

Numerieke simulatie van een lokale en niet-lokale aanpak bij een continuüm damage model

Citation for published version (APA):

Feron, R. P. M. J. (1992). *Numerieke simulatie van een lokale en niet-lokale aanpak bij een continuüm damage model*. (DCT rapporten; Vol. 1992.047). Technische Universiteit Eindhoven.

Document status and date:

Gepubliceerd: 01/01/1992

Document Version:

Uitgevers PDF, ook bekend als Version of Record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

**Numerieke simulatie van een
lokale en niet-lokale aanpak
bij een continuüm damage model**

R.P.M.J. Feron

Rapport nr.: WFW.92.047

**Numerieke simulatie van een lokale en niet-lokale aanpak
bij een continuüm damage model**

Auteur: Raymond Feron

Stagebegeleider: Marcel Brekelmans

Eindhoven, april 1992

VAKGROEP FUNDAMENTELE WERKTUIGKUNDE
FACULTEIT WERKTUIGBOUWKUNDE
TECHNISCHE UNIVERSITEIT EINDHOVEN

INHOUDSOPGAVE

SAMENVATTING.....	1
SYMBOLENLIJST.....	2
1. INLEIDING.....	4
2. ALGEMENE WERKWIJZE.....	5
2.1. Algemeen	
2.2. Eindige elementenmethode	
2.3. Numerieke incrementstapmethode	
3. LINEAIR ELASTISCH.....	14
3.1. Modelling	
3.2. Numerieke resultaten	
4. HOMOGENE SCHADE.....	15
4.1. Modelling	
4.2. Numerieke resultaten	
5. INHOMOGENE SCHADE: LOKALE WERKWIJZE.....	18
5.1. Modelling	
5.2. Clause in het numerieke model	
5.3. Incrementstapmethode	
5.4. Numerieke resultaten	
5.4.1. Invoer	
5.4.2. Aantonen reële/irreële overgang	
5.4.3. Invloed van het aantal elementen	
5.5. Conclusie	
6. INHOMOGENE SCHADE: NIET-LOKALE WERKWIJZE.....	26
6.1. Modelling	
6.2. Numerieke implementatie van niet-lokale schade	
6.2.1. Numeriek wegen van de rek	
6.2.2. Numerieke schade-toename	
6.3. Incrementstapmethode	
6.4. Numerieke resultaten	
6.4.1. Invoer	
6.4.2. Beperkingen door numerieke werkwijze	
6.4.3. Resultaten met $l_D=30$ mm.	
6.4.4. Resultaten met $l_D=10$ mm.	

Inhoudsopgave

7. CONCLUSIE.....	34
LITERATUUR.....	35
BIJLAGEN.....	36
Bijlage 1. Gebruikte programmatuur bij inhomogene schade	
Bijlage 1A. Lokale werkwijze: invoerls.m en rekenls.m	
Bijlage 1B. Niet-lokale werkwijze: invoernl.m en rekennl.m	
Bijlage 2. Homogene schade	
Bijlage 3. Inhomogene schade met lokale aanpak: aantonen reële/irreële overgang	
Bijlage 4. Inhomogene schade met lokale aanpak: invloed van het aantal elementen	
Bijlage 5. Inhomogene schade met niet-lokale aanpak en $l_D=30$ mm	
Bijlage 6. Inhomogene schade met niet-lokale aanpak en $l_D=10$ mm	
Bijlage 7. Beperking door numerieke werkwijze.	

SAMENVATTING

Een onderzoek wordt in het algemeen afgesloten met een schriftelijke weergave. Deze bevat vaak slechts een globale representatie van het gedane en een conclusietrekking. Dit was ook het geval bij een artikel [4], afkomstig van de universiteit Parijs 6, waarin gesteld wordt dat een niet-lokale aanpak van de schade bij een continuüm damage model een bruikbare methode is, die de tekortkomingen van de lokale aanpak opheft. Mijn stageverslag beoogt deze stelling op een eenvoudige manier te onderzoeken en te bewijzen.

Voor de numerieke modellering werd, rekening houdende met de algemeen toegankelijke omgeving, gebruik gemaakt van PC-Matlab.

Zo werd allereerst, om enige ervaring op te doen met het numeriek programmeren in PC-Matlab, een volledig elastisch model berekend en vervolgens een model met homogene schade. Na dit leerzame werk werden vervolgens de inhomogene schade modellen met lokale aanpak en niet-lokale aanpak beschouwd.

Het uiteindelijke resultaat is, dat de onderzochte stelling slechts geldt onder een bepaalde voorwaarde en dus niet algemeen aanvaard kan worden.

SYMBOLENLIJST

<u>Grootheid</u>	<u>Eenheid</u>	<u>Verklaring</u>
A	mm ²	dwarsdoorsnede-oppervlakte
D	[-]	schade parameter
D _i	[-]	schade in element i
E	N/mm ²	elasticiteitsmodulus
E _i	N/mm ²	E-modulus van element i
E ₀	N/mm ²	begin E-modulus
E _{0i}	N/mm ²	begin E-modulus van element i
ε	[-]	rek
ε _c	[-]	critische rek
ε _{end}	[-]	parameter die van invloed is op D
ε _i	[-]	rek in element i
ε _{max}	[-]	rek bij breuk
ε _{xx}	[-]	axiale rek
ε̃	[-]	gewogen rek
ε̃ _i	[-]	gewogen rek in element i
F _{max}	N	maximale normaalkracht
F _{imax}	N	max. knpts.-kracht in knptn. 2 t/m n
g	[-]	weegfunctie die rek weegt
g	[-]	incrementstapgrootte-parameter
G	[-]	incrementstapgrootte-parameter
j	[-]	incrementstapaanduiding
j _c	[-]	critische incrementstap
J _c	[-]	critische incrementstap
j _{max}	[-]	j waarbij staaf breekt
j _{maxreal}	[-]	j waarbij model irreëel wordt
<u>k</u>	mm	kolom met knooppuntskrachten
k _i	N/mm	elementstijfheid
K	N	kracht aan uiteinde staaf
<u>K</u>	N/mm	ge-assembleerde stijfheidsmatrix
<u>K_i</u>	N/mm	elementstijfheidsmatrix
l	mm	lengte staaf
l _p	mm	halve representatieve lengte
l _i	mm	lengte van een element
m	[-]	parameter die van invloed is op D
n	[-]	aantal elementen
N	N	normaalkracht
N _i	[-]	interpolatiefunctie
q	N/mm	verdeelde belasting
q	[-]	parameter die van invloed is op D
R	N	reactiekracht aan begin staaf

<u>Grootheid</u>	<u>Eenheid</u>	<u>Verklaring</u>
ρ	kg/mm ³	dichtheid
s	[-]	aantal incrementstappen
σ	N/mm ²	spanning
σ_c	N/mm ²	critische spanning
σ_i	N/mm ²	spanning in element i
t	sec	tijd
u	mm	axiale verplaatsing
\tilde{u}	mm	kolom met knooppuntsverplaatsingen
u_i	mm	verplaatsing in knooppunt i
u_l	mm	eindverlenging uiteinde
u_{lj}	mm	incrementstapverplaatsing uiteinde
u_p	mm	voorgeschreven verplaatsing
\tilde{u}_y	mm	kolom met vrije knpts.-verplaatsingen
w	[-]	weegfunctie die beweg.vrgl. weegt
\tilde{w}	[-]	kolom met waarden van w in knptn.
w_i	[-]	waarde van w in knooppunt i
x	mm	axiale coördinaat
x_k	mm	lokale axiale coördinaat
X	[-]	incrementstapgrootte-parameter

1. INLEIDING

Een tak van de vakgroep WFW (fundamentele werktuigkunde) van de faculteit werktuigbouwkunde aan de Technische Universiteit Eindhoven (TUE), houdt zich bezig met de continuüm damage mechanica (CDM). Kern van de continuüm damage mechanica is, dat een schade parameter geïntroduceerd wordt bij het beschrijven van het constitutieve gedrag van een vaste stof materiaal, in de vorm van een interne variabele.

Over de hele wereld vindt onderzoek plaats naar deze relatief jonge stroming van de vaste stof mechanica (pas sinds ± 1980 op grote schaal in ontwikkeling). Ook aan de universiteit van Parijs 6 staat de CDM volop in de belangstelling, getuige de literatuur die daar wordt gepubliceerd. Een artikel dat van deze universiteit afkomstig is, wordt in dit stage-verslag onder de loep genomen. Het betreft 'A multiscale approach to distributed damage and its usefulness for capturing structural size effect', geschreven door C. Saouridis en J. Mazars (zie [4]). Hierin wordt gesteld, dat de CDM met gebruik van een lokale schade variabele (schade afhankelijk van lokale rek) numeriek leidt tot objectiviteitsproblemen met betrekking tot elementgrootte en meshoriëntatie. Dit in tegenstelling tot het gebruik van een niet-lokale schade-definitie (zogenaamde 'multiscale approach'), waarbij de resultaten veel minder elementgrootte- en meshoriëntatie-afhankelijk zouden zijn.

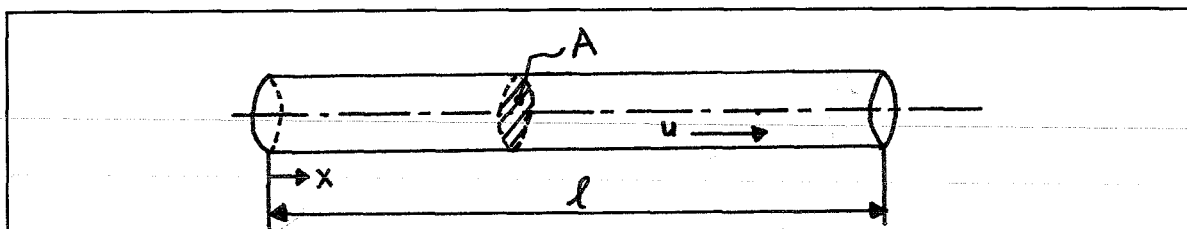
In het artikel worden slechts de algemene principes van de lokale en niet-lokale aanpak summier aangegeven en resultaten in de vorm van plaatjes getoond. Hoe de numerieke implementatie plaatsvindt blijft verder buiten beschouwing. Hier ligt de basis van mijn stage. Deze had namelijk tot doel om met een eenvoudig voorbeeld (één-dimensionale trekstaaf) te onderzoeken welke problemen je zoal tegenkomt bij de numerieke implementatie en of de strekking van het Parijse artikel onderstreept kan worden.

In hoofdstuk 2 wordt de algemene werkwijze belicht. De hoofdstukken 3 en 4 hebben als onderwerp respectievelijk een lineair elastisch model en een model met homogene schade, terwijl hoofdstuk 5 een model met inhomogene schade en lokale aanpak behandelt. Tenslotte komt in hoofdstuk 6 de inhomogene schade met niet-lokale aanpak aan de orde en wordt dit verslag afgesloten met enkele conclusies (hfdst. 7).

2. ALGEMENE WERKWIJZE

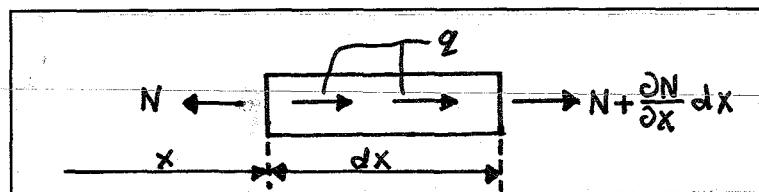
2.1. Algemeen

Er wordt uitgegaan van een trekstaaf met lengte l , dwarsdoorsnede A en dichtheid ρ . Verder blijven vlakke dwarsdoorsneden vlak, hetgeen betekent dat de axiale verplaatsing u (zie fig.1.1) uitsluitend een functie is van de axiale, materiële coördinaat x en van de tijd. De verplaatsing op een bepaald punt van de staaf wordt dus vastgelegd door $u=u(x,t)$:



figuur 2.1

Op een infinitesimaal stukje dx van de staaf werken de volgende krachten:



figuur 2.2

Hierbij is N de normaalkracht en q de verdeelde belasting, per lengte-eenheid (lijnbelasting).

De volgende set beschrijvende vergelijkingen kan worden gedefinieerd:

* Rek/verplaatsingsrelatie:

$$\epsilon_{xx} = \frac{\partial u}{\partial x} \quad \text{voor } 0 \leq x \leq l, t \geq 0 \quad (2.1)$$

* Constitutieve vergelijking (Wet van Hooke):

$$N = E * A * \epsilon_{xx} \quad \text{voor } 0 \leq x \leq l, t \geq 0 \quad (2.2)$$

* Bewegingsvergelijking (quasi statisch):

$$\frac{\partial N}{\partial x} + q(x,t) = \rho * A * \frac{\partial^2 u}{\partial t^2} = 0 \quad \text{voor } 0 \leq x \leq l, t \geq 0 \quad (2.3)$$

* Randvoorwaarden (kinematische):

$$u(x=0,t) = 0 \quad \text{voor } t \geq 0 \quad (2.4)$$

$$u(x=l,t) = u_l(t) \quad \text{voor } t \geq 0 \quad (2.5)$$

Uit dit stelsel kunnen de axiale rek ε_{xx} en de normaalkracht N worden geëlimineerd, waardoor een differentiaalvergelijking ontstaat waarin de verplaatsing $u=u(x,t)$ als primaire onbekende wordt opgevat:

$$\frac{\partial}{\partial x} (E * A * \frac{\partial u}{\partial x}) + q(x,t) = 0 \quad \text{voor } 0 \leq x \leq l, t \geq 0 \quad (2.6)$$

Vergelijking (2.6) wordt gewogen met een weegfunctie $w(x)$ en geïntegreerd over het gehele interval. Vervolgens vindt er een partiële integratie plaats, hetgeen resulteert in:

$$\int_{x=0}^l \frac{\partial w}{\partial x} * E * A * \frac{\partial u}{\partial x} dx = \int_{x=0}^l w * q dx + (w * E * A * \frac{\partial u}{\partial x}) \Big|_{x=0}^l \quad (2.7)$$

De eis die aan zowel de verplaatsingsfunctie u als aan de weegfunctie w wordt gesteld is, dat deze functies met betrekking tot het verloop als functie van de coördinaat x , stuksgewijs differentieerbaar moeten zijn. Verder moet voor alle $w(x)$ aan vergelijking (2.7) zijn voldaan en moet $u(x,t)$ uiteraard ook aan de randvoorwaarden voldoen.

Zonder lijnbelasting resteert:

$$\int_{x=0}^l \frac{\partial w}{\partial x} * E * A * \frac{\partial u}{\partial x} = (w * E * A * \frac{\partial u}{\partial x})_{x=l} - (w * E * A * \frac{\partial u}{\partial x})_{x=0} \quad (2.8)$$

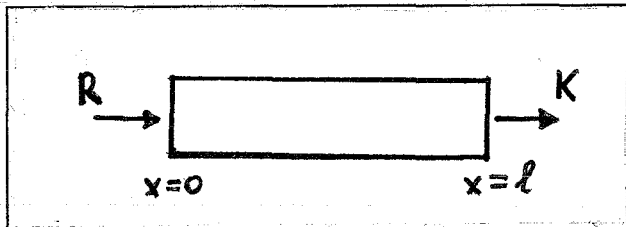
Aangezien het stelsel van bewegingsvergelijking (2.10) en de randvoorwaarden (2.4) en (2.5) op elk willekeurig tijdstip t kan worden opgelost zonder rekening te houden met voorafgaande tijdstippen, speelt de tijd een ondergeschikte rol en kan volstaan worden met een momentane beschouwing. De verplaatsing u is dus uitsluitend een functie van de coördinaat x .

De randvoorwaarden worden dan geformuleerd als:

$$u(x=0) = 0 \quad (2.9)$$

$$u(x=l) = u_i \quad (2.10)$$

Met betrekking tot de twee termen van het rechterlid van (2.8) worden de krachten K en R geïntroduceerd, die respectievelijk ter plaatse van $x=l$ en $x=0$ (inklemming) op de staaf werken de staaf zullen werken. Met de volgende tekenafpraak:



figuur 2.3

zal voor K en R gelden:

$$K = N(x=l) = E * A * \left(\frac{du}{dx} \right)_{x=l} \quad (2.11)$$

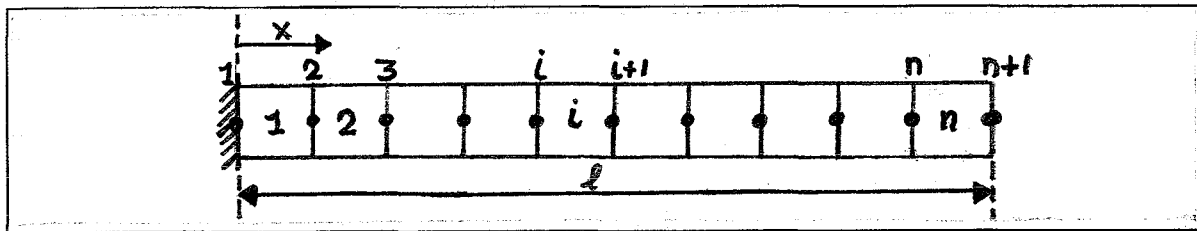
$$R = -N(x=0) = -E * A * \left(\frac{du}{dx} \right)_{x=0} \quad (2.12)$$

,waarmee (2.8) overgaat in:

$$\int_{x=0}^l \left(\frac{dw}{dx} \right) * E * A * \left(\frac{du}{dx} \right) dx = w|_{x=l} * K + w|_{x=0} * R \quad (2.13)$$

2.2. Eindige elementenmethode

De evenwichtsvergelijking (2.13) geldt als uitgangspunt voor de methode der eindige elementen. De staaf wordt hierbij opgedeeld in een eindig aantal (n) elementen. Aan het begin en einde van elk element zit een knooppunt, zodat er in totaal n+1 knooppunten zijn. De staaf wordt dus als volgt gediscretiseerd:



figuur 2.4

Als parameters van het verplaatsingsveld worden de verplaatsingen van de knooppunten gekozen ($u_i = u(x_i)$). Hiermee is het mogelijk om continuïteit van de verplaatsing over de elementsgrenzen te verkrijgen. Per element wordt de verplaatsing tussen de bij dat element behorende knooppunten geïnterpoleerd met behulp van interpolatiefuncties $N_i(x)$:

$$u(x) = \sum_{i=1}^{n+1} N_i(x) * u_i \quad (2.14)$$

Aangezien er twee knooppunten per element zijn, zijn de interpolatiefuncties van de eerste orde. Kenmerk van een interpolatiefunctie $N_k(x)$ is dat deze de waarde 1 heeft in het knooppunt k ($i=k$) en 0 in de overige knooppunten ($i \neq k$). Bij een lineaire interpolatie bestaat deze functie tussen enerzijds de knooppunten $k-1$ en k en anderzijds de knooppunten k en $k+1$ uit rechte lijnen, met bij $i=k$ een top van waarde 1, terwijl hij elders de waarde 0 heeft.

De methode van Galerkin gebruikt, om de weegfunctie $w(x)$ vast te leggen, dezelfde interpolatiefuncties als voor de verplaatsing:

$$w(x) = \sum_{i=1}^{n+1} N_i(x) * w_i \quad (2.15)$$

Hierbij zijn de factoren w_i ($i=1,2,\dots,n+1$) gelijk aan de functiewaarden van de weegfunctie $w(x)$ in de knooppunten: $w_i = w(x=x_i)$.

Voor de eenvoud wordt verondersteld dat A onafhankelijk is van de coördinaat x , terwijl E per element kan verschillen, dus E_i ($i=1\dots n$) i.p.v. E . Verder wordt er dus gewerkt met lineaire interpolatiefuncties, zodat $N_i(x) = (x_{i+1} - x) / l_i$ en $N_{i+1}(x) = (x - x_i) / l_i$ op element i ($x_i \leq x \leq x_{i+1}$). Zo kan voor het verplaatsingsveld in element i worden geschreven:

$$u(x) = \begin{bmatrix} N_i(x) & N_{i+1}(x) \end{bmatrix} \begin{bmatrix} u_i \\ u_{i+1} \end{bmatrix} \quad \text{voor } x_i \leq x \leq x_{i+1} \quad (2.16)$$

De weegfunctie $w(x)$ kent een soortgelijke uitdrukking.

$$w(x) = [N_i(x) \quad N_{i+1}(x)] \begin{bmatrix} u_i \\ u_{i+1} \end{bmatrix} \quad \text{voor} \quad x_i \leq x \leq x_{i+1} \quad (2.17)$$

De integraal in het linkerlid van (2.13) als som van integralen op elementniveau worden geschreven:

$$\sum_{i=1}^n \int_{x=x_i}^{x=x_{i+1}} \frac{dw}{dx} * E_i * A * \frac{du}{dx} dx = K * w_{n+1} + w_1 * R \quad (2.18)$$

Met gebruik van (2.16) en (2.17) verandert het linkerlid van (2.18) in:

$$\sum_{i=1}^n \int_{x=x_i}^{x=x_{i+1}} [w_i \quad w_{i+1}] \begin{bmatrix} \frac{-1}{l_i} \\ \frac{1}{l_i} \end{bmatrix} \begin{bmatrix} \frac{-1}{l_i} & \frac{1}{l_i} \end{bmatrix} \begin{bmatrix} u_i \\ u_{i+1} \end{bmatrix} * E_i * A dx \quad (2.19)$$

Als definitie van de "elementstijfheidsmatrix" \underline{K}_i voor element i wordt geponeerd:

$$\underline{K}_i = \frac{E_i * A}{l_i} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (2.20)$$

zodat het linkerlid van (2.18) wordt:

$$\sum_{i=1}^n \int_{x=x_i}^{x=x_{i+1}} \frac{dw}{dx} * E_i * A * \frac{du}{dx} dx = \sum_{i=1}^n [w_i \quad w_{i+1}] [\underline{K}_i] \begin{bmatrix} u_i \\ u_{i+1} \end{bmatrix} \quad (2.21)$$

Tenslotte wordt (2.13) dus m.b.v. (2.21):

$$\sum_{i=1}^n [w_i \quad w_{i+1}] [\underline{K}_i] \begin{bmatrix} u_i \\ u_{i+1} \end{bmatrix} = w_{n+1} * K + w_1 * R \quad (2.22)$$

Voor de verdere uitwerking worden kolommen geïntroduceerd waarin de verplaatsingen en weegfunctiewaarden van alle knooppunten zijn opgenomen:

$$\underline{u}' = [u_1 \ u_2 \ \dots \ u_k \ \dots \ u_n \ u_{n+1}] \quad (2.23)$$

$$\underline{w}' = [w_1 \ w_2 \ \dots \ w_k \ \dots \ w_n \ w_{n+1}] \quad (2.24)$$

Daarmee kan (2.22) uiteindelijk genoteerd worden als:

$$\underline{w}' [\underline{K}] \underline{u} = \underline{w}' \underline{k} \quad (2.25)$$

Hierin is \underline{K} een matrix met $n+1$ rijen en kolommen en is \underline{k} een kolom met $n+1$ componenten. De bijdrage van element i aan \underline{K} is:

$$\begin{bmatrix} k_i & -k_i \\ -k_i & k_i \end{bmatrix} \text{ met } k_i = \frac{E_i * A}{l_i} \quad (2.26)$$

Deze matrix komt op de rijen i en $i+1$ en de kolommen i en $i+1$ van \underline{K} te staan, dus de bijdrage van een element i aan de totale stijfheidsmatrix beperkt zich tot deze rijen en kolommen. Na assemblage van de bijdragen van alle elementen krijgt \underline{K} dus de vorm van een bandmatrix:

$$\underline{K} = \begin{bmatrix} k_1 & -k_1 & 0 & \cdot & \cdot & \cdot & 0 & 0 & 0 \\ -k_1 & k_1+k_2 & -k_2 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ 0 & -k_2 & k_2+k_3 & -k_3 & 0 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & 0 & -k_{n-2} & k_{n-2}+k_{n-1} & -k_{n-1} & 0 \\ 0 & 0 & \cdot & \cdot & \cdot & 0 & -k_{n-1} & k_{n-1}+k_n & -k_n \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 & -k_n & k_n \end{bmatrix} \quad (2.27)$$

De kolom \underline{k} , die de knooppuntskrachten bevat, ziet er als volgt uit:

$$\underline{k}' = [R \ 0 \ 0 \ 0 \ \dots \ 0 \ 0 \ 0 \ K] \quad (2.28)$$

Voor alle mogelijke componenten van w moet aan vergelijking (2.22) worden voldaan en dus moet gelden:

$$\left[\underline{K} \right] \underline{u} = \underline{k} \quad (2.29)$$

Dit is een stelsel van $n+1$ vergelijkingen met $n+1$ onbekenden. De onbekenden zijn de knooppuntsverplaatsingen u_k , voor $k=2,3,\dots,n$ en de eerste en laatste componenten van de kolom k . De bekende componenten van u zijn $u_1 (=0)$ en $u_{n+1} (=u_l)$, volgens de kin. randvoorwaarden.

De eerste vergelijking van het stelsel wordt weggelaten, omdat deze alleen interessant is om de kracht R uit te rekenen (u_1 is al bekend). Het stelsel dat dan overblijft telt n onbekenden bij n vergelijkingen en wordt opgedeeld naar bekende en onbekende componenten:

$$\begin{bmatrix} \underline{K}_{vv} & \underline{K}_{vp} \\ \underline{K}_{pv} & \underline{K}_{pp} \end{bmatrix} \begin{bmatrix} \underline{u}_v \\ u_p \end{bmatrix} = \begin{bmatrix} 0 \\ K \end{bmatrix} \quad (2.30)$$

Hierin is \underline{K}_{vv} een $n-1$ bij $n-1$ matrix, \underline{K}_{vp} een $n-1$ kolom, \underline{K}_{pv} een $n-1$ rij en \underline{K}_{pp} een getal, \underline{u}_v een $n-1$ kolom met vrije verplaatsingen, terwijl u_p gelijk is aan de voorgeschreven u_l .

Dit stelsel kan uitgeschreven worden:

$$\underline{K}_{vv} \underline{u}_v + \underline{K}_{vp} u_p = 0 \quad (2.31)$$

$$\underline{K}_{pv} \underline{u}_v + \underline{K}_{pp} u_p = K \quad (2.32)$$

Uitwerken van (2.31) geeft:

$$\underline{K}_{vv} \underline{u}_v = -\underline{K}_{vp} u_p \text{ met } -\underline{K}_{vp} u_p = \underline{f} \rightarrow \boxed{\underline{K}_{vv} \underline{u}_v = \underline{f}} \quad (2.33)$$

Deze laatste vergelijking wordt in de verschillende programma's gebruikt om de vrije verplaatsingen te berekenen.

2.3. Numerieke incrementstapmethode

De eindverplaatsing u_i wordt incrementeel bereikt, volgens $u_{ij}(j)=u_{ij}(j-1)+\delta u_{ij}$ ($u_{ij}(0)=0$). Deze eindverplaatsingen worden incrementstappen genoemd, waarvan er s zijn (j loopt van 1 tot en met s). Dus $u_{ij}(s)=u_i$.

Allereerst worden de vrije verplaatsingen van de knooppunten 2 t/m n bepaald met vergelijking (2.33), waarbij voor de eerste incrementstap beginelasticiteitsmoduli voor alle afzonderlijke elementen (n stuks) moeten worden opgegeven. Dit gebeurt middels een invoerprogramma. Dan wordt voor alle elementen de rek berekend, waarmee de schade in de huidige incrementstap kan worden bepaald. In elke incrementstap wordt bekeken in welk element de schade nog toeneemt en in welke element niet. Dat is afhankelijk van de schade definitie: bij de lokale definitie van de schade neemt de schade in een element toe zolang de rek in dat element toeneemt, terwijl dat bij de niet-lokale definitie het geval is zolang de gewogen rek van het element toeneemt. Als de rek (lokaal of gewogen) dus begint af te nemen in een bepaald element, dan wordt vanaf die incrementstap de schade in dat element gelijk gezet aan de schade in de vorige incrementstap.

Met deze schade per element worden vervolgens de nieuwe elasticiteitsmoduli (t.b.v. volgende stap) en de spanningen in de elementen bepaald. Dan wordt met de volgende incrementstap begonnen.

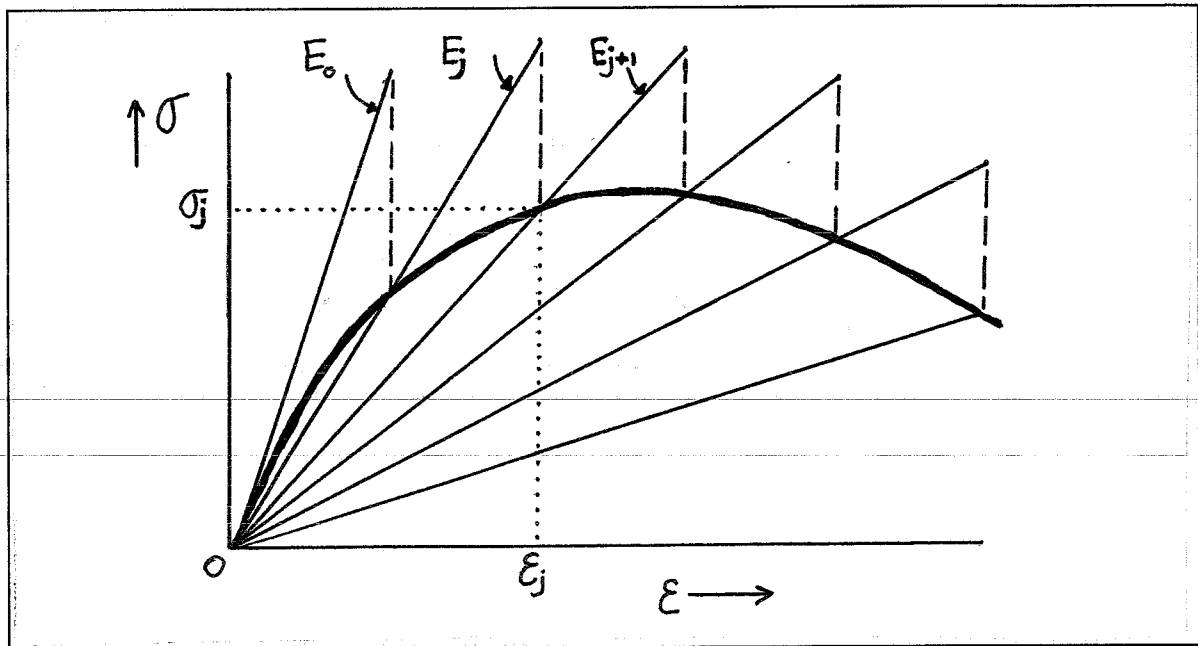
Het programma eindigt als in een der elementen de maximale rek (zelf in te voeren parameter) wordt overschreden.

In figuur 2.5 op de volgende pagina wordt getracht om grafisch de gebruikte rekenmethode weer te geven. Daarbij wordt uitgebeeld hoe de elasticiteitsmodulus voor één element varieert per incrementstap als functie van de opgelegde rekverandering en hoe daaruit de werkelijk optredende spanning in het betreffende element wordt bepaald.

Deze figuur geeft duidelijk het benaderingskarakter van de incrementstapmethode weer; Hoe kleiner de stappen, hoe nauwkeuriger.

Deze figuur geldt eigenlijk alleen bij het model met homogene schade, want dan kent elk element hetzelfde spanning-rek verloop. Daarbij treedt een volledige foutencompensatie na elke stap op en worden de spanningen perfect berekend.

Dit in tegenstelling tot het model met inhomogene schade. Hierbij zijn de spanning-rek verlopen van de afzonderlijke elementen verschillend en zal de foutencompensatie op het einde van elke incrementstap niet perfect zijn. Dit komt doordat er een onbalans optreedt in vergelijking (2.29) die residukrachten oplevert.



figuur 2.5

3. LINEAIR ELASTISCH

3.1. Modelling

Om het werken met PC-MATLAB in de vingers te krijgen en om inzicht te krijgen in het programmeren van een eindige elementenmodel, is de eerste berekening van het volledig elastische soort. Het kenmerk van een volledig elastisch model is dat de elasticiteitsmoduli van alle elementen constant zijn. De berekening kan dus eigenlijk met één incrementstap (er is géén schade-toename, dus geen E-verandering) worden uitgevoerd. Als de staaf bovendien homogeen elastisch is, dan is de elasticiteitsmodulus in alle elementen gelijk.

3.2. Numerieke resultaten

Wanneer het invoerprogramma wordt aangeroepen vraagt het om de lengte van de staaf l , het aantal elementen n , de eindlengte u_1 , de elasticiteitsmodulus E en de dwarsdoorsnede A in te geven. De lengte van één element l_1 wordt vervolgens met deze gegevens berekend. Het hoofdprogramma berekent daarna in één incrementstap de vrije verplaatsingen (mits $n \neq 1$), waaruit de rekken en de spanningen worden bepaald.

Als voorbeeld is het programma één keer doorlopen met de volgende invoergegevens:

- * $l=100$ mm
- * $n=10$
- * $u_1=5$ mm
- * $E=2 \cdot 10^3$ N/mm²
- * $A=10$ mm²

Natuurlijk kan de numerieke berekening nog gemakkelijk analytisch getoetst worden. Er geldt namelijk $\varepsilon_1 = \Delta l_1 / l_1 = \Delta l / l = 5 / 100 = 0.05 = \varepsilon$, zodat $\sigma_1 = E \cdot \varepsilon_1 = E \cdot \varepsilon = 2 \cdot 10^3 \cdot 0.05 = 100 = \sigma$ en dat blijkt numeriek ook uit te komen.

4. HOMOGENE SCHADE

4.1. Modelling

Er wordt nu gewerkt met een model waarbij sprake is van schadetoename in een element bij toenemende lokale rek:

$$D_i = f(\varepsilon_i) \quad (4.1)$$

Dit heeft tot gevolg dat ook de elasticiteitsmodulus E_i (per element) zal afnemen bij stijgende ε_i , want er geldt:

$$E_i = (1 - D_i) * E_{0i} \quad (4.2)$$

Hierbij is E_{0i} de beginelasticiteitsmodulus van element i . Uiteindelijk resultaat hiervan is dat de spanning-rek relatie een niet-lineair verloop krijgt:

$$\sigma_i = E_i * \varepsilon_i = (1 - D_i) * E_{0i} * \varepsilon_i \quad (4.3)$$

Door middel van de keuze van het verloop van schade D_i als functie van de lokale rek ε_i , kan dus het spanning-rek verloop in een element worden voorgeschreven.

Aangezien er hier nog sprake is van homogene schade, is de schade definitie voor alle elementen gelijk. Er zou dus eigenlijk kunnen worden volstaan met één enkel element. Wel zal er vanaf nu met een eindig aantal incrementstappen gewerkt moeten worden, want de schade in een element is afhankelijk van de rek in dat element en die varieert weer per incrementstap. Immers $u_{ij}(j) = u_{ij}(j-1) + \delta u_j$, waarbij j de incrementstap aangeeft, en $\varepsilon_i(j) = f(u_{ij}(j))$.

Voor de schade definitie wordt gebruik gemaakt van [2] (hoofdstuk 7). Hierin is D gedefinieerd als:

$$D = 1 - \left[\frac{\kappa_c^q - \kappa^q}{\kappa_c^q - \kappa_i^q} \right]^{\frac{1}{m}} \quad (4.4)$$

Voor de uniaxiaal belaste staaf geldt:

$$\kappa = \max \left[\kappa_i ; \varepsilon(J) \mid J \leq j \right] \quad (4.5)$$

Hierbij zijn κ_c , κ_i , q en m materiaalparameters. Als in eerste instantie ter illustratie het materiaal polystyreen (PS) wordt gekozen, dan hebben deze de waarden: $\kappa_c = 0.02$, $\kappa_i = 0$, $q = 3$ en $m = 3.25$.

Deze waarden kunnen, rekening houdende met (4.5), ingevuld worden in (4.4), hetgeen resulteert in:

$$D = 1 - \left[1 - \frac{\varepsilon^3}{8 \cdot 10^{-6}} \right]^{\frac{1}{3.25}} \quad (4.6)$$

Deze formule is slechts geldig tot en met een rek ε van 0.02 ($=\varepsilon_c$), want voor hogere rekken zou de schade negatief worden en dat is fysisch niet reëel. Bij $\varepsilon = 0.02$ is D gelijk 1 en dat is de waarde van de schade waarbij de staaf breekt ($D_c=1$).

4.2. Numerieke resultaten

Het model met homogene schade wordt nu numeriek berekend. Allereerst moet er een invoerprogramma (invoerhs.m) worden aangeroepen, waarin E_0 , q , m en A moeten zijn gedefinieerd en dat vraagt om de lengte l , het aantal elementen n , de eindverplaatsing u_1 en het aantal incrementstappen s in te geven. Het rekenprogramma rekenhs.m berekent vervolgens per incrementstap en per element achtereenvolgens de vrije verplaatsingen $u_{v,i}$, de rek, de schade, de werkelijk bij deze verplaatsingen behorende elasticiteitsmodulus (zie figuur 2.5) en daarmee de werkelijk optredende spanning. Deze variabelen worden per element i en per incrementstap j vastgelegd in de matrices U , Eps , D , E en $Span$, die de afmetingen n (aantal elementen) bij s (aantal incrementstappen) hebben.

Het totale numerieke programma wordt doorlopen met de parameters:

- * $E_0 = 3.5 \cdot 10^2 \text{ N/mm}^2$ (PS)
- * $q = 3$ (PS)
- * $m = 3.25$ (PS)
- * $A = 10 \text{ mm}^2$
- * $l = 100 \text{ mm}$
- * $n = 5$
- * $u_1 = 2 \text{ mm}$

De eerste keer wordt er gerekend met 5 incrementstappen ($s=5$) en de tweede keer met 100 ($s=100$). De resultaten zijn in bijlage 2 te zien. Enkele conclusies:

-Het verschil tussen de berekening met $s=5$ en die met $s=100$ is in de plots van de spanning als functie van de rek duidelijk te zien: bij $s=100$ worden 20 keer zoveel incrementstappen berekend als bij $s=5$, zodat het verloop een vloeiender karakter heeft bij eerstgenoemde.

-De elasticiteitsmodulus verandert bij kleine rek ($\varepsilon \leq 0.006$) nauwelijks, terwijl deze bij grote rek juist aanzienlijk verandert ($\varepsilon \geq 0.015$). Je kunt dus stellen dat het materiaal zich bij kleine rek min of meer elastisch gedraagt en dat er vanaf een bepaalde grenswaarde van de rek pas relevante schade toename is, die alsmaar goter wordt. Dit heeft zijn oorzaak in de definitie van de schade.

-Uit [1] zijn richtlijnen voor het gedrag van polystyreen onder trekbelasting genomen:

$$* E = 3 \div 3.5 \cdot 10^2 \text{ N/mm}^2$$

$$* \sigma_c = 45 \div 60 \text{ N/mm}^2$$

$$* \varepsilon_c = 3 \div 4 \text{ \%}$$

Aan de plot van de spanning tegen de rek is te zien dat de maximale spanning σ_c ongeveer 45 N/mm² is, wat goed overeenkomt met de verwachte literatuurwaarde voor polystyreen. Echter de rek bij breuk $\varepsilon_{\max} = 0.02$ wijkt daarvan enigszins af.

5. INHOMOGENE SCHADE: LOKALE WERKWIJZE

5.1. Modelling

De schade definitie zoals die bij het model met homogene schade het geval was (4.4), blijft gehandhaafd, alleen wordt er nu afgeweken van de keuze van polystyreen als materiaal. De parameter κ blijft, voor de eenvoud, gelijk 0, maar de parameters κ_c , q en m kunnen nu naar gelieve worden ingevuld. Hierdoor wordt de algemene schade definitie:

$$D = 1 - \left[1 - \left(\frac{\kappa}{\varepsilon_{end}} \right)^q \right]^{\frac{1}{m}} \quad (5.1)$$

met

$$\kappa = \max[\varepsilon(J) \mid J \leq j] \quad (5.2)$$

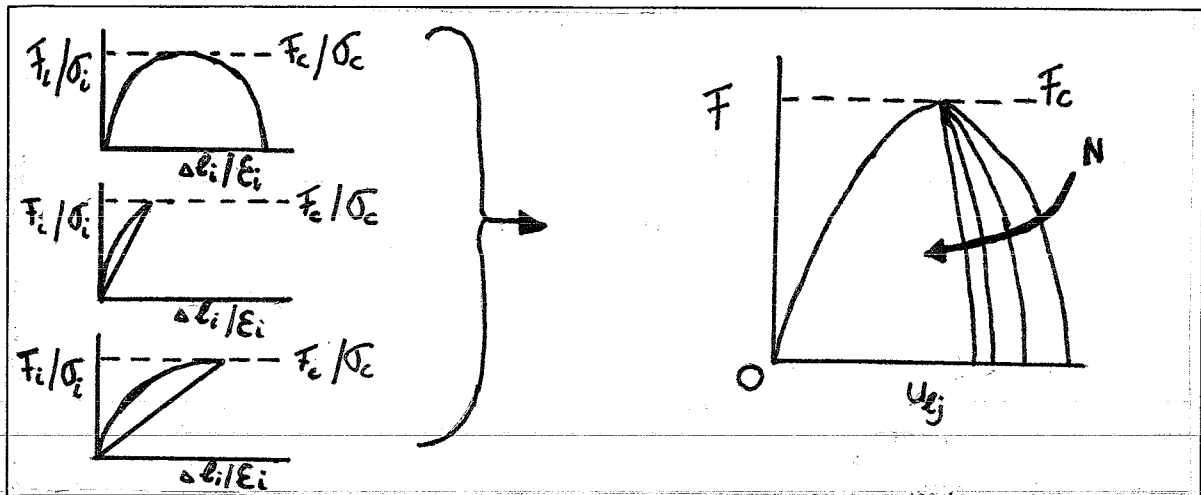
Voor κ_c is in (5.1) ε_{end} genoteerd, omdat die variabele gelijk is aan de rek waarbij de definitie van de schade nog net fysisch reëel is en waarbij deze schade dus de waarde 1 (=D_c) heeft.

Het grote verschil met het model met homogene schade, is dat de schade in een element nu een ander verloop als functie van de in dat element heersende rek heeft dan bij een ander element. Dit wordt verwezenlijkt door voor de verschillende elementen een andere keuze van de parameters q , m en ε_{end} te nemen.

5.2. Clause in het numerieke model

Er kunnen bij het lokale model van inhomogene schade, numeriek irreële oplossingen ontstaan. Wanneer namelijk één element zwakker is dan de andere, dan zal dat zwakste element (element met laagste maximale spanning σ_c) bij een bepaalde incrementstap ($j=j_c$) beginnen te verzwakken, terwijl de overige elementen gaan relaxeren, hetgeen tot objectiviteitsproblemen m.b.t. het aantal elementen leidt (zie figuur 5.1).

Het totale kracht-verlenging verloop kan dus grafisch als volgt worden opgebouwd uit de afzonderlijk (per element) kracht-verlenging grafieken:



figuur 5.1

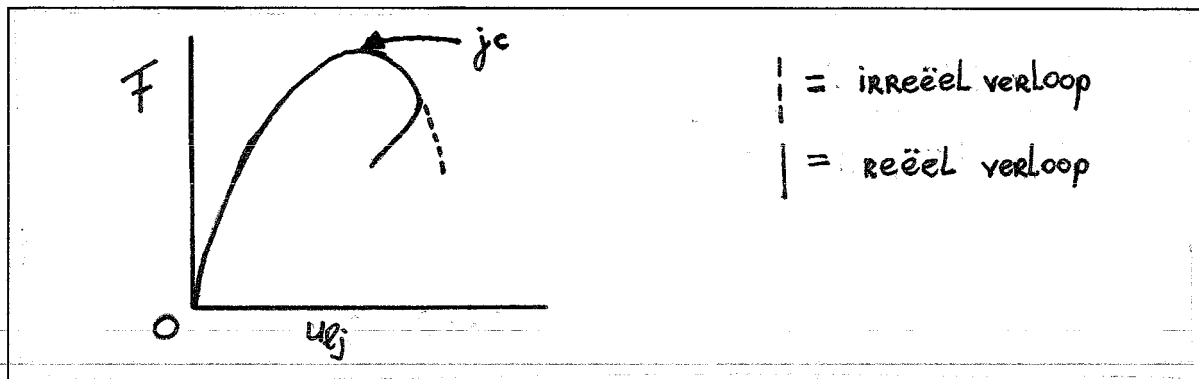
De numerieke berekening zal altijd een oplossing blijven geven, totdat in één element de maximale rek wordt overschreden en daar de staaf breekt ($j=j_{\max}$). Echter in de modelvorming zal de staaf slechts een beperkte lengteverandering kunnen ondergaan, die afhankelijk is van het aantal elementen. Wanneer namelijk het zwakste element begint te verzwakken en de overige elementen gaan relaxeren ($j=j_c$), dan zal de berekening bij een bepaalde incrementstap irreëel worden. Dan ontstaat er een situatie waarbij een verdere verlenging van de staaf niet meer mogelijk is, omdat de relaxatie bij de sterkere elementen de verlenging van het zwakste element overtreft ("snap-back punt" is overschreden; zie figuur 5.2). Dit zal des te sneller optreden naarmate:

1. Het aantal elementen groter is
2. De elasticiteitsmoduli van de sterkere elementen bij relaxatie minder groot zijn t.o.v die van het zwakste element
3. De elasticiteitsmodulus van het zwakste element sneller zal zakken bij toenemende rek.

Maar wat levert het numerieke model dan nog nadat het snap-back punt is overschreden?

In het numerieke programma wordt per incrementstap de eindverlenging iets groter dan bij de vorige incrementstap. Dit wordt voorgeschreven. Dus het programma zal nooit een situatie kunnen ontdekken waarbij een verdere verlenging niet meer mogelijk is. Nadat het snap-back punt is overschreden zullen de residukrachten (krachten in de knooppunten 2 t/m n) die na iedere stap worden bepaald groot zijn. Echter in het model zijn geen krachten aangebracht bij de knooppunten 2 t/m n, zodat de berekening foutief (irreëel) is.

De reële/irreële overgang kan als volgt grafisch worden uitgebeeld:



figuur 5.2

De overgang reëel naar irreëel gedrag kan dus worden bepaald door naar de knooppuntskrachten $2 \text{ t/m } n$ te kijken. Daartoe wordt de kolom \underline{k} die alle knooppuntskrachten bevat iedere incrementstap berekend volgens $\underline{K}u = \underline{k}$. Bij elke stap zullen enkele van de knooppuntskrachten $2 \text{ t/m } n$ iets van de perfecte waarde 0 afwijken, omdat de verplaatsingen in de huidige incrementstap, met de elasticiteitsmoduli die bepaald zijn aan de hand van de rekken in de vorige incrementstap, worden berekend (zie figuur 2.5). Nadat de verplaatsingen berekend zijn, vindt pas aanpassing van de elasticiteitsmoduli aan de rekken van de huidige incrementstap plaats.

Gesteld wordt, dat de overgang plaats vindt als de maximale absolute waarde van een knooppuntskracht in de knooppunten $2 \text{ t/m } n$ een bepaalde waarde overschrijdt.

5.3. Incrementstapgrootte

Er is in het numerieke programma rekenls.m een procedure opgenomen die het mogelijk maakt dat er met een variabele stapgrootte kan worden gewerkt:

Van $j=1$ tot en met $j=j_c$ worden de verplaatsingen voorgeschreven volgens:

$$u_j(j) = X^*(j/s) * u_1 \quad (5.3)$$

waarbij X een in te voeren parameter die de stapgrootte beïnvloedt.

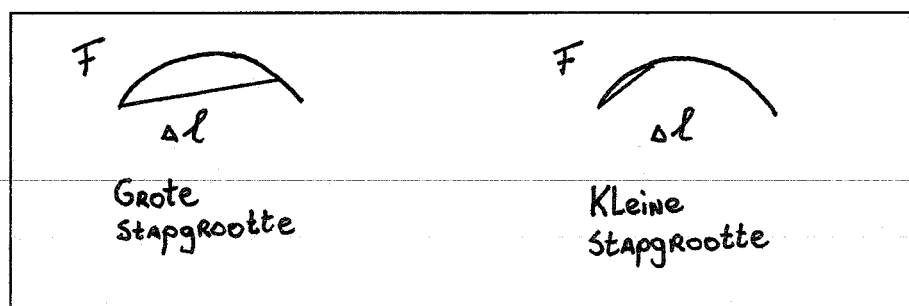
Van $j=j_c$ tot en met s geldt:

$$u_j(j) = u_j(j-1) + ((j-j_c-1) / (g * s_{rest})) * u_{lrest} \quad (5.4)$$

waarbij $s_{rest} = s - (j_c - 1)$, $u_{lrest} = u_1 - u_j(j_c - 1)$ en g de stapgrootte bepalende factor. Door voor zowel X als g een getal in te vullen dat groter dan 1 is, wordt de incrementstapgrootte vanaf j_c kleiner dan ervoor. Stap j_c wordt dus tweemaal doorlopen. De eerste keer dat j_c wordt doorlopen dient slechts ter globale indicatie van de overschrijding van de maximale spanning in een element. Het programma gaat dan de volgende loop in (van $j=j_c$ tot s), waarin de schade in de relaxerende

elementen constant wordt gehouden.

Het voordeel van het twee keer doorlopen van stap jc is, dat zo de reële/irreële overgang nauwkeuriger wordt bepaald. Op de eerste plaats is het zo, dat wanneer je pas vanaf stap $jc+1$ met kleinere stapgrootte gaat werken, dat je dan bij stap jc (grote stapgrootte) al ver in het gebied van insnoering bent beland en het best zo kan zijn dat het model dan al irreëel is geworden:



figuur 5.3

Verder werkt een kleinere stapgrootte op zich natuurlijk altijd nauwkeuriger, ook ter bepaling van de overgang.

5.4. Numerieke resultaten

Als de maximale spanning in het zwakste element wordt overschreden ($j=jc$), blijft in het numerieke programma voor de sterkere elementen de schade (en daarmee dus ook de elasticiteitsmodulus) in de volgende incrementstappen gelijk aan de waarde die deze variabele in stap $jc-1$ heeft. Immers de schade in een element neemt alleen toe als de rek in dat element toeneemt, hetgeen niet meer het geval is bij de sterkere elementen vanaf $j=jc$. Dan gaat namelijk alleen het zwakste element verzwakken en beginnen de overige elementen te relaxeren.

5.4.1. Invoer

De benodigde invoergegevens worden aangemaakt door het programma invoerls.m aan te roepen. Hierin moeten het dwarsdoorsnedeoppervlakte A , de beginelasticiteitsmoduli E_{0i} van alle elementen, de waarden q , m , ε_{end} en ε_{max} van alle elementen zijn ingevuld, zodat elk element een eigen spanning-rek verloop heeft. Verder vraagt dit invoerprogramma nadat het aangeroepen is om de lengte l , het aantal elementen n , de eindverlenging u_j , het aantal incrementstappen s en de stapgrootte-na- jc -bepalende factor g in te geven. De lengte l_i wordt dan berekend.

5.4.2. Aantonen reële/irreële overgang

Om de vermelde overgang te laten zien wordt een model gekozen waarbij de elementen slechts een beperkte verzwakking hebben. Dan kan het zwakste element namelijk niet zoveel voorgeschreven verlenging opvangen (E -modulus zwakke en sterke elementen is ongeveer gelijk), zodat te verwachten valt dat het model al snel fysisch irreëel wordt. Na een onderzoek naar de wijze waarop de schade D en het spanning-rek verloop veranderen bij verandering van de

parameters E_0 , m , q en ε_{end} (zie (5.1) en (4.3)), wordt de volgende tabel verkregen:

Soort	E_0 (N/mm ²)	m	q	ε_{end}	σ_c (N/mm ²)	ε_c
1	6000	1.1	1.1	0.02	34.03	0.0107
2	3500	1.1	1.1	0.02	19.85	0.0106
3	3500	1.1	1.2	0.02	20.95	0.0108

De eerste soort elementen is het veruit het sterkst, de tweede het zwakst en de derde zit daar tussenin. De staaf wordt opgedeeld in 5 ($n=5$) elementen. Het vijfde element is van het tweede soort, dus het zwakst, het derde element van het eerste soort, dus het sterkst, en de overige elementen van het derde soort. Verder geldt: $l=100$ mm, $A=10$ mm², $s=300$, $X=1.5$ en $g=50$. Er is dus een aanzienlijk verschil in stapgrootte! De resultaten van deze proef staan in bijlage 3.

Van een gewone trekstaaf zullen de afzonderlijke gedeeltes natuurlijk niet zo'n grote verschillen in spanning-rek verloop kennen. Dit voorbeeld dient echter slechts ter contole van de werking van het numerieke programma en om de overgang te laten zien.

Conclusies:

- Inderdaad treedt de reële/irreële overgang al vrij snel na het begin van verzwakking van het zwakste element op.
- De maximale knooppuntskracht in de knooppunten 2 t/m n t.o.v. de maximale normaalkracht is bijna 1.5 % (3 t.o.v 198.5) en neemt toe bij grotere stappen.
- De maximale normaalkracht (=198.5) is exact gelijk aan de maximale spanning in element 5 (=19.85) maal het oppervlakte A (=10).

5.4.3. Invloed van het aantal elementen

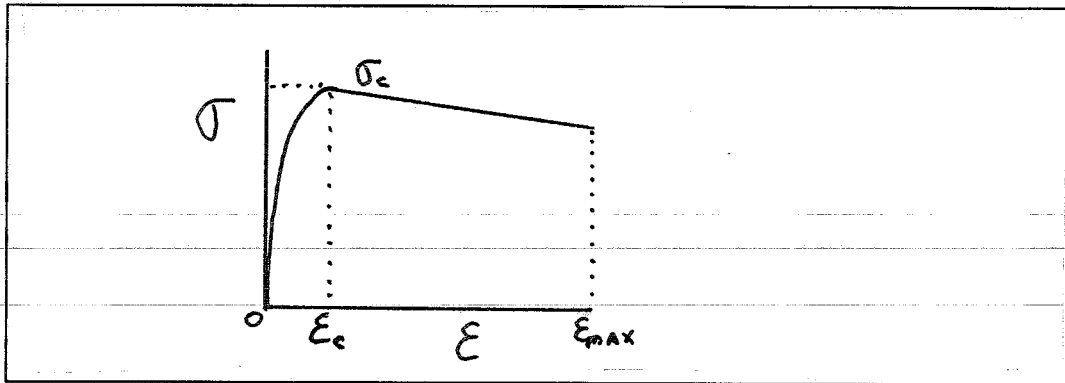
De doelstelling van deze stage was het aantonen van de sterke afhankelijkheid van het aantal elementen (dus elementgrootte) bij een lokale schade aanpak en een veel minder sterke afhankelijkheid bij de niet-lokale schade definiëring (misschien zelfs convergerend naar onafhankelijkheid). Na enig voorbereidend en illustrerend werk zijn we dus nu aangekomen bij het eigenlijke begin van de daadwerkelijke opdracht.

De afhankelijkheid kan natuurlijk alleen worden aangetoond als het model niet meteen na insnoering irreëel wordt. Er moet dus worden gezocht naar een numeriek model, waarvan (zie onderdeel 5.2):

- 1.De elasticiteitsmodulus van de sterke elementen bij insnoering groot is t.o.v die van het zwakste element.

2. De elasticiteitsmodulus van het zwakste element vanaf het begin van de insnoering niet snel zal zakken.

Hieraan voldoet het volgende spanning-rek verloop:



figuur 5.4

Zo blijft zelfs bij een groot aantal elementen het numeriek model reëel. De voorgeschreven eindverlenging blijft namelijk mogelijk, doordat per incrementstap de verlenging van het zwakste element de som van relaxaties van de sterkere elementen overtreft.

Een spanning-rek verloop zoals in figuur 5.4 kan met de definiëring van de schade volgens (5.1) worden verkregen. Wel moet dan een kleine aanpassing plaatsvinden. Tot nu toe eindigde de berekening namelijk als voor een element ε_{end} (zie (5.1) en tekst beneden (4.6)) was overschreden, dan gold voor dat element namelijk $D_c=1$. Dus was de maximale rek in een element ($=\varepsilon_{\text{max}}$) gelijk aan de schadeparameter ε_{end} . Van deze koppeling wordt nu afgestapt. Wanneer namelijk ε_{max} gelijk wordt gehouden aan 0.02 en ε_{end} groter is dan 0.02, dan is het gewenste spanning-rek verloop te verwezenlijken. Gevolg hiervan is wel dat D_c niet meer gelijk is aan 1, maar iets kleiner wordt. Aldus worden 2 soorten elementen gedefiniëerd:

Soort	E_0 (N/mm ²)	m	q	ε_{end}	ε_{max}	σ_c (N/mm ²)	ε_c
1	100e2	0.2	0.2	0.1	0.02	0.9766	3.13e-3
2	105e2	0.2	0.2	0.1	0.02	0.1025	3.13e-3

Elementsoort 1 is iets zwakker dan elementsoort 2. Om de afhankelijkheid van het aantal elementen bij het verloop van de totale kracht-verlenging grafiek duidelijk aan te tonen, wordt gekozen voor slechts één zwak element en n-1 sterke elementen. Element n/2 (bij een even aantal elementen n) of het middelste element (bij een oneven aantal elementen) wordt van het zwakke

soort (bij dit model met lokale schade is dat een willekeurige keuze).

Na 13 berekeningen (achtereenvolgens met $n=1, n=2, n=3, \dots, n=13$), wordt de volgende tabel met resultaten opgesteld. In deze tabel is F_{max} de maximale normaalkracht, F_{imax} de maximale residukracht, $u_{ij}(j_c-1)$ de verlenging van de staaf net voordat in het zwakste element de maximale spanning wordt overschreden en $u_{ij}(j_{max})$ de verlenging van de staaf op het moment van breken. In de kolom van $\Delta u_{ij}(j_{max})$ staat het verschil tussen $u_{ij}(j_{max})$ bij $n=p-1$ en bij $n=p$. Bij $n=1$ is gewerkt met alleen het zwakste element.

n	F_{max} (N)	F_{imax} (N)	$u_{ij}(j_c-1)$ (mm)	$u_{ij}(j_{max})$ (mm)	$\Delta u_{ij}(j_{max})$ (* 10^{-1} mm)
1	9.766	0	0.3108	2.003	-
2	9.766	0.1034	0.2588	1.033	9.7
3	9.837	0.0747	0.2328	0.7088	3.24
4	9.86	0.1034	0.2231	0.5472	1.62
5	9.882	0.1419	0.2181	0.4508	0.964
6	9.893	0.1811	0.2146	0.3865	0.643
7	9.906	0.2435	0.2119	0.3403	0.462
8	9.861	0.1936	0.2104	0.3056	0.347
9	9.853	0.2294	0.2088	0.2786	0.27
10	9.853	0.3292	0.2075	0.2569	0.217
11	9.86	0.4221	0.2072	0.2396	0.173
12	9.865	0.5312	0.2072	0.2249	0.147
13	9.865	0.6813	0.2063	0.2145	0.104

Vermeld dient te worden, dat tot en met de berekening met $n=7$ g de waarde 1 heeft, en daarna per berekening toeneemt, wat inhoudt dat vanaf $n=8$ na j_c telkens een kleinere stapgrootte zou worden toegepast (zie 5.3). Echter het aantal incrementstappen s neemt af met toenemend aantal elementen n , wat het gevolg is van het beperkte geheugen van de gebruikte computer. In het numerieke rekenprogramma komen namelijk veel matrices voor die de afmetingen n bij s hebben.

Uiteindelijk werken de toenemende waarde van g en de afnemende waarde van s elkaar dusdanig tegen, dat de stapgrootte voor de verschillende berekeningen ongeveer even groot is.

Gesteld kan worden dat, rekening houdende met de stapgrootte, de getallen in de tabel op 3 cijfers nauwkeurig zijn.

Grafieken zijn te vinden in bijlage 4.

5.5. Conclusie

Als eindconclusie kan getrokken worden dat inderdaad het verloop van de kracht als functie van de opgelegde verlenging sterk afhankelijk is van het aantal elementen. Tot het begin van verzwakking van het zwakste element is van deze afhankelijkheid niet veel te bemerken, omdat het zwakke en de sterke elementen tot dat moment nagenoeg dezelfde spanning-rek verlopen hebben. Pas daarna wordt de afhankelijkheid duidelijk, hetgeen uiteindelijk resulteert in een veranderlijke eindverlenging. Deze eindverlenging verandert met toenemend aantal elementen wel steeds minder, maar hij blijft dalen.

6. INHOMOGENE SCHADE: NIET-LOKALE WERKWIJZE

6.1. Modelling

In tegenstelling tot het model met lokale schade, waarbij de schade in een element een functie is van de rek in hetzelfde element (zie (5.1)), wordt bij het model met niet-lokale schade ter bepaling van de schade in een element de rek gewogen. De schade-definitie volgens [3] wordt gehanteerd:

$$D = f(\tilde{\varepsilon}) \quad (6.1)$$

In deze schade definitie is de functie f dezelfde als die in (5.1) is gebruikt en is $\tilde{\varepsilon}$ de gewogen rek:

$$\tilde{\varepsilon} = \frac{\int_{-l_D}^{l_D} g * \varepsilon \, dx_k}{\int_{-l_D}^{l_D} g \, dx_k} \quad (6.2)$$

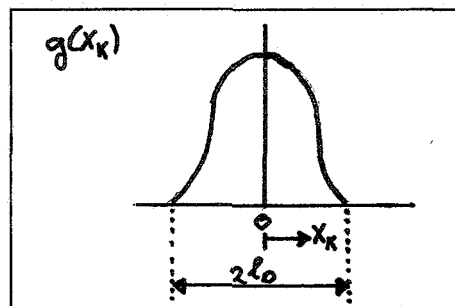
waarin:

$$g(x_k) = \exp \left[\left(\frac{-2 * x_k}{l_D} \right)^2 \right] \quad \text{als } |x_k| < l_D \quad (6.3)$$

$$g(x_k) = 0 \quad \text{als } |x_k| > l_D \quad (6.4)$$

en x_k de lokale x-coördinaat is.

De functie $g(x_k)$ ziet er grafisch uit als:

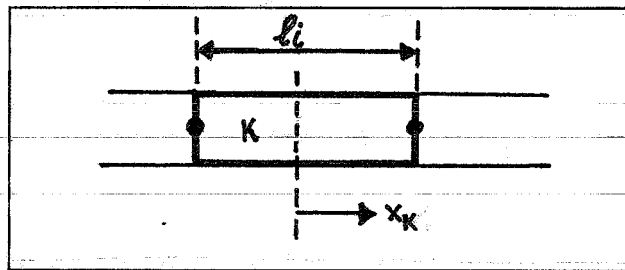


figuur 6.1

Ter bepaling van de schade in een punt wordt de rek dus gewogen over een afstand l_D naar zowel links als rechts. De rek in een punt wordt als het ware uitgesmeerd over een lengte $2 l_D$, de representatieve lengte.

6.2. Numerieke implementatie van de niet-lokale werkwijze

Numeriek wordt er met een eindig aantal (n) elementen gerekend. Elk element heeft zijn lokale x -coördinaat x_k . De nulwaarde van x_k ligt op het midden van element k :



figuur 6.2

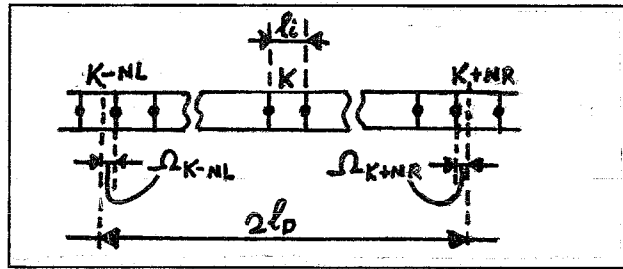
6.2.1. Numeriek wegen van de rek

Om de gewogen rek in element k te bepalen, wordt de integraal in de teller van (6.2) opgedeeld in een aantal deelintegralen:

$$\tilde{\varepsilon}_k = \frac{\sum_{i=k-nl}^{k+nr} \int_{\Omega_i} g dx_k * \varepsilon_i}{\int_{l_{Di}}^{l_{Dr}} g dx_k} \quad (6.5)$$

waarbij nl en nr afhankelijk zijn van de plaats waar element k zich bevindt (zie figuur 6.3); als k in het midden ligt, dan zijn nl en nr aan elkaar gelijk i.t.t. de situatie dat k in de buurt van een rand ligt, want dan geldt deze gelijkheid niet. Iets dergelijks geldt ook voor l_{Di} en l_{Dr} .

De rek ε mag bij deze sommatie over elementen buiten de integraal worden gelaten, omdat deze constant is over een element. Niet rekening houdende met randeffecten ($nl=nr$), is de lengte Ω_i gelijk aan de elementlengte l_i voor $i=k-(nl-1)$ tot en met $i=k+(nr-1)$ (zie figuur 6.3). Voor $i=k-nl$ en $i=k+nr$ hoeft Ω_i echter niet de gehele elementlengte te zijn. Voor deze buitenste elementen die bij het wegen van de rek meedoen, is de lengte Ω_i namelijk afhankelijk van de representatieve lengte l_D :



figuur 6.3

De lengte Ω_i voor $i=k-nl$ en $k+nr$ is dus afhankelijk van zowel de lengte l_D als van het aantal elementen n , terwijl deze voor de overige elementen alleen afhankelijk is van het aantal elementen en de totale staallengte l .

Als de lengte l_D bijvoorbeeld 30 mm is, de totale lengte l 100 mm en het aantal elementen n is 3, dan wordt de rek numeriek als volgt gewogen (ε_i is de rek in element i):

$$\tilde{\varepsilon}_1 = \frac{\int_{-16 \frac{2}{3}}^{16 \frac{2}{3}} g dx_1 * \varepsilon_1 + \int_{16 \frac{2}{3}}^{30} g dx_1 * \varepsilon_2}{\int_{-16 \frac{2}{3}}^{30} g dx_1} \quad (6.6)$$

$$\tilde{\varepsilon}_2 = \frac{\int_{-30}^{16 \frac{2}{3}} g dx_2 * \varepsilon_1 + \int_{-16 \frac{2}{3}}^{16 \frac{2}{3}} g dx_2 * \varepsilon_2 + \int_{16 \frac{2}{3}}^{30} g dx_2 * \varepsilon_3}{\int_{-30}^{30} g dx_2} \quad (6.7)$$

$$\tilde{\varepsilon}_3 = \frac{\int_{-30}^{-16 \frac{2}{3}} g dx_3 * \varepsilon_2 + \int_{-16 \frac{2}{3}}^{16 \frac{2}{3}} g dx_3 * \varepsilon_3}{\int_{-30}^{16 \frac{2}{3}} g dx_3} \quad (6.8)$$

Het gebruikte rekenprogramma is nu minder algemeen bruikbaar, omdat je telkens in dat programma de gewogen rek van de verschillende elementen moet definiëren. Daarvoor is in het programma rekennl.m namelijk geen automatische ingebouwd.

6.2.2. Numerieke schade toename

Zolang in een element de gewogen rek $\tilde{\epsilon}_i$ toeneemt, neemt ook de schade D_i in dat element toe. Pas als in een bepaalde incrementstap de gewogen rek in een element niet meer stijgt, dan blijft de schade in het betreffende element vanaf die incrementstap gelijk aan de schade in de vorige incrementstap.

6.3. Incrementstapgrootte

Om de maximale spanning en de daarbij behorende incrementstap jc nauwkeuriger te kunnen bepalen (zie figuur 5.3), wordt in het interval vlak voor jc ook al met een kleinere stapgrootte gewerkt. Ook wordt zo de maximale fout in de kolom met knooppuntskrachten k lager. Er zijn nu drie verschillende stapgroottes in het programma ingebouwd. De eerste loopt van $j=1$ tot $j=Jc$:

$$u_{ij}(j) = x \cdot (j/s) \cdot u_i \quad (6.9)$$

Met Jc wordt bedoeld die incrementstap waarbij met een grote stap wordt bepaald dat ergens de maximale spanning wordt overschreden. Vervolgens wordt van $j=Jc-a$ tot en met jc met een stapgrootte die bepaald wordt door de factor G gerekend:

$$u_{ij}(j) = u_{ij}(Jc-(a+1)) + \{(j-(Jc-(a+1)))\} / \{G \cdot srest\} \cdot ulrest \quad (6.10)$$

Hierbij is $ulrest = u_i - u_{ij}(Jc-(a+1))$ en is $srest = s - (Jc-(a+1))$. Het getal a is nog te kiezen, al naar gelang het aantal incrementstappen dat je wilt gebruiken om jc nauwkeuriger te bepalen. De incrementstap jc is dus die incrementstap waarbij met kleinere stapgrootte wordt bepaald dat de insnoering begint. Tenslotte wordt van $j=jc$ tot en met s (als de staaf nog niet gebroken is) gebruikt:

$$u_{ij}(j) = u_{ij}(jc-1) + \{(j-(jc-1))\} / \{g \cdot srest\} \cdot ulrest \quad (6.11)$$

Natuurlijk moeten dan $ulrest$ en $srest$ worden aangepast: $ulrest = u_i - u_{ij}(jc-1)$ en $srest = s - (jc-1)$.

6.4. Numerieke resultaten

Als er nu evenals bij lokale schade gewerkt wordt met één zwakker element en de rest sterkere, dan kent het rekenprogramma `rekennl.m` (behorende bij niet-lokale aanpak) t.o.v het programma `rekenls.m` (behorende bij lokale aanpak) een groot verschil in numerieke programmering. Bij het laatstgenoemde programma wordt vanaf incrementstap jc (overschrijding maximale spanning in zwakste element) de schade in de sterkere elementen constant gehouden. De rek in de sterkere elementen neemt vanaf die incrementstap namelijk af. Bij de niet-lokale aanpak echter zal de schade pas niet meer toenemen als de gewogen rek niet meer toeneemt. Dus moet ter bepaling of in een element de schade constant moet worden gehouden, bij elke incrementstap vanaf $j=jc$, in elk element bepaald worden of de gewogen rek in de huidige incrementstap groter is dan in de vorige. Zo ja, schade toename in dat element volgens (6.1), zo nee blijft vanaf de eerste keer dat dat geconstateerd wordt de schade constant.

6.4.1. Invoer

Het enige verschil met de invoer van het model met lokale schade, is dat bij het invoerprogramma bij niet-lokale schade (invoernl.m) ook de parameter G moet worden ingegeven. Deze G is de parameter die de stapgrootte bepaalde waarmee incrementstap jc wordt bepaald.

6.4.2. Beperking door numerieke werkwijze

Als er gerekend wordt met rekennl.m dan blijkt dat de gevolgde werkwijze met betrekking tot het numeriek wegen een beperking oplevert. De uitkomsten van een berekening met een even aantal elementen zijn namelijk nooit volgens verwachting. De staaf breekt bij een even aantal elementen altijd bij het eerste element, terwijl het n/2-de element in feite het zwakste is. Dit heeft zijn oorzaak in de manier van numeriek wegen van de rek, waardoor randeffecten ontstaan. Aangezien het n/2-de element niet het middelste element is, maar iets links van het midden, verplaatst het zwaartepunt van de gewogen rek zich helemaal naar links (uitwerking in bijlage 7). Hierdoor neemt daar ook de schade het snelst toe en zal de staaf in element 1 breken, wat niet logisch is. Bij een oneven aantal elementen heb je van dit probleem geen last, omdat daarbij precies het middelste element het zwakste is. Dat resulteert in een symmetrisch verdeelde gewogen rek; het middelste element m heeft in elke elementstap de grootste gewogen rek, de elementen m-1 en m+1 hebben een lagere gewogen rek die voor beide gelijk is, etc. Bij het niet-lokale model zijn dus alleen berekeningen uitgevoerd met een oneven aantal elementen.

6.4.3. Resultaten met $l_p=30$ mm

In eerste instantie heeft de representatieve lengte $2 l_D$ de waarde 60. Ter bepaling van de gewogen rek in een bepaald knooppunt, wordt de rek dus over een afstand van zowel 30 naar links als naar rechts gewogen (als dat mogelijk is; zie 6.2.1). Als voorbeeld volgt hier hoe de rek in element 4 met n=7 wordt gewogen:

$$\tilde{\epsilon}_4 = \frac{\int_{-21}^{-30} g dx_4 * \epsilon_2 + \int_{-21}^{-7} g dx_4 * \epsilon_3 + \int_{-7}^{7} g dx_4 * \epsilon_4 + \int_{7}^{21} g dx_4 * \epsilon_5 + \int_{21}^{30} g dx_4 * \epsilon_6}{\int_{-30}^{30} g dx_4} \quad (6.12)$$

Er worden 7 berekeningen uitgevoerd, waarbij n een oneven getal is (n=3, 5, 7, 9, 11, 13, 15). De berekening met n=1 levert hetzelfde resultaat op als bij het lokale schade model; er is slechts één element, dus kan er niet gewogen worden. De volgende tabel wordt verkregen, waarin de in het model met lokale schade opgesomde grootheden hier dezelfde betekenis hebben (zie 5.4.3). De grootheid $\Delta u_{ij}(jmax)$ is gelijk aan het verschil tussen $u_{ij}(jmax)$ bij n=p en n=p-1. De vermelde stapgrootte gelijk is aan de incrementstapgrootte vlak voor breken.

n	Fmax (N)	Fimax (N)	$u_{ij}(j_c-1)$ (mm)	$u_{ij}(j_{max})$ (mm)	$\Delta u_{ij}(j_{max})$ (*10 ⁻³ mm)	Stapgrootte (*10 ⁻⁴ mm)
3	10.03	0.0474	0.2947	0.7521	-	6
5	10.14	0.0450	0.3080	0.7360	-16.1	9
7	10.17	0.0363	0.3102	0.7644	28.4	11
9	10.19	0.0353	0.3105	0.7669	2.5	15
11	10.20	0.0403	0.3101	0.7692	2.3	19
13	10.21	0.03616	0.3101	0.7692	0	19
15	10.22	0.03561	0.3097	0.7727	3.5	20

Reeds bij $n=3$ geldt $g>G>X$ ($3>2>1$), zodat de incrementstapgrootte in het begin van de opgelegde verlenging het grootst is, dan vlak voor incrementstap j_c iets kleiner wordt, waarna hij tenslotte na j_c nog kleiner wordt (zie 6.3). Zowel X , G als g nemen toe met het aantal elementen en hebben als eindwaarde respectievelijk de waardes 1.5, 3 en 6. Het aantal incrementstappen neemt om dezelfde reden als bij de lokale aanpak af met het aantal elementen, zodat ook hier uiteindelijk de stapgrootte bij de verschillende elementen nagenoeg constant blijft.

Grafieken van deze berekeningen zijn in bijlage 5 te zien

Enkele conclusies:

- Aan de waardes voor Fimax in de tabel is te zien dat het opdelen van incrementstapgroottegebieden in drie intervallen wel degelijk succes heeft, want deze waardes blijken erg klein te zijn.
- Inderdaad convergeert de kracht-verlenging grafiek naar een stabiele waarde. Met een stabiele grafiek wordt bedoeld dat hij zowel qua vorm als ook qua karakteriserende waardes ($u_{ij}(j_c-1)$, $u_{ij}(j_{max})$, etc.) niet meer verandert t.o.v. de berekening voorafgaand. De grafieken van twee verschillende berekeningen kunnen dan over elkaar heen worden gelegd. Het 'niet meer veranderen' dient nader gespecificeerd te worden. Omdat de numerieke werkwijze van het benaderingssoort is, waarbij sprake is van een incrementstapmethode, zullen er altijd kleine verschillen tussen de grafieken van twee opeenvolgende berekeningen zijn. Maar als deze verschillen miniem en constant van grootte zijn, dan kan gesteld worden dat de grafiek stabiel is. Bij dit niet-lokale model wordt deze stabiele grafiek dus bij $n=9$ bereikt (zie ter verduidelijking de waardes voor $\Delta u_{ij}(j_{max})$ in de tabel). Echter ook bij $n=3$, 5 en 7 is de niet-lokale aanpak al duidelijk zichtbaar.
- De vorm van de grafiek van het verloop van de kracht als functie van de verlenging is anders dan die met de lokale aanpak wordt aangemaakt.

6.4.4. Resultaten met $l_D=10$ mm.

Nu wordt er gekeken hoe de stabiele eindwaarde, die met de niet-lokale aanpak wordt verkregen, van de representatieve lengte $2 l_D$ afhangt. Daarvoor wordt voor l_D de waarde 10 mm genomen.

Welke gevolgen heeft dit voor het numeriek wegen van de rek? Voor zowel $n=3$ als $n=5$ is de gewogen rek van een element precies hetzelfde als de gewone rek. De elementlengte van beide l_i is dan $33 \frac{1}{3}$ mm respectievelijk 20 mm, zodat de representatieve lengte $2 l_D=20$ mm nog over de elementlengte valt en er helemaal niet gewogen wordt. Pas vanaf $n=7$ ($l_i=14 \frac{2}{7}$ mm) is de representatieve lengte groter dan de elementlengte en zal het model naar een stabiele krachtverlenging grafiek beginnen te convergeren.

Verwacht wordt dus dat, nu de gewogen rek van de overige elementen (naar links en rechts) minder invloed zal ondervinden van de insnoering van het zwakke element, het model een lokaler karakter zal krijgen. De eerste berekeningen zullen volkomen gelijk zijn aan die met een lokale aanpak, zodat deze niet in onderstaande tabel zijn opgenomen. De uiteindelijke stabiele krachtverlenging grafiek zal er eentje zijn met een kleinere eindverlenging dan bij $l_D=30$.

Er worden 10 berekeningen verricht die de volgende tabel opleveren:

n	Fmax (N)	Fimax (N)	$u_{ij}(j_c-1)$ (mm)	$u_{ij}(j_{max})$ (mm)	$\Delta u_{ij}(j_{max})$ (* 10^{-3} mm)	Stapgrootte (* 10^{-4} mm)
7	9.961	0.1319	0.2312	0.351	-	4
9	10.09	0.2541	0.2558	0.3157	-35.3	3
11	10.15	0.5486	0.2726	0.3256	9.9	3
13	10.19	0.5552	0.2807	0.3484	22.8	4
15	10.20	0.4057	0.2843	0.3667	18.3	6
17	10.22	0.2658	0.2875	0.3741	7.4	7
19	10.22	0.2083	0.2897	0.3772	3.1	7
21	10.23	0.2699	0.2918	0.3816	4.4	8
23	10.24	0.3023	0.2935	0.3871	5.5	9
25	10.24	0.2904	0.2964	0.3877	0.6	10

De beginwaarde bij $n=7$ van zowel X, G als g is 1. Daarna zullen deze geleidelijk toenemen tot ze bij $n=25$ uiteindelijk 2, 2 en 7 zijn. Steeds zal gelden $g > G \geq X$. Met betrekking tot de invloed van het afnemend aantal incrementstappen en de toenemende X, G en g, op de stapgrootte kan weer hetzelfde gezegd worden als bij 6.4.3.

Grafieken zijn te vinden in bijlage 6.

Enkele conclusies:

- Volgens verwachting heeft het niet-lokale model met $l_D=10$ mm een lokaler karakter dan het niet-lokale model met $l_D=30$ mm. Dit uit zich in het pas bij een hoger aantal elementen (zeg maar (bij $n=17$)) stabiel worden van de kracht-verlenging grafiek.
- Ook de veronderstelling dat de stabiele eindverlenging bij $l_D=10$ mm kleiner is dan bij $l_D=30$ mm wordt bewezen.

7. CONCLUSIE

Een CDM-model met een niet-lokale aanpak (schade afhankelijk van gewogen rek) is bij de berekening van een lineaire trekstaaf een slechte methode. Ze leidt namelijk alleen tot juiste, objectieve resultaten als er gerekend wordt met een oneven aantal elementen, waarbij de rekverdeling symmetrisch is, dus als het zwakste element het middelste is.

De schrijvers van het Parijse artikel gaan echter uit van een inhomogenere situatie. Mogelijk dat als elk element een ander spanning-rek verloop kent, dat dan betere resultaten worden verkregen.

Met het hier beschouwde één-dimensionale voorbeeld kan alleen de invloed van de elementstapgrootte op de objectiviteit worden aangetoond (lokaal) en soms worden opgelost (niet-lokaal). Aan te bevelen is daarom een twee-dimensionaal onderzoek, zodat tevens de rol van de meshoriëntatie bij het optreden van objectiviteitsproblemen kan worden bepaald.

Ook kan nog worden nagegaan hoe de niet-lokale aanpak in een standaard-elementenmethode pakket kan worden geïmplementeerd en hoe een iteratieve procedure t.b.v een nog verdere reductie van de residuele krachten er uit moet zien.

LITERATUUR

- [1] Adriaansen, W.
Kunststof en Rubber jaarboek 1988.
Kunststof- en Rubberinstituut TNO, Rotterdam 1988.

 - [2] Brekelmans, W.A.M.
Continuum Damage Mechanica,
collegedictaat trimester 4.1.
TUE, 1990.

 - [3] Brekelmans W.A.M. en Schreurs P.J.G.
Elasticiteit en inleiding plasticiteit,
collegedictaat trimester 2.1.
TUE, 1985.

 - [4] Saouridis C. en Mazars J.
A multiscale approach to distributed damage and its
usefulness for capturing structural size effect.
Laboratoire de Mécanique et Technologie /
Université Paris 6, Cachan Frankrijk.
-

BIJLAGEN**BIJLAGE 1. GEBRUIKTE PROGRAMMATUUR BIJ INHOMOGENE SCHADE.****Bijlage 1A. Lokale werkwijze.**Invoerprogramma invoerls.m:

```

%programma invoerls.m

l=input('lengte van de staaf in millimeters: ');
n=input('aantal elementen: ');
ul=input('verplaatsing uiteinde van de staaf in mms.: ');
s=input('aantal incrementstappen: ');
%De parameter G die de stapgrootte vlak voor j=jc bepaalt.
G=input('G (=parameter voor incrementstapgrootte vlak voor jc): ');
%De parameter g die de stapgrootte na j=jc bepaalt.
g=input('g (=parameter voor incrementstapgrootte na jc): ');
%
%
%De doorsnede is overal constant.
%
A=10;
%
%Aanmaken van begin-elasticiteitsmodulus-kolom
%          -q-kolom
%          -m-kolom
%          -Epsmax-kolom
%
E0=zeros(n,1);
q=zeros(n,1);
m=zeros(n,1);
Epsmax=zeros(n,1);
Epsend=zeros(n,1);
for i=1:n
    E0(i)=105e2;    %Default waardes
    q(i)=0.2;
    m(i)=0.2;
    Epsmax(i)=0.02;
    Epsend(i)=0.1;
end;
%
```

```

%Mogelijkheid tot veranderen van E-modulus, q, m en Epsc van element i
%N.B.: # q>1
%      # m>1
%
M=round(n/2+0.25);
E0(M)=100e2;
%Epsmax()=;
%
%lengte van een element in referentietoestand
%
li=l/n;
%
%save invoerls
%
incrstep=zeros(s+1,1);
incrstep(1)=0;
for j=2:s
    incrstep(j)=j;
end;

```

Rekenprogramma rekenls.m:

```

%Programma rekenls.m.
%
%Plaats reserveren.
%
Kvv=zeros(n-1,n-1);
Kpv=zeros(1,n-1);
Kvp=zeros(n-1,1);
f=zeros(n-1,1);
uv=zeros(n,1);
u=zeros(n+1,1);
U=zeros(n+1,s);
Eps=zeros(n,s);
eps=zeros(n,1);
D=zeros(n,s);
E=zeros(n,s+1);
F=zeros(s-1,1);
Fi=zeros(n+1,s-1);
Fimax=zeros(s-1,1);
fi=zeros(n+1,1);
Span=zeros(n,s);
ulj=zeros(s,1);

```

```

%Dit zijn default waardes:
a=1;          %Om bij te houden in welke incrementstap het programma
              %ongeveer is.
c=zeros(n,1); %Om aan te geven welk(e) element(en) het zwakst zijn.
d=0;          %Om programma te stoppen als staaf is gebroken.
jc=1e3;       %Om de kritische incrementstap vast te leggen.
jmax=1;       %Om de maximale incrementstap vast te leggen.
jmaxreal=1e3; %Om de maximale reele incrementstap vast te leggen.
Fimaxreal=1e3; %Om de overgang reeel-irreel te bepalen.
e=1;          %Ter indicatie van de reele-irrele overgang en
%            om programma dan eventueel te stoppen.
%
%Begin met tijdmeting en voer berekening uit per iteratiestap.
%Bepaal middelste element m.
%
t0=clock;
%
%Voer eerst de berekening uit per incrementstap tot en met jc.
%Reset K voor iedere incrementstap.
%
for j=1:s
    if j<jc
        if j==a*100; %Ter indicatie in welke incrementstap het programma is.
            b=a*100;
            B=(a+1)*100;
            disp([num2str(b), ' < j < ', num2str(B)]);
            a=a+1;
        end;
        K=zeros(n+1,n+1);
%
%    Voer berekening uit per element.
%
        for i=1:n
            % Beginelasticiteitsmodulus aanmaken.
            if j==1
                E(i,j)=E0(i);
            end;
%
%            Bepaal de elementstijfheidsmatrices per incrementstap.
%
            k=E(i,j)*A/li;
            Ke=[k -k;
                -k k];
        end;
    end;
end;

```

```

%
% Assembleer de elementstijfheidsmatrices tot K.
%
% K([i i+1],[i i+1])=K([i i+1],[i i+1])+Ke;
end;
%
% Bereken de knooppuntskrachten uit de vorige incrementstap.
% Bepaal de maximale knooppuntskrachten voor de knptn. 2 t/m n
% per incrementstap.
%
if j~=1
    fi=K*u;
    if n~=1
        Fimax(j-1,1)=max(abs(fi(2:n)));
    end;
    F(j-1,1)=fi(n+1);
    for i=1:n+1
        Fi(i,j-1)=fi(i);
    end;
end; %if j~=1
%
% Haal de benodigde deelmatrices uit K.
%
if n~=1
    Kvv=K([2:n],[2:n]);
    Kpv=K(n+1,[2:n]);
    Kvp=K([2:n],n+1);
end;
Kpp=K(n+1,n+1);
%
% Bepaal verplaatsing uiteinde en het rechterlid f per
% incrementstap.
%
ulj(j)=j*ul/s;
f=-Kvp*ulj(j);
%
% Bepaal de vrije verpaatsingen per incrementstap.
%
if n~=1
    uv=Kvv\f;
end;
%
% Stel verplaatsingskolom u samen per incrementstap.

```

```

%      Maak tevens een matrix U, waarvan de j-de kolom overeen-
%      komt met de kolom u bij de j-de incrementstap.
%
    if n==1
        u=[0;
           ulj(j)];
    else
        u=[0;
           uv;
           ulj(j)];
    end;
    for i=1:n+1
        U(i,j)=u(i);
    end;

%
%      Bepaal rek per element.
%
    for i=1:n
        Eps(i,j)=(u(i+1)-u(i))/li;
    end;

%
%      Bepaal schade, elasticiteitsmodulus, spanning en krachten
%      per element en per incrementstap.
%
    for i=1:n
%
%      Schade-definitie voor alle elementen tot j=jc.
%      Bepaal tevens nieuwe elasticiteitsmodulus en werkelijke
%      spanning.
%
        D(i,j)=1-(1-(Eps(i,j)/Epsend(i))^(q(i)))^(1/(m(i)));
        E(i,j+1)=(1-D(i,j))*E0(i);
        Span(i,j)=10*E(i,j+1)*Eps(i,j);
    end;

%
%      Bepaal of ergens de maximale spanning wordt overschreden.
%
    if j~=1
        for i=1:n
            if Span(i,j)<Span(i,j-1)
                jc=j;
                disp(['Vanaf incrementstap ',num2str(jc),' begint de insnoering.'])
            end;
        end;
    end;

```

```

        end;
        end;      %if j~=1
    end;        %if j<jc
end;          %for j=1:s
if U(2,s)~=0 & n~=1
    if ulj(s)~=ul
        disp(['De incrementstapgrootte voor j=jc is te klein gekozen:']);
    else
        disp(['De staaf snoert niet in, omdat ul te klein is.']);
    end;
end;
%
%Bepaling van zwakke element(-en);
%
ulrest=ul-ulj(jc-1);      %Resterende verlenging.
srest=s-(jc-1);          %Resterende incrementstappen.
if n~=1;
    Ulj=ulj(jc-1)+ulrest/(50*srest);
    f=-Kvp*Ulj;
    uv=Kvv\f;
    u=[0;
        uv;
        Ulj];
    for i=1:n
        eps(i)=(u(i+1)-u(i))/li;
    end;
    for i=1:n
        if eps(i)>Eps(i,jc-1)
            c(i)=1;      %c(i)=1 legt vast wat de/het zwakke
                Zwaki=i;  %element(-en) is/zijn.
        end;
    end;
end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=jc:s      %Incrementstap jc wordt opnieuw doorlopen!
    if d~=1 & e==1
        if j==a*100; %Ter indicatie waar het programma is.
            b=a*100;
            B=(a+1)*100;
            disp([num2str(b), ' < j < ', num2str(B)]);
            a=a+1;
        end;
    end;
end;

```

```

K=zeros(n+1,n+1);
%
%   Voer berekening uit per element.
%
for i=1:n
%
%   Bepaal de elementstijfheidsmatrices per incrementstap.
%
    k=E(i,j)*A/li;
    Ke=[k -k;
        -k k];
%
%   Assembleer de elementstijfheidsmatrices tot K.
%
    K([i i+1],[i i+1])=K([i i+1],[i i+1])+Ke;
end;
%
%   Bereken de knooppuntskrachten uit de vorige incrementstap.
%   Bepaal de maximale knooppuntskrachten voor de knptn. 2 t/m n
%   per incrementstap.
%
if j~=jc
    fi=K*u;
    if n~=1
        Fimax(j-1,1)=max(fi(2:n));
        if j==jc+21           %De norm voor de overgang van
            Fimaxreal=max(abs(Fimax(jc-5:jc+20)));
            Fimaxreal=3;      %reel naar irreeel modelgedrag wordt
        end;                 %vastgelegd.
        if Fimax(j-1,1)>Fimaxreal
            if e==1           %Als je na jmaxreal doorreken, wordt
                jmaxreal=j-2; %jmaxreal slechts 1 keer bepaald en
                               %mededeling slechts 1 keer geplot.
                disp(['Het model wordt irreeel na incrementstap ',num2str(jmaxreal),'.']);
            end;
            e=2;
            D(i,(jmaxreal+1))=0;
            Eps(i,(jmaxreal+1))=0;
            E(i,jmaxreal+2)=0;
            Span(i,jmaxreal+1)=0;
            ulj(jmaxreal+1)=0;
            U(i,(jmaxreal+1))=0;
            if i==n

```

```

        U(n+1,(jmaxreal+1))=0;
    end;
end;
end; %if n~=1
if e==1          %Als reele/irrele overgang nog niet
    F(j-1,1)=fi(n+1); %is bereikt worden deze variabelen
    for i=1:n+1    %gevuld.
        Fi(i,j-1)=fi(i);
    end;
end;
end; %if j~=jc
%
% Haal de benodigde deelmatrices uit K.
%
% if e==1          %Als de berekening nog reel is.
%     if n~=1
%         Kvv=K([2:n],[2:n]);
%         Kpv=K(n+1,[2:n]);
%         Kvp=K([2:n],n+1);
%     end;
%     Kpp=K(n+1,n+1);
%
% Bepaal verplaatsing uiteinde en het rechterlid f per
% incrementstap. Mogelijkheid tot gebruik van een nog kleinere
% stapgrootte vlak na jc.
%
%     if j<=jc+50
%         ulj(j)=ulj(jc-1)+(j-(jc-1))*ulrest/(g*srest);
%     else
%         ulrest2=ul-ulj(jc+50);
%         srest2=s-(jc+50);
%         ulj(j)=ulj(jc+50)+(j-(jc+50))*ulrest2/(g*srest2);
%     end;
%     f=-Kvp*ulj(j);
%
% Bepaal de vrije verpaatsingen per incrementstap.
%
%     if n~=1
%         uv=Kvv\f;
%     end;
%
% Stel verplaatsingskolom u samen per incrementstap.
% Maak tevens een matrix U, waarvan de j-de kolom overeen-
```



```

%      komt met de kolom u bij de j-de incrementstap.
%
if n==1
    u=[0;
        ulj(j)];
else
    u=[0;
        uv;
        ulj(j)];
end;
for i=1:n+1
    U(i,j)=u(i);
end;
if U(2,s)~=0 & n~=1
    if ulj(s)~=ul
        disp(['De incrementstapgrootte vanaf j=jc is te klein gekozen:'])
        g=input(['g= ',num2str(g),' , maak g kleiner, geef een nieuwe waarde in: ']);
    else
        disp(['De staaf breekt niet, omdat ul te klein is.']);
    end;
end;

%
%      Bepaal rek per element.
%
for i=1:n
    Eps(i,j)=(u(i+1)-u(i))/li;
    if Eps(i,j)>Epsmax(i)
        %d=1 is een indicatie dat ergens de
        d=1;          %maximale rek wordt overschreden en
    end;            %zorgt ervoor dat de verdere incre-
end;                %mentstappen niet worden berekend.

%
%      Geef D van het/de elemen(-ten) waarvoor geldt Eps(i,jmax)>Epsmax(i)
%      de waarde 1 en handhaaf voor alle elementen de shade zoals die
%      bij deze incrementstap is, als de staaf breekt (d=1).
%
if d==1
    for i=1:n
        jmax=j;
        if i==1
            disp(['De staaf breekt bij incrementstap ',num2str(jmax),'.']);
        end;
        for r=jmax:s

```

```

    if Eps(i,jmax)>Epsmax(i)
        D(i,r)=1;
        if r==jmax
            Eps(i,r)=inf;    %Geef de rek van de/het zwakste
        else                %elementen de waarde inf voor
            Eps(i,r)=Epsmax(i); %j=jmax en, i.v.m. plotjes, de
        end;                %waarde Epsmax(i) voor "verdere
    else                    %incrementstappen".
        if r==jmax
            Eps(i,r)=0;    %Geef de rek van de "sterke"
        end;              %elementen de waarde 0 voor
        D(i,r)=D(i,r-1);  %j=jmax.
        E(i,r+1)=E(i,r);
    end;
end;
end;    %for i=1:n
else %Dus Eps(i,j)<=Epsmax(i).
%
%    Bepaal schade, elasticiteitsmodulus, spanning en krachten
%    per element en per incrementstap.
%
for i=1:n
    if c(i)==1 | n==1
        D(i,j)=1-(1-(Eps(i,j)/Epsend(i))^(q(i)))^(1/(m(i)));
        E(i,j+1)=(1-D(i,j))*E0(i);
        Span(i,j)=10*E(i,j+1)*Eps(i,j);
%
%    Vanaf j=jc-1 blijft de schade in de sterkere elementen
%    onveranderd.
%
    else %dus c(i)=0;
        D(i,j)=D(i,j-1);
%
%    E en spanning worden berekend voor de sterke elementen.
%
        E(i,j+1)=(1-D(i,j))*E0(i);
        Span(i,j)=10*E(i,j+1)*Eps(i,j);
    end;
end;
end;    %if d=1
end;    %if e=1
end;    %if d~=1 & e=1
end;    %for j=jc:s

```

```

%
%Bepaal rekestijd
%
t=etime(clock,t0);
disp(['De rekestijd is ',num2str(t),' sec.'])
%
%save rekenls

```

Bijlage 1B. Niet-lokale werkwijze.

Invoerprogramma invoerls.m:

```

%programma invoerls.m

l=input('lengte van de staaf in millimeters: ');
n=input('aantal elementen: ');
ul=input('verplaatsing uiteinde van de staaf in mms.: ');
s=input('aantal incrementstappen: ');
%De parameter G die de stapgrootte vlak voor j=jc bepaalt.
G=input('G (=parameter voor incrementstapgrootte vlak voor jc): ');
%De parameter g die de stapgrootte na j=jc bepaalt.
g=input('g (=parameter voor incrementstapgrootte na jc): ');
%
%
%De doorsnede is overal constant.
%
A=10;
%
%Aanmaken van begin-elasticiteitsmodulus-kolom
%           -q-kolom
%           -m-kolom
%           -Epsmax-kolom
%
E0=zeros(n,1);
q=zeros(n,1);
m=zeros(n,1);
Epsmax=zeros(n,1);
Epsend=zeros(n,1);
for i=1:n
    E0(i)=105e2;    %Default waardes
    q(i)=0.2;
    m(i)=0.2;
    Epsmax(i)=0.02;

```

```

    Epsend(i)=0.1;
end;
%
%Mogelijkheid tot veranderen van E-modulus, q, m en Epsc van element i
%N.B.: # q>1
%      # m>1
%
M=round(n/2+0.25);
E0(M)=100e2;
%Epsmax()=;
%
%lengte van een element in referentietoestand
%
li=1/n;
%
%save invoerls
%
incrstep=zeros(s+1,1);
incrstep(1)=0;
for j=2:s+1
    incrstep(j)=j;
end;

```

Rekenprogramma rekennl.m voor $l_D=30$ mm en $n=5$:

```

%Programma rekennl.m.
%
%Plaats reserveren.
%
Kvv=zeros(n-1,n-1);
Kpv=zeros(1,n-1);
Kvp=zeros(n-1,1);
f=zeros(n-1,1);
uv=zeros(n,1);
u=zeros(n+1,1);
U=zeros(n+1,s);
Eps=zeros(n,s);      %gewone rek
EPS=zeros(n,s);     %gewogen rek
D=zeros(n,s);
E=zeros(n,s+1);
Fi=zeros(n+1,s-1);
Fimax=zeros(s-1,1);
fi=zeros(n+1,1);

```

```

Span=zeros(n,s);
ulj=zeros(s,1);
%Dit zijn default waardes:
a=1;          %Om bij te houden in welke incrementstap het programma
              %ongeveer is.
c=zeros(n,1); %Om aan te geven welk(e) element(en) het zwakst zijn.
d=0;         %Om programma te stoppen als staaf is gebroken.
Jc=1e3;      %Om de kritische incrementstap vast te leggen.
jc=1e3;      %idem.
jmax=1;      %Om de maximale incrementstap vast te leggen.
jmaxreal=1e3; %Om de maximale reele incrementstap vast te leggen.
Fimaxreal=1e3; %Om de overgang reeel-irreeel te bepalen.
e=1;         %Ter indicatie van de reele-irreele overgang en
%           om programma dan eventueel te stoppen.
%
%Begin met tijdmeting en voer berekening uit per iteratiestap.
%
t0=clock;
%
%Voer eerst de berekening uit per incrementstap tot en met Jc.
%Reset K voor iedere incrementstap.
%
for j=1:s
    if j<Jc
        if j==a*100; %Ter indicatie in welke incrementstap het programma is.
            b=a*100;
            B=(a+1)*100;
            disp([num2str(b),' < j < ',num2str(B)]);
            a=a+1;
        end;
        K=zeros(n+1,n+1);
%
%   Voer berekening uit per element.
%
        for i=1:n
            % Beginelasticiteitsmodulus aanmaken.
            if j==1
                E(i,j)=E0(i);
            end;
%
%   Bepaal de elementstijfheidsmatrices per incrementstap.
%
            k=E(i,j)*A/li;

```

```

        Ke=[k -k;
            -k k];
%
%       Assembleer de elementstijfheidsmatrices tot K.
%
        K([i i+1],[i i+1])=K([i i+1],[i i+1])+Ke;
end;
%
%       Bereken de knooppuntskrachten uit de vorige incrementstap.
%       Bepaal de maximale knooppuntskrachten voor de knptn. 2 t/m n
%       per incrementstap.
%
        if j~=1
            fi=K*u;
            if n~=1
                Min=abs(min(fi(2:n)));
                Max=max(fi(2:n));
                Fimax(j-1,1)=max([Min Max]);
            end;
            for i=1:n+1
                Fi(i,j-1)=fi(i);
            end;
        end;   %if j~=1
%
%       Haal de benodigde deelmatrices uit K:
%
        if n~=1
            Kvv=K([2:n],[2:n]);
            Kpv=K(n+1,[2:n]);
            Kvp=K([2:n],n+1);
        end;
        Kpp=K(n+1,n+1);
%
%       Bepaal verplaatsing uiteinde en het rechterlid f per
%       incrementstap.
%
        ulj(j)=j*ul/s;
        f=-Kvp*ulj(j);
%
%       Bepaal de vrije verpaatsingen per incrementstap.
%
        if n~=1
            uv=Kvv\f;

```

```

end;
%
%   Stel verplaatsingskolom u samen per incrementstap.
%   Maak tevens een matrix U, waarvan de j-de kolom overeen-
%   komt met de kolom u bij de j-de incrementstap.
%
if n==1
    u=[0;
        ulj(j)];
else
    u=[0;
        uv;
        ulj(j)];
end;
for i=1:n+1
    U(i,j)=u(i);
end;
%
%   Bepaal rek per element.
%
for i=1:n
    Eps(i,j)=(u(i+1)-u(i))/li;
end;
%
%   Bepaal integralen waarmee de rek gewogen kan worden.
%
if j==1
    int1=quad('g',-10,10);
    int2=quad('g',10,30);
    int=int1+int2;
    Int=int+int2;
end;
%
%   Bepaal gewogen rek.
%
EPS(1,j)=[int1*Eps(1,j)+int2*Eps(2,j)]/int;
EPS(2,j)=[int2*Eps(1,j)+int1*Eps(2,j)+int2*Eps(3,j)]/Int;
EPS(3,j)=[int2*Eps(2,j)+int1*Eps(3,j)+int2*Eps(4,j)]/Int;
EPS(4,j)=[int2*Eps(3,j)+int1*Eps(4,j)+int2*Eps(5,j)]/Int;
EPS(5,j)=[int2*Eps(4,j)+int1*Eps(5,j)]/int;
%
%   Bepaal schade, elasticiteitsmodulus, spanning en krachten
%   per element en per incrementstap.

```

```

%
%   for i=1:n
%
%       Schade-definitie voor alle elementen tot j=Jc.
%       Bepaal tevens nieuwe elasticiteitsmodulus en werkelijke
%       spanning.
%
%       D(i,j)=1-(1-(EPS(i,j)/Epsend(i))^(q(i)))^(1/(m(i)));
%       E(i,j+1)=(1-D(i,j))*E0(i);
%       Span(i,j)=10*E(i,j+1)*Eps(i,j);
%   end;
%
%   Bepaal of ergens de maximale spanning wordt overschreden.
%   Bepaal tevens in welk element.
%
%   if j~=1
%       for i=1:n
%           if Span(i,j)<Span(i,j-1)
%               Jc=j;
%           end;
%       end;
%   end;
%   %if j~=1
% end;
%   %if j<Jc
% end;
%   %Dus Jc is bereikt!
ulrest=ul-ulj(Jc-21);
srest=s-(Jc-21);
for j=Jc-20:s
    %Bepaal de kritische incrementstap jc nauwkeuriger.
    if j<Jc
        if j==a*100; %Ter indicatie in welke incrementstap het programma is.
            b=a*100;
            B=(a+1)*100;
            disp([num2str(b),' < j < ',num2str(B)]);
            a=a+1;
        end;
        K=zeros(n+1,n+1);
%
%   Voer berekening uit per element.
%
%   for i=1:n
%
%       Bepaal de elementstijfheidsmatrices per incrementstap.
%
%       k=E(i,j)*A/li;

```



```

        Ke=[k -k;
           -k k];
%
%       Assembleer de elementstijfheidsmatrices tot K.
%
        K([i i+1],[i i+1])=K([i i+1],[i i+1])+Ke;
end;
%
%       Bereken de knooppuntskrachten uit de vorige incrementstap.
%       Bepaal de maximale knooppuntskrachten voor de knptn. 2 t/m n
%       per incrementstap.
%
if j~=Jc-20
    fi=K*u;
    if n~=1
        Min=abs(min(fi(2:n)));
        Max=max(fi(2:n));
        Fimax(j-1,1)=max([Min Max]);
    end;
    for i=1:n+1
        Fi(i,j-1)=fi(i);
    end;
end; %if j~=Jc-20
%
%       Haal de benodigde deelmatrices uit K.
%
if n~=1
    Kvv=K([2:n],[2:n]);
    Kpv=K(n+1,[2:n]);
    Kvp=K([2:n],n+1);
end;
Kpp=K(n+1,n+1);
%
%       Bepaal verplaatsing uiteinde en het rechterlid f per
%       incrementstap.
%
ulj(j)=ulj(Jc-21)+(j-(Jc-21))*ulrest/(G*srest);
f=-Kvp*ulj(j);
%
%       Bepaal de vrije verpaatsingen per incrementstap.
%
if n~=1

```

```

    uv=Kvv\*f;
end;
%
%   Stel verplaatsingskolom u samen per incrementstap.
%   Maak tevens een matrix U, waarvan de j-de kolom overeen-
%   komt met de kolom u bij de j-de incrementstap.
%
if n==1
    u=[0;
        ulj(j)];
else
    u=[0;
        uv;
        ulj(j)];
end;
for i=1:n+1
    U(i,j)=u(i);
end;
%
%   Bepaal rek per element.
%
for i=1:n
    Eps(i,j)=(u(i+1)-u(i))/li;
end;
%
%   Bepaal gewogen rek.
%
EPS(1,j)=[int1*Eps(1,j)+int2*Eps(2,j)]/int;
EPS(2,j)=[int2*Eps(1,j)+int1*Eps(2,j)+int2*Eps(3,j)]/Int;
EPS(3,j)=[int2*Eps(2,j)+int1*Eps(3,j)+int2*Eps(4,j)]/Int;
EPS(4,j)=[int2*Eps(3,j)+int1*Eps(4,j)+int2*Eps(5,j)]/Int;
EPS(5,j)=[int2*Eps(4,j)+int1*Eps(5,j)]/int;
%
%   Bepaal schade, elasticiteitsmodulus, spanning en krachten
%   per element en per incrementstap.
%
for i=1:n
%
%   Schade-definitie voor alle elementen tot j=jc.
%   Bepaal tevens nieuwe elasticiteitsmodulus en werkelijke
%   spanning.
%
    D(i,j)=1-(1-(EPS(i,j)/Epsend(i))^(q(i)))^(1/(m(i)));

```

```
E(i,j+1)=(1-D(i,j))*E0(i);
Span(i,j)=10*E(i,j+1)*Eps(i,j);
end;
%
%   Bepaal of ergens de maximale spanning wordt overschreden.
%
%   if j~=1
%       for i=1:n
%           if Span(i,j)<Span(i,j-1);
%               jc=j;
%               disp(['Vanaf incrementstap ',num2str(jc),' begint de insnoering.'])
%           end;
%       end;
%   end;
end;
%if j<jc
%for j=Jc-20:s
if U(2,s)~=0 & n~=1
if ulj(s)~=ul
disp(['De incrementstapgrootte voor j=jc is te klein gekozen.']);
else
disp(['De staaf snoert niet in, omdat ul te klein is.']);
end;
end;
ulrest=ul-ulj(jc-1);    %Resterende verlenging.
srest=s-(jc-1);        %Resterende incrementstappen.

%%%
% Berekening vanaf de kritische incrementstap, dus als insnoering is begonnen:
%%%

for j=jc:s
if d~=1 & e==1
if j==a*100;
b=a*100;
B=(a+1)*100;
disp([num2str(b),' < j < ',num2str(B)]);
a=a+1;
end;
K=zeros(n+1,n+1);
%
%   Voer berekening uit per element.
```

```

%
  for i=1:n
%
%       Bepaal de elementstijfheidsmatrices per incrementstap.
%
    k=E(i,j)*A/li;
    Ke=[k -k;
        -k k];
%
%       Assembleer de elementstijfheidsmatrices tot K.
%
    K([i i+1],[i i+1])=K([i i+1],[i i+1])+Ke;
  end;
%
%       Bereken de knooppuntskrachten uit de vorige incrementstap.
%       Bepaal de maximale knooppuntskrachten voor de knptn. 2 t/m n
%       per incrementstap.
%
  if j~=jc
    fi=K*u;
    if n~=1
      Min=abs(min(fi(2:n)));
      Max=max(fi(2:n));
      Fimax(j-1,1)=max([Min Max]);
      if j==jc+1           %De norm voor de overgang van
        Fimaxreal=2;      %reel naar irreeel modelgedrag wordt
      end;                 %vastgelegd.
      if Fimax(j-1,1)>Fimaxreal
        if e==1           %Als je na jmaxreal doorreken, wordt
          jmaxreal=j-2;   %jmaxreal slechts 1 keer bepaald en
                          %mededeling slechts 1 keer geplot.
          disp(['Het model wordt irreeel na incrementstap ',num2str(jmaxreal),'.']);
        end;
        e=2;
        D(i,(jmaxreal+1))=0;
        Eps(i,(jmaxreal+1))=0;
        E(i,jmaxreal+2)=0;
        Span(i,jmaxreal+1)=0;
        ulj(jmaxreal+1)=0;
        U(i,(jmaxreal+1))=0;
        if i==n
          U(n+1,(jmaxreal+1))=0;
        end;
      end;
    end;
  end;

```

```

    end;
end;      %if n~=1
if e==1      %Als reele/irreele overgang nog niet
    for i=1:n+1      %is bereikt wordt deze variabele
        Fi(i,j-1)=fi(i); %gevuld.
    end;
end;
end;      %if j~=jc
%
% Haal de benodigde deelmatrices uit K.
%
if e==1      %Als de berekening nog reeel is.
    if n~=1
        Kvv=K([2:n],[2:n]);
        Kpv=K(n+1,[2:n]);
        Kvp=K([2:n],n+1);
    end;
    Kpp=K(n+1,n+1);
%
% Bepaal verplaatsing uiteinde en het rechterlid f per
% incrementstap.
%
ulj(j)=ulj(jc-1)+(j-(jc-1))*ulrest/(g*srest);
f=-Kvp*ulj(j);
%
% Bepaal de vrije verpaatsingen per incrementstap.
%
if n~=1
    uv=Kvv\f;
end;
%
% Stel verplaatsingskolom u samen per incrementstap.
% Maak tevens een matrix U, waarvan de j-de kolom overeen-
% komt met de kolom u bij de j-de incrementstap.
%
if n==1
    u=[0;
        ulj(j)];
else
    u=[0;
        uv;
        ulj(j)];
end;

```

```

for i=1:n+1
    U(i,j)=u(i);
end;
if U(2,s)~=0 & n~=1
    if ulj(s)~=ul
        disp(['De incrementstapgrootte vanaf j=jc is te klein gekozen:'])
        g=input(['g= ',num2str(g),' , maak g kleiner, geef een nieuwe waarde in: ']);
    else
        disp(['De staaf breekt niet, omdat ul te klein is.']);
    end;
end;

%
%      Bepaal rek per element.
%
for i=1:n
    Eps(i,j)=(u(i+1)-u(i))/li;
    if Eps(i,j)>Epsmax(i)
        Zwaki=i;          %d=1 is een indicatie dat ergens de
        d=1;              %maximale rek wordt overschreden en
    end;                  %zorgt ervoor dat de verdere incre-
end;                      %mentstappen niet worden berekend.

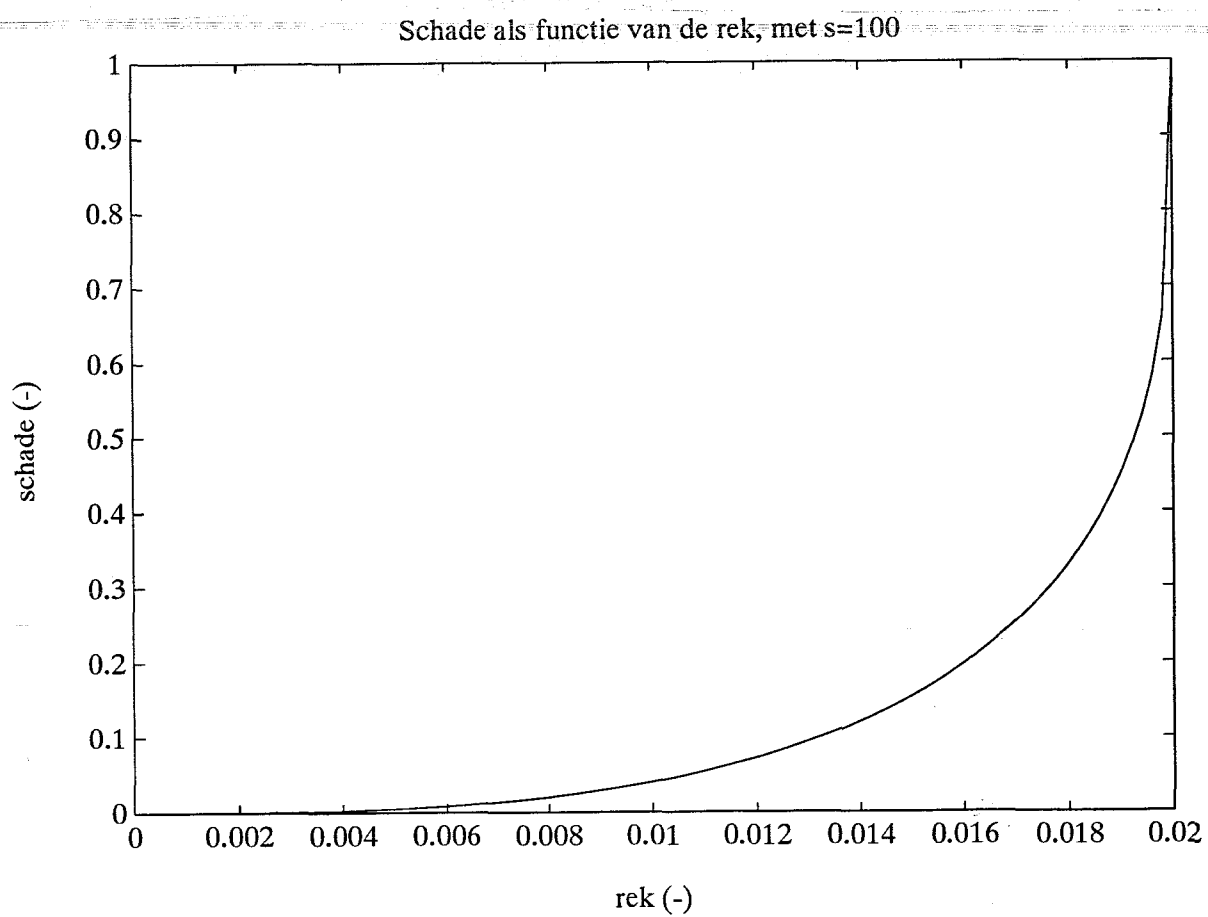
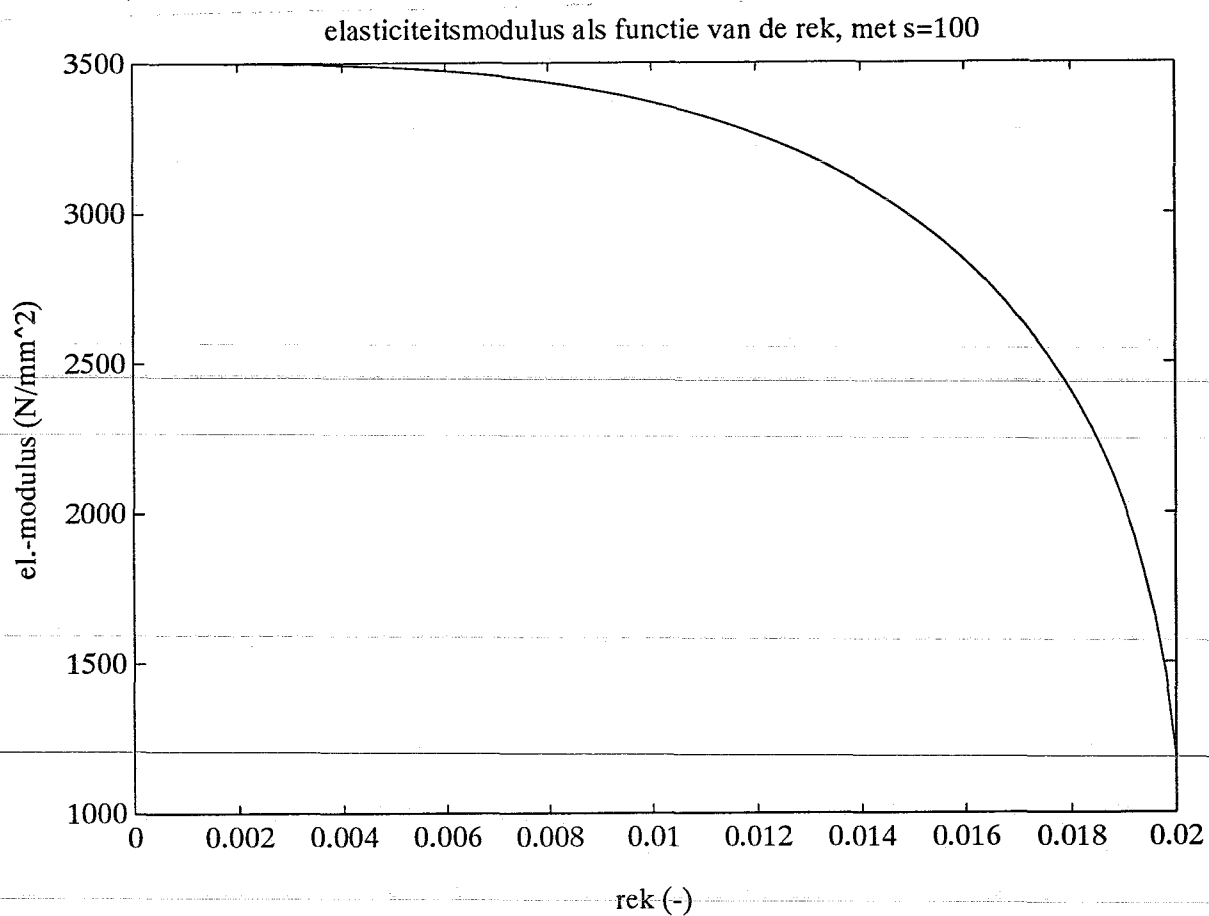
%
%      Geef D van het/de elemen(-ten) waarvoor geldt Eps(i,jmax)>Epsmax(i)
%      de waarde 1 en handhaaf voor alle elementen de shade zoals die
%      bij deze incrementstap is, als de staaf breekt (d=1).
%
if d==1
    for i=1:n
        jmax=j;
        if i==1
            disp(['De staaf breekt bij incrementstap ',num2str(jmax),'']);
        end;
        for r=jmax:s
            if Eps(i,jmax)>Epsmax(i)
                D(i,r)=1;
                if r==jmax
                    Eps(i,r)=inf;    %Geef de rek van de/het zwakste
                else                  %elementen de waarde inf voor
                    Eps(i,r)=Epsmax(i); %j=jmax en, i.v.m. plotjes, de
                end;                  %waarde Epsmax(i) voor "verdere
            else                       %incrementstappen".
                if r==jmax
                    Eps(i,r)=0;      %Geef de rek van de "sterke"
                end;
            end;
        end;
    end;
end;

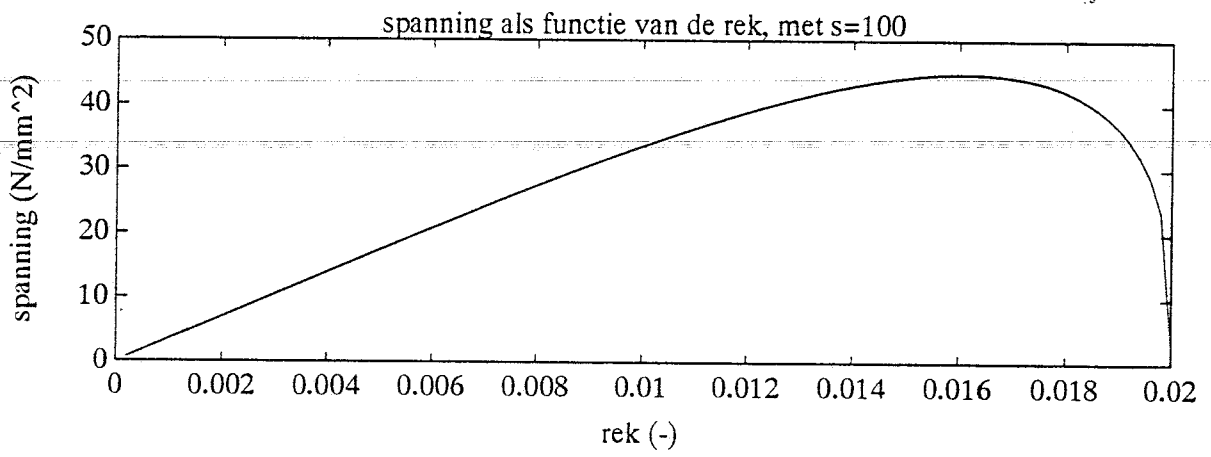
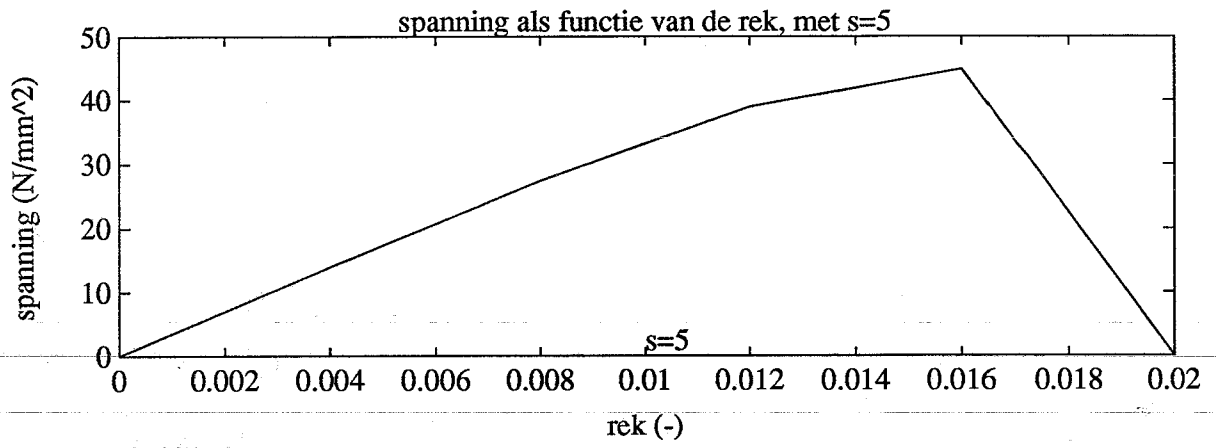
```

```

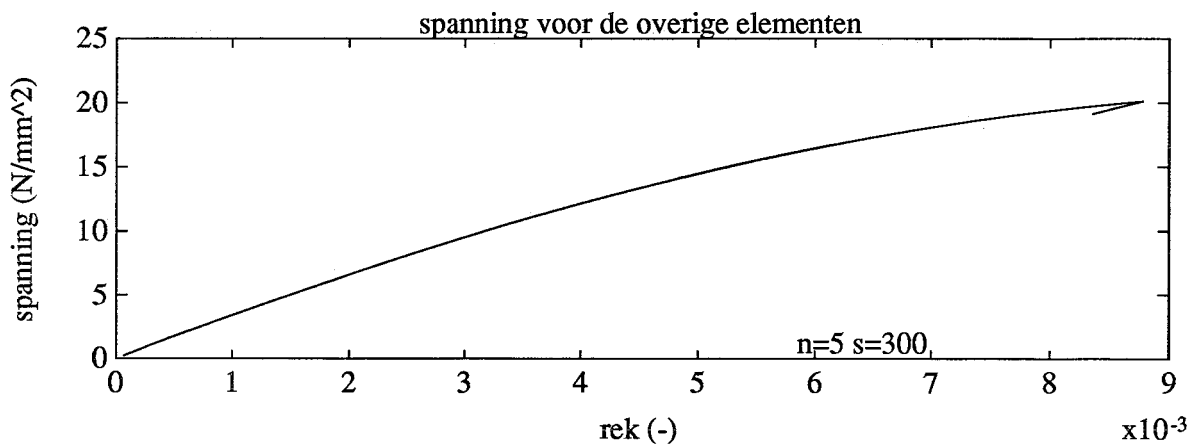
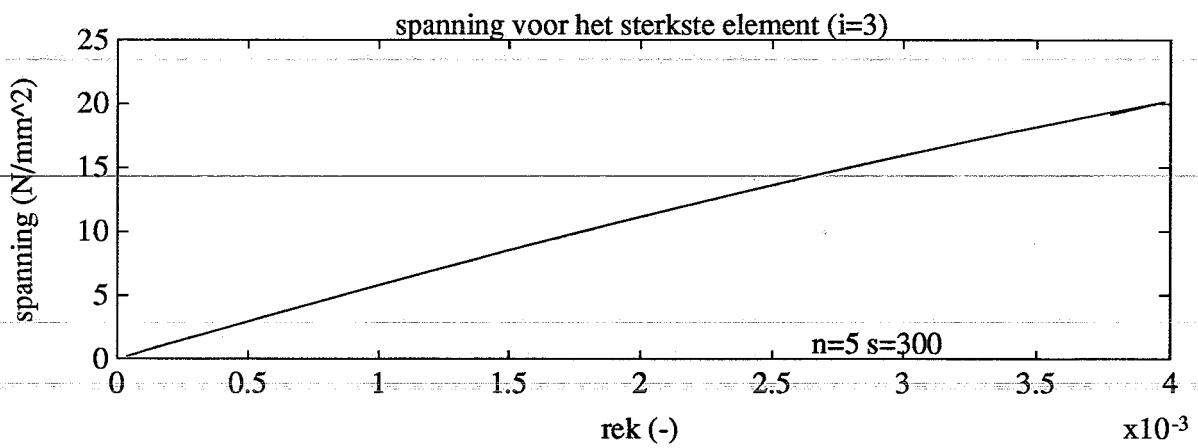
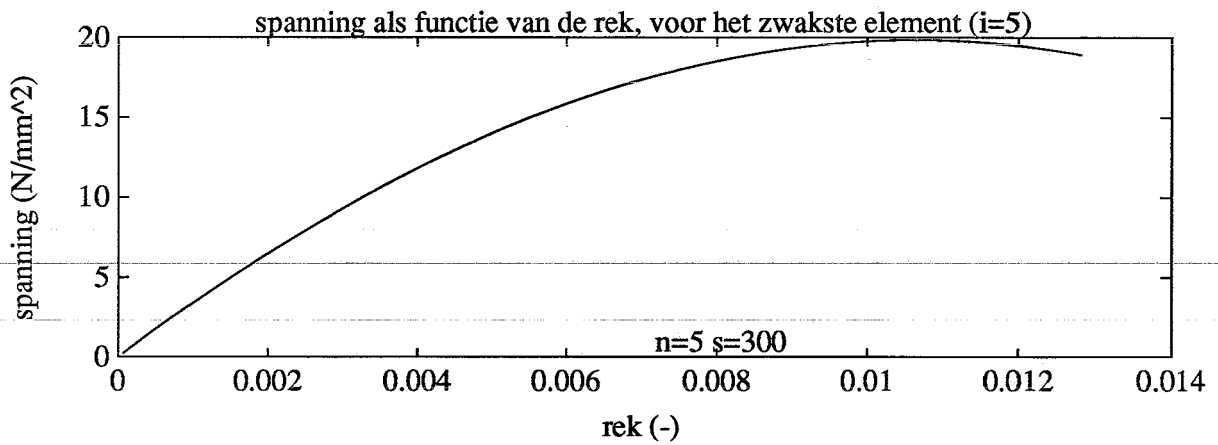
        end;                %elementen de waarde 0 voor
        D(i,r)=D(i,r-1);    %j=jmax.
        E(i,r+1)=E(i,r);
    end;
end;
end;    %for i=1:n
else %Dus Eps(i,j)<=Epsmax(i).
%
%    Bepaal gewogen rek.
%
EPS(1,j)=[int1*Eps(1,j)+int2*Eps(2,j)]/int;
EPS(2,j)=[int2*Eps(1,j)+int1*Eps(2,j)+int2*Eps(3,j)]/Int;
EPS(3,j)=[int2*Eps(2,j)+int1*Eps(3,j)+int2*Eps(4,j)]/Int;
EPS(4,j)=[int2*Eps(3,j)+int1*Eps(4,j)+int2*Eps(5,j)]/Int;
EPS(5,j)=[int2*Eps(4,j)+int1*Eps(5,j)]/int;
%
%    Bepaal in welk element de schade nog toeneemt.
%
for i=1:n
    if j~=1
        if EPS(i,j)>EPS(i,j-1)
            c(i)=1;
        end;
    end;
end;
%
%    Bepaal schade, elasticiteitsmodulus, spanning en krachten
%    per element en per incrementstap.
%
for i=1:n
    if c(i)==1 | n==1
        D(i,j)=1-(1-(EPS(i,j)/Epsend(i))^(q(i)))^(1/(m(i)));
        E(i,j+1)=(1-D(i,j))*E0(i);
        Span(i,j)=10*E(i,j+1)*Eps(i,j);
%
%    Vanaf j=jc-1 blijft de schade in de sterkere elementen
%    onveranderd.
%
    else    %dus c(i)=0;
        D(i,j)=D(i,j-1);
%
%    E en spanning worden berekend voor de sterke elementen.
%
```

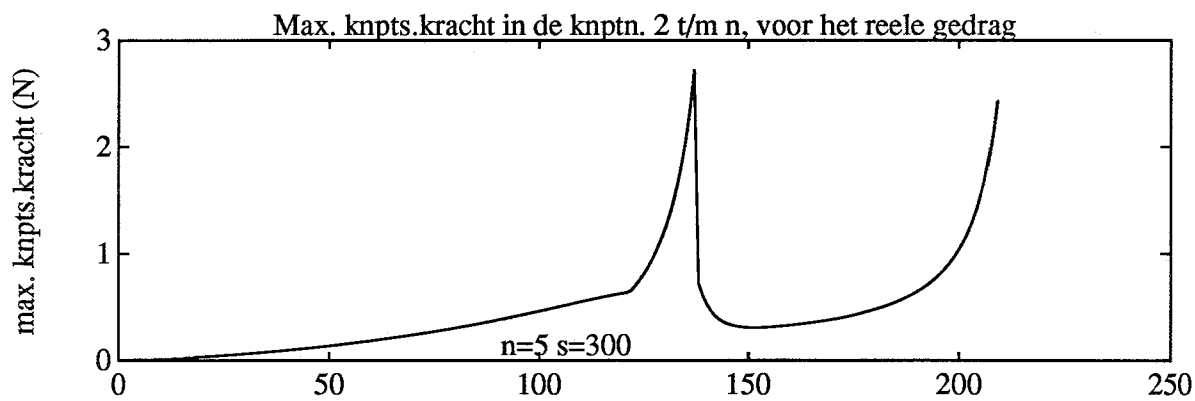
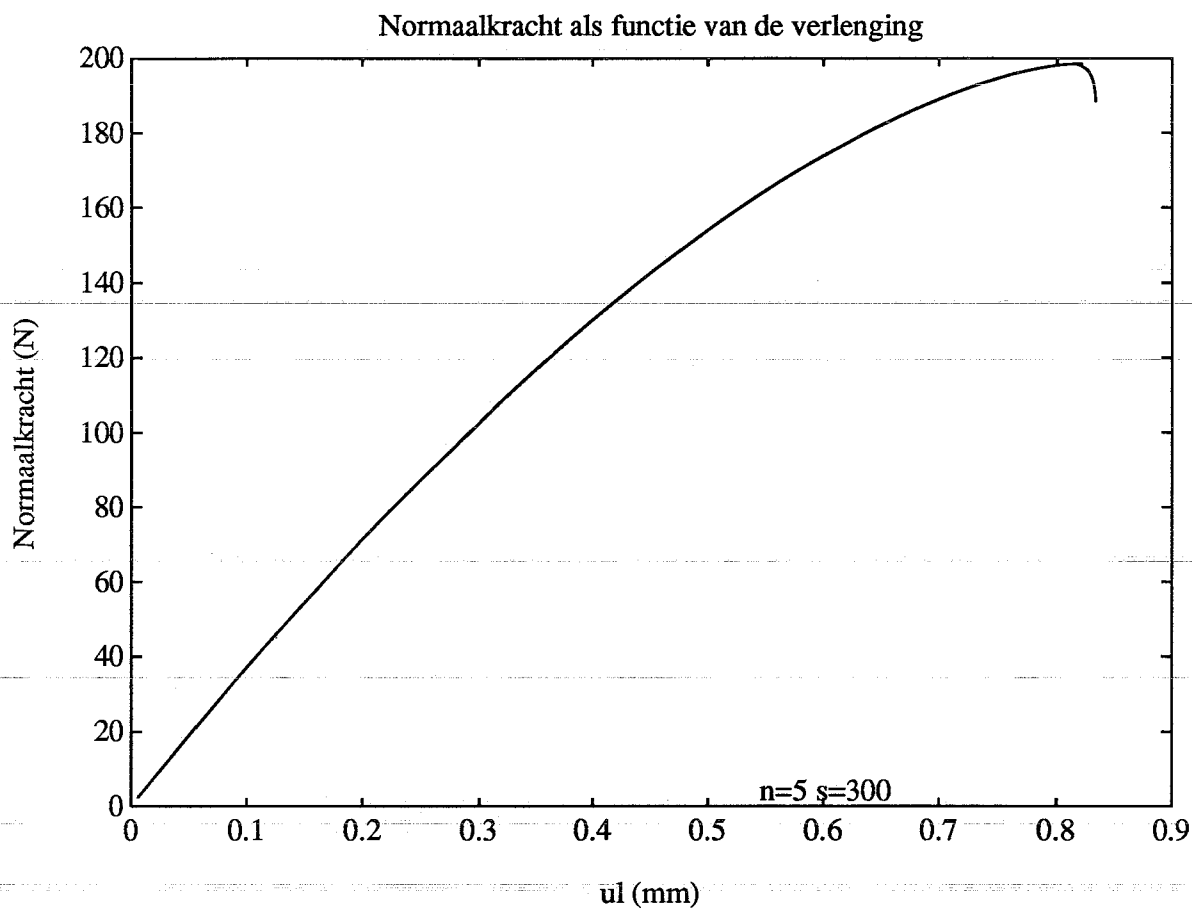
```
        E(i,j+1)=(1-D(i,j))*E0(i);
        Span(i,j)=10*E(i,j+1)*Eps(i,j);
    end;
end;    %for i=1:n
c=zeros(n,1);
end;    %if Eps<Epsmax
end;    %if e=1
end;    %if d=1
end;    %for j=jc:s
%
%Bepaal rekestijd
%
t=etime(clock,t0);
disp(['De rekestijd is ',num2str(t),' sec.'])
%
%save rekestijd
```

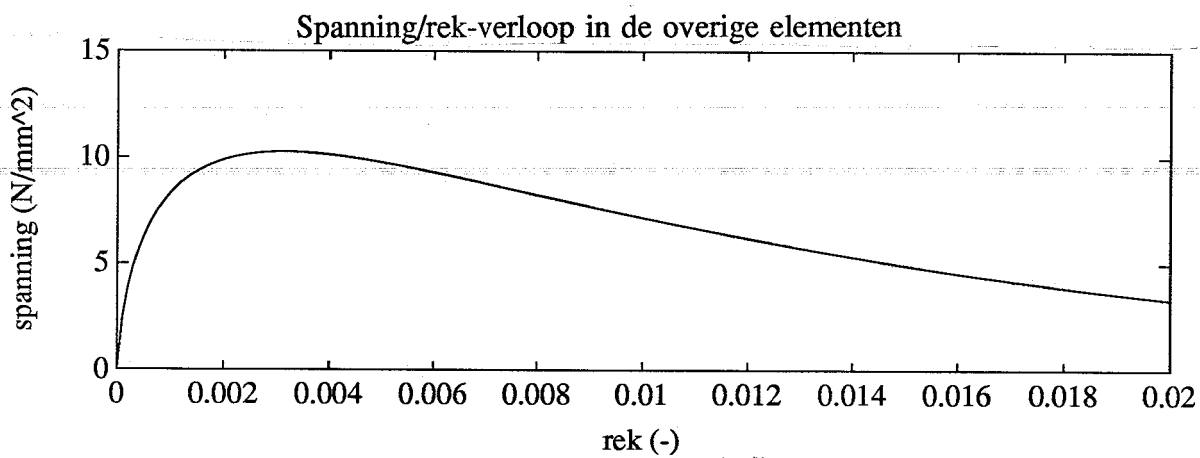
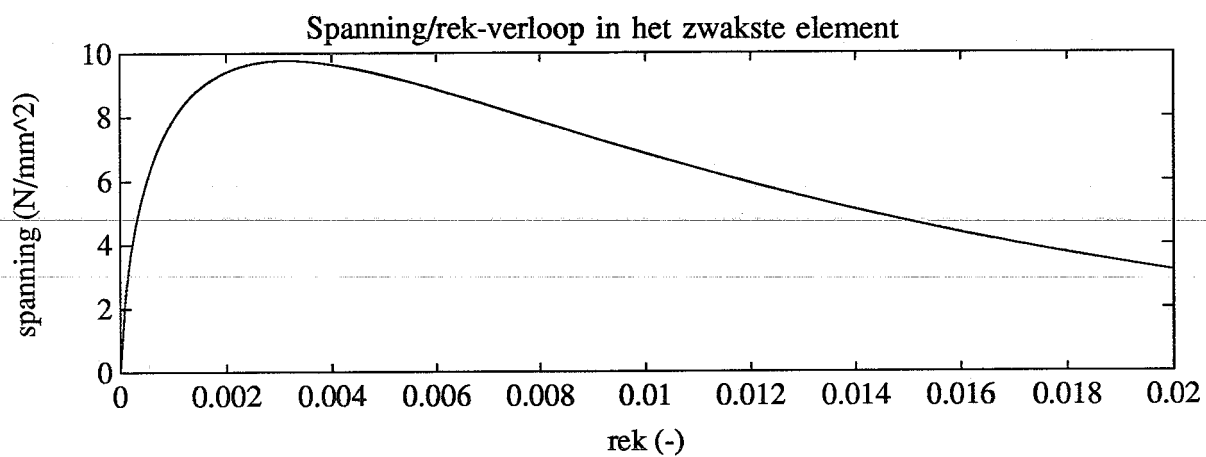

BIJLAGE 2. HOMOGENE SCHADE.

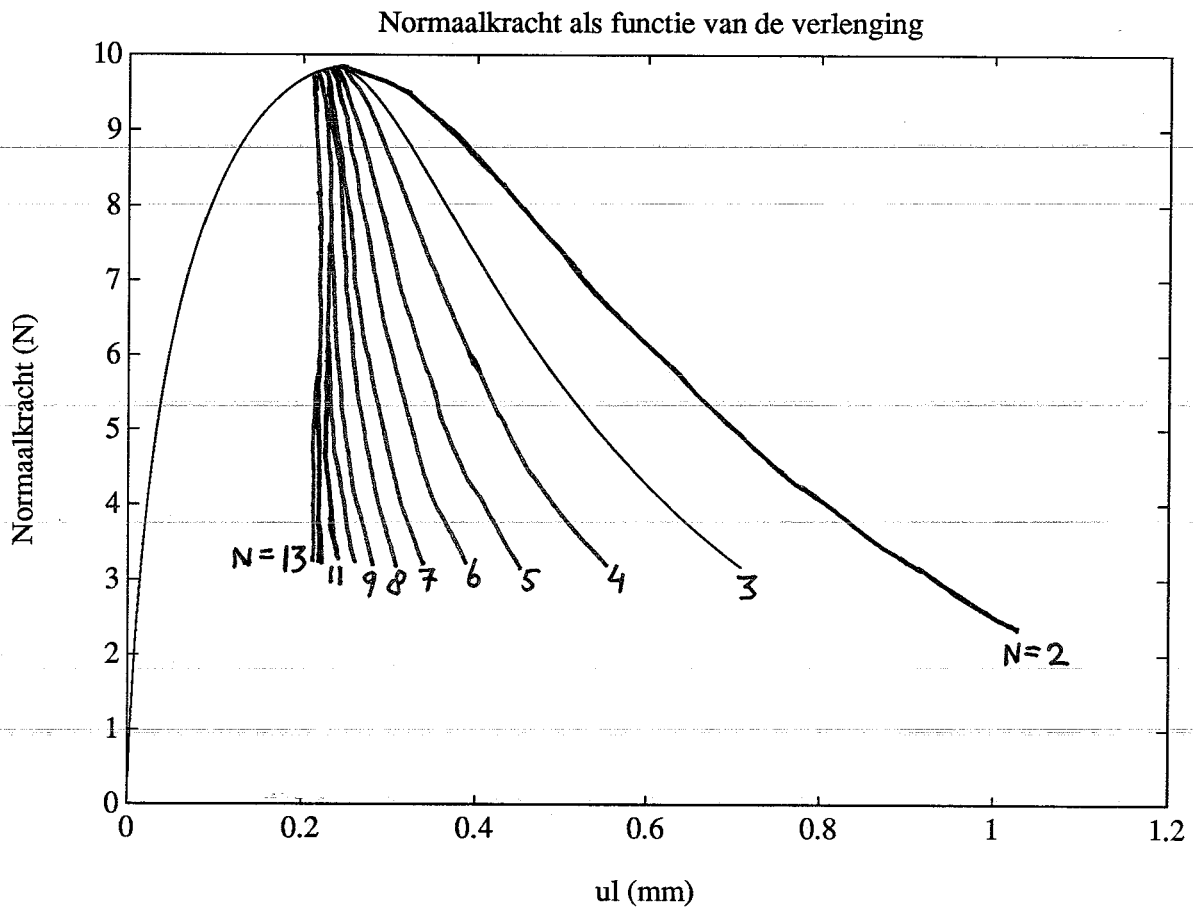


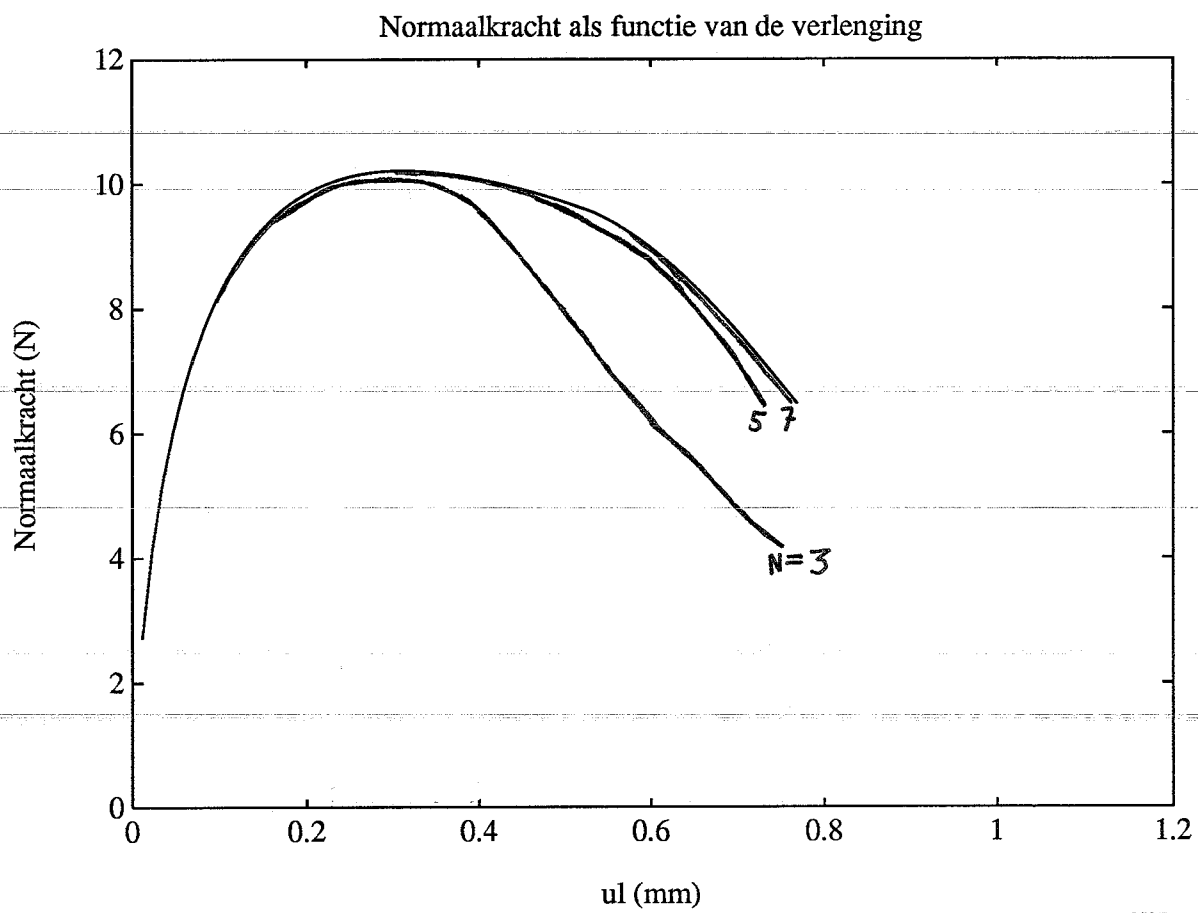
**BIJLAGE 3. INHOMOGENE SCHADE MET LOKALE AANPAK:
AANTONEN REELE/IRREELE OVERGANG.**

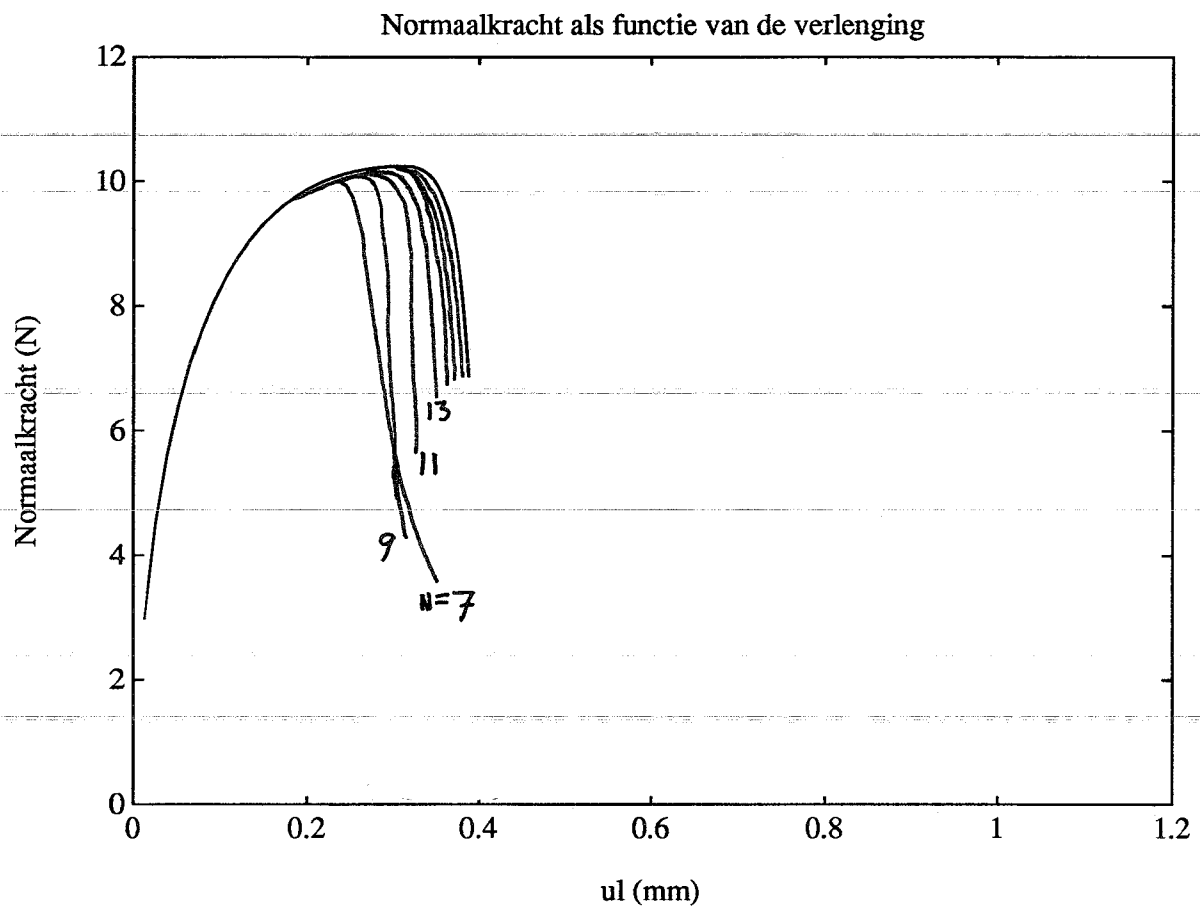




**BIJLAGE 4. INHOMOGENE SCHADE MET LOKALE AANPAK:
INVLOED VAN HET AANTAL ELEMENTEN.**



BIJLAGE 5. INHOMOGENE SCHADE MET NIET-LOKALE AANPAK EN $I_p=30$ mm.

BIJLAGE 6. INHOMOGENE SCHADE MET NIET-LOKALE AANPAK EN $l_p=10$ mm.

BIJLAGE 7. BEPERKING DOOR NUMERIEKE WERKWIJZE.

Het doel van deze bijlage is om aan te tonen dat de numerieke niet-lokale aanpak bij een even aantal elementen tot een foutieve oplossing leidt en bij een oneven aantal elementen tot een juiste oplossing leidt.

De gewogen rek in element k wordt numeriek als volgt bepaald uit de gewone rek van element k en van de omringende elementen (Zie 6.2.1.):

$$\tilde{\varepsilon}_k = \frac{\sum_{i=k-nl}^{i=k+nr} \int_{\Omega_i} g(x_k) dx_k * \varepsilon_i}{\int_{-l_{Dk}}^{l_{Dr}} g(x_k) dx_k} \quad (1)$$

Hierbij is g een weegfunctie die een verloop kent zoals in figuur 6.1 is weergegeven.

Ter vereenvoudiging wordt in deze bijlage echter niet met deze weegfunctie gewerkt, maar met weegfactoren en is de representatieve lengte $2 * l_D$ gelijk aan 5 keer de elementlengte. Dit resulteert in:

$$\tilde{\varepsilon}_k = \sum_{i=k-nl}^{i=k+nr} g_i * \varepsilon_i \quad (2)$$

waarbij

$$\sum_{i=k-nl}^{i=k+nr} g_i = 1 \quad (3)$$

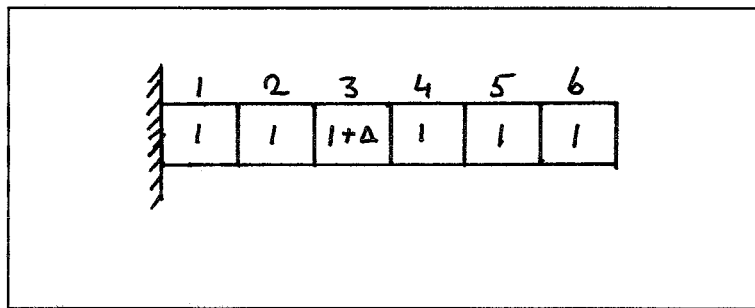
en nl en nr afhankelijk zijn van de plaats waar het element k zich bevindt, hetgeen veroorzaakt wordt door de randen. Voorbeeld: is element k een element in het midden, dan is zowel nl als nr gelijk aan 2 ($2 * l_D = 5 * l_e$), is k het eerste element, dan is nl gelijk 0 en nr gelijk 2 en als k het laatste (n -de) element is, dan is nl gelijk 2 en nr gelijk 0.

De weegfactoren g_i zijn gelijk aan: $g_k = 1/3$, $g_{k-1} = 1/4 = g_{k+1}$ en $g_{k-2} = 1/12 = g_{k+2}$.

Het principe is dus hetzelfde als beschreven in 6.2.1., alleen wordt een berg rekenwerk bespaard.

Even aantal elementen

Er wordt uitgegaan van 6 elementen. Verder is het $n/2$ -de element (dus het derde) het zwakst, zodat na één incrementstap de rekverdeling er bijvoorbeeld als volgt uit kan zien:



De gewogen rek in elk element wordt hiermee:

$$\tilde{\epsilon}_1 = (1/3 + 1/4 + 1/12*(1 + \Delta))/(2/3) = 1 + 1/8*\Delta$$

$$\tilde{\epsilon}_2 = (1/4 + 1/3 + 1/4*(1 + \Delta) + 1/12)/(11/12) = 1 + 3/11*\Delta$$

$$\tilde{\epsilon}_3 = (1/12 + 1/4 + 1/3*(1 + \Delta) + 1/4 + 1/12)/1 = 1 + 1/3*\Delta$$

$$\tilde{\epsilon}_4 = (1/12 + 1/4*(1 + \Delta) + 1/3 + 1/4 + 1/12)/1 = 1 + 1/4*\Delta$$

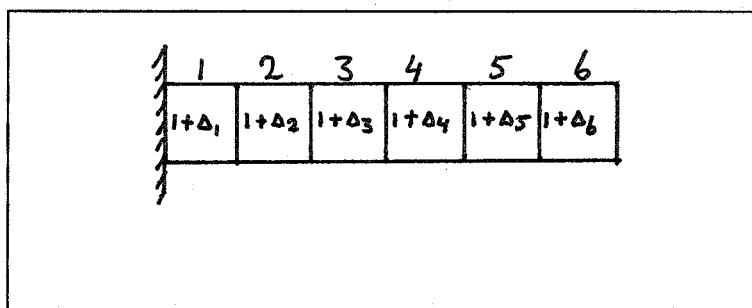
$$\tilde{\epsilon}_5 = (1/12*(1 + \Delta) + 1/4 + 1/3 + 1/4)/(11/12) = 1 + 1/11*\Delta$$

$$\tilde{\epsilon}_6 = (1/12 + 1/4 + 1/3)/(2/3) = 1$$

Hieruit blijkt:

$$\tilde{\epsilon}_3 > \tilde{\epsilon}_2 > \tilde{\epsilon}_4 > \tilde{\epsilon}_1 > \tilde{\epsilon}_5 > \tilde{\epsilon}_6$$

zodat de rekverdeling in stap 2 er kan uitzien als:



met $\Delta_3 > \Delta_2 > \Delta_4 > \Delta_1 > \Delta_5 > \Delta_6$.

Vervolgens wordt weer de gewogen rek berekend:

$$\begin{aligned}\tilde{\varepsilon}_1 &= (1/3*(1 + \Delta_1) + 1/4*(1 + \Delta_2) + 1/12*(1 + \Delta_3))/(2/3) \\ &= 1 + 1/8*\Delta_3 + 3/8*\Delta_2 + 1/2*\Delta_1\end{aligned}$$

$$\begin{aligned}\tilde{\varepsilon}_2 &= (1/4*(1 + \Delta_1) + 1/3*(1 + \Delta_2) + 1/4*(1 + \Delta_3) + 1/12*(1 + \Delta_4))/(11/12) \\ &= 1 + 3/11*\Delta_3 + 4/11*\Delta_2 + 1/11*\Delta_4 + 3/11*\Delta_1\end{aligned}$$

$$\begin{aligned}\tilde{\varepsilon}_3 &= (1/12*(1 + \Delta_1) + 1/4*(1 + \Delta_2) + 1/3*(1 + \Delta_3) + 1/4*(1 + \Delta_4) + 1/12*(1 + \Delta_5))/1 \\ &= 1 + 1/3*\Delta_3 + 1/4*\Delta_2 + 1/4*\Delta_4 + 1/12*\Delta_1 + 1/12*\Delta_5\end{aligned}$$

$$\begin{aligned}\tilde{\varepsilon}_4 &= (1/12*(1 + \Delta_2) + 1/4*(1 + \Delta_3) + 1/3*(1 + \Delta_4) + 1/4*(1 + \Delta_5) + 1/12*(1 + \Delta_6))/1 \\ &= 1 + 1/4*\Delta_3 + 1/12*\Delta_2 + 1/3*\Delta_4 + 1/4*\Delta_5 + 1/12*\Delta_6\end{aligned}$$

$$\begin{aligned}\tilde{\varepsilon}_5 &= (1/12*(1 + \Delta_3) + 1/4*(1 + \Delta_4) + 1/3*(1 + \Delta_5) + 1/4*(1 + \Delta_6))/(11/12) \\ &= 1 + 1/11*\Delta_3 + 3/11*\Delta_4 + 4/11*\Delta_5 + 3/11*\Delta_6\end{aligned}$$

$$\tilde{\varepsilon}_6 = (1/12*(1 + \Delta_4) + 1/4*(1 + \Delta_5) + 1/3*(1 + \Delta_6))/(2/3) = 1 + 1/8*\Delta_4 + 3/8*\Delta_5 + 1/2*\Delta_6$$

Als de gewogen rek in incrementstap 1 en 2 worden vergeleken, kan worden geconcludeerd dat per incrementstap het verschil tussen $\tilde{\varepsilon}_2$ en $\tilde{\varepsilon}_4$ groter wordt en daarmee ook het verschil tussen Δ_2 en Δ_4 .

Kan in de tweede stap reeds gelden dat $\tilde{\varepsilon}_2 > \tilde{\varepsilon}_3$?

Dus kan:

$$3/11*\Delta_3 + 4/11*\Delta_2 + 1/11*\Delta_4 + 3/11*\Delta_1 > 1/3*\Delta_3 + 1/4*\Delta_2 + 1/4*\Delta_4 + 1/12*\Delta_1 + 1/12*\Delta_5 \quad (4)$$

$$\rightarrow 5/44*\Delta_2 - 1/44*\Delta_4 + 1/132*\Delta_1 - 1/12*\Delta_5 > 2/33*\Delta_3 \quad (5)$$

Stel $\Delta_4 = \Delta_2$ en $\Delta_5 = \Delta_1$, hetgeen een pessimistische veronderstelling is \rightarrow

$$1/11*\Delta_2 - 10/132*\Delta_1 > 2/33*\Delta_3 \quad (6)$$

Als $10/132*\Delta_1$ wordt verwaarloosd, dan kan reeds in de tweede stap $\tilde{\varepsilon}_2 > \tilde{\varepsilon}_3$ zijn als:

$$\Delta_2 > 1/2*\Delta_3 \quad (7)$$

Indien de term $10/132*\Delta_1$ niet wordt verwaarloosd, dan hoeft het in deze tweede incrementstap nog niet zo te zijn dat $\tilde{\varepsilon}_2 > \tilde{\varepsilon}_3$. Maar er is reeds geconcludeerd dat per stap Δ_2 relatief groter wordt t.o.v. Δ_4 , dus wordt per incrementstap de kans groter dat aan (5) wordt voldaan en dat dus $\tilde{\varepsilon}_2 > \tilde{\varepsilon}_3$ is.

Dit bewijst dat de maximale waarde van de gewogen rek naar links kan gaan lopen als het verschil tussen het zwakste element en de overige elementen slechts gering is en er gerekend wordt met een even aantal elementen. Uiteindelijk zal hierdoor in het numerieke model de maximale rek worden overschreden in element 1 en niet in het zwakste ($n/2$ -de) element, wat een foutieve oplossing is.

Oneven aantal elementen

Als er 5 elementen zijn, dan ziet de gewogen rekverdeling in stap 2 er als volgt uit:

$$\begin{aligned}\varepsilon_1 &= (1/3*(1 + \Delta_1) + 1/4*(1 + \Delta_2) + 1/12*(1 + \Delta_3))/(2/3) \\ &= 1 + 1/8*\Delta_3 + 3/8*\Delta_2 + 1/2*\Delta_1\end{aligned}$$

$$\begin{aligned}\varepsilon_2 &= (1/4*(1 + \Delta_1) + 1/3*(1 + \Delta_2) + 1/4*(1 + \Delta_3) + 1/12*(1 + \Delta_4))/(11/12) \\ &= 1 + 3/11*\Delta_3 + 4/11*\Delta_2 + 1/11*\Delta_4 + 3/11*\Delta_1\end{aligned}$$

$$\begin{aligned}\varepsilon_3 &= (1/12*(1 + \Delta_1) + 1/4*(1 + \Delta_2) + 1/3*(1 + \Delta_3) + 1/4*(1 + \Delta_4) + 1/12*(1 + \Delta_5))/1 \\ &= 1 + 1/3*\Delta_3 + 1/4*\Delta_2 + 1/4*\Delta_4 + 1/12*\Delta_1 + 1/12*\Delta_5\end{aligned}$$

$$\begin{aligned}\varepsilon_4 &= (1/12*(1 + \Delta_2) + 1/4*(1 + \Delta_3) + 1/3*(1 + \Delta_4) + 1/4*(1 + \Delta_5))/(11/12) \\ &= 1 + 4/11*\Delta_3 + 1/11*\Delta_2 + 4/11*\Delta_4 + 3/11*\Delta_5\end{aligned}$$

$$\begin{aligned}\varepsilon_5 &= (1/12*(1 + \Delta_3) + 1/4*(1 + \Delta_4) + 1/3*(1 + \Delta_5))/(2/3) \\ &= 1 + 1/8*\Delta_3 + 3/8*\Delta_4 + 1/2*\Delta_5\end{aligned}$$

Met $\Delta_3=\alpha$, $\Delta_2=\Delta_4=\beta$ en $\Delta_1=\Delta_5=\gamma$, waarbij $\alpha>\beta>\gamma$ geeft dit:

$$\varepsilon_1 = \varepsilon_5 = 1 + 1/8*\alpha + 3/8*\beta + 1/2*\gamma$$

$$\varepsilon_2 = \varepsilon_4 = 1 + 3/11*\alpha + 5/11*\beta + 3/11*\gamma$$

$$\varepsilon_3 = 1 + 1/3*\alpha + 1/2*\beta + 1/6*\gamma$$

De gewogen rek ε_2 is groter dan de gewogen rek ε_3 als:

$$-1/22*\beta + 7/66*\gamma > 2/33*\alpha \quad (8)$$

ofwel:

$$-3*\beta + 7*\gamma > 4*\alpha \quad (9)$$

en aangezien altijd $\alpha>\beta>\gamma$ zal ε_3 altijd groter blijven dan ε_2 en treedt er bij een oneven aantal elementen nooit een verschuiving van de maximale rek als gevolg van randeffecten op.