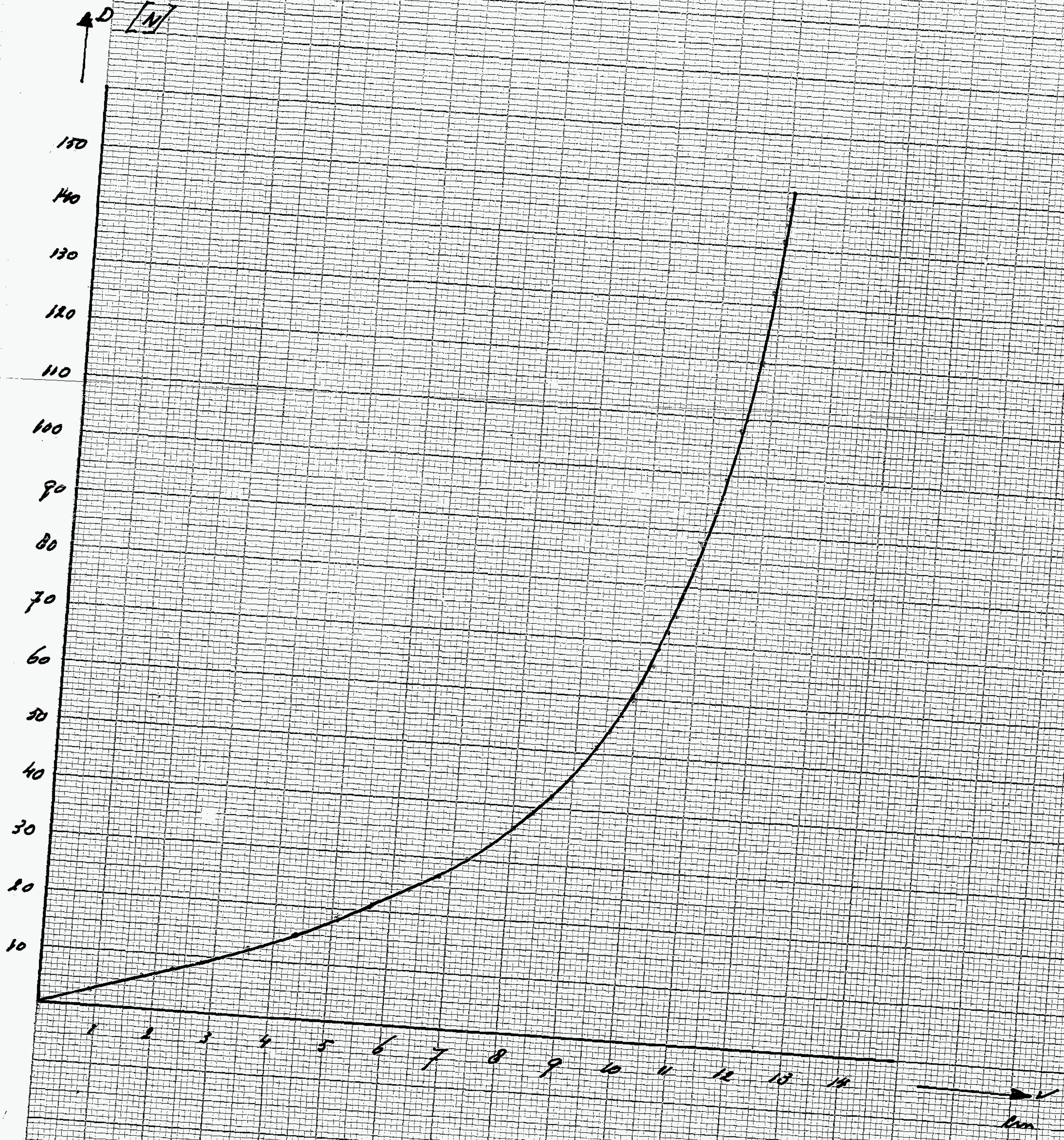
$F = .5_{10}{-4} \; m^2$

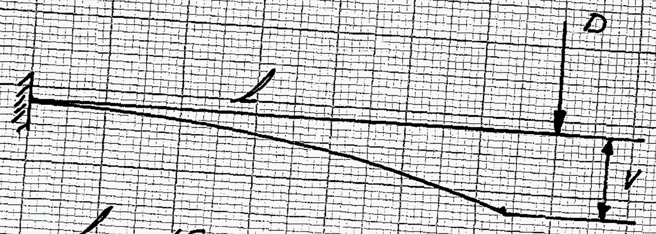
$l = .15 \; m$

max. MOMENT = 46,75 Nm.



GRAF IV





$E = .21_{10}{}^{12}\ N/m^2$

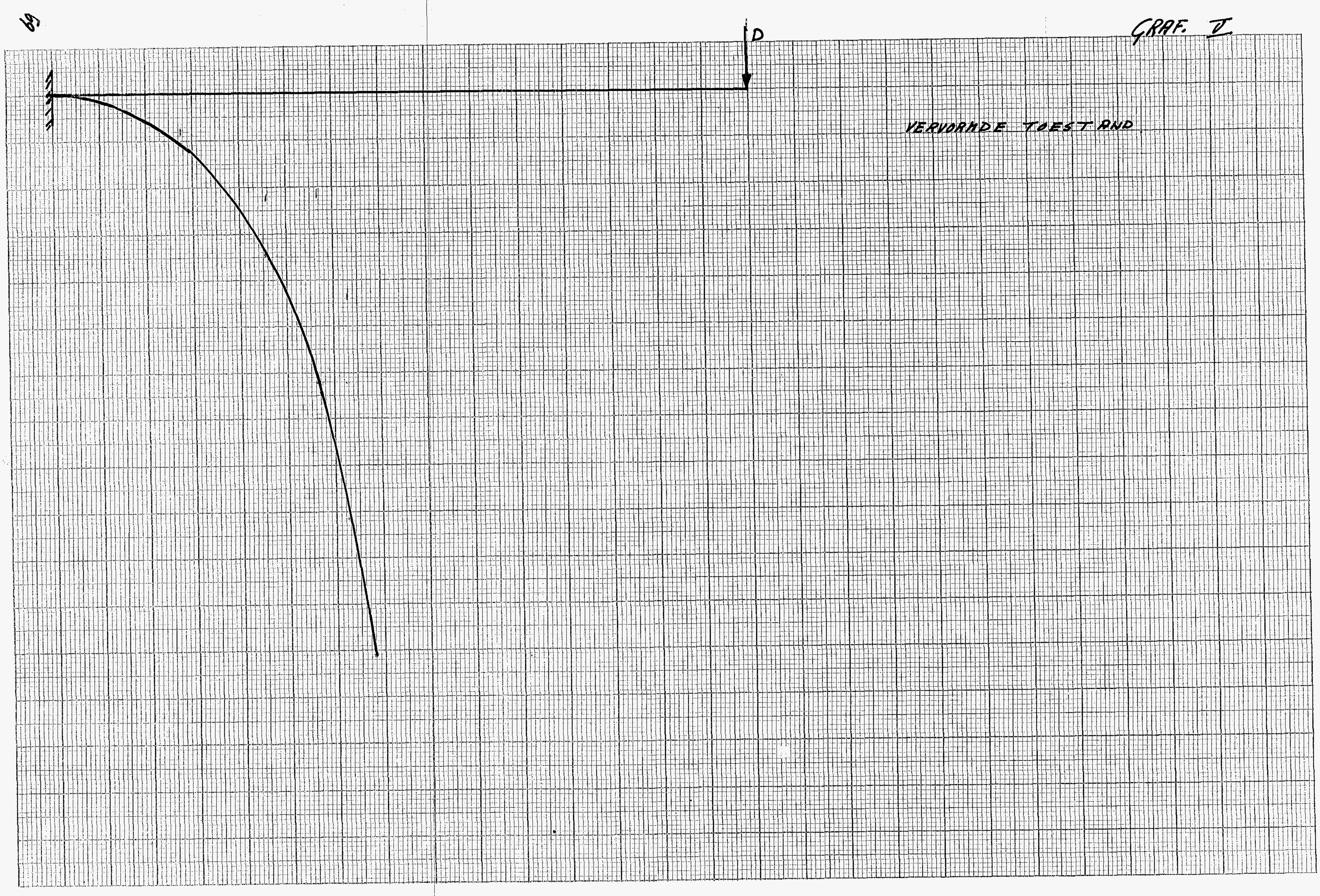
$l = .15\ m$

$I = .166666_{10}{}^{-11}$

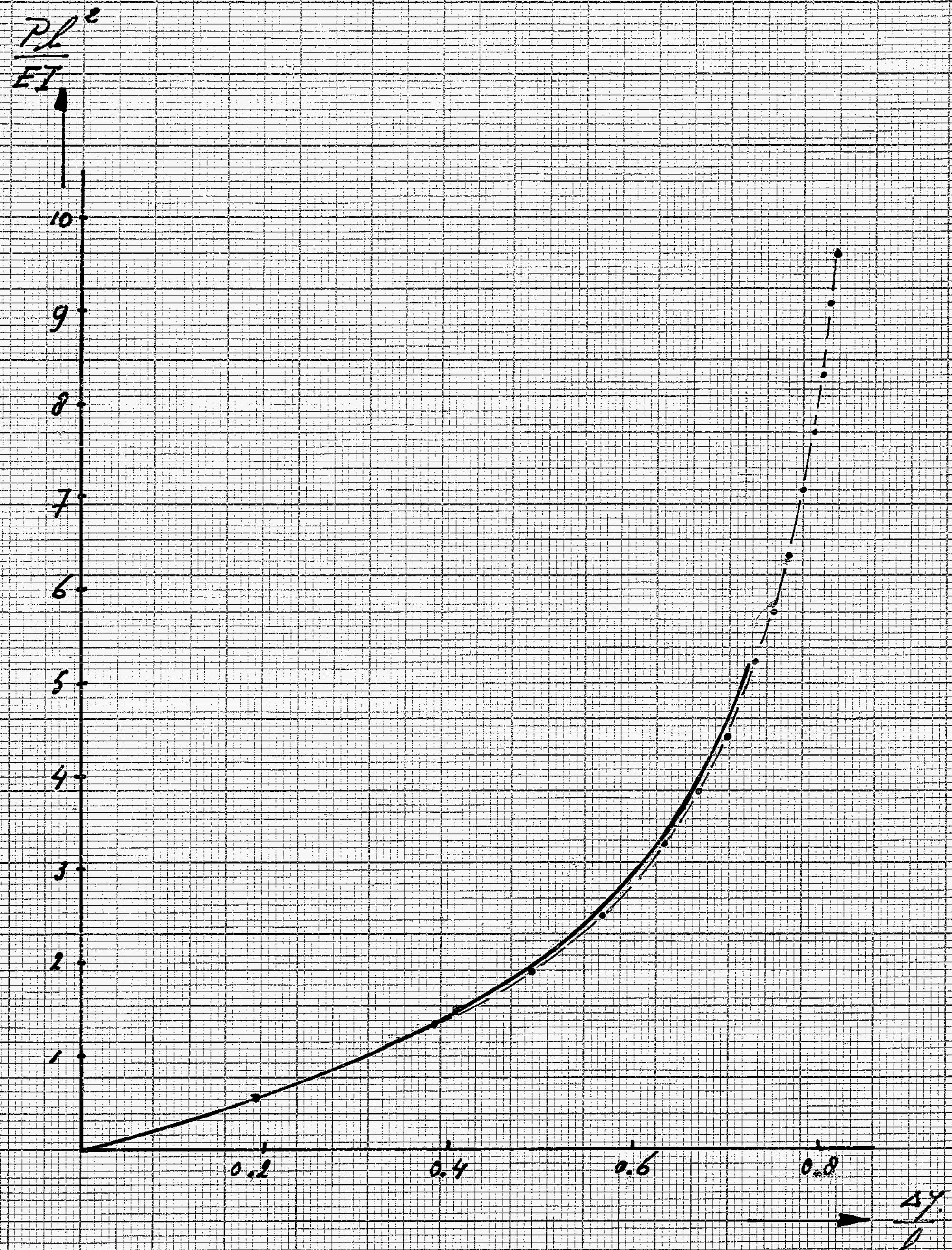
$A = .5_{10}{}^{-4}$

5 elementen

GRAF. V



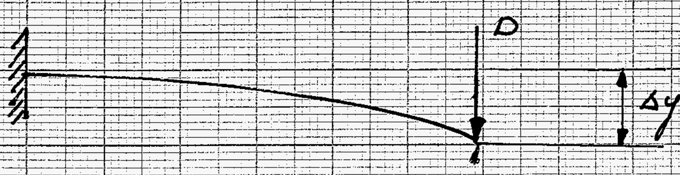
GRAF VI



—— : oplossing met elleptische integralen

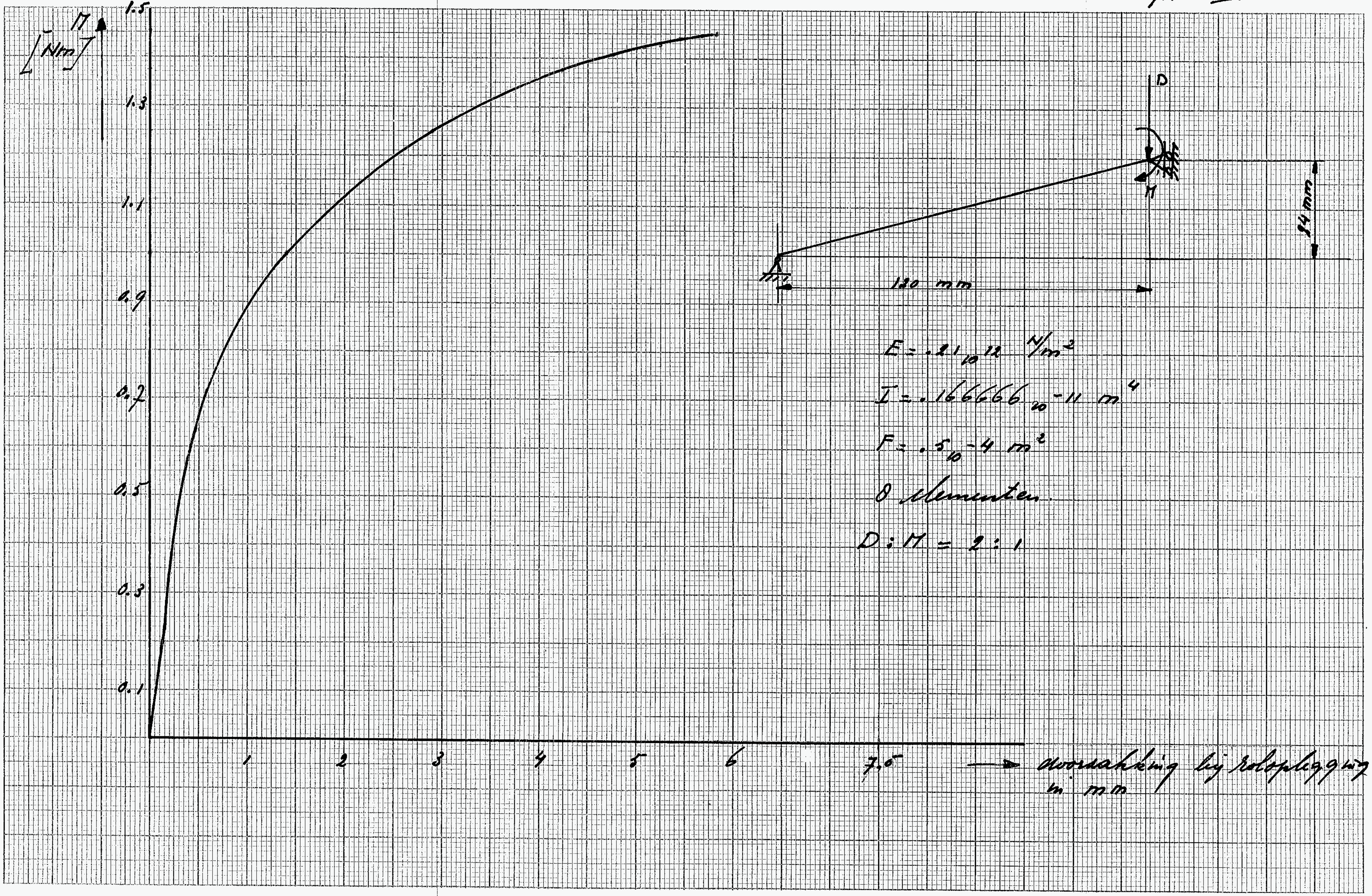
—·— : resultaten met programma 05064608

(de punten zijn de berekende punten)

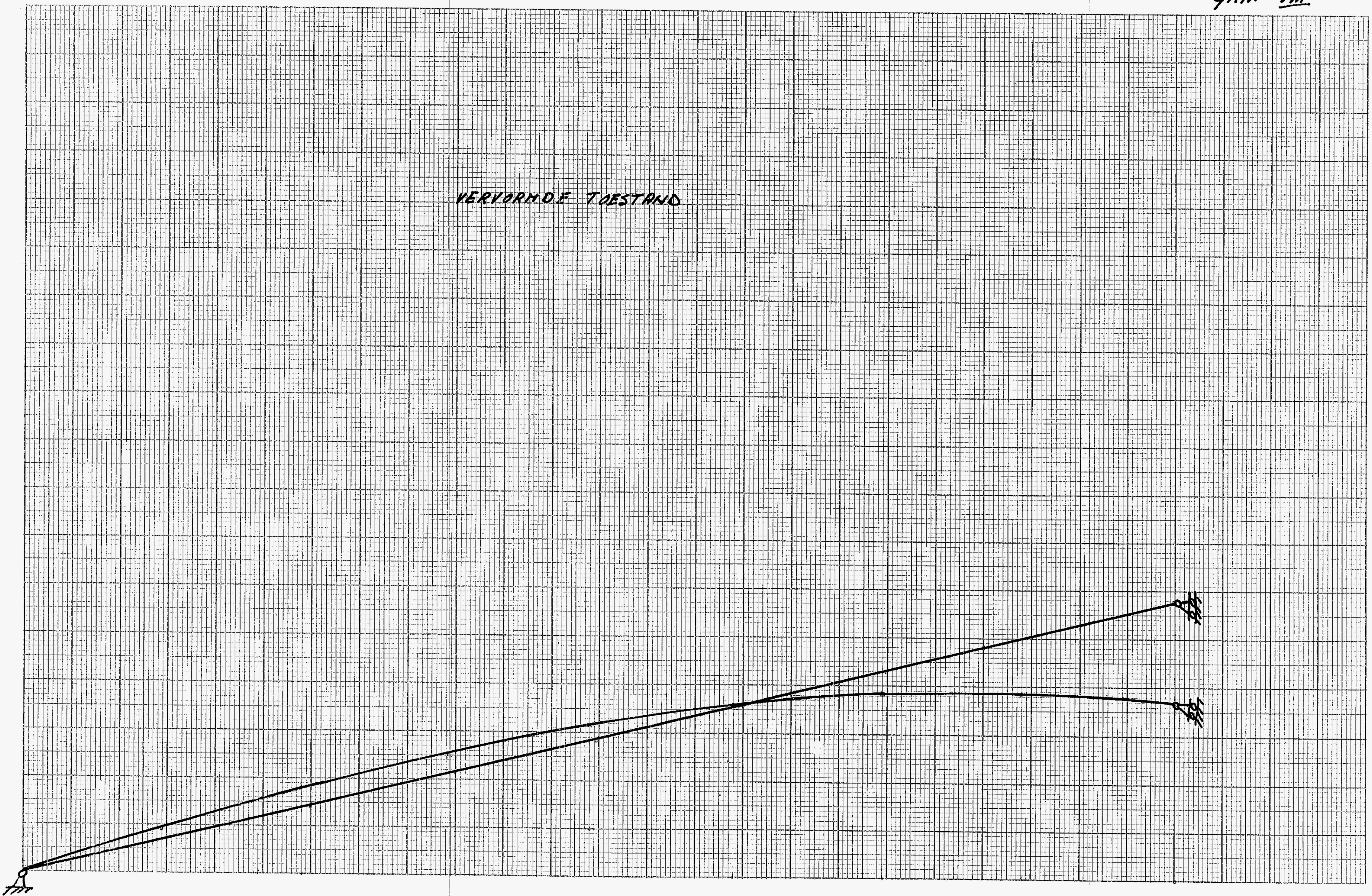


l - oorspronkelijke lengte

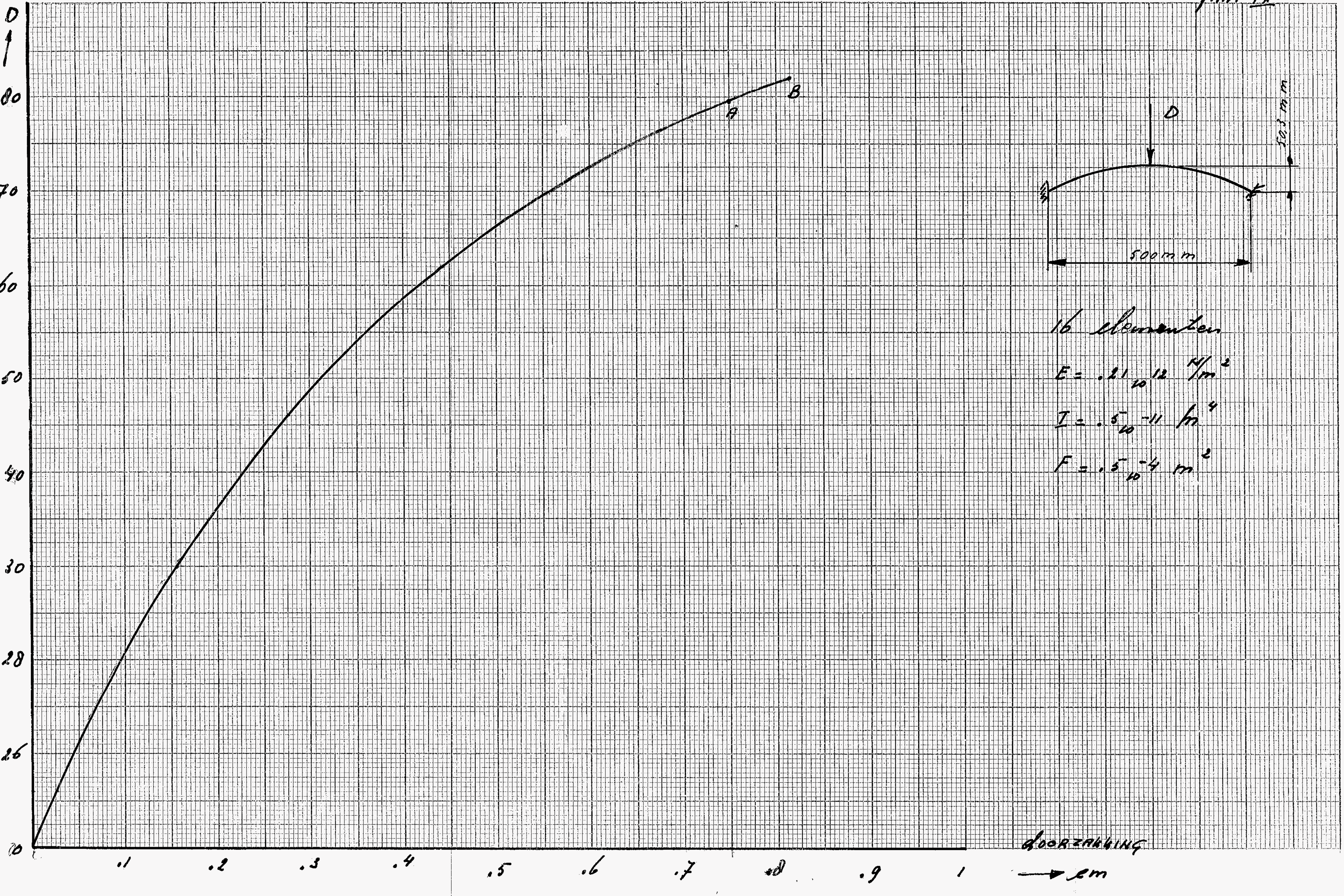
GRAF VII


M [Nm]
1.5
1.3
1.1
0.9
0.7
0.5
0.3
0.1
1
2
3
4
5
6
7.5
doorzakking bij roloplegging in mm
D
M
120 mm
84 mm
E = .21₁₀12 N/m²
I = .166666₁₀-11 m⁴
F = .5₁₀-4 m²
8 elementen
D : M = 2 : 1

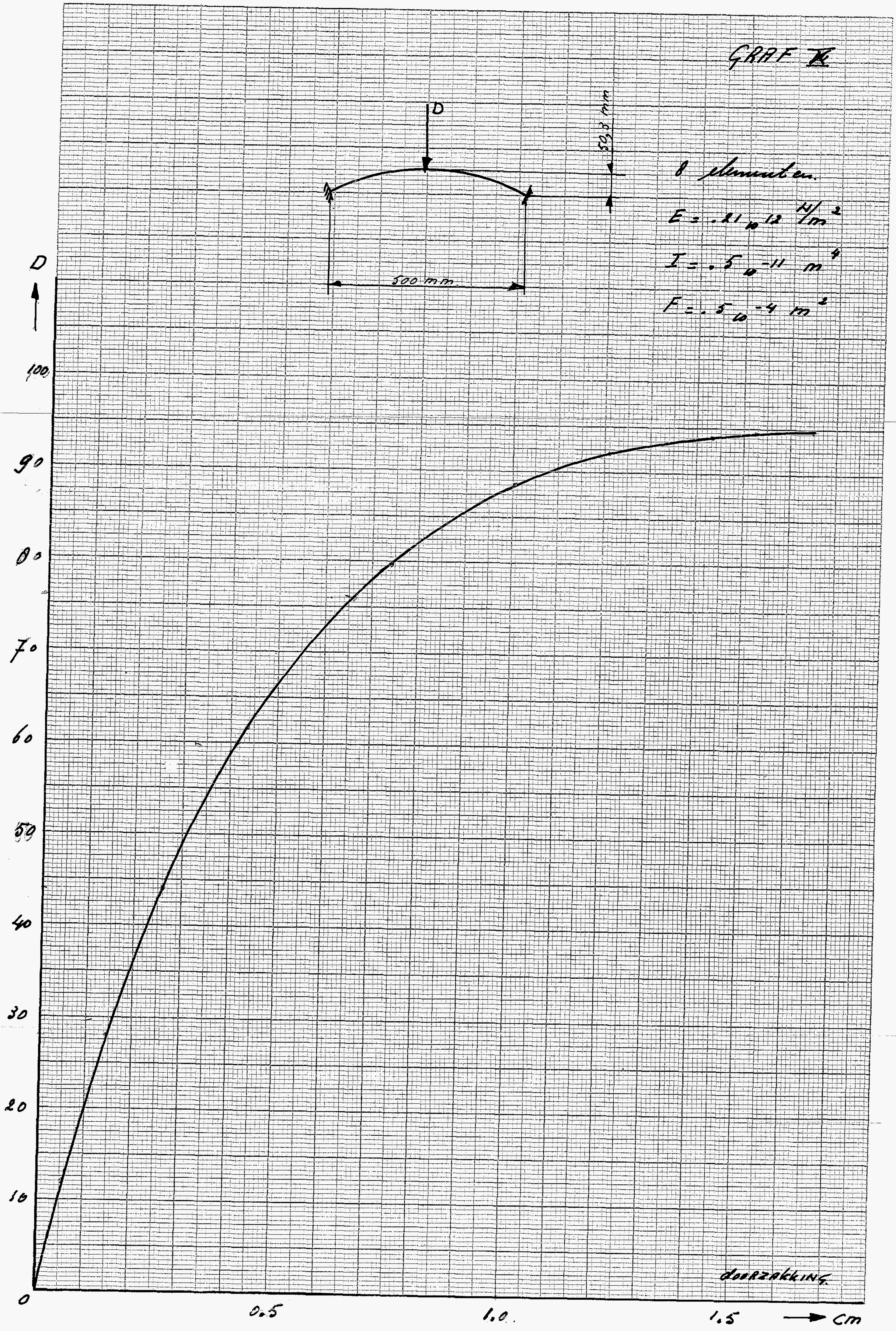
GRAF VIII

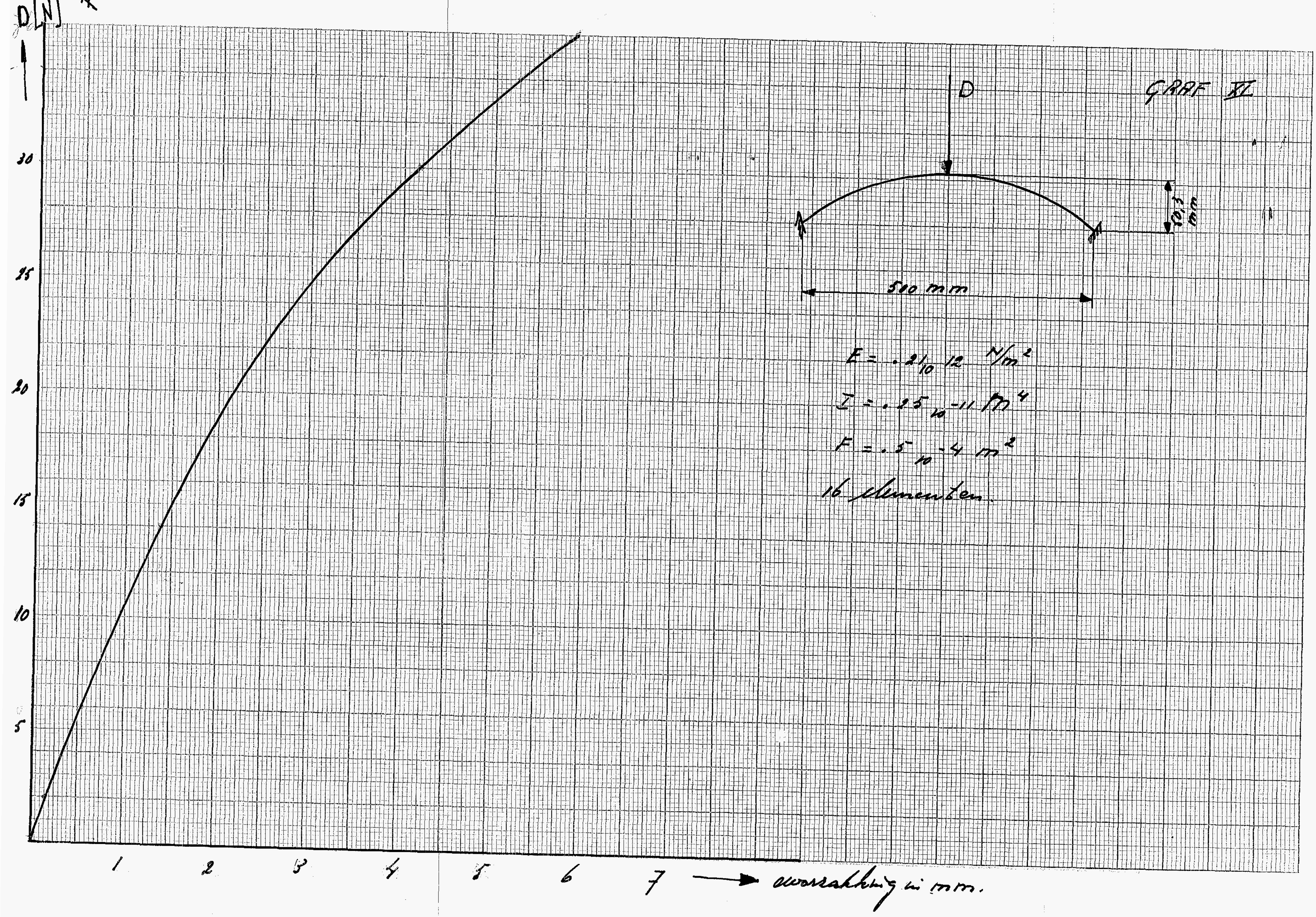


GRAF IX



D
A
B
D
500mm
50,3 mm
16 elementen
E = .21 10^12 N/m^2
I = .5 10^-11 m^4
F = .5 10^-4 m^2
doorzakking
cm


GRAF IX
D
50,3 mm
500 mm
8 elementen.
E = .21 10 12 N/m 2
I = .5 10 -11 m 4
F = .5 10 -4 m 2
D
100
90
80
70
60
50
40
30
20
10
0
0.5
1.0
1.5
cm
doorzakking



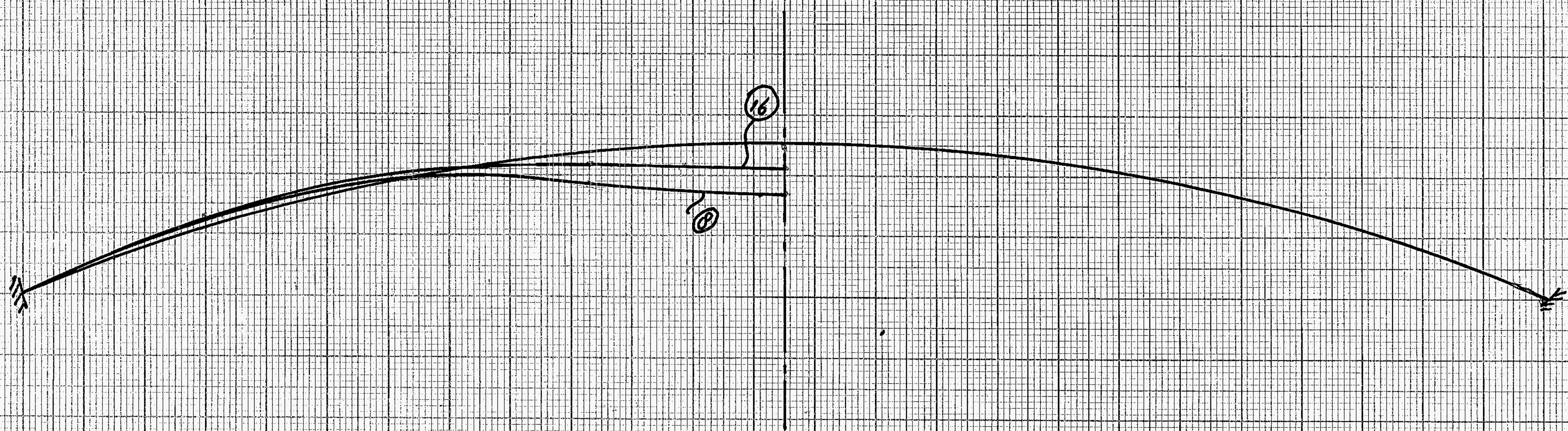
D[N]
5
10
15
20
25
30
1
2
3
4
5
6
7
doorzakking in mm.
GRAF XI
D
500 mm
50,3 mm
E = .21 10 12 N/m2
I = .25 10 -11 m4
F = .5 10 -4 m2
16 elementen.

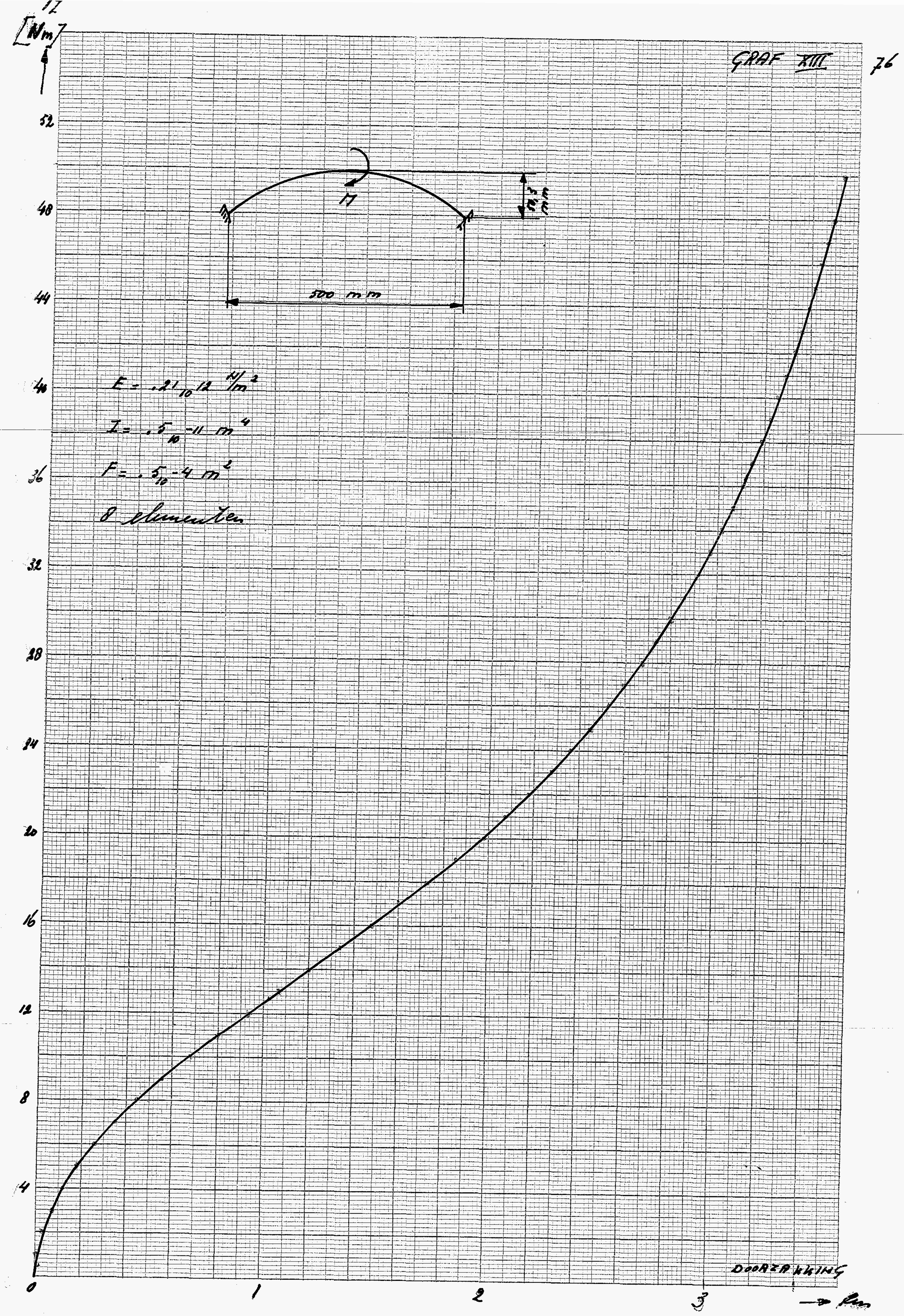
GRAF XII

VERVORMDE TOESTAND

HOREND BY GRAFIEK no IV met 16 elem. (16)

EN GRAFIEK no V met 8 element (8)





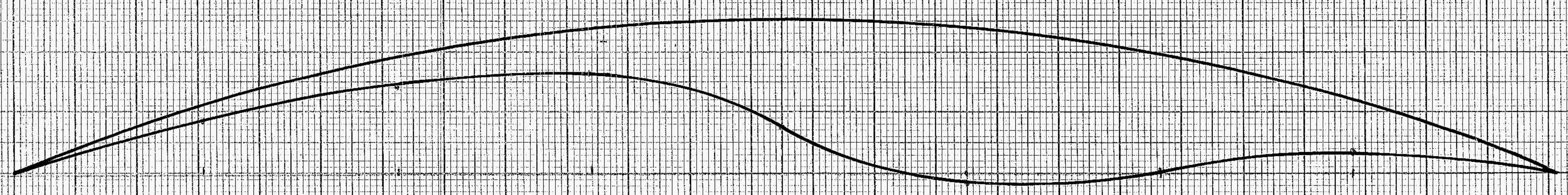
M [Nm]
GRAF XIII
76
52
48
44
40
36
32
28
24
20
16
12
8
4
0
1
2
3
M
3 mm
500 mm
E = .21 10 12 N/m 2
I = .5 10 -11 m 4
F = .5 10 -4 m 2
8 elementen
DOORZAKKING
mm

GRAF XIV

VERVORMDE TOESTAND

HORENDE BY GRAFIEK No XIII

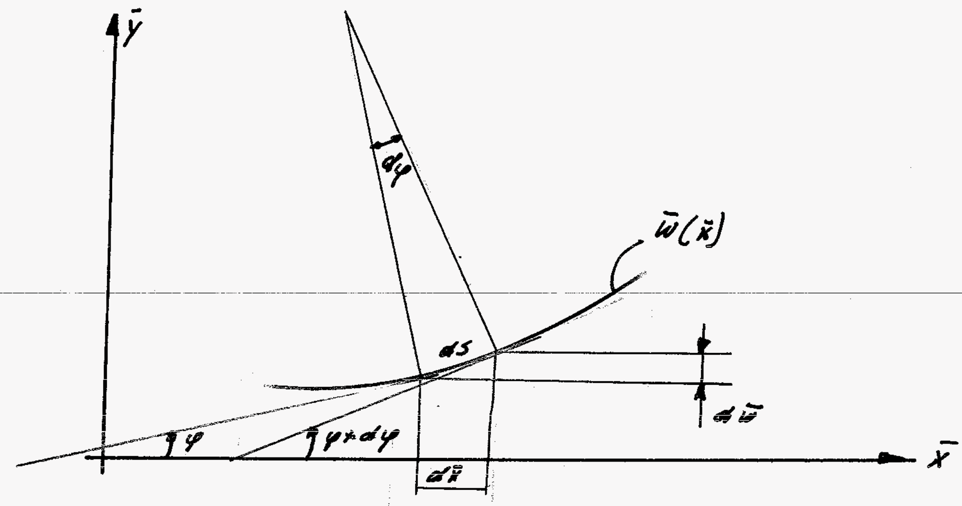
8 elementen



APPENDIX II

Bepaling van $\int_{x=0}^{l} \left(\frac{d\bar{w}}{d\bar{x}}\right)^2 d\bar{x}$

We beschouwen een element in vervormde toestand:



We kunnen de lengte van het balkelement in vervormde toestand berekenen.

$$l_{vervormd} = \int_0^{l'} ds$$

Verder kunnen we de volgende relaties afleiden:

$$ds = \left(d\bar{x}^2 + d\bar{w}^2\right)^{1/2}$$

$$ds = \left(1 + \left(\frac{d\bar{w}}{d\bar{x}}\right)^2\right)^{1/2} d\bar{x}$$

$$ds \approx \left\{1 + \frac{1}{2}\left(\frac{d\bar{w}}{d\bar{x}}\right)^2\right\} d\bar{x}$$

$$l_{vervormd} = \int_0^{l'} \left\{1 + \frac{1}{2}\left(\frac{d\bar{w}}{d\bar{x}}\right)^2\right\} d\bar{x}$$

De oorspronkelijke lengte van het element is l, zodat de lengteverandering t.g.v. het $\bar{w}$-veld gelijk is aan:

$$l_{vervormd} - l = \frac{1}{2}\int_0^{l'} \left(\frac{d\bar{w}}{d\bar{x}}\right)^2 d\bar{x}$$

Het $\bar{w}$-veld hebben we met een 3^de^ graads-polynoom benaderd; terwijl we het $\bar{u}$-veld met een eerste-graads polynoom benaderd hebben.

$$\bar{w} = a_3 + a_4\bar{x} + a_5\bar{x}^2 + a_6\bar{x}^3$$

$$\bar{u} = a_1 + a_2\bar{x}$$

De randvoorwaarden zijn:

$\bar{x}=0$ $\bar{u}=0$ $\bar{x}=l'$ $\bar{u} = l - l' = u_2$

$\bar{w} = 0$ $\bar{w} = 0$

$\dfrac{d\bar{w}}{d\bar{x}} = \bar{\varphi}_1 = \varepsilon_2$ $\dfrac{d\bar{w}}{d\bar{x}} = -\bar{\varphi}_2 = -\varepsilon_3$

Met deze randvoorwaarden kunnen we voor het $\bar{w}$-veld de volgende vergelijking afleiden:

$$\bar{w} = \varepsilon_2\,\bar{x}\left\{1-\frac{\bar{x}}{l'}\right\}^2 + \varepsilon_3\,\frac{\bar{x}^2}{l'}\left\{1-\frac{\bar{x}}{l'}\right\}$$

of in vektornotatie:

$$\bar{w} = \left[0 \quad \bar{x}\left\{1-\frac{\bar{x}}{l'}\right\}^2 \quad \frac{\bar{x}^2}{l'}\left\{1-\frac{\bar{x}}{l'}\right\}\right]\varepsilon$$

$$\frac{d\bar{w}}{d\bar{x}} = \left[0 \quad 1-\frac{4\bar{x}}{l'}+3\left(\frac{\bar{x}}{l'}\right)^2 \quad \frac{2\bar{x}}{l'}-3\left(\frac{\bar{x}}{l'}\right)^2\right]\varepsilon$$

We kunnen nu de $\int_0^l \left(\frac{d\bar{w}}{d\bar{x}}\right)^2 d\bar{x}$ als volgt schrijven:

$$\int_0^{l'}\left(\frac{d\bar{w}}{d\bar{x}}\right)^2 d\bar{x} = \int_0^{l'} \varepsilon^T \begin{bmatrix} 0 \\ 1-4\frac{\bar{x}}{l'}+3\left(\frac{\bar{x}}{l'}\right)^2 \\ \frac{2\bar{x}}{l'}-3\left(\frac{\bar{x}}{l'}\right)^2 \end{bmatrix} \left[0 \quad 1-\frac{4\bar{x}}{l'}+3\left(\frac{\bar{x}}{l'}\right)^2 \quad \frac{2\bar{x}}{l'}-3\left(\frac{\bar{x}}{l'}\right)^2 \quad \varepsilon\right] d\bar{x}$$

$$= \int_0^{l'} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \left[1-\frac{4\bar{x}}{l'}+3\left(\frac{\bar{x}}{l'}\right)^2\right]^2 & \frac{2\bar{x}}{l'}-11\left(\frac{\bar{x}}{l'}\right)^2+18\left(\frac{\bar{x}}{l'}\right)^3-9\left(\frac{\bar{x}}{l'}\right)^4 \\ 0 & \frac{2\bar{x}}{l'}-11\left(\frac{\bar{x}}{l'}\right)^2+18\left(\frac{\bar{x}}{l'}\right)^3-9\left(\frac{\bar{x}}{l'}\right)^4 & \left[\frac{2\bar{x}}{l'}-3\left(\frac{\bar{x}}{l'}\right)^2\right]^2 \end{bmatrix} \varepsilon\, d\bar{x}$$

Bij uitwerking volgt nu:

$$\int_0^{l} \left(\frac{d\bar{w}}{d\bar{x}}\right)^2 = \varepsilon^T \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{2}{15} l^3 & \frac{l^3}{30} \\ 0 & \frac{l^3}{30} & \frac{2}{15} l^3 \end{bmatrix} \varepsilon$$

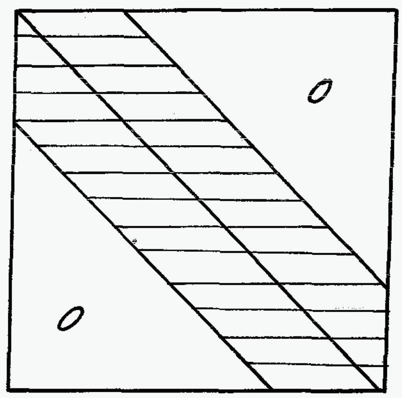
$$= \frac{l^3}{30}\left[4\bar{\psi}_1^{\,2} - 2\bar{\psi}_1\bar{\psi}_2 + 4\bar{\psi}_2^{\,2}\right]$$

APPENDIX III

Beschrijving van de procedure CHOLBD(n, m, dec, a, b, fail)
Gegeven is het stelsel vergelijkingen met reële componenten,

$$Ax = b$$

waarin A een n-rijige symmetrische positief-definiete matrix is, die een bandstructuur heeft. Dus de interessante termen van de matrix A zijn om de hoofddiagonaal geconcentreerd.



De bandbreedte m van de matrix A definiëren we nu als het getal dat aangeeft welke laatste co-diagonaal in het rechtsbovengedeelte van A nog termen ongelijk aan nul bevat, bezien vanaf de hoofddiagonaal van A. Dit betekent dat

$$A[i,j] = 0 \text{ voor } |i-j| > m .$$

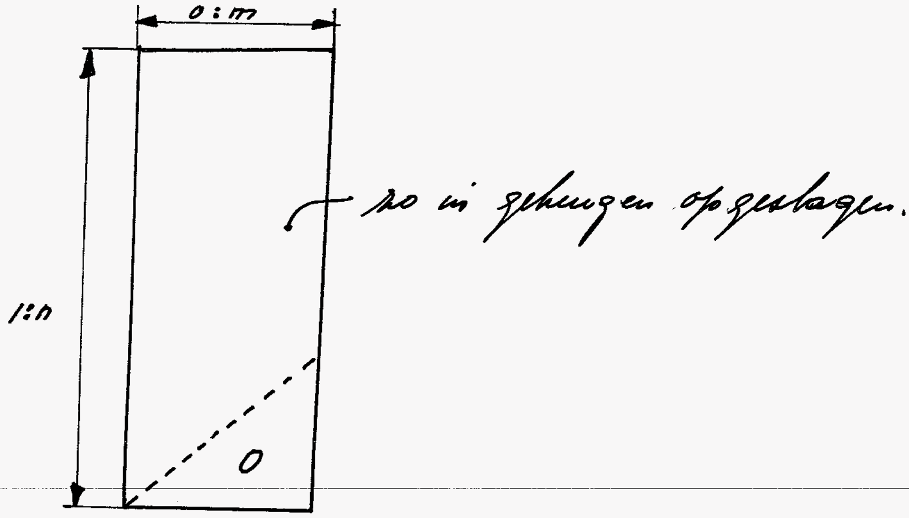
De matrix A ontbinden we in 2 matrices L en R en wel zodanig, dat

$$A = LR$$
$$L = R^T$$

en R is een 3hoeksmatrix, d.w.z. alleen het rechter bovengedeelte is gevuld.

Omdat A een symmetrische matrix is, hoeft men slechts het rechtsbovengedeelte te kennen. Deze interessante gegevens bergen we daarom in de matrix

$a[1:n, 0:m]$ op.



De diagonaalelementen van A staan nu op $a[i,0]$. De termen waarvoor $i+j>n$ komen in de matrix a termen voor, die gelijk aan nul zijn.
Al naar gelang de keuze van de parameter dec, voert de procedure de volgende bewerkingen uit:

dec =	0	1	2	3	4	5	6	7
Invoer matrix a en vektor b inlezen of berekenen.	a	a	a	a	a	a	a	a
	b	b	b	b	b	b	x	x
uitvoer a overschreven met.	R	a	R	a	R	a	R	a
uitvoer b overschreven met	$A^{-1}b$	$A^{-1}b$	$R^{-1}b$	$R^{-1}b$	$L^{-1}b$	$L^{-1}b$	b	b

De label fail vormt de uitgang als de matrix A niet positief definiet is.

APPENDIX IV

Bij de procedure REKSPANITTERATIE is de opmerking geplaatst, dat we na elke stap in de uitwendige belastingsvektor, of na elke iteratie, de hoek die de verbindingslijn van de 2 knooppunten van element i maakt met de x-as opnieuw moeten berekenen. Dit moeten we telkens doen, omdat deze hoek in de rekdefinities gebruikt wordt.

bijv: $\varepsilon_2 = -\varphi_1 + \arctan \dfrac{y_2-y_1}{x_2-x_1} - \arctan \dfrac{y_2-y_1+w_2-w_1}{x_2-x_1+u_2-u_1}$

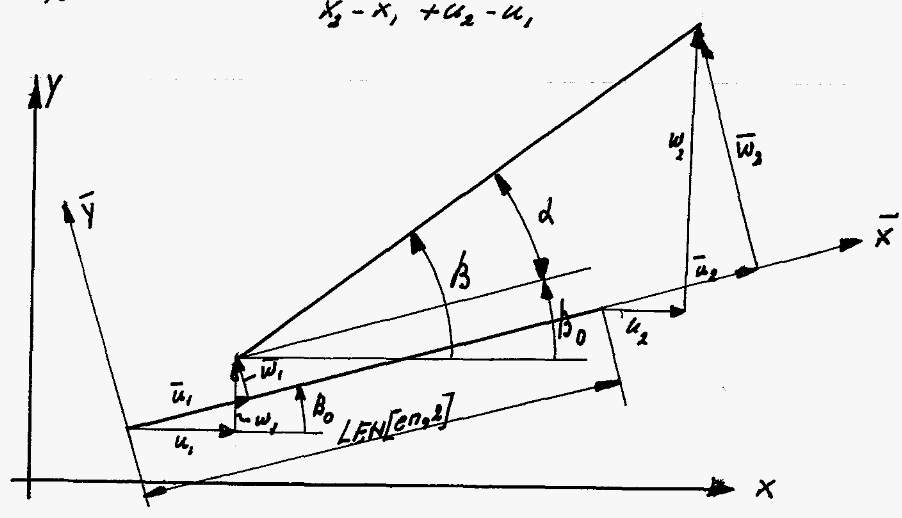
φ_1 is hierin de absolute hoekverdraaiing dat element i in het eerste knooppunt ondergaat t.g.v. belastings- of ~~iteratie~~ stap.

$\arctan \dfrac{y_2-y_1}{x_2-x_1}$ is de hoek, welke het element in uitgangspositie maakt met de x-as. Deze kunnen we door een eenmalige berekening vastleggen. (procedure UITGANGSHOEK)

$\arctan \dfrac{y_2-y_1+w_2-w_1}{x_2-x_1+u_2-u_1}$ moeten we ook eenduidig vastleggen en wel op een zodanige manier, dat de respectievelijke ε_2 en ε_3 continue funkties van de verplaatsingen zijn.

De werkwijze gaat nu als volgt:

Stel $\beta = \arctan \dfrac{y_2-y_1+w_2-w_1}{x_2-x_1+u_2-u_1}$



Uit de figuur zien we, dat geldt:

$$\beta = \beta_0 + \alpha$$

We nemen aan, dat β_0 bekend is. We willen nu hoek α bepalen. We transformeren hiervoor de verplaatsingsvektor $\vec{u} = [u_1\ w_1\ \varphi_1 u_2\ w_2\ \varphi_2]$ naar het lokale assenstelsel $\bar{x}-\bar{y}$.

Nu geldt:

$$\begin{aligned}
\bar{u}_1 &= u_1 \cos\beta_0 + v_1 \sin\beta_0 \\
\bar{v}_1 &= -u_1 \sin\beta_0 + v_1 \cos\beta_0 \\
\bar{u}_2 &= u_2 \cos\beta_0 + v_2 \sin\beta_0 \\
\bar{v}_2 &= -u_2 \sin\beta_0 + v_2 \cos\beta_0
\end{aligned}$$

Nu berekenen we de hoek α als volgt:

$$\alpha = \arctan \frac{\bar{v}_2 - \bar{v}_1}{LEN[en,2] + \bar{u}_2 - \bar{u}_1}$$

We stellen nu:

$$\begin{aligned}
\bar{v}_2 &= y_2 \\
\bar{v}_1 &= y_1 \\
LEN[en,2] + \bar{u}_2 &= x_2 \\
\bar{u}_1 &= x_1
\end{aligned}$$

Dan definieren we $\alpha = \arctan \frac{y_2 - y_1}{x_2 - x_1}$ als volgt:

1. $x_2 > x_1$ dan $\alpha = \arctan \frac{y_2 - y_1}{x_2 - x_1}$
2. $x_2 < x_1$ en $y_2 > y_1$ dan $\alpha = \arctan \frac{y_2 - y_1}{x_2 - x_1} + \pi$
3. $x_2 < x_1$ en $y_2 < y_1$ dan $\alpha = \arctan \frac{y_2 - y_1}{x_2 - x_1} - \pi$
4. $x_2 = x_1$ en $y_2 > y_1$ dan $\alpha = \frac{\pi}{2}$
5. $x_2 = x_1$ en $y_2 < y_1$ dan $\alpha = -\frac{\pi}{2}$.

De hoek α is nu eenduidig gedefinieerd in

$$-\pi \leq \alpha \leq \pi$$

Nu berekenen we hoek β als volgt:

$$\beta = \beta_0 + \alpha$$

en hierna krijgt p_0 de waarde van p. Met andere woorden, we hebben ervoor gezorgd dat de atelan $\dfrac{y_2 - y_1 + w_2 - w_1}{x_2 - x_1 + u_2 - u_1}$ een continue funktie van de knooppuntsverplaatsingen is, omdat we de hoek telkens t.o.v. zijn juist voorafgaande stap berekenen.

APPENDIX V

Afleiding van de matrices $\frac{\partial \varepsilon_i}{\partial u_p}$, $\frac{\partial^2 \varepsilon_x}{\partial u_q \partial u_p}$, $\frac{\partial^2 \varepsilon_y}{\partial u_q \partial u_p}$ en $\frac{\partial^2 \gamma_{xy}}{\partial u_q \partial u_p}$

Enkele hulpdifferentiaties:

$$\frac{\partial \bar{u}_2}{\partial u_1} = -\cos\beta \cos\beta_0 \qquad \frac{\partial \bar{v}_2}{\partial u_1} = -\cos\beta \sin\beta_0$$

$$\frac{\partial \bar{u}_2}{\partial u_2} = \cos\beta \cos\beta_0 \qquad \frac{\partial \bar{v}_2}{\partial v_1} = -\sin\beta \sin\beta_0$$

$$\frac{\partial \bar{u}_2}{\partial v_1} = -\sin\beta \cos\beta_0 \qquad \frac{\partial \bar{v}_2}{\partial u_2} = \cos\beta \sin\beta_0$$

$$\frac{\partial \bar{u}_2}{\partial v_2} = \sin\beta \cos\beta_0 \qquad \frac{\partial \bar{v}_2}{\partial v_2} = \sin\beta \sin\beta_0$$

$$\frac{\partial \bar{u}_2}{\partial u_3} = 0 \qquad \frac{\partial \bar{v}_2}{\partial u_3} = 0$$

$$\frac{\partial \bar{u}_2}{\partial v_3} = 0 \qquad \frac{\partial \bar{v}_2}{\partial v_3} = 0$$

$$\frac{\partial \bar{u}_3}{\partial u_1} = -\cos\gamma \cos(\gamma-\alpha) - l_{13}' \sin(\gamma-\alpha)\left[\frac{\sin\gamma}{l_{13}'} - \frac{\sin\beta}{l_{12}'}\right]$$

$$\frac{\partial \bar{u}_3}{\partial v_1} = -\sin\gamma \cos(\gamma-\alpha) - l_{13}' \sin(\gamma-\alpha)\left[\frac{-\cos\gamma}{l_{13}'} + \frac{\cos\beta}{l_{12}'}\right]$$

$$\frac{\partial \bar{u}_3}{\partial u_2} = -\frac{l_{13}'}{l_{12}'} \sin(\gamma-\alpha)\sin\beta$$

$$\frac{\partial \bar{u}_3}{\partial v_2} = \frac{l_{13}'}{l_{12}'} \sin(\gamma-\alpha)\cos\beta$$

$$\frac{\partial \bar{u}_3}{\partial u_3} = \cos\gamma \cos(\gamma-\alpha) + \sin(\gamma-\alpha)\sin\gamma$$

$$\frac{\partial \bar{u}_3}{\partial v_3} = \sin\gamma \cos(\gamma-\alpha) - \sin(\gamma-\alpha)\cos\gamma$$

$$\frac{\partial \bar{v}_3}{\partial u_1} = -\cos\gamma \sin(\gamma-\alpha) + l_{13}' \cos(\gamma-\alpha)\left[\frac{\sin\gamma}{l_{13}'} - \frac{\sin\beta}{l_{12}'}\right]$$

$$\frac{\partial \bar{v}_3}{\partial v_1} = -\sin\gamma \sin(\gamma-\alpha) + l_{13}' \cos(\gamma-\alpha)\left[\frac{-\cos\gamma}{l_{13}'} + \frac{\cos\beta}{l_{12}'}\right]$$

$$\frac{\partial \bar{v}_3}{\partial u_2} = \frac{l_{13}'}{l_{12}'} \cos(\gamma-\alpha)\sin\beta$$

$$\frac{\partial \bar{v}_3}{\partial v_2} = -\frac{l_{13}'}{l_{12}'} \cos(\gamma-\alpha)\cos\beta$$

$$\frac{\partial \bar{v}_3}{\partial u_3} = \cos\gamma\sin(\gamma-\alpha) - \cos(\gamma-\alpha)\sin\gamma$$

$$\frac{\partial \bar{v}_3}{\partial v_3} = \sin\gamma\sin(\gamma-\alpha) + \cos(\gamma-\alpha)\cos\gamma$$

Verder gebruiken we nog:

$$\gamma = \arctan\frac{y_3-y_1 \quad +v_3-v_1}{x_3-x_1 \quad +u_3-u_1}$$

$$\frac{\partial\gamma}{\partial u_1} = \frac{\sin\gamma}{l_{13}'} \qquad \frac{\partial\gamma}{\partial u_2} = 0 \qquad \frac{\partial\gamma}{\partial u_3} = -\frac{\sin\gamma}{l_{13}'}$$

$$\frac{\partial\gamma}{\partial v_1} = -\frac{\cos\gamma}{l_{13}'} \qquad \frac{\partial\gamma}{\partial v_2} = 0 \qquad \frac{\partial\gamma}{\partial v_3} = \frac{\cos\gamma}{l_{13}'}$$

$$\alpha = \arctan\frac{y_2-y_1 \, +v_2-v_1}{x_2-x_1 \, +u_2-u_1} - \arctan\frac{y_2-y_1}{x_2-x_1}$$

$$\frac{\partial\alpha}{\partial u_1} = \frac{\sin\beta}{l_{12}'} \qquad \frac{\partial\alpha}{\partial u_2} = -\frac{\sin\beta}{l_{12}'} \qquad \frac{\partial\alpha}{\partial u_3} = 0$$

$$\frac{\partial\alpha}{\partial v_1} = -\frac{\cos\beta}{l_{12}'} \qquad \frac{\partial\alpha}{\partial v_2} = \frac{\cos\beta}{l_{12}'} \qquad \frac{\partial\alpha}{\partial v_3} = 0$$

$$\beta = \arctan\frac{y_2-y_1 \, +v_2-v_1}{x_2-x_1 \, +u_2-u_1}$$

$$\frac{\partial\beta}{\partial u_1} = \frac{\sin\beta}{l_{12}'} \qquad \frac{\partial\beta}{\partial u_2} = -\frac{\sin\beta}{l_{12}'} \qquad \frac{\partial\alpha}{\partial u_3} = 0$$

$$\frac{\partial\beta}{\partial v_1} = -\frac{\cos\beta}{l_{12}'} \qquad \frac{\partial\beta}{\partial v_2} = \frac{\cos\beta}{l_{12}'} \qquad \frac{\partial\alpha}{\partial v_3} = 0$$

We zullen nu beginnen met de matrix $\dfrac{\partial \varepsilon_i}{\partial u_p}$

$$\frac{\partial \varepsilon_i}{\partial u_p} = \begin{bmatrix} \frac{\partial \varepsilon_x}{\partial u_1} & \frac{\partial \varepsilon_x}{\partial v_1} & \frac{\partial \varepsilon_x}{\partial u_2} & \frac{\partial \varepsilon_x}{\partial v_2} & \frac{\partial \varepsilon_x}{\partial u_3} & \frac{\partial \varepsilon_x}{\partial v_3} \\ \frac{\partial \varepsilon_y}{\partial u_1} & \frac{\partial \varepsilon_y}{\partial v_1} & \frac{\partial \varepsilon_y}{\partial u_2} & \frac{\partial \varepsilon_y}{\partial v_2} & \frac{\partial \varepsilon_y}{\partial u_3} & \frac{\partial \varepsilon_y}{\partial v_3} \\ \frac{\partial \varepsilon_t}{\partial u_1} & \frac{\partial \varepsilon_t}{\partial v_1} & \frac{\partial \varepsilon_t}{\partial u_2} & \frac{\partial \varepsilon_t}{\partial v_2} & \frac{\partial \varepsilon_t}{\partial u_3} & \frac{\partial \varepsilon_t}{\partial v_3} \\ \frac{\partial \gamma_{xy}}{\partial u_1} & \frac{\partial \gamma_{xy}}{\partial v_1} & \frac{\partial \gamma_{xy}}{\partial u_2} & \frac{\partial \gamma_{xy}}{\partial v_2} & \frac{\partial \gamma_{xy}}{\partial u_3} & \frac{\partial \gamma_{xy}}{\partial v_3} \end{bmatrix}$$

$$\frac{\partial \varepsilon_x}{\partial u_1} = \frac{1}{A}\left[(y_3-y_1)\frac{\partial \bar{u}_2}{\partial u_1} + (y_1-y_2)\frac{\partial \bar{u}_3}{\partial u_1}\right]$$

$$\frac{\partial \varepsilon_x}{\partial v_1} = \frac{1}{A}\left[(y_3-y_1)\frac{\partial \bar{u}_2}{\partial v_1} + (y_1-y_2)\frac{\partial \bar{u}_3}{\partial v_1}\right]$$

$$\frac{\partial \varepsilon_x}{\partial u_2} = \frac{1}{A}\left[(y_3-y_1)\frac{\partial \bar{u}_2}{\partial u_2} + (y_1-y_2)\frac{\partial \bar{u}_3}{\partial u_2}\right]$$

$$\frac{\partial \varepsilon_x}{\partial v_2} = \frac{1}{A}\left[(y_3-y_1)\frac{\partial \bar{u}_2}{\partial v_2} + (y_1-y_2)\frac{\partial \bar{u}_3}{\partial v_2}\right]$$

$$\frac{\partial \varepsilon_x}{\partial u_3} = \frac{1}{A}\left[(y_3-y_1)\frac{\partial \bar{u}_2}{\partial u_3} + (y_1-y_2)\frac{\partial \bar{u}_3}{\partial u_3}\right]$$

$$\frac{\partial \varepsilon_x}{\partial v_3} = \frac{1}{A}\left[(y_3-y_1)\frac{\partial \bar{u}_2}{\partial v_3} + (y_1-y_2)\frac{\partial \bar{u}_3}{\partial v_3}\right]$$

$$\frac{\partial \varepsilon_y}{\partial u_1} = \frac{1}{A}\left[(x_1-x_3)\frac{\partial \bar{v}_2}{\partial u_1} + (x_2-x_1)\frac{\partial \bar{v}_3}{\partial u_1}\right]$$

$$\frac{\partial \varepsilon_y}{\partial v_1} = \frac{1}{A}\left[(x_1-x_3)\frac{\partial \bar{v}_2}{\partial v_1} + (x_2-x_1)\frac{\partial \bar{v}_3}{\partial v_1}\right]$$

$$\frac{\partial \varepsilon_y}{\partial u_2} = \frac{1}{A}\left[(x_1-x_3)\frac{\partial \bar{v}_2}{\partial u_2} + (x_2-x_1)\frac{\partial \bar{v}_3}{\partial u_2}\right]$$

$$\frac{\partial \varepsilon_y}{\partial v_2} = \frac{1}{A}\left[(x_1-x_3)\frac{\partial \bar{v}_2}{\partial v_2} + (x_2-x_1)\frac{\partial \bar{v}_3}{\partial v_2}\right]$$

$$\frac{\partial \varepsilon_y}{\partial u_3} = \frac{1}{A}\left[(x_1-x_3)\frac{\partial \bar{v}_2}{\partial u_3} + (x_2-x_1)\frac{\partial \bar{v}_3}{\partial u_3}\right]$$

$$\frac{\partial \varepsilon_y}{\partial v_3} = \frac{1}{A}\left[(x_1-x_3)\frac{\partial \bar{v}_2}{\partial v_3} + (x_2-x_1)\frac{\partial \bar{v}_3}{\partial v_3}\right]$$

$$\frac{\partial \varepsilon_t}{\partial u_1} = \frac{3}{A(x_1+x_2+x_3)}\left[(x_2y_3 - x_3y_2)\right] + \frac{1}{A}(y_2-y_3) + \frac{1}{A}\cdot\frac{y_1+y_2+y_3}{x_1+x_2+x_3}(x_3-x_2)$$

$$\frac{\partial \varepsilon_t}{\partial v_1} = 0$$

$$\frac{\partial \varepsilon_t}{\partial u_2} = \frac{3}{A(x_1+x_2+x_3)}\left[(x_3y_1 - x_1y_3)\right] + \frac{1}{A}(y_3-y_1) + \frac{1}{A}\,\frac{y_1+y_2+y_3}{x_1+x_2+x_3}(x_1-x_3)$$

$$\frac{\partial \varepsilon_t}{\partial v_2} = 0$$

$$\frac{\partial \varepsilon_t}{\partial u_3} = \frac{3}{A(x_1+x_2+x_3)}\left[(x_1y_2 - x_2y_1)\right] + \frac{1}{A}(y_1-y_2) + \frac{1}{A}\,\frac{y_1+y_2+y_3}{x_1+x_2+x_3}(x_2-x_1)$$

$$\frac{\partial \varepsilon_t}{\partial v_3} = 0$$

$$\frac{\partial \gamma_{xy}}{\partial u_1} = \frac{1}{A}\left[(x_1-x_3)\frac{\partial \bar{u}_2}{\partial u_1} + (x_2-x_1)\frac{\partial \bar{u}_3}{\partial u_1} + (y_3-y_1)\frac{\partial \bar{v}_2}{\partial u_1} + (y_1-y_2)\frac{\partial \bar{v}_3}{\partial u_1}\right]$$

$$\frac{\partial \gamma_{xy}}{\partial u_2} = \frac{1}{A}\left[(x_1-x_3)\frac{\partial \bar{u}_2}{\partial u_2} + (x_2-x_1)\frac{\partial \bar{u}_3}{\partial u_2} + (y_3-y_1)\frac{\partial \bar{v}_2}{\partial u_2} + (y_1-y_2)\frac{\partial \bar{v}_3}{\partial u_2}\right]$$

$$\frac{\partial \gamma_{xy}}{\partial u_3} = \frac{1}{A}\left[(x_1-x_3)\frac{\partial \bar{u}_2}{\partial u_3} + (x_2-x_1)\frac{\partial \bar{u}_3}{\partial u_3} + (y_3-y_1)\frac{\partial \bar{v}_2}{\partial u_3} + (y_1-y_2)\frac{\partial \bar{v}_3}{\partial u_3}\right]$$

$$\frac{\partial \gamma_{xy}}{\partial v_1} = \frac{1}{A}\left[(x_1-x_3)\frac{\partial \bar{u}_2}{\partial v_1} + (x_2-x_1)\frac{\partial \bar{u}_3}{\partial v_1} + (y_3-y_1)\frac{\partial \bar{v}_2}{\partial v_1} + (y_1-y_2)\frac{\partial \bar{v}_3}{\partial v_1}\right]$$

$$\frac{\partial \gamma_{xy}}{\partial v_2} = \frac{1}{A}\left[(x_1-x_3)\frac{\partial \bar{u}_2}{\partial v_2} + (x_2-x_1)\frac{\partial \bar{u}_3}{\partial v_2} + (y_3-y_1)\frac{\partial \bar{v}_2}{\partial v_2} + (y_1-y_2)\frac{\partial \bar{v}_3}{\partial v_2}\right]$$

$$\frac{\partial \gamma_{xy}}{\partial v_3} = \frac{1}{A}\left[(x_1-x_3)\frac{\partial \bar{u}_2}{\partial v_3} + (x_2-x_1)\frac{\partial \bar{u}_3}{\partial v_3} + (y_3-y_1)\frac{\partial \bar{v}_2}{\partial v_3} + (y_1-y_2)\frac{\partial \bar{v}_3}{\partial v_3}\right]$$

Voor de matrices $\frac{\partial^2 \varepsilon_x}{\partial u_q \partial u_p}$, $\frac{\partial^2 \varepsilon_y}{\partial u_q \partial u_p}$ en $\frac{\partial^2 \gamma_{xy}}{\partial u_q \partial u_p}$ hebben we de hierna volgende hulpdifferentiaties nodig:

$$\frac{\partial \bar{u}_2}{\partial u_1 \partial u_1} = \cos\beta_0 \frac{\sin^2\beta}{l_{12}} \qquad \frac{\partial \bar{v}_2}{\partial u_1 \partial u_1} = \sin\beta_0 \frac{\sin^2\beta}{l_{12}}$$

$$\frac{\partial \bar{u}_2}{\partial u_1 \partial v_1} = -\cos\beta_0 \frac{\sin 2\beta}{2 l_{12}} \qquad \frac{\partial \bar{v}_2}{\partial u_1 \partial v_1} = -\sin\beta_0 \frac{\sin 2\beta}{2 l_{12}}$$

$$\frac{\partial \bar{u}_2}{\partial u_1 \partial u_2} = -\cos\beta_0 \, \frac{\sin^2\beta}{l_{12}^3}$$

$$\frac{\partial \bar{u}_2}{\partial u_1 \partial v_2} = \cos\beta_0 \, \frac{\sin 2\beta}{2 l_{12}^3}$$

$$\frac{\partial \bar{u}_2}{\partial u_1 \partial u_3} = \frac{\partial \bar{u}_2}{\partial u_1 \partial v_3} = 0$$

$$\frac{\partial \bar{v}_2}{\partial u_1 \partial u_2} = -\sin\beta_0 \, \frac{\sin^2\beta}{l_{12}^3}$$

$$\frac{\partial \bar{v}_2}{\partial u_1 \partial v_2} = \sin\beta_0 \, \frac{\sin 2\beta}{2 l_{12}^3}$$

$$\frac{\partial \bar{v}_2}{\partial u_1 \partial u_3} = \frac{\partial \bar{v}_2}{\partial u_1 \partial v_3} = 0$$

$$\frac{\partial \bar{u}_2}{\partial v_1 \partial v_1} = \cos\beta_0 \, \frac{\cos^2\beta}{l_{12}^3}$$

$$\frac{\partial \bar{u}_2}{\partial v_1 \partial u_2} = \cos\beta_0 \, \frac{\sin 2\beta}{2 l_{12}^3}$$

$$\frac{\partial \bar{u}_2}{\partial v_1 \partial v_2} = -\cos\beta_0 \, \frac{\cos^2\beta}{l_{12}^3}$$

$$\frac{\partial \bar{u}_2}{\partial v_1 \partial u_3} = \frac{\partial \bar{u}_2}{\partial v_1 \partial v_3} = 0$$

$$\frac{\partial \bar{v}_2}{\partial v_1 \partial v_1} = \sin\beta_0 \, \frac{\cos^2\beta}{l_{12}^3}$$

$$\frac{\partial \bar{v}_2}{\partial v_1 \partial u_2} = \sin\beta_0 \, \frac{\sin 2\beta}{2 l_{12}^3}$$

$$\frac{\partial \bar{v}_2}{\partial v_1 \partial v_2} = -\sin\beta_0 \, \frac{\cos^2\beta}{l_{12}^3}$$

$$\frac{\partial \bar{v}_2}{\partial v_1 \partial u_3} = \frac{\partial \bar{v}_2}{\partial v_1 \partial v_3} = 0$$

$$\frac{\partial \bar{u}_2}{\partial u_2 \partial u_2} = \cos\beta_0 \, \frac{\sin^2\beta}{l_{12}^3}$$

$$\frac{\partial \bar{u}_2}{\partial u_2 \partial v_2} = -\cos\beta_0 \, \frac{\sin 2\beta}{2 l_{12}^3}$$

$$\frac{\partial \bar{u}_2}{\partial u_2 \partial u_3} = \frac{\partial \bar{u}_2}{\partial u_2 \partial v_3} = 0$$

$$\frac{\partial \bar{v}_2}{\partial u_2 \partial u_2} = \sin\beta_0 \, \frac{\sin^2\beta}{l_{12}^3}$$

$$\frac{\partial \bar{v}_2}{\partial u_2 \partial v_2} = -\sin\beta_0 \, \frac{\sin 2\beta}{2 l_{12}^3}$$

$$\frac{\partial \bar{v}_2}{\partial u_2 \partial u_3} = \frac{\partial \bar{v}_2}{\partial u_2 \partial v_3} = 0$$

$$\frac{\partial \bar{u}_2}{\partial v_2 \partial v_2} = \cos\beta_0 \, \frac{\cos^2\beta}{l_{12}^3}$$

$$\frac{\partial \bar{u}_2}{\partial v_2 \partial u_3} = \frac{\partial \bar{u}_2}{\partial v_2 \partial v_3} = 0$$

$$\frac{\partial \bar{v}_2}{\partial v_2 \partial v_2} = \sin\beta_0 \, \frac{\cos^2\beta}{l_{12}^3}$$

$$\frac{\partial \bar{v}_2}{\partial v_2 \partial u_3} = \frac{\partial \bar{v}_2}{\partial v_2 \partial v_3} = 0$$

$$\frac{\partial \bar u_2}{\partial u_3 \partial u_3} = \frac{\partial \bar u_2}{\partial u_3 \partial v_3} = \frac{\partial \bar u_2}{\partial v_3 \partial v_3} = 0 \qquad \frac{\partial \bar v_2}{\partial u_3 \partial u_3} = \frac{\partial \bar v_2}{\partial u_3 \partial v_3} = \frac{\partial \bar v_2}{\partial v_3 \partial v_3} = 0$$

Voor de verplaatsingscomponent $\bar u_3$ volgen de de hulp-differentiatieschema:

$$\frac{\partial \bar u_3}{\partial u_1 \partial u_1} = \cos(\gamma-\alpha)\left[\frac{2\sin\gamma\sin\beta}{l_{12}'} - \frac{\sin^2\beta\, l_{13}'}{l_{12}'^2}\right] + \sin(\gamma-\alpha)\left[-\frac{\cos\gamma\sin\beta}{l_{12}'} + \frac{\sin 2\beta\, l_{13}'}{2 l_{12}'^2}\right]$$

$$\frac{\partial \bar u_3}{\partial v_1 \partial u_1} = \cos(\gamma-\alpha)\left[-\frac{\sin\gamma\cos\beta}{l_{12}'} - \frac{\sin\beta\cos\gamma}{l_{12}'} + \frac{\sin 2\beta\, l_{13}'}{2 l_{12}'^2}\right]$$

$$+ \sin(\gamma-\alpha)\left[-\frac{\sin\gamma\sin\beta}{l_{12}'} + \frac{\sin^2\beta\, l_{13}'}{l_{12}'^2}\right]$$

$$\frac{\partial \bar u_3}{\partial u_2 \partial u_1} = \cos(\gamma-\alpha)\left[-\frac{\sin\beta\sin\gamma}{l_{12}'} + \frac{\sin^2\beta\, l_{13}'}{l_{12}'^2}\right] - \sin(\gamma-\alpha)\cdot\frac{\sin 2\beta\, l_{13}'}{2 l_{12}'^2}$$

$$\frac{\partial \bar u_3}{\partial v_2 \partial u_1} = \cos(\gamma-\alpha)\left[\frac{\cos\beta\sin\gamma}{l_{12}'} - \frac{\sin 2\beta\, l_{13}'}{2 l_{12}'^2}\right] - \sin(\gamma-\alpha)\cdot\frac{\sin^2\beta\, l_{13}'}{l_{12}'^2}$$

$$\frac{\partial \bar u_3}{\partial u_3 \partial u_1} = \cos(\gamma-\alpha)\left[-\frac{\sin\gamma\sin\beta}{l_{12}'}\right] + \sin(\gamma-\alpha)\cdot\frac{\cos\gamma\sin\beta}{l_{12}'}$$

$$\frac{\partial \bar u_3}{\partial v_3 \partial u_1} = \cos(\gamma-\alpha)\cdot\frac{\cos\gamma\sin\beta}{l_{12}'} + \sin(\gamma-\alpha)\,\frac{\sin\beta\sin\gamma}{l_{12}'}$$

$$\frac{\partial \bar u_3}{\partial v_1 \partial v_1} = \cos(\gamma-\alpha)\left[\frac{2\cos\gamma\cos\beta}{l_{12}'} - \frac{\cos^2\beta\, l_{13}'}{l_{12}'^2}\right] + \sin(\gamma-\alpha)\left[\frac{\sin\gamma\cos\beta}{l_{12}'} - \frac{\sin 2\beta\, l_{13}'}{2 l_{12}'^2}\right]$$

$$\frac{\partial \bar u_3}{\partial v_1 \partial u_2} = \cos(\gamma-\alpha)\left[\frac{\sin\beta\cos\gamma}{l_{12}'} - \frac{\sin 2\beta\, l_{13}'}{2 l_{12}'^2}\right] + \sin(\gamma-\alpha)\cdot\frac{\cos^2\beta\, l_{13}'}{l_{12}'^2}$$

$$\frac{\partial \bar u_3}{\partial v_1 \partial v_2} = \cos(\gamma-\alpha)\left[-\frac{\cos\beta\cos\gamma}{l_{12}'} + \frac{\cos^2\beta\, l_{13}'}{l_{12}'^2}\right] + \sin(\gamma-\alpha)\,\frac{\sin 2\beta\, l_{13}'}{2 l_{12}'^2}$$

$$\frac{\partial \bar{u}_3}{\partial u_1 \partial u_3} = \cos(\gamma-\alpha)\left[\frac{\sin\gamma \cos\beta}{l_{12}^{3}}\right] - \sin(\gamma-\alpha)\cdot \frac{\cos\gamma \cos\beta}{l_{12}^{3}}$$

$$\frac{\partial \bar{u}_3}{\partial v_1 \partial v_3} = \cos(\gamma-\alpha)\left[-\frac{\cos\gamma \cos\beta}{l_{12}^{3}}\right] + \sin(\gamma-\alpha)\left[-\frac{\sin\gamma \cos\beta}{l_{12}^{3}}\right]$$

$$\frac{\partial \bar{u}_3}{\partial u_2 \partial u_2} = \cos(\gamma-\alpha)\left[-\frac{\sin^2\beta \, l_{13}^{3}}{l_{12}^{3}}\right] + \sin(\gamma-\alpha)\frac{\sin 2\beta \, l_{13}^{3}}{2 l_{12}^{32}}$$

$$\frac{\partial \bar{u}_3}{\partial u_2 \partial v_2} = \cos(\gamma-\alpha)\frac{\sin 2\beta \, l_{13}^{3}}{2 l_{12}^{32}} + \sin(\gamma-\alpha)\frac{\sin^2\beta \, l_{13}^{3}}{l_{12}^{32}}$$

$$\frac{\partial \bar{u}_3}{\partial u_2 \partial u_3} = \cos(\gamma-\alpha)\frac{\sin\gamma \sin\beta}{l_{12}^{3}} - \sin(\gamma-\alpha)\frac{\cos\gamma \sin\beta}{l_{12}^{3}}$$

$$\frac{\partial \bar{u}_3}{\partial u_2 \partial v_3} = -\cos(\gamma-\alpha)\frac{\cos\gamma \sin\beta}{l_{12}^{3}} - \sin(\gamma-\alpha)\frac{\sin\gamma \sin\beta}{l_{12}^{3}}$$

$$\frac{\partial \bar{u}_3}{\partial v_2 \partial v_2} = \cos(\gamma-\alpha)\left[-\frac{\cos^2\beta \, l_{13}^{3}}{l_{12}^{32}}\right] - \sin(\gamma-\alpha)\frac{\sin 2\beta \, l_{13}^{3}}{2 l_{12}^{32}}$$

$$\frac{\partial \bar{u}_3}{\partial v_2 \partial u_3} = \cos(\gamma-\alpha)\left[-\frac{\sin\gamma \cos\beta}{l_{12}^{3}}\right] + \sin(\gamma-\alpha)\frac{\cos\gamma \cos\beta}{l_{12}^{3}}$$

$$\frac{\partial \bar{u}_3}{\partial v_2 \partial v_3} = \cos(\gamma-\alpha)\frac{\cos\gamma \cos\beta}{l_{12}^{3}} + \sin(\gamma-\alpha)\frac{\sin\gamma \cos\beta}{l_{12}^{3}}$$

$$\frac{\partial \bar{u}_3}{\partial u_3 \partial u_3} = \frac{\partial \bar{u}_3}{\partial u_3 \partial v_3} = 0$$

$$\frac{\partial \bar{u}_3}{\partial v_3 \partial v_3} = 0$$

Voor de verplaatsingscomponent $\bar{v}_3$ volgen de hulpdifferentiaties hierna:

$$\frac{\partial \bar{v}_3}{\partial u_1 \partial u_1} = \cos(\gamma-\alpha)\left[\frac{\cos\gamma \sin\beta}{l_{12}'} - \frac{\sin 2\beta\, l_{13}'}{2 l_{12}'^{2}}\right] + \sin(\gamma-\alpha)\left[\frac{2\sin\gamma \sin\beta}{l_{12}'} - \frac{\sin^2\beta\, l_{13}'}{l_{12}'^{2}}\right]$$

$$\frac{\partial \bar{v}_3}{\partial u_1 \partial v_1} = \cos(\gamma-\alpha)\left[\frac{\sin\beta \sin\gamma}{l_{12}'} - \frac{\sin^2\beta\, l_{13}'}{l_{12}'^{2}}\right] +$$

$$\sin(\gamma-\alpha)\left[-\frac{\sin\gamma \cos\beta}{l_{12}'} - \frac{\sin\beta \cos\gamma}{l_{12}'} + \frac{\sin 2\beta\, l_{13}'}{2 l_{12}'^{2}}\right]$$

$$\frac{\partial \bar{v}_3}{\partial u_1 \partial u_2} = \cos(\gamma-\alpha)\,\frac{\sin 2\beta\, l_{13}'}{2 l_{12}'^{2}} + \sin(\gamma-\alpha)\left[-\frac{\sin\beta \sin\gamma}{l_{12}'} + \frac{\sin^2\beta\, l_{13}'}{l_{12}'^{2}}\right]$$

$$\frac{\partial \bar{v}_3}{\partial u_1 \partial v_2} = \cos(\gamma-\alpha)\,\frac{\sin^2\beta\, l_{13}'}{l_{12}'^{2}} + \sin(\gamma-\alpha)\left[\frac{\cos\beta \sin\gamma}{l_{12}'} - \frac{\sin 2\beta\, l_{13}'}{2 l_{12}'^{2}}\right]$$

$$\frac{\partial \bar{v}_3}{\partial u_1 \partial u_3} = \cos(\gamma-\alpha)\left[-\frac{\cos\gamma \sin\beta}{l_{12}'}\right] + \sin(\gamma-\alpha)\left[-\frac{\sin\gamma \sin\beta}{l_{12}'}\right]$$

$$\frac{\partial \bar{v}_3}{\partial u_1 \partial v_3} = \cos(\gamma-\alpha)\left[-\frac{\sin\gamma \sin\beta}{l_{12}'}\right] + \sin(\gamma-\alpha)\,\frac{\cos\gamma \sin\beta}{l_{12}'}$$

$$\frac{\partial \bar{v}_3}{\partial v_1 \partial v_1} = \cos(\gamma-\alpha)\left[-\frac{\sin\gamma \cos\beta}{l_{12}'} + \frac{\sin 2\beta}{2 l_{12}'^{2}}\, l_{13}'\right] +$$

$$\sin(\gamma-\alpha)\left[\frac{2\cos\gamma \cos\beta}{l_{12}'} - \frac{\cos^2\beta\, l_{13}'}{l_{12}'^{2}}\right]$$

$$\frac{\partial \bar{v}_3}{\partial v_1 \partial u_2} = \cos(\gamma-\alpha)\left[-\frac{\cos^2\beta\, l_{13}'}{l_{12}'^{2}}\right] + \sin(\gamma-\alpha)\left[\frac{\sin\beta \cos\gamma}{l_{12}'} - \frac{\sin 2\beta\, l_{13}'}{2 l_{12}'^{2}}\right]$$

$$\frac{\partial \bar{v}_3}{\partial v_1 \partial v_2} = \cos(\gamma-\alpha)\left[-\frac{\sin 2\beta\, l_{13}'}{2 l_{12}'^{2}}\right] + \sin(\gamma-\alpha)\left[-\frac{\cos\beta \cos\gamma}{l_{12}'} + \frac{\cos^2\beta\, l_{13}'}{l_{12}'^{2}}\right]$$

$$\frac{\partial \bar{v}_3}{\partial v_1 \partial u_3} = \cos(\gamma-\alpha)\,\frac{\cos\gamma \cos\beta}{l_{12}'} + \sin(\gamma-\alpha)\left[\frac{\sin\gamma \cos\beta}{l_{12}'}\right]$$

$$\frac{\partial \bar{v}_3}{\partial v_1 \partial v_3} = \cos(\gamma-\alpha)\,\frac{\sin\gamma\cos\beta}{l_{12}^{3}} + \sin(\gamma-\alpha)\left[-\frac{\cos\gamma\cos\beta}{l_{12}^{3}}\right]$$

$$\frac{\partial \bar{v}_3}{\partial u_2 \partial u_2} = \cos(\gamma-\alpha)\left[-\frac{\sin 2\beta\, l_{13}^{3}}{2 l_{12}^{32}}\right] + \sin(\gamma-\alpha)\left[-\frac{\sin^2\beta\, l_{13}^{3}}{l_{12}^{32}}\right]$$

$$\frac{\partial \bar{v}_3}{\partial u_2 \partial v_2} = \cos(\gamma-\alpha)\left[-\frac{\sin^2\beta\, l_{13}^{3}}{l_{12}^{32}}\right] + \sin(\gamma-\alpha)\,\frac{\sin 2\beta\, l_{13}^{3}}{2 l_{12}^{32}}$$

$$\frac{\partial \bar{v}_3}{\partial u_2 \partial u_3} = \cos(\gamma-\alpha)\,\frac{\cos\gamma\sin\beta}{l_{12}^{3}} + \sin(\gamma-\alpha)\,\frac{\sin\gamma\sin\beta}{l_{12}^{3}}$$

$$\frac{\partial \bar{v}_3}{\partial u_2 \partial v_3} = \cos(\gamma-\alpha)\,\frac{\sin\gamma\sin\beta}{l_{12}^{3}} + \sin(\gamma-\alpha)\left[-\frac{\cos\gamma\sin\beta}{l_{12}^{3}}\right]$$

$$\frac{\partial \bar{v}_3}{\partial v_2 \partial v_2} = \cos(\gamma-\alpha)\,\frac{\sin 2\beta\, l_{13}^{3}}{2 l_{12}^{32}} + \sin(\gamma-\alpha)\left[-\frac{\cos^2\beta\, l_{13}^{3}}{l_{12}^{32}}\right]$$

$$\frac{\partial \bar{v}_3}{\partial v_2 \partial u_3} = \cos(\gamma-\alpha)\left[-\frac{\cos\gamma\cos\beta}{l_{12}^{3}}\right] + \sin(\gamma-\alpha)\left[-\frac{\sin\gamma\cos\beta}{l_{12}^{3}}\right]$$

$$\frac{\partial \bar{v}_3}{\partial v_2 \partial v_3} = \cos(\gamma-\alpha)\left[-\frac{\sin\gamma\cos\beta}{l_{12}^{3}}\right] + \sin(\gamma-\alpha)\left[\frac{\cos\gamma\cos\beta}{l_{12}^{3}}\right]$$

$$\frac{\partial \bar{v}_3}{\partial u_3 \partial u_3} = \frac{\partial \bar{v}_3}{\partial u_3 \partial v_3} = 0$$

$$\frac{\partial \bar{v}_3}{\partial v_3 \partial v_3} = 0$$

Met behulp van formule (VIII 4) zijn we nu in staat om de matrices $\frac{\partial^2 \varepsilon_x}{\partial u_q \partial u_p}$, $\frac{\partial^2 \varepsilon_y}{\partial u_q \partial u_p}$ en $\frac{\partial^2 \gamma_{xy}}{\partial u_q \partial u_p}$ te vullen en wel op de volgende manier:

$$\frac{\partial^2 \varepsilon_x}{\partial u_q \partial u_p} = \frac{1}{A}\left[(y_3 - y_1)\frac{\partial^2 \bar{u}_2}{\partial u_q \partial u_p} + (y_1 - y_2)\frac{\partial^2 \bar{u}_3}{\partial u_q \partial u_p}\right]$$

$$\frac{\partial^2 \varepsilon_y}{\partial u_q \partial u_p} = \frac{1}{A}\left[(x_1 - x_3)\frac{\partial^2 \bar{v}_2}{\partial u_q \partial u_p} + (x_2 - x_1)\frac{\partial^2 \bar{v}_3}{\partial u_q \partial u_p}\right]$$

$$\frac{\partial^2 \varepsilon_z}{\partial u_q \partial u_p} = 0$$

$$\frac{\partial^2 \gamma_{xy}}{\partial u_q \partial u_p} = \frac{1}{A}\left[(x_1 - x_3)\frac{\partial^2 \bar{u}_2}{\partial u_q \partial u_p} + (x_2 - x_1)\frac{\partial^2 \bar{u}_3}{\partial u_q \partial u_p} + (y_3 - y_1)\frac{\partial^2 \bar{v}_2}{\partial u_q \partial u_p} + (y_1 - y_2)\frac{\partial^2 \bar{v}_3}{\partial u_q \partial u_p}\right]$$

De opbouw van de hiervoor genoemde matrices is voor elk gelijk, zodat we het alleen voor $\frac{\partial^2 \varepsilon_x}{\partial u_q \partial u_p}$ demonstreren:

$$\frac{\partial^2 \varepsilon_x}{\partial u_q \partial u_p} = \begin{bmatrix} \frac{\partial^2 \varepsilon_x}{\partial u_1 \partial u_1} & \frac{\partial^2 \varepsilon_x}{\partial v_1 \partial u_1} & \frac{\partial^2 \varepsilon_x}{\partial u_2 \partial u_1} & \frac{\partial^2 \varepsilon_x}{\partial v_2 \partial u_1} & \frac{\partial^2 \varepsilon_x}{\partial u_3 \partial u_1} & \frac{\partial^2 \varepsilon_x}{\partial v_3 \partial u_1} \\ & \frac{\partial^2 \varepsilon_x}{\partial v_1 \partial v_1} & \frac{\partial^2 \varepsilon_x}{\partial u_2 \partial v_1} & \frac{\partial^2 \varepsilon_x}{\partial v_2 \partial v_1} & \frac{\partial^2 \varepsilon_x}{\partial u_3 \partial v_1} & \frac{\partial^2 \varepsilon_x}{\partial v_3 \partial v_1} \\ & & \frac{\partial^2 \varepsilon_x}{\partial u_2 \partial u_2} & \frac{\partial^2 \varepsilon_x}{\partial v_2 \partial u_2} & \frac{\partial^2 \varepsilon_x}{\partial u_3 \partial u_2} & \frac{\partial^2 \varepsilon_x}{\partial v_3 \partial u_2} \\ & & & \frac{\partial^2 \varepsilon_x}{\partial v_2 \partial v_2} & \frac{\partial^2 \varepsilon_x}{\partial u_3 \partial v_2} & \frac{\partial^2 \varepsilon_x}{\partial v_3 \partial v_2} \\ & \text{sym.} & & & \frac{\partial^2 \varepsilon_x}{\partial u_3 \partial u_3} & \frac{\partial^2 \varepsilon_x}{\partial v_3 \partial u_3} \\ & & & & & \frac{\partial^2 \varepsilon_x}{\partial v_3 \partial v_3} \end{bmatrix}$$

Geheel analoge structuur voor $\frac{\partial^2 \varepsilon_y}{\partial u_q \partial u_p}$ en $\frac{\partial^2 \gamma_{xy}}{\partial u_q \partial u_p}$

APPENDIX VI

COMPUTER PROGRAMMA'S

```
Lalgol 05064608 Beijers
begin comment Berekening van de veerkarakteristiek met behulp van de elementenmethode,
            waarin het criterium van grote verplaatsingen is gebruikt;

      procedure STYFEL(E, I, A, l, S);
      value E, I, l, A; real E, I, l, A; array S;
      begin S[1, 1] := E × A/l; S[1, 2] := S[1, 3] := S[2, 1] := S[3, 1] := 0;
            S[2, 2] := S[3, 3] := 4 × E × I/l; S[2, 3] := S[3, 2] := - 2 × E × I/l;
      end STYFEL;

      procedure HULP( a, b, c, d, e, f, LEN, st, en, le, KN);
      value en, st; integer en, st; real a, b, c, d, e, f; integer array le;  array LEN, KN ;
      begin integer i, j; real x1, x2, z1, z2, l;
            i := le [en,1]; j:= le[en,2];
            x1 := KN[i,1]; x2 := KN[j,1]; z1 := KN[i,2]; z2 := KN[j,2];
            if st = 1 then begin l := ((z2 - z1) × (z2 - z1) + (x2 - x1) × (x2 - x1)) ↑ 0.5;
                                 LEN[en, 1] := LEN[en, 2] := l;
                           end;
            if st ≠ 1 then begin l := LEN[en, 2] end;
            a := (z2 - z1)/l; b := (x2 -x1)/l; c := a × a; d := b × b; e := 2 × a × b; f := d - c;
      end HULP;

      procedure BELASTINGVEKTOR(T, lnp, f);
      value T; integer T; integer array lnp; array f;
      begin integer i, j, ab, n, ae; real q;
            for i := 1 step 1 until T do f[i] := 0; ab := read; for i := 1 step 1 until ab do
            begin n := read; NLCR; PRINTTEXT(⟨knooppunt=⟩); ABSFIXT(3, 0, n); ae:= read;
                  for j := 1 step 1 until ae do begin q := f[lnp[n, read]] := read; SPACE(5); FLOT(5, 3, q);
                                                end;
            end
      end BELASTINGSVEKTOR;

      procedure ONDERSTEUNING(N, T, lnp);
      value N; integer N, T; integer array lnp;
      begin integer n, v, a0, i, j, av, t;
            t := 0; for n := 1 step 1 until N do for v := 1, 2, 3 do lnp[n, v] := 1;
            a0 := READ; for i := 1 step 1 until a0 do
            begin n := READ; av := READ; for j := 1 step 1 until av do
                  begin v := READ; lnp[n, v] := 0;
                  end
            end inlezen voorgeschreven verplaatsingen;
```

```
        for n := 1 step 1 until N do for v := 1, 2, 3 do if lnp[n, v] = 1 then
        begin t := t + 1;
              lnp[n, v] := t;
        end;
        T := t;
end bepaling lnp vektor met T vrijheidsgraden;
procedure INLE(Ne, le);
value Ne; integer Ne; integer array le;
begin integer k;
      for k := 1 step 1 until Ne do
      begin le[k, 1] := READ; le[k, 2] := READ;
      end
end inlezen le vektor;

procedure GOM(l, sigma1, sigma2, sigma3, G1, G2, G3, G, a, b, c, d, e, f);
value l, sigma1, sigma2, sigma3, a, b, c, d, e, f; real l, sigma1, sigma2, sigma3, a, b, c, d, e, f;
array G1, G2, G3, G;
begin integer i, j; real hulp1, hulp2, hulp3;
      for i := 1 step 1 until 6 do for j := 1 step 1 until 6 do G1[i, j] := G2[i, j] := G3[i, j] := G[i, j] := 0;
      G1[1, 1] := sigma1 × 6/(5 × l) × c; G1[4, 4] := G1[1, 1]; G1[1, 4] := − G1[1, 1];
      G1[1, 2] := − sigma1 × 3/(5 × l) × e; G1[1, 5] := G1[2, 4] := − G1[1, 2]; G1[4, 5] := G1[1, 2];
      G1[1, 3] := 0.1 × sigma1 × a; G1[1, 6] := G1[1, 3]; G1[3, 4] := G1[4, 6] := − G1[1, 3];
      G1[2, 2] := sigma1 × 6/(5 × l) × d; G1[5, 5] := G1[2, 2]; G1[2, 5] := − G1[2, 2];
      G1[2, 3] := − 0.1 × sigma1 × b; G1[2, 6] := G1[2, 3]; G1[5, 6] := G1[3, 5] := − G1[2, 3];
      G1[3, 3] := sigma1 × 4 × l/30; G1[6, 6] := G1[3, 3]; G1[3, 6] := − G1[3, 3]/4;
      hulp2 := l/(l × l) × sigma2;
      G2[1, 1] := − hulp2 × e; G2[1, 4] := G2[2, 2] := G2[5, 5] := − G2[1, 1]; G2[2,5] := G2[4, 4] := G2[1, 1];
      G2[1, 2] := hulp2 × f; G2[4, 5] := G2[1, 2]; G2[1, 5] := G2[2, 4] := − G2[1, 2];
      hulp3 := l/(l × l) × sigma3;
      G3[1, 1] := hulp3 × e; G3[1, 4] := G3[2, 2] := G3[5, 5] := − G3[1, 1]; G3[2,5] := G3[4, 4] := G3[1, 1];
      G3[1, 2] := − hulp3 × f; G3[4, 5] := G3[1, 2]; G3[1, 5] := G3[2, 4] := − G3[1, 2];
      for i := 1 step 1 until 6 do for j := i step 1 until 6 do G[i, j] := G1[i, j] + G2[i, j] + G3[i, j];
      for i := 2 step 1 until 6 do for j := 1 step 1 until i − 1 do G[i, j] := G[j, i];
end GOMETRISCHE MATRIX;

procedure ORDEEPS(a, b, l, eps1, eps2, eps3, Oeps);
value a, b, l, eps1, eps2, eps3; real a, b, l, eps1, eps2, eps3; array Oeps;
begin integer i, j;
      for i := 1 step 1 until 3 do for j := 1 step 1 until 6 do Oeps[i, j] := 0;
      Oeps[1, 1] := 0.1 × a × (− eps2 + eps3); Oeps[1, 4] := − Oeps[1, 1];
      Oeps[1, 2] := − 0.1 × b × (− eps2 + eps3); Oeps[1, 5] := − Oeps[1, 2];
       Oeps[1, 3] := l/30 × (− 4 × eps2 − eps3); Oeps[1, 6] := l/30 × (eps2 + 4 × eps3);
     for i := 1,2 do for j := 1 step 1 until 6 do Oeps[i+1,j] := 0 ;
```

```
end ORDEEPS;

procedure DEPSDU(a, b, l, Q1, Q1T);
value a, l, b; real a, b, l; array Q1, Q1T;
begin integer i, j; real hulp;
      for i := 1 step 1 until 3 do for j := 1 step 1 until 6 do Q1[i, j] := 0;
      hulp := 1/l;
      Q1[1, 1] := - b; Q1[1, 4] := - Q1[1, 1]; Q1[1, 2] := - a; Q1[1, 5] := - Q1[1, 2];
      Q1[2, 1] := - a × hulp; Q1[2, 2] := hulp × b; Q1[2, 4] := Q1[3, 1] := - Q1[2, 1] ; Q1[3, 4] := Q1[2, 1];
      Q1[2, 5] := Q1[3, 2] := - Q1[2, 2]; Q1[3, 5] := Q1[2, 2]; Q1[2, 3] := - 1; Q1[3, 6] := 1;
      for i := 1 step 1 until 3 do for j := 1 step 1 until 6 do Q1T[j, i] := Q1[i, j];
end DEPSDU;

procedure VERMENIGVULDIGING(A, B, C, D, DT);
array A, B, C, D, DT;
begin integer i, j, k; array HU[1:6, 1:3];
      for i := 1 step 1 until 6 do for j := 1 step 1 until 3 do HU[i, j] := 0;
      for i := 1 step 1 until 6 do for j := 1 step 1 until 6 do D[i, j] := DT[i, j] := 0;
      for i := 1 step 1 until 6 do for j := 1 step 1 until 3 do
      begin for k := 1 step 1 until 3 do HU[i, j] := HU[i, j] + A[i, k] × B[k, j]; end;
      for i := 1 step 1 until 6 do for j := 1 step 1 until 6 do
      begin for k := 1 step 1 until 3 do D[i, j] := D[i, j] + HU[i, k] × C[k, j]; end;
      for i := 1 step 1 until 6 do for j := 1 step 1 until 6 do DT[i, j] := D[j, i];
end VERMENIGVULDIGING;

procedure OPTELLING(A, B, C, D, Qtot);
array A, B, C, D, Qtot;
begin integer i, j;
      for i := 1 step 1 until 6 do for j := 1 step 1 until 6 do Qtot[i, j] := 0;
      for i := 1 step 1 until 6 do for j := 1 step 1 until 6 do Qtot[i, j] := A[i, j] + B[i, j] + C[i, j] + D[i, j];
end OPTELLING;

procedure QKONSTRUKTIE(en, Qtot, le, lnp, Qkon);
value en; integer en; array Qtot, le, lnp, Qkon;
begin integer s1, s2, v1, v2, l1, l2, i;
      for s1:= 1, 2 do for v1 := 1, 2, 3 do
      begin i := lnp[le[en, s1], v1]; l1 := (s1 - 1) × 3 + v1; if i > 0 then
            begin for s2 := 1, 2 do for v2 := 1, 2, 3 do
                  begin j := lnp[le[en, s2], v2]; l2 := (s2 - 1) × 3 + v2;
                        if j ≥ i then Qkon[i, j - i] := Qkon[i, j - i] + Qtot[l1, l2];
                  end
            end
```

```
            end
    end QKONSTRUKTIE;

    procedure CHOLBD(n, m, dec, a, b, fail);
    value n, m, dec; integer n, m, dec; array a, b; label fail;
    begin integer k, m1, i, j;
            m1 := m;
            if dec = 0 ∨ dec = 2 ∨ dec = 4 ∨ dec = 6 then
                for k := 1 step 1 until n do
                begin if a[k, 0] ≤ 0 then goto fail;
                        a[k, 0] := sqrt(a[k, 0]);
                        if m > n - k then m1 := n - k;
                        for i := 1 step 1 until m1 do a[k, i] := a[k, i]/a[k, 0];
                        for i := m1 + 1 step 1 until m do a[k, i] := 0;
                        for j := 1 step 1 until m1 do for i := 0 step 1 until m1 - j do a[k + j, i] := a[k + j, i] - a[k,j] × a[k, j +i];
                end k;
            m1 := m;
            if dec = 0 ∨ dec = 1 ∨ dec = 4 ∨ dec = 5 then
                for k := 1 step 1 until n do
                begin b[k] := b[k]/a[k, 0];
                        if m > n - k then m1 := n - k;
                        for j := 1 step 1 until m1 do b[k + j] := b[k + j] - a[k, j] × b[k]
                end k;
            m1 := m;
            if dec < 4 then
                for k := n step - 1 until 1 do
                begin b[k] := b[k]/a[k, 0];
                        if k < m + 1 then m1 := k - 1;
                        for j := 1 step 1 until m1 do b[k - j] := b[k - j] - a[k - j, j] × b[k]
                end k
    end CHOLBD;

    procedure UITVOERFFTOT(N, fftot, lnp);
    valueN; integer N; array fftot, lnp;
    begin integer i, j, k, a; real array mn[1:3];
            for i := 1 step 1 until N do
            begin for j := 1,2,3 do
                    begin a := lnp[i,j]; mn[j] :=0;
                            if a ≠ 0 then mn[j] :=fftot[a];
                    end;
                    if mn[1] ≠ 0 ∨ mn[2] ≠ 0 ∨mn[3] ≠0 then
                    begin NLCR; PRINTTEXT(≮ knooppunt = ≯); ABSFIXT(3,0,i);
                            for k := 1 step 1 until 3 do
```

```
                    begin SPACE(5); FLOT(5,3,mn[k]) end;
                end;
            end;
        endUITFFTOT;

        procedure UITVOER KN SI R(KN, SI, R, Ne, k, st, N);
        value N, Ne, st; integer N,Ne, st, k; array KN, SI, R;
        begin CARRIAGE(5); SPACE(10);
            PRINTTEXT(⟨knooppuntskoordinaten—elementnr—xkoordinaat—ykoordinaat na stap nr⟩);
            ABSFIXT(3, 0, st); CARRIAGE(2);
            UITVOER(N, 2, KN);
            CARRIAGE(5); SPACE(10);
            PRINTTEXT(⟨de krachten in de elementen—elementnr—N—M1—M2 na stap nr⟩);
            ABSFIXT(3, 0, st); CARRIAGE(2); UITVOER(Ne, 3, SI);
            CARRIAGE(5); SPACE(10);
            PRINTTEXT(⟨de rekken van de elementen—elementnr—eps1—eps2—eps3 na stap nr⟩);
            ABSFIXT(3, 0, st); CARRIAGE(2); UITVOER(Ne, 3, R);
        end UITVOER KN SI R;

        procedure UITGANGSHOEK(en, KN, le, lnp, HOEK);
        value en; integer en; array KN, le, lnp, HOEK;
        begin real x1, x2, z1, z2, hulp2;
            x1 := KN[le[en, 1], 1]; x2 := KN[le[en, 2], 1];
            z1 := KN[le[en, 1], 2]; z2 := KN[le[en, 2], 2];
            if x2 > x1 then hulp2 := arctan((z2 - z1)/(x2 - x1));
            if x2 < x1 then hulp2 := arctan((z2 - z1)/(x2 - x1)) + pi;
            if x2 = x1 ∧ z2 > z1 then hulp2 := pi/2;
            if x2 = x1 ∧ z2 < z1 then hulp2 := - pi/2;
            HOEK[en] := hulp2;
        end UITGANGSHOEK;

        procedure REKSPANITTERATIE(en, ue, R, KN, KNH, LEN, le, lnp, u, SI, S, E, I, A, PSI, PSIH, HOEK);
        value en, E, I, A; integer en; real E, I, A; array ue, R, KN, KNH, LEN, le, lnp, u, SI, S,PSI, PSIH, HOEK;
        begin integer i, j, l1,l2; real x1, x2, z1, z2, psi1, psi2, hulp1, hulp2,hulp3, hulp4, hulp5, pi; array eps[1:3];
            pi := 3.14159265358;
            x1 := KN[le[en, 1], 1]; x2 := KN[le[en, 2], 1]; psi1 := PSI[ en,1];
            z1 := KN[le[en, 1], 2]; z2 := KN[le[en, 2], 2]; psi2 := PSI[ en,2];
            for i := 1, 2 do for j := 1, 2, 3 do
            begin l1 := (i - 1) × 3 + j; l2 := lnp[le[en, i], j];
                ue[l1] := if l2 = 0 then 0 else u[l2];
            end;
            hulp2 := (z2 - z1)/ LEN[en,2] ; hulp3 := (x2 - x1)/ LEN[en,2];
           x1 := x1 + ue[1]; KNH[le[en, 1], 1] := x1;
           x2 := x2 + ue[4]; KNH[le[en, 2], 1] := x2;
```

```
        z1 := z1 + ue[2]; KNH[le[en, 1], 2] := z1;
        z2 := z2 + ue[5]; KNH[le[en, 2], 2] := z2;
        psi1 := psi1 + ue[3]; PSI[en,1] := psi1;
        psi2 := psi2 + ue[6]; PSI[en,2] := psi2;
        hulp5 := ((x2 - x1) ⋀ 2 + (z2 - z1) ⋀ 2) ⋀ 0.5;
        hulp1 := PSIH[en];
        x1 := hulp3 × ue[1] + hulp2 × ue[2];
        z1 := - hulp2 × ue[1] + hulp3 × ue[2];
        x2 := hulp3 × ue[4] + hulp2 × ue[5];
        z2 := - hulp2 × ue[4] + hulp3 × ue[5];
        x2 := x2 + LEN[en,2];
        LEN[en,2] := hulp5;
        if x2 > x1  then hulp4 := arctan((z2 - z1)/(x2 - x1));
        if x2 < x1 ∧ z2 > z1 then hulp4 := arctan((z2 - z1 )/(x2 -x1)) + pi;
        if x2 < x1 ∧ z2 < z1 then hulp4 := arctan((z2 - z1)/(x2 - x1)) - pi;
        if x2 = x1 ∧ z2 > z1 then hulp4 := pi/2;
        if x2 = x1 ∧ z2 < z1 then hulp4 := - pi / 2 ;
        hulp1 := hulp1 + hulp4; PSIH[en] := hulp1;
        eps[2]:= - psi1 + HOEK [en] - hulp1;
        eps[3] := psi2 - HOEK[en] + hulp1;
        eps[1] := hulp5× (1+ 1/30 × (2 × eps[2] × eps[2] + eps[2] × eps[3] + 2 × eps[3] × eps[3])) - LEN[en,1];
        for i := 1, 2, 3 do R[en, i] := eps[i];
        STYFEL(E, I, A, LEN[en,1], S);
        for i := 1, 2, 3 do SI[ en,i ] := 0 ;
        for i := 1, 2, 3 do for j := 1, 2, 3 do SI[en, i] := SI[en, i] + S[i, j] × R[en, j];
end REKSPANITTERATIE;

 procedure UITVOER(Ne, k, M);
 value Ne; integer Ne, k; array M;
 begin integer i, j;
        for i := 1 step 1 until Ne do
        begin CARRIAGE(2); ABSFIXT(3, 0, i);
              for j := 1 step 1 until k do
              begin SPACE(5); FLOT(5, 3, M[i, j]);
              end
        end
 end UITVOER;

 procedure VERMVEKTOR( SI, DEDU, vektor,  en); value en; integer en;
 array SI, DEDU, vektor;
 begin integer i, j;
        for i := 1 step 1 until 6 do vektor[i] := 0;
         for i := 1 step 1 until 6 do for j := 1, 2, 3 do vektor[i] := vektor[i] + SI[en,j] × DEDU[j,i];
      end VERMVEKTOR;
```

```
procedure RECHTERLID(vektor, rechter, en, le, lnp);
value en; integer en; array vektor, rechter, le, lnp;
begin integer i, j, l1, l2;
      for i := 1, 2 do for j := 1, 2, 3 do
      begin l1 := (i-1) × 3 + j; l2 := lnp[le[en, i], j];
            if l2 > 0 then rechter[l2] := rechter[l2] + vektor[l1];
      end;
end RECHTERLID;

procedure BANDBREEDTE(ae, ake, avk, lnp, le2, bb); value ae, ake, avk; integer ae, ake, avk, bb; integer array lnp, le2;
begin integer en, max, min, i, j, k;
      bb := 0;
      for en := 1 step 1 until ae do
      begin max := 0; min := ae × ake × avk;
            for i := 1 step 1 until ake do for j := 1 step 1 until avk do
            begin k := lnp[le2[en,i],j]; if k > 0 then
                  begin if k > max then max := k;
                        if k < min then min := k;
                  end
            end;
            if (max - min) > bb then bb := max - min;
      end;
      CARRIAGE(10); PRINTTEXT(⟨Bandbreedte stijfheidsmatrix:⟩); ABSFIXT(5,0,bb);
end BANDBREEDTE;


integer N, Ne, T, en, stap, uitvoer, test, i, j, st, sp, itteratie, cond;
real macheps, E, I, A, a, b, c, d, e, f, co, si, absol, procent, pi, nu, na, nain, fout, conditie;
real array S[1:3, 1:3], Q1, Qeps[1:3, 1:6], G1, G2, G3, Gtot, Qelm1, Qelm2, Qelm3, Qtot, W[1:6, 1:6], Q1T[1:6,1:3],
           ue[1:6], DEDU[1:3,1:6], vektor[1:6];
CARRIAGE(2); PRINTTEXT(⟨ De starttijd = ⟩); FLOT(5,3,TIME/100);
N := read; NLCR; PRINTTEXT(⟨aantal knooppunten=⟩); ABSFIXT(3, 0, N); CARRIAGE(2);
Ne := read;PRINTTEXT(⟨aantal elementen=⟩); ABSFIXT(3, 0, Ne); CARRIAGE(2);
E := read; PRINTTEXT(⟨elasticiteitsmodules=⟩); FLOT(5, 3, E); CARRIAGE(2);
I := read; PRINTTEXT(⟨oppervlaktetraagheidsmoment=⟩); FLOT(5, 3, I); CARRIAGE(2);
A := read; PRINTTEXT(⟨oppervlakte element=⟩); FLOT(5, 3, A); CARRIAGE(2);
absol := read; PRINTTEXT(⟨de afwijking t.o.v. 0 mag zijn in de belastingsvektor⟩); FLOT(5,3,absol); CARRIAGE(2);
procent := read ; PRINTTEXT(⟨de procentuele afwijking in de belastingsvektor mag zijn⟩); FLOT(5,3,procent); CARRIAGE(2);
macheps := 2 × ₁₀-12 ;
stap := read; PRINTTEXT(⟨aantal geschatte belastingsstappen voor de berekening=⟩); ABSFIXT(3, 0, stap);
cond := read; CARRIAGE(2); PRINTTEXT(⟨aantal toegelaten itteratiestappen=⟩); ABSFIXT(3,0,cond);
CARRIAGE(5);
begin integer bb; integer array lnp[1:N, 1:3], le[1:Ne, 1:2];
      real array R[1:Ne, 1:3], KNH, KNRE, KN[1:N, 1:2], LEN[1:Ne, 1:2], SI[1:Ne, 1:3],
```

```
            LENRE, PSIHRE, HOEK[1:Ne], PSIH[1:Ne], PSI,PSIRE[1:Ne,1:2];
for i := 1 step 1 until N do for j := 1 step 1 until 2 do KN[i, j] := read;
SPACE(10); PRINTTEXT(⟨de knooppuntskoordinaten in uitgangspositie—elementen—x koord—y koord⟩);
UITVOER(N, 2, KN);
for i := 1 step 1 until Ne do for j := 1, 2, 3 do R[i, j] := SI[i, j] := 0;
for i := 1 step 1 until Ne do for j := 1, 2 do LEN[i, j] := PSI[i,j] :=  0;
for i := 1 step 1 until Ne do HOEK[i] := PSIH[i]  := 0;
ONDERSTEUNING(N, T, lnp); CARRIAGE(5); PRINTTEXT(⟨de lnp—vektor van de constructie⟩);
UITVOER(N,3,lnp); CARRIAGE(5);
INLE(Ne, le);  PRINTTEXT(⟨de le—vektor van de constructie⟩); UITVOER(Ne,2,le); CARRIAGE(5);
for i := 1 step 1 until Ne do UITGANGSHOEK(i, KN, le, lnp, HOEK);
for i := 1 step 1 until Ne do PSIH[i] := HOEK[i];
BANDBREEDTE (Ne, 2, 3, lnp, le, bb); CARRIAGE(4);
itteratie := 0;
begin integer con, hu, teller, conhulp, controle; real array Qkon, Qkonhulp[1:T, 0:bb], ff, rechter, fftot, ffkon,
                                                  rechterlid, rechterlidhulp[1:T], u1,u2,deltau[1:T];

;
          hu:= 0; st:= 0;  con:= 1; teller:= 0; conhulp := 0;
          for i:= 1 step 1 until T do fftot[i]:= rechterlid[i]:= u2[i] := u1[i] := deltau[i] := 0;
          CARRIAGE(5); SPACE(5);
          PRINTTEXT(⟨de totale belastingsvektor van de construktie⟩);
          BELASTINGVEKTOR(T, lnp, ffkon);
          CARRIAGE(2); PRINTTEXT(⟨ de nodig voor verwerken invoergegevens⟩); FLOT(5,3,TIME/100);
          for i:= 1 step 1 until T do ff[i]:= ffkon[i]/stap;
VERDER:   for i:= 1 step 1 until T do rechter[i]:= 0;
          for i := 1 step 1 until 6 do for j := 1 step 1 until 6 do W[i,j] := 0 ;
          for i:= 1 step 1 until T do for j:= 0 step 1 until bb do Qkon[i,j]:= 0;
          CARRIAGE(2); PRINTTEXT(⟨ tijd start opbouwen stijfheidsmatrix en rechterlid⟩); FLOT(5,3,TIME/100);
          for en:= 1 step 1 until Ne do
          begin HULP(a, b, c, d, e, f,LEN, con, en, le, KN);
                STYFEL(E, I, A, LEN[en, 1],S);
                GOM(LEN[en,1], SI[en,1], SI[en,2], SI[en,3], G1, G2, G3, Gtot, a, b, c, d, e, f);
                ORDEEPS(a, b, LEN[en,2], R[en,1], R[en,2], R[en,3], Oeps);
                DEPSDU(a, b, LEN[en,2], Q1, Q1T);
                for i:= 1,2,3 do for j:= 1 step 1 until 6 do DEDU[i,j]:= Q1[i,j] + Oeps[i,j];
                for i := 1,2,3 do for j := 1 step 1 until 6 do Q1T[j,i] := DEDU[i,j];
                VERMENIGVULDIGING(Q1T, S, DEDU, Qelm1,Qelm2);
                OPTELLING (Qelm1,W, W, Gtot, Qtot);
                QKONSTRUKTIE (en, Qtot, le, lnp, Qkon);
                VERMVEKTOR(SI, DEDU, vektor, en);
                RECHTERLID(vektor, rechter, en, le, lnp);
          end;
          CARRIAGE(2); PRINTTEXT(⟨tijd=⟩); FLOT(5,3,TIME/100);
          for i := 1 step 1 until T do rechterlid[i] := fftot[i] — rechter[i];
```

```
            if st ≠ 0 then
            begin for i:= 1 step 1 until T do
                  begin if fftot[i] = 0 then
                        begin if abs (rechterlid[i]) > absol then
                              begin hu:= hu + 1; goto OPLOSSEN end;
                        end;
                        if fftot[i] ≠ 0 then
                        begin if abs (rechterlid[i] × 100/fftot[i]) > procent then
                              begin hu:= hu + 1; goto OPLOSSEN end;
                        end
                  end;
                  if hu < 2 then
                  begin for i:= 1 step 1 until T do ff[i]:= ff[i] × 2 end;
                  if hu > 3 then
                  begin for i:= 1 step 1 until T do ff[i]:= ff[i]/2 end;
            end;
            hu := 0;
STAP:       for i:= 1 step 1 until T do
            begin if abs(fftot[i]) > abs( ffkon[i]) then goto EINDE;
                  fftot[i]:= fftot[i] + ff[i];
                  rechterlid[i]:= fftot[i] - rechter[i];
            end;
            if st ≠ 0 then UITVOERKNSIR(KN, SI, R, Ne,3, st, N);
            st:= st + 1; con := 2;
            for i:= 1 step 1 until T do for j:= 0 step 1 until bb do Qkon hulp[i,j]:= Qkon[i,j];
            for i:= 1 step 1 until T do rechterlidhulp[i]:= rechterlid[i];
            for i := 1 step 1 until N do for j := 1 step 1 until 2 do KNRE[i,j] := KN[i,j];
            for i := 1 step 1 until Ne do begin LENRE[i] :=LEN[i,2]; PSIHRE[i] := PSIH[i];
                                          for j := 1, 2 do PSIRE[i,j] := PSI[i,j];
                                    end;
            CARRIAGE(3); PRINTTEXT(⟨De belastingskomponenten in de knooppunten na een aantal stappen van⟩);
            ABSFIXT(3,0,st); CARRIAGE(2);
            UITVOERFFTOT(N,fftot,lnp);
OPNIEUW:    CHOLBD (T, bb, 0, Qkon, rechterlid, fail);
            for en:= 1 step 1 until Ne do REKSPANITTERATIE (en, ue, R, KN, KNH, LEN, le, lnp, rechterlid,
                                                    SI, S, E,I,A,PSI,PSIH, HOEK);
            for i:= 1 step 1 until N do for j:= 1,2 do KN[i,j]:= KNH[i,j];
            goto VERDER;
OPLOSSEN:   if hu < cond then
            begin CHOLBD (T, bb, 0, Qkon, rechterlid, fail);
                  for en:= 1 step 1 until Ne do
                  REKSPANITTERATIE (en, ue, R, KN, KNH, LEN, le, lnp, rechterlid, SI, S, E, I, A, PSI, PSIH, HOEK);
                   for i:= 1 step 1 until N do for j:= 1,2 do KN[i,j]:= KNH[i,j];
```

```
            con:= 2;
            NLCR; PRINTTEXT(⪡ er wordt uitgevoerd itteratiestap no:⪢); ABSFIXT(3,0,hu);
             goto VERDER;
        end;
        if hu > cond then
        begin for i:= 1 step 1 until T do ff[i]:= ff[i]/2;
            for i:= 1 step 1 until T do
             begin  rechterlid[i]:= rechterlidhulp[i] - ff[i]; rechterlidhulp[i] := rechterlid[i] end;
            for i:= 1 step 1 until T do for j:= 0 step 1 until bb do Qkon[i,j]:= Qkonhulp[i,j];
            for i := 1 step 1 until N do for j := 1 step 1 until 2 do KN[i,j] := KNRE[i,j];
            for i := 1 step 1 until Ne do begin LEN[i,2] := LENRE[i]; PSIH[i] := PSIHRE[i];
                                              for j := 1, 2 do PSI[i,j] := PSIRE[i,j];
                                          end;
            CARRIAGE(5);
            PRINTTEXT(⪡wegens teveel iteraties wordt opnieuw met de voorgaande stap gestart met halve stapgrootte⪢);
            con :=  2; hu := 0 ;
            for i := 1 step 1 until T do fftot[i] := fftot [i] - ff[i];
            CARRIAGE(5); PRINTTEXT(⪡De totale uitwendige belastingsvektor is nu⪢); UITVOERFFTOT(N, fftot, lnp);
            goto OPNIEUW;
        end;
fail:    CARRIAGE(5);  PRINTTEXT(⪡De stijfheidsmatrix is niet meer positief definiet⪢);
        NLCR; PRINTTEXT(⪡ hetgeen duidt op de niet - lineaire stabiliteitswaarden⪢);
        if st ≠ conhulp then
        begin conhulp := st; teller := 0end;
        if st = conhulp then
        begin teller := teller + 1;
            for i := 1 step 1 until T do ff[i] := ff[i]/2;
            for i := 1 step 1 until T do fftot[i] := fftot[i] - ff[i];
            if teller < 5 then
            begin for i:= 1 step 1 until T do
                begin rechterlid[i]:= rechterlidhulp[i] - ff[i]; rechterlidhulp[i] := rechterlid[i]; end;
                for i:= 1 step 1 until T do for j:= 0 step 1 until bb do Qkon[i,j]:= Qkonhulp[i,j];
                for i := 1 step 1 until N do for j := 1 step 1 until 2 do KN[i,j] := KNRE[i,j];
                for i := 1 step 1 until Ne do begin LEN[i,2] := LENRE[i]; PSIH[i] := PSIHRE[i];
                                                  for j := 1, 2 do PSI[i,j] := PSIRE[i,j];
                                              end;
                CARRIAGE(5);
                PRINTTEXT(⪡Om een betere benadering voor de niet lineaire stabiliteitswaarden ⪢);
```

```
                        NLCR; PRINTTEXT(⪡te krijgen wordt de stap gehalveerd⪢);
                        NLCR;PRINTTEXT(⪡ en opnieuw geittereerd⪢);
                        hu := 0;
                        CARRIAGE(5); PRINTTEXT(⪡ De totale belastingsvektor is nu : ⪢); UITVOERFFTOT(N, fftot, lnp);
                        goto OPNIEUW;
                    end;
                    CARRIAGE(5); PRINTTEXT(⪡ De matrix is 5× achter elkaar niet positief definiet geweest ⪢);
                    NLCR; PRINTTEXT(⪡en daarom wordt het proces gestopt⪢);
                end;
                goto EINDE;
            end;
        end;
    EINDE :
    end
    progend
```

Invoervolgorde voor programma 05064608

N = aantal knooppunten
Ne = aantal elementen
E = elasticiteitsmodulus
I = oppervlaktetraagheidsmoment van de dwarsdoorsnede
A = oppervlakte van de dwarsdoorsnede
absol = absolute afwijking t.o.v. nul
procnt = procentuele afwijking
stap = aantal geschatte stappen in de uitwendige belasting
cond = aantal iteratiestappen, dat toegelaten wordt voordat de belastingsstap gehalveerd wordt.

– : Inlezen van de knooppuntscoördinaten. De coördinaten inlezen in dezelfde volgorde dan de knooppuntsnummering.

– : Het vullen van de lnp-vektor.
ao = aantal knooppunten met voorgeschreven verplaatsingen
n = knooppuntnummer
av = aantal voorgeschreven richtingen.
v = richting, die voorgeschreven is.

– : vullen van de le-vektor. In volgorde van de elementnummering de knooppunten per element.

– : vullen van de belastingsvektor.
ab = aantal knooppunten met voorgeschreven belasting
n = knooppuntnummer
ae = aantal voorgeschreven belastingen per knooppunt
hierna richting en grootte van de voorgeschreven belastingen.

```
Lalgol05064355Beyers
begin comment berekening van de knikkracht van een vakwerk, met behulp van de elementenmethode, waarin tevens de verplaatsingen
              en de krachten in de knooppunten bepaald kunnen worden;
      library CHOLESKI DECOMPOSITION, CHOLESKI SOLUTION, INPROD, TRIDIAGONALIZATION, EIGENVALUE,
              EIGENVECTOR, BACKTRANSFORMATION;
      procedure DEPSDU(a, b, l, Q1, Q1T);
      value a, l, b; real a, b, l; array Q1, Q1T;
      begin integer i, j; real hulp;
            for i := 1 step 1 until 3 do for j := 1 step 1 until 6 do Q1[i, j] := 0;
            hulp := 1/l;
            Q1[1, 1] := - b; Q1[1, 4] := - Q1[1, 1]; Q1[1, 2] := - a; Q1[1, 5] := - Q1[1, 2];
            Q1[2, 1] := - a × hulp; Q1[2, 2] := hulp × b; Q1[2, 4] := Q1[3, 1] := - Q1[2, 1] ; Q1[3, 4] := Q1[2, 1];
            Q1[2, 5] := Q1[3, 2] := - Q1[2, 2]; Q1[3, 5] := Q1[2, 2]; Q1[2, 3] := - 1; Q1[3, 6] := 1;
            for i := 1 step 1 until 3 do for j := 1 step 1 until 6 do Q1T[j, i] := Q1[i, j];
      end DEPSDU;

      procedure VERMENIGVULDIGING(A, B, C, D, DT);
      array A, B, C, D, DT;
      begin integer i, j, k; array HU[1:6, 1:3];
            for i := 1 step 1 until 6 do for j := 1 step 1 until 3 do HU[i, j] := 0;
            for i := 1 step 1 until 6 do for j := 1 step 1 until 6 do D[i, j] := DT[i, j] := 0;
            for i := 1 step 1 until 6 do for j := 1 step 1 until 3 do
            begin for k := 1 step 1 until 3 do HU[i, j] := HU[i, j] + A[i, k] × B[k, j]; end;
            for i := 1 step 1 until 6 do for j := 1 step 1 until 6 do
            begin for k := 1 step 1 until 3 do D[i, j] := D[i, j] + HU[i, k] × C[k, j]; end;
            for i := 1 step 1 until 6 do for j := 1 step 1 until 6 do DT[i, j] := D[j, i];
      end VERMENIGVULDIGING;

      procedure STYFEL(E, I, A, l, S);
      value E, I, l, A; real E, I, l, A; array S;
      begin S[1, 1] := E × A/l; S[1, 2] := S[1, 3] := S[2, 1] := S[3, 1] := 0;
            S[2, 2] := S[3, 3] := 4 × E × I/l; S[2, 3] := S[3, 2] := - 2 × E × I/l;
      end STYFEL;

      procedure HULP( a, b, c, d, e, f, LEN, st, en, le, KN);
      value en, st; integer en, st; real a, b, c, d, e, f; integer array le;  array LEN, KN ;
      begin integer i, j; real x1, x2, z1, z2, l;
            i := le [en,1]; j:= le[en,2];
            x1 := KN[i,1]; x2 := KN[j,1]; z1 := KN[i,2]; z2 := KN[j,2];
            l := ((z2 - z1) × (z2 - z1) + (x2 - x1) × (x2 - x1)) ↑ 0.5; LEN[en] := l;
            a := (z2 - z1)/l; b := (x2 -x1)/l; c := a × a; d := b × b; e := 2 × a × b; f := d - c;
      end HULP;
```

```
procedure PROFIELKEUZE(Ne, keuze, di, F, I, b, h);
value Ne; integer Ne, keuze; array  di, F, I, b, h;
begin integer i; real c, pi; c := arctan(1)/45; pi := 4 × arctan(1);
      for i := 1 step 1 until Ne do
      begin keuze := read;
            if keuze = 1 then begin di[i] := read; F[i] := pi/4 × di[i] × di[i];
                                    I[i] := pi/64 × (di[i] ⋀ 4); NLCR; ABSFIXT(3, 0, i);
                                    SPACE(4); FLOT(5, 3, F[i]); SPACE(4);
                                    FLOT(5, 3, I[i]); SPACE(36); FLOT(5, 3, di[i]);
                              end;
            if keuze = 2 ∨ keuze = 3 then begin  b[i] := read; h[i] := read; F[i] := b[i] × h[i];
                                                 if keuze = 3 then I[i] := 1/12 × h[i] × (b[i] ⋀ 3)
                                                              else I[i] := 1/12 × b[i] × (h[i] ⋀ 3);
                                                 NLCR; ABSFIXT(3, 0, i); SPACE(4);
                                                 FLOT(5, 3, F[i]); SPACE(4); FLOT(5, 3, I[i]); SPACE(3);
                                                 FLOT(5, 3, b[i]); SPACE(3); FLOT(5, 3, h[i]);
                                           end;
            if keuze = 4 then begin  F[i] := read;h[i] := read; I[i] := 2 × F[i] × h[i] × h[i]/4;
                                    NLCR; ABSFIXT(3, 0, i); SPACE(4);
                                    FLOT(5, 3, F[i]); SPACE(4); FLOT(5, 3, I[i]); SPACE(19);
                                    FLOT(5, 3, h[i]);
                              end;
            if keuze = 5 then begin  F[i] := read; I[i] := read; NLCR; ABSFIXT(3, 0, i);
                                    SPACE(4); FLOT(5, 3, F[i]); SPACE(4);
                                    FLOT(5, 3, I[i]);
                              end;
end
end PROFIELKEUZE;

procedure GENERAL EIGENVALUE PROBLEM(n, A, C, lambda, X, k1, k2, non definite, eivec);
value n, k1, k2, eivec; integer n, k1, k2; array A, C, lambda, X; label non definite; Boolean eivec;
begin integer i, j, l; array diag, codiag[1:n];
       for j := 1 step 1 until n do for i := 1 step 1 until n do
       A[i, j] := INPROD(l, 1, i-1, -C[i, l], A[l, j], A[i, j])/C[i, i];
       for j := 1 step 1 until n do for i := 1 step 1 until j do
       A[i, j] := A[j, i] := INPROD(l, 1, i-1, -C[i, l], A[j, l], A[j, i])/C[i, i];
       TRIDIAGONALIZATION(n, A, A, diag, codiag);
       EIGENVALUE(n, diag, codiag, k1, k2, lambda);
       if eivec then begin EIGENVECTOR(n, diag, codiag, k1, k2, lambda, X);
                           BACKTRANSFORMATION(n, A, k1, k2, X);
                           for j := k1 step 1 until k2 do for i := n step -1 until 1 do
                           X[i, j] := INPROD(l, i+1, n, -C[l, i], X[l, j], X[i, j])/C[i, i ]
                     end
 end EIGENVALUE PROBLEM;
```

```
procedure UITVOERFFTOT(N, fftot, lnp);
valueN; integer N; array fftot, lnp;
begin integer i, j, k, a; real array mn[1:3];
    for i := 1 step 1 until N do
    begin for j := 1,2,3 do
        begin a := lnp[i,j]; mn[j] :=0;
            if a ≠ 0 then mn[j] :=fftot[a];
        end;
        if mn[1] ≠ 0 ∨ mn[2] ≠ 0 ∨mn[3] ≠0 then
        begin NLCR; PRINTTEXT(≮ knooppunt = ≯); ABSFIXT(3,0,i);
            for k := 1 step 1 until 3 do
            begin SPACE(5); FLOT(5,3,mn[k]) end;
        end;
    end;
endUITFFTOT;

  procedure TRANSFORMATIE(ue, ueg, a, b);
  real a, b;  array ue, ueg;
  begin integer i, j; array R[1:3, 1:3];
      R[1, 1] := R[2, 2] := b; R[2, 1] := —a;R[1, 2] := —R[2, 1];
      for i := 1, 2 do R[i, 3] := R[3, i] := 0; R[3, 3] := 1;
      for i := 1 step 1 until 6 do ueg[i] := 0;
      for i := 1 step 1 until 3 do for j := 1 step 1 until 3 do
      begin ueg[i] := ueg[i] +R[i, j] × ue[j];
            ueg[i+3] := ueg[i+3] + R[i, j] × ue[j+3];
      end
  end TRANSFORMATIE;

procedure BELASTINGVEKTOR(T, lnp, f);
value T; integer T; integer array lnp; array f;
begin integer i, j, ab, n, ae; real q;
    for i := 1 step 1 until T do f[i] := 0; ab := read; for i := 1 step 1 until ab do
    begin n := read; NLCR; PRINTTEXT(≮knooppunt=≯); ABSFIXT(3, 0, n); ae:= read;
        for j := 1 step 1 until ae do begin q := f[lnp[n, read]] := read; SPACE(5); FLOT(5, 3, q);
                                      end;
    end
end BELASTINGSVEKTOR;

procedure ONDERSTEUNING(N, T, lnp);
value N; integer N, T; integer array lnp;
begin integer n, v, a0, i, j, av, t;
    t := 0; for n := 1 step 1 until N do for v := 1, 2, 3 do lnp[n, v] := 1;
```

```
        a0 := READ; for i := 1 step 1 until a0 do
        begin n := READ; av := READ; for j := 1 step 1 until av do
              begin v := READ; lnp[n, v] := 0;
              end
        end inlezen voorgeschreven verplaatsingen;
        for n := 1 step 1 until N do for v := 1, 2, 3 do if lnp[n, v] = 1 then
        begin t := t + 1;
              lnp[n, v] := t;
        end;
        T := t;
  end bepaling lnp vektor met T vrijheidsgraden;

  procedure INLE(Ne, le);
  value Ne; integer Ne; integer array le;
  begin integer k;
        for k := 1 step 1 until Ne do
        begin le[k, 1] := READ; le[k, 2] := READ;
        end
  end inlezen le vektor;

  procedure URM(A, r0, r, k0, k); value r0, r, k0, k; integer r0, r, k0, k; array A;
  begin comment Uitvoer Real Matrix, voor nadere gegevens zie: Toelichting op WE-procedures;
        integer m, n, i, j, l;
        m := (k -k0 +1) : 10; m := if k -k0 +1 = 10 × m  then m else m + 1;
        for n := 0 step 1 until m - 1 do
        begin NLCR; SPACE(2); for l := 10 × n + k0, l + 1 while l ≤ k0 - 1 + 10 × (n + 1) ∧ l ≤ k do
              begin SPACE(8); ABSFIXT(3, 0, l) end; NLCR;
              for i := r0 step 1 until r do
              begin NLCR; ABSFIXT(3, 0, i); SPACE(2);
                    for j := 10 × n + k0 step 1 until l-1 do
                    begin integer n; real x; x:= abs(A[i,j]);
                          if x < ₁₀-100 ∨ x ≥ ₁₀99 then n := (if x = 0 then 1 else 3) else
                          if x < ₁₀-10  ∨ x ≥ ₁₀9  then n := 2 else n := 1;
                          FLOT(5,n,A[i,j]); SPACE(3-n)
                    end
              end; CARRIAGE(5)
        end
  end URM;
```

```
procedure GOM(l, sigma1, sigma2, sigma3, G1, G2, G3, G, a, b, c, d, e, f);
value l, sigma1, sigma2, sigma3, a, b, c, d, e, f; real l, sigma1, sigma2, sigma3, a, b, c, d, e, f;
array G1, G2, G3, G;
begin integer i, j; real hulp1, hulp2, hulp3;
      for i := 1 step 1 until 6 do for j := 1 step 1 until 6 do G1[i, j] := G2[i, j] := G3[i, j] := G[i, j] := 0;
      G1[1, 1] := sigma1 × 6/(5 × l) × c; G1[4, 4] := G1[1, 1]; G1[1, 4] := − G1[1, 1];
      G1[1, 2] := − sigma1 × 3/(5 × l) × e; G1[1, 5] := G1[2, 4] := − G1[1, 2]; G1[4, 5] := G1[1, 2];
      G1[1, 3] := 0.1 × sigma1 × a; G1[1, 6] := G1[1, 3]; G1[3, 4] := G1[4, 6] := − G1[1, 3];
      G1[2, 2] := sigma1 × 6/(5 × l) × d; G1[5, 5] := G1[2, 2]; G1[2, 5] := − G1[2, 2];
      G1[2, 3] := − 0.1 × sigma1 × b; G1[2, 6] := G1[2, 3]; G1[5, 6] := G1[3, 5] := − G1[2, 3];
      G1[3, 3] := sigma1 × 4 × l/30; G1[6, 6] := G1[3, 3]; G1[3, 6] := − G1[3, 3]/4;
      hulp2 := 1/(l × l) × sigma2;
      G2[1, 1] := − hulp2 × e; G2[1, 4] := G2[2, 2] := G2[5, 5] := − G2[1, 1]; G2[2,5] := G2[4, 4] := G2[1, 1];
      G2[1, 2] := hulp2 × f; G2[4, 5] := G2[1, 2]; G2[1, 5] := G2[2, 4] := − G2[1, 2];
      hulp3 := 1/(l × l) × sigma3;
      G3[1, 1] := hulp3 × e; G3[1, 4] := G3[2, 2] := G3[5, 5] := − G3[1, 1]; G3[2,5] := G3[4, 4] := G3[1, 1];
      G3[1, 2] := − hulp3 × f; G3[4, 5] := G3[1, 2]; G3[1, 5] := G3[2, 4] := − G3[1, 2];
      for i := 1 step 1 until 6 do for j := i step 1 until 6 do G[i, j] := G1[i, j] + G2[i, j] + G3[i, j];
      for i := 2 step 1 until 6 do for j := 1 step 1 until i − 1 do G[i, j] := G[j, i];
end GOMETRISCHE MATRIX;

procedure QKONSTRUKTIE(en, Qtot, le, lnp, Qkon);
value en; integer en; array Qtot, le, lnp, Qkon;
begin integer s1, s2, v1, v2, l1, l2, i;
      for s1:= 1, 2 do for v1 := 1, 2, 3 do
      begin i := lnp[le[en, s1], v1]; l1 := (s1 − 1) × 3 + v1; if i > 0 then
            begin for s2 := 1, 2 do for v2 := 1, 2, 3 do
                  begin j := lnp[le[en, s2], v2]; l2 := (s2 − 1) × 3 + v2;
                        if j ≥ i then Qkon[i, j] := Qkon[i, j] + Qtot[l1, l2];
                  end
            end
      end
end QKONSTRUKTIE;
procedure UITVOER(Ne, k, M);
value Ne; integer Ne, k; array M;
begin integer i, j;
      for i := 1 step 1 until Ne do
      begin CARRIAGE(2); ABSFIXT(3, 0, i);
            for j := 1 step 1 until k do
            begin SPACE(5); FLOT(5, 3, M[i, j]);
            end
      end
end UITVOER;
```

```
integer N,Ne, test, an, eigenvektoren, i, j, keuze, T,en, k, l1, l2;
real macheps, E,a,b,c,d,e,f,PK;
real array S[1:3, 1:3], Q1[1:3, 1:6], G1, G2, G3, Gtot,Qelm1, W[1:6, 1:6], Q1T[1:6, 1:3], ue, ueg[1:6],
        eps[1:3];
N:= read; NLCR; PRINTTEXT(<aantal knooppunten =>); ABSFIXT(3,0,N); CARRIAGE(2);
Ne:= read; PRINTTEXT(<aantal elementen =>); ABSFIXT(3,0,Ne); CARRIAGE(2);
E:= read; PRINTTEXT(<elasticiteitsmodules =>); FLOT(5,3,E); CARRIAGE(2);
macheps:= 2 × 10-12; test:= read; an:= read; eigenvektoren:= read;
begin integer array lnp[1:N, 1:3], le[1:Ne, 1:2];
    real array I, F,di, h, br[1:Ne], KN[1:N, 1:2], SI[1:Ne, 1:3], LEN[1:Ne];
    for i:= 1 step 1 until N do for j:= 1, 2 do KN[i,j]:= read;
    PRINTTEXT(<de knooppuntskoordinaten van de konstruktie>); CARRIAGE(2);
    UITVOER(N,2,KN); CARRIAGE(5);
    PRINTTEXT(<        oppervlak     traagheidsmoment     breedte        hoogte          diameter>);
    PROFIELKEUZE(Ne, keuze, di, F, I, br, h);
    ONDERSTEUNING(N,T,lnp); CARRIAGE(5);
    PRINTTEXT(<de lnp - vektor van de construktie>); CARRIAGE(2);
    UITVOER(N,3,lnp);
    INLE(Ne, le); CARRIAGE(5);
    PRINTTEXT(<de le - vektor van de construktie>); CARRIAGE(2);
    UITVOER(Ne, 2, le); CARRIAGE(5);
    begin array X,Qkon, Gkon[1:T, 1:T], ff, u, lambda[1:T];
        PRINTTEXT(<de belastingsgrootheden in de volgende knooppunten>);
        BELASTINGVEKTUR(T,lnp,ff); CARRIAGE(5);
        for i:= 1 step 1 until T do for j:= 1 step 1 until T do Qkon[i,j]:= Gkon[i,j]:= 0;
        for en:= 1 step 1 until Ne do
        begin HULP(a,b,c,d,e,f,LEN,1,en, le,KN);
            STYFEL(E,I[en],F[en],LEN[en],S);
            DEPSDU(a,b,LEN[en],Q1,Q1T);
            VERMENIGVULDIGING(Q1T,S,Q1,Qelm1,W);
            QKONSTRUKTIE(en, Qelm1, le, lnp, Qkon);
        end;
        for i:= 2 step 1 until T do for j:= 1 step 1 until i - 1 do Qkon[i,j]:= Qkon[j,i];
        CHOLESKIDECOMPOSITION(T,Qkon,Qkon,non definite);
        CHOLESKISOLUTION(T,Qkon,ff,u);
        PRINTTEXT(<verplaatsingsvektoren in het globaal assenstelsel --u1--w1--psi1--u2--w2--psi2-- >);
        CARRIAGE(2);
        for en:= 1 step 1 until Ne do
        begin for i:= 1,2 do for j:= 1,2,3 do
            begin l1:= (i - 1) × 3 + j; l2:= lnp [le[en, i],j];
                ue[l1]:= if l2 = 0 then 0 else u[l2];
            end;
            HULP(a,b,c,d,e,f,LEN, 1,en,le,KN);
```

```
                TRANSFORMATIE(ue,ueg,a,b);
                ABSFIXT(3,0,en);
                SPACE(5);
                for j:= 1 step 1 until 6 do
                begin FLOT(5,3,ue[j]); SPACE(5) end;
                eps[1]:= ueg[4] - ueg[1];
                eps[2]:= (ueg[2] - ueg[5])/LEN[en] - ueg[3];
                eps[3]:= (- ueg[2] + ueg[5])/LEN[en] + ueg[6];
                STYFEL(E,I[en], F[en],LEN[en],S);
                for j:= 1,2,3 do SI[en, j]:= 0;
                for j:= 1,2,3 do for k := 1,2,3 do SI[en,j]:= SI[en,j] + S[j,k] × eps[k];
                GOM(LEN[en],SI[en,1],SI[en,2],SI[en,3],G1,G2,G3,Gtot,a,b,c,d,e,f);
                QKONSTRUKTIE(en,Gtot,le,lnp,Gkon);
                CARRIAGE(2);
            end;
            for i:= 2 step 1 until T do for j:= 1 step 1 until i - 1 do Gkon[i,j]:= Gkon[j,i];
            CARRIAGE(5);
            PRINTTEXT(≮de belastingsvektoren van de elementen - - - N - - M1 - - M2 - - - ≯);
            CARRIAGE(2); UITVOER(Ne,3,SI);
            CARRIAGE(2);
            GENERAL EIGEN VALUE PROBLEM(T,Gkon,Qkon,lambda, X,1,T,non definite, eigenvektoren = 1);
            if eigenvektoren = 1 then
            begin CARRIAGE(5);
                PRINTTEXT(≮de eigenvektoren van de construktie≯); CARRIAGE(2);
                URM(X,1,T,1,T); CARRIAGE(5);
            end;
            for j:= 1 step 1 until T do
            begin if an = 1 then
                begin PRINTTEXT(≮eigenwaarde lambda =≯); FLOT(5,3,lambda[j]);
                    PK:= 1/lambda[j]; SPACE(5);
                    PRINTTEXT(≮de vermenigvuldigingsfactor = ≯); FLOT(5,3,PK); CARRIAGE(2);
                 end;
            end;
            if abs(lambda[1]) > abs(lambda[T]) then PK := 1/lambda[1] else PK := 1 / lambda[T];
            CARRIAGE(5); PRINTTEXT(≮de kleinste vermenigvuldigingsfactor = ≯); FLOT(5,3,PK);
            for i := 1 step 1 until T do ff[i] := -ff[i] × PK;
            CARRIAGE(5); PRINTTEXT(≮ De knikbelasting van de constructie = ≯);
            CARRIAGE(2);
            UITVOERFFTOT(N, ff, lnp);
        end;
    end; goto EINDE;
    non definite: PRINTTEXT(≮stijfheidsmatrix van de konstruktie is niet positief definiet≯);
    EINDE:
 end
progend
```

Invoervolgorde programma 05064355

„Stabiliteit waarin het effect van het moment is meegenomen."

N = aantal knooppunten

Ne = aantal elementen

E = elasticiteitsmodulus

test = variabele die gebruikt wordt voor hulpuitvoer, is niet in het definitieve programma gebruikt.

an = 1 dan worden de eigenwaardes uitgevoerd.

eigenvektoren = 1 dan uitvoer van de eigenvektoren.

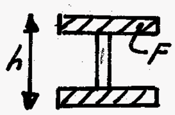
—: inlezen knooppuntscoördinaten

keuze = 1 cirkelvormige dwarsdoorsnede
per element inlezen: d[i] = diameter

keuze = 2 rechthoekige dwarsdoorsnede $b > h$
inlezen: b[i] = breedte element i
h[i] = hoogte element i

keuze = 3 rechthoekige dwarsdoorsnede $b < h$
inlezen b[i] = breedte element i
h[i] = hoogte element i

keuze = 4 elementen hebben de volgende dwarsdoorsnede



$$I = 2 \cdot \frac{h^2 F}{4}$$

F[i] = oppervlakte element i
h[i] = hoogte element i

keuze = 5 Willekeurige dwarsdoorsnede:
F[i] = oppervlak element i
I[i] = oppervlaktetraagheid.

Het op de bekende wijze vullen van de knp. vektor.
a0 = aantal knooppunten met voorgeschreven verplaatsingen
n = knooppuntnummer
av = aantal voorgeschreven verplaatsingen
v = richting voorgeschreven verplaatsing

Inlezen van de le-vektor.
Vullen van de belastingsvektor