

# Dieptrekken : de flens : bepaling geometrie middels optimalisatie naar drie parameters a.d.h.v. het bovengrenstheorema

**Citation for published version (APA):**

Du Bois, J. M. A. (1991). *Dieptrekken : de flens : bepaling geometrie middels optimalisatie naar drie parameters a.d.h.v. het bovengrenstheorema*. (TH Eindhoven. Afd. Werktuigbouwkunde, Vakgroep Produktietechnologie : WPB; Vol. WPA1118-1119). Technische Universiteit Eindhoven.

**Document status and date:**

Gepubliceerd: 01/01/1991

**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](http://www.tue.nl/taverne)

**Take down policy**

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

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

I. L. J. A. Du Bois

Dieptrekken : de flens  
Bepaling geometrie middels optimalisatie naar  
drie parameters a.d.h.v. het bovengrenstheorema.

ir. J.M.A. Du Bois

juni '91

IOPM-Dieptrekken 010

WPA 1118

## SYMBOLENLIJST

$\kappa$	modelparameter	[—]
$\lambda$	modelparameter	[—]
$\nu$	modelparameter	[—]
$\mu$	wrijvingscoëfficiënt	[—]
$\epsilon$	logaritmische rek	[—]
$\epsilon_0$	voordeformatie	[—]
$\bar{\epsilon}$	effektieve rek	[—]
$\dot{\epsilon}$	reksnelheid	[T <sup>-1</sup> ]
$\frac{\dot{\cdot}}{\bar{\epsilon}}$	effektieve reksnelheid	[T <sup>-1</sup> ]
n	verstevingsexponent	[—]
$\sigma_f$	vloeispanning	[ML <sup>-1</sup> T <sup>-2</sup> ]
p	plooihouderdruk	[ML <sup>-1</sup> T <sup>-2</sup> ]
r, $\varphi$ , z	cilinder coördinaten	[L, —, L]
$r_i$	binnenrand flens	[L]
$r_u$	momentane positie buitenrand flens	[L]
$r_{u0}$	initiële positie buitenrand flens	[L]
s	momentane flensdikte	[L]
$s_0$	initiële flensdikte	[L]
t	tijd	[T]
$\Delta t$	kleine tijdseenheid	[T]
$\dot{u}$	voorgeschreven snelheid binnenrand flens	[LT <sup>-1</sup> ]
B	integratieconstante	[—]
C	karacteristieke deformatieweerstand	[ML <sup>-1</sup> T <sup>-2</sup> ]
K	modelparameter	[—]
L	modelparameter	[—]
$P_{def}$	deformatievermogen	[ML <sup>2</sup> T <sup>-3</sup> ]
$P_{wr}$	wrijvingsvermogen	[ML <sup>2</sup> T <sup>-3</sup> ]
*	aanduiding voor een dimensieloze grootheid	

INHOUDSOPGAVE.

1 Inleiding	[2]
2 Modelvorming	[3]
3 Numerieke oplossing	[8]
4 Resultaten en conclusies	[9]
Literatuur	[15]

1 INLEIDING.

Dit rapport is een vervolg op het IOPM-Dieptrek rapport 009, [1]. Daar wordt uitgegaan van een flens die gedurende het gehele dieptrekproces planparallel blijft. In dit rapport wordt ervanuit gegaan dat dit niet het geval is en dat de dikte van de flens een functie is van de momentane positie van een materieel deeltje in de flens, zie figuur 1,:

$$s = s(r,t) \quad (1.1)$$

Omdat onbekend is in welke mate de dikte van de flens van de straal  $r$  afhangt wordt de orde van de straal  $r$ ,  $\nu$ , als een te optimaliseren parameter in de modelvorming meegenomen. Bij de modelvorming wordt verder uitgegaan van de volgende aannamen:

- de flens blijft cirkelsymmetrisch gedurende het hele proces.
- isotroop materiaal
- wrijving volgens het Coulomb wrijvingsmodel
- de snelheid aan de binnenrand van de flens is voorgeschreven.
- het materiaal is exponentieel verstevigend
- er wordt gebruik gemaakt van de bovengrensmethode [2],[3]

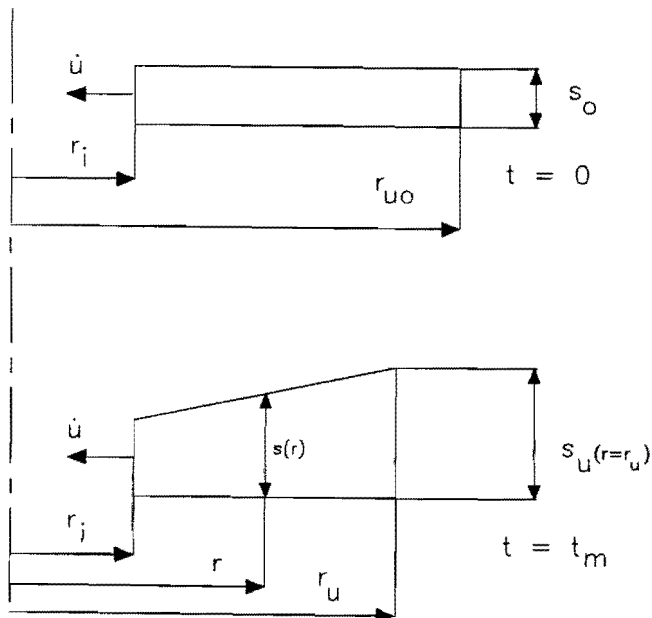


Fig.1 Model van de flens.

2 MODELFORMING.

Uitgaande van de bovenstaande aannamen wordt het volgende snelheidsveld in axiale richting gedefinieerd:

$$\dot{u}_z(r,z,t) = \kappa(t) \cdot \left\{ \frac{r}{r_i} \right\}^\nu \cdot \frac{z}{s_0} \cdot \dot{u} + \lambda(t) \cdot \frac{z}{s_0} \cdot \dot{u} \quad (2.1)$$

Hierin zijn  $\kappa$ ,  $\lambda$  en  $\nu$  de te optimaliseren modelparameters. De termen voor de reksnelheden in de hoofdrichtingen  $r$ ,  $\varphi$  en  $z$  kunnen als volgt geschreven worden [2]:

$$\begin{aligned} \dot{\epsilon}_r(r,t) &= \frac{\delta \dot{u}_r(r,t)}{\delta r} \\ \dot{\epsilon}_\varphi(r,t) &= \frac{\dot{u}_\varphi(r,t)}{r} \end{aligned} \quad (2.2)$$

$$\dot{\epsilon}_z(r,t) = \frac{\delta \dot{u}_z(r,z,t)}{\delta z}$$

$$\dot{\epsilon}_{rz}(r,z,t) = 0.5 \cdot \frac{\delta \dot{u}_z(r,z,t)}{\delta r}$$

Met behulp van deze definities en gebruikmakend van lokale volume-invariantie, de som van de reksnelheden in de hoofdrichtingen is gelijk aan nul, kan de volgende relatie afgeleid worden:

$$-\frac{1}{r} \cdot \frac{\delta}{\delta r} \cdot \{ r \cdot \dot{u}_r(r,t) \} = \frac{\delta \dot{u}_z(r,z,t)}{\delta z} \quad (2.3)$$

Om het rekenwerk te vereenvoudigen worden de volgende modelparameters gedefinieerd:

$$K(t) = \frac{\kappa(t) \cdot r_i}{s_0} \quad (2.4)$$

$$L(t) = \frac{\lambda(t) \cdot r_i}{s_0}$$

De relevante termen worden in dimensieloze vorm gepresenteerd. Hierbij wordt gebruik gemaakt van de volgende gegeneraliseerde definities in het geval van:

$$\begin{aligned}
 - \text{geometrische grootheden:} & \quad x^* = \frac{x}{r_i} \\
 - \text{snelheden:} & \quad \dot{x}^* = \frac{\dot{x}}{\dot{u}} \\
 - \text{reksnelheden:} & \quad \dot{x}^* = \frac{\dot{x} \cdot r_i}{\dot{u}} \quad (2.5) \\
 - \text{spanningen:} & \quad x^* = \frac{x}{C} \\
 - \text{vermogens:} & \quad x^* = \frac{x}{2\pi \cdot r_i \cdot s_0 \cdot \dot{u} \cdot C} \\
 - \text{tijd:} & \quad t^* = t \cdot \frac{\dot{u}}{r_i}
 \end{aligned}$$

Oplossen van (2.3) door integratie na substitutie van (2.1) en toepassing van de definities (2.4) en (2.5) levert de volgende vergelijking voor de dimensieloos geschreven radiale snelheid van een materieel deeltje:

$$\dot{u}_r^*(r^*, t^*) = -\frac{K(t^*)}{\nu + 2} \cdot \{r^*\}^{\nu+1} - \frac{L(t^*)}{2} \cdot r^* + \frac{B(t^*)}{r^*} \quad (2.6)$$

Met behulp van de randvoorwaarde,  $\dot{u}_r^*(r^*=1, t^*) = -1$ , kan de integratieconstante  $B(t^*)$  worden bepaald:

$$B(t^*) = \frac{K(t^*)}{\nu + 2} + \frac{L(t^*)}{2} - 1 \quad (2.7)$$

Met behulp van (2.6) en (2.7) kan de volgende term voor voor het dimensieloos geschreven radiale snelheidsveld gevonden worden:

$$\dot{u}_r^*(r^*, t^*) = -\frac{K(t^*)}{\nu + 2} \cdot \left[ \{r^*\}^{\nu+1} - \frac{1}{r^*} \right] - \frac{L(t^*)}{2} \cdot \left[ r^* - \frac{1}{r^*} \right] - \frac{1}{r^*} \quad (2.8)$$

Het axiale snelheidsveld wordt als volgt dimensieloos geschreven:

$$\dot{u}_z^*(r^*, z^*, t^*) = K(t^*) \cdot \{r^*\}^\nu \cdot z^* + L(t^*) \cdot z^* \quad (2.9)$$

Met behulp van (2.8) en (2.9) en de definities (2.2) en (2.5) kan nu het reksnelheidsveld bepaald worden:

$$\begin{aligned}\dot{\epsilon}_{\varphi}^*(r^*, t^*) &= -\frac{K(t^*)}{\nu + 2} \cdot \left[ \{r^*\}^{\nu} - \left\{\frac{1}{r^*}\right\}^2 \right] - \frac{L(t^*)}{2} \cdot \left[ 1 - \left\{\frac{1}{r^*}\right\}^2 \right] - \left\{\frac{1}{r^*}\right\}^2 \\ \dot{\epsilon}_z^*(r^*, t^*) &= K(t^*) \cdot \{r^*\}^{\nu} + L(t^*) \\ \dot{\epsilon}_r^*(r^*, t^*) &= -\{ \dot{\epsilon}_{\varphi}^*(r^*, t^*) + \dot{\epsilon}_z^*(r^*, t^*) \} \\ \dot{\epsilon}_{rz}^*(r^*, z^*, t^*) &= 0.5 \cdot \nu \cdot K(t^*) \cdot \{r^*\}^{\nu-1} \cdot z^*\end{aligned}\tag{2.10}$$

De term voor de effectieve reksnelheid wordt als volgt gedefinieerd:

$$\dot{\bar{\epsilon}}^*(r^*, z^*, t^*) = \left\{ \frac{2}{3} \cdot \left\{ \dot{\epsilon}_r^{*2}(r^*, t^*) + \dot{\epsilon}_{\varphi}^{*2}(r^*, t^*) + \dot{\epsilon}_z^{*2}(r^*, t^*) \right\} + \frac{4}{3} \cdot \dot{\epsilon}_{rz}^{*2}(r^*, z^*, t^*) \right\}^{\frac{1}{2}}\tag{2.11}$$

Het te dissiperen deformatie vermogen wordt als volgt gedefinieerd:

$$P_{\text{def}} = \int_V \sigma_f \cdot \dot{\bar{\epsilon}} \cdot dV\tag{2.12}$$

De vloeispanning wordt als volgt gedefinieerd:

$$\sigma_f = (\epsilon_0 + \bar{\epsilon})^n\tag{2.13}$$

In dimensieloze vorm wordt het deformatievermogen als volgt geschreven:

$$P_{\text{def}}(t^*) = \frac{1}{s_0} \cdot \int_1^{r_u^*(t^*)} \int_0^{s(r^*, t^*)} (\epsilon_0 + \bar{\epsilon})^n \cdot \dot{\bar{\epsilon}}^*(r^*, z^*, t^*) \cdot r^* \cdot dr^* \cdot dz^*\tag{2.14}$$



Als er aangenomen wordt dat de reksnelheidsterm  $\dot{\epsilon}_{rz}^*$ , in de term voor de effectieve reksnelheid (2.11), te verwaarlozen is kan de integratie over  $z$  analytisch opgelost worden door gebruik te maken van een standaard integraal. De reksnelheidsterm wordt hiertoe gesplitst in een functie in  $r$ ,  $\chi_r$ , en een functie in  $r$  en  $z$ ,  $\chi_{rz}$ :

$$\dot{\epsilon}(r^*, z^*, t^*) = \chi_r(r^*, t^*) \cdot \chi_{rz}(r^*, t^*, z^*) \quad (2.15)$$

Hierin worden  $\chi_r$  en  $\chi_{rz}$  als volgt gedefinieerd:

$$\chi_r(r^*, t^*) = \left| \sqrt{\frac{1}{3}} \cdot \nu \cdot K(t^*) \cdot r^{*\nu-1} \right| \quad (2.16)$$

en

$$\chi_{rz}(r^*, t^*, z^*) = \left\{ f(r^*, t^*) + z^{*2} \right\}^{\frac{1}{2}} \quad (2.17)$$

Met  $f$  een functie in  $r$ :

$$f(r^*, t^*) = \frac{2}{\nu^2 \cdot K^2(t^*) \cdot r^{*2\nu-2}} \cdot \left\{ \dot{\epsilon}_r^{*2}(r^*, t^*) + \dot{\epsilon}_\varphi^{*2}(r^*, t^*) + \dot{\epsilon}_z^{*2}(r^*, t^*) \right\} \quad (2.18)$$

De integratie over  $z$ ,  $\text{Int}_z$ , van  $\chi_{rz}$  wordt als volgt gedefinieerd:

$$\text{Int}_z(r^*, t^*) = \int_0^{s(r^*, t^*)} \chi_{rz}(r^*, z^*, t^*) \cdot dz^* \quad (2.19)$$

Hierin is  $s(r^*, t^*)$  de momentane dikte van de flens op de plaats  $r^*$ . Substitutie van (2.17) levert nu de volgende standaard oplossing:

$$\begin{aligned} \text{Int}_z(r^*, t^*) = \frac{f(r^*, t^*)}{2} \cdot \text{Ln} \left\{ \frac{\sqrt{f(r^*, t^*) + s(r^*, t^*)} + s(r^*, t^*)}{\sqrt{f(r^*, t^*)}} \right\} + \\ + \frac{s(r^*, t^*)}{2} \cdot \sqrt{f(r^*, t^*) + s^2(r^*, t^*)} \end{aligned} \quad (2.20)$$

Het dimensieloos geschreven deformatievermogen kan nu als volgt geschreven worden:

$$P_{\text{def}}^*(t^*) = \frac{1}{s_0} \cdot \int_1^{r_u^*(t^*)} \chi_r^*(r^*, t^*) \cdot \text{Int}_z^*(r^*, t^*) \cdot r^* \cdot dr^* \quad (2.21)$$

Deze integraal is alleen numeriek op te lossen. Met  $\chi_r$  volgens (2.16) en  $\text{Int}_z$  volgens (2.20)

Bij het bepalen van het wrijvingsvermogen wordt ervanuit gegaan dat de wrijving alleen aan de buitenrand van de flens aangrijpt ten gevolge van het taps worden van de flens. Het wrijvingsvermogen wordt als volgt gedefinieerd [4]:

$$P_{\text{wr}}(r=r_u) = \mu \cdot p \cdot \pi \cdot (r_{u0}^2 - r_i^2) \cdot |\dot{u}_r(r=r_u)| \quad (2.21)$$

In dimensieloze vorm wordt het wrijvingsvermogen als volgt geschreven:

$$P_{\text{wr}}^*(r^*=r_u^*) = \frac{\mu}{2} \cdot p^* \cdot (r_{u0}^{*2} - 1) \cdot \frac{\dot{u}_r^*(r^*=r_u^*)}{s_0}$$

Met  $\dot{u}_r^*$  volgens (2.8)

### 3 NUMERIEKE OPLOSSING

De numerieke oplossing verloopt in principe op dezelfde manier als die besproken in [1]. Uitgaande van een initiële geometrie worden die waarden van  $K[t=0]$ ,  $L[t=0]$  en  $\nu[t=0]$  bepaald waarvoor het te dissiperen deformatievermogen minimaal is. Met behulp van deze waarden wordt dan een momentane geometrie op het tijdstip  $t = t + \Delta t$  bepaald waarna middels optimalisatie weer het minimaal te dissiperen vermogen en de waarden van de modelparameters  $K[t=t+\Delta t]$ ,  $L[t=t+\Delta t]$  en  $\nu[t + \Delta t]$  worden bepaald. Deze procedure wordt herhaald totdat de flens naar binnen getrokken is.

Het tijdje  $\Delta t$  wordt bepaald door de momentane geometrie met een nauwkeurigheid van 0.1 maal de nauwkeurigheid waarmee het vermogen wordt bepaald te bepalen. De flens wordt dan in een aantal stappen naar binnen getrokken. Als men de geometrie met een nauwkeurigheid van 0.001 bepaald zijn dit ongeveer 12 stappen, bij een nauwkeurigheid van 0.0001 zijn dit ongeveer 35 stappen. In tijd uitgedrukt respectievelijk 3 en 42 minuten. Deze gegevens gelden voor een initiële dieptrekverhouding van  $r_{u0}^* = 2.2$ . Voor een kleinere dieptrekverhouding zullen de rekentijden kleiner zijn.

De buitenrand van de flens ,dit is de bovenste integratiegrens, wordt als volgt bepaald:

$$r_u^*(t^*) = r_u^*(t^* - \Delta t^*) + \Delta t^* \cdot \dot{u}_r^*(r^* = r_u^*(t^* - \Delta t^*), t^* - \Delta t^*) \quad (3.1)$$

De momentane dikte van de flens wordt bepaald door van een materiaal deeltje in de flens de positie op voorgaande tijdstippen te bepalen. Dit gebeurt met behulp van een nulpuntzoeker. De momentane dikte van de flens wordt dan als volgt bepaald:

$$s^*(r^*(t^*)) = s^*(r^*(t^* - \Delta t^*)) + \Delta t^* \cdot \dot{u}_z^*(r^*(t^* - \Delta t^*), z^* = s^*(r^*(t^* - \Delta t^*))) \quad (3.2)$$

#### 4 RESULTATEN EN CONCLUSIES.

In de nu volgende grafieken worden de waarden getoond die de te optimaliseren parameters aannemen afhankelijk van de nauwkeurigheid waarmee het te dissiperen vermogen wordt bepaald

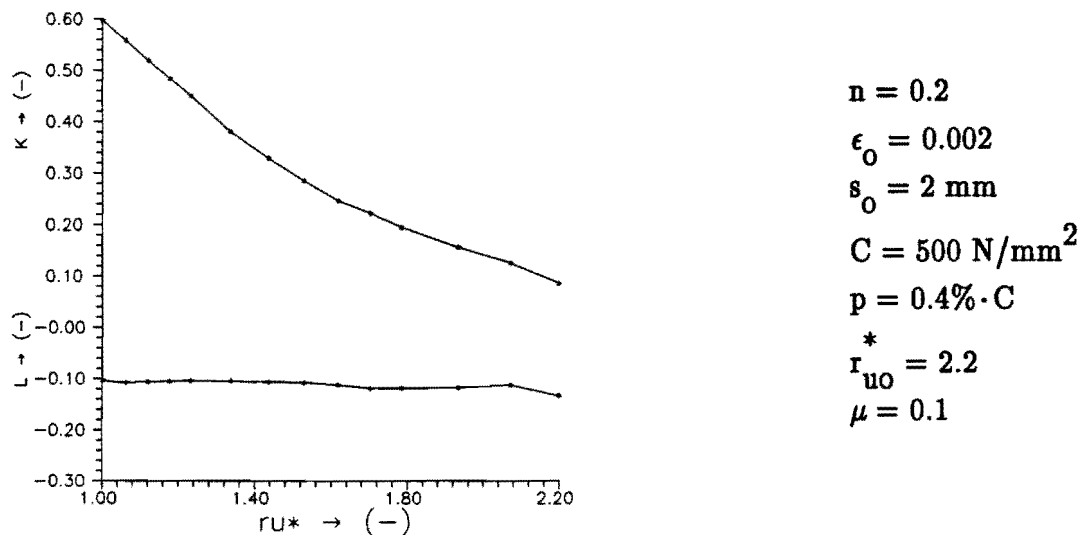


Fig 4.1 Waarden van  $K$  en  $L$  als het vermogen bepaald wordt met een nauwkeurigheid van 0.01.

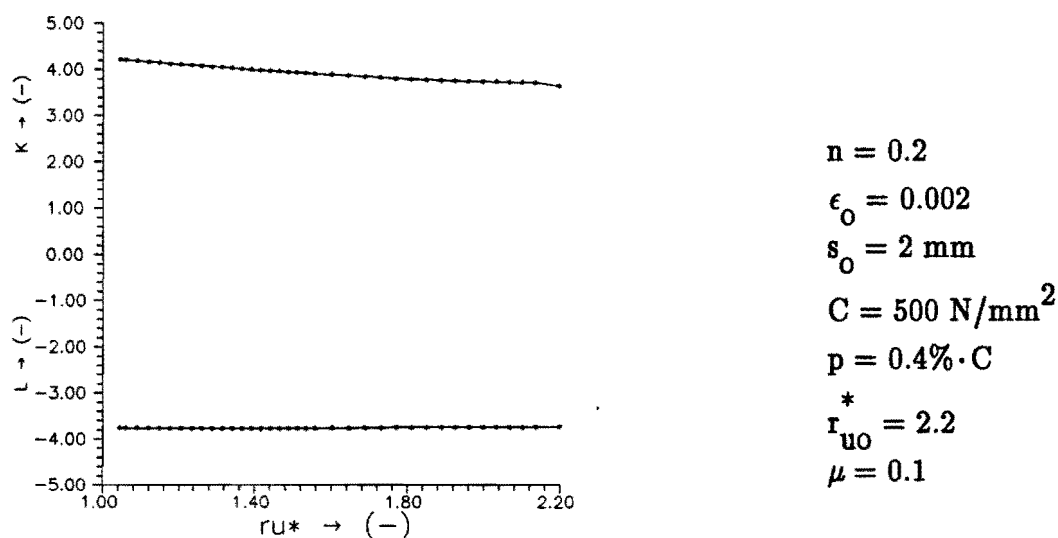
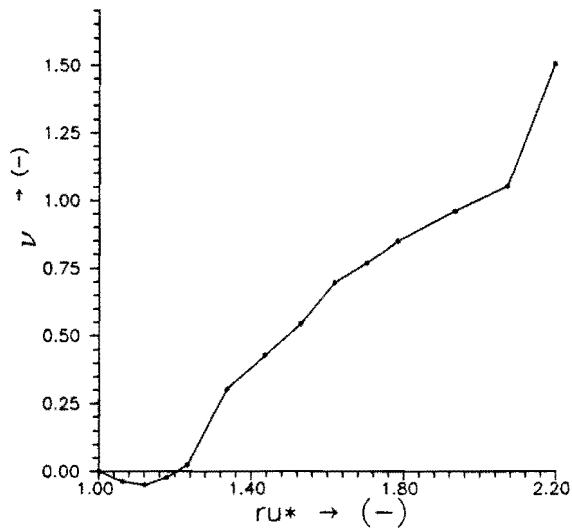
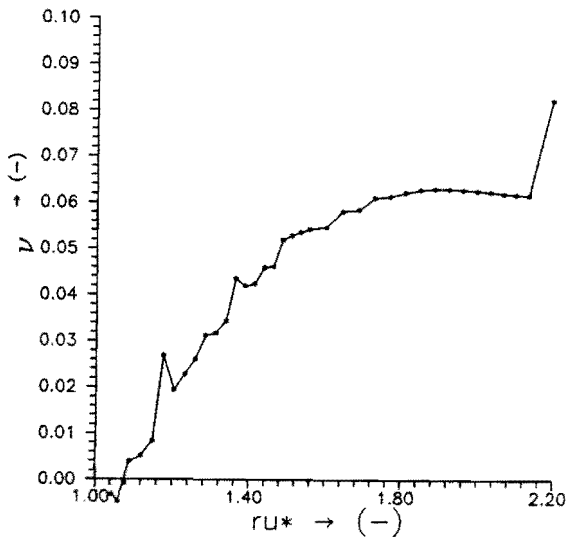


Fig 4.2 Waarden van  $K$  en  $L$  als het vermogen bepaald wordt met een nauwkeurigheid van 0.001.



$$\begin{aligned} n &= 0.2 \\ \epsilon_0 &= 0.002 \\ s_0 &= 2 \text{ mm} \\ C &= 500 \text{ N/mm}^2 \\ p &= 0.4\% \cdot C \\ r_{u0}^* &= 2.2 \\ \mu &= 0.1 \end{aligned}$$

Fig 4.3 Waarde van  $\nu$  als het vermogen bepaald wordt met een nauwkeurigheid van 0.01.



$$\begin{aligned} n &= 0.2 \\ \epsilon_0 &= 0.002 \\ s_0 &= 2 \text{ mm} \\ C &= 500 \text{ N/mm}^2 \\ p &= 0.4\% \cdot C \\ r_{u0}^* &= 2.2 \\ \mu &= 0.1 \end{aligned}$$

Fig 4.4 Waarden van  $\nu$  als het vermogen bepaald wordt met een nauwkeurigheid van 0.001.

In de figuren is te zien dat de waarden voor de modelparameters waarmee het te dissiperen deformatievermogen is bepaald met een nauwkeurigheid van 0.001 het meest constante verloop hebben. Het grootste verloop van de data ziet men in de grafieken 4.3 en 4.4. Daarin is ook te zien dat zich een scherpe correctie instelt met betrekking tot de waarde van  $\nu$  na de initiële fase. Opvallend is verder het verschil in de waarde die  $\nu$  aanneemt afhankelijk van de nauwkeurigheid waarmee het vermogen wordt bepaald. In figuur 4.4 is te zien dat zich een lichte stijging voordoet van de waarde van  $\nu$  tot een waarde van  $r_u^*$  van ongeveer 1.9 daarna volgt een lichte daling tot een waarde van  $r_u^*$  van ongeveer 1.7 waarna de waarde van  $\nu$  sterk afneemt tot 0. Als men de relatieve afwijking van de planparalleliteit uitzet tegenover de positie van de buitenrand van de flens verkrijgt men het verloop zoals in figuur 4.5. Als men deze figuur gaat vergelijken met figuur 4.6 in welke de data staat weergegeven die verkregen is door het proces door te rekenen met als aanname :  $\nu = 1$ , ziet men dat door  $\nu$  vrij te laten het model een lagere maximale relatieve afwijking van de planparalleliteit levert. Dit was te verwachten doordat de waarde voor  $\nu$  die doormiddel van optimalisatie wordt bepaald veel kleiner is dan 1. Dat de relatieve afwijking van de planparalleliteit na het bereiken van een maximum weer snel naar nul nadert is te verklaren uit het feit dat het verschil in diktetoename ten gevolge van stuik aan de binnen- en de buitenrand van de flens zal afnemen naarmate het proces verder gevorderd is.

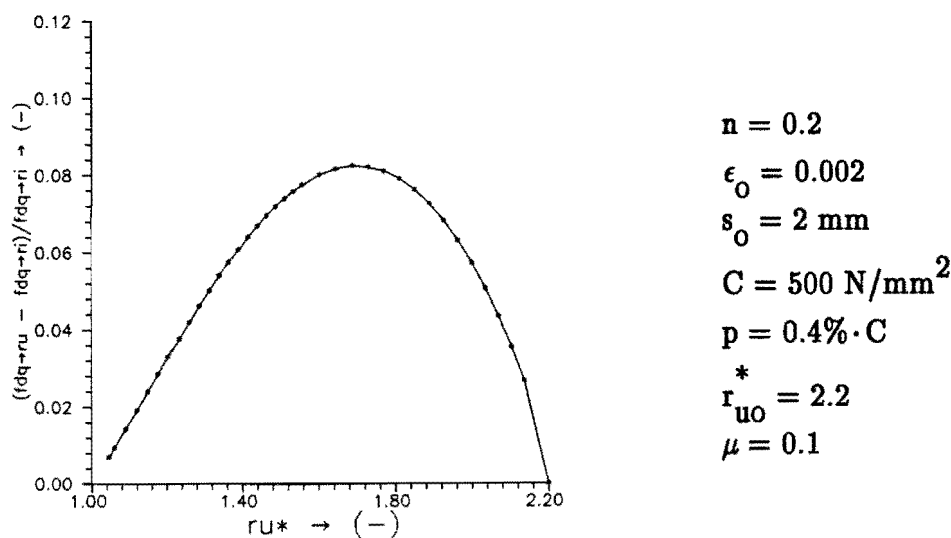


Fig 4.5 De relatieve afwijking ten opzichte van planparalleliteit als functie van de buitenrand van de flens. Het deformatievermogen is bepaald met een nauwkeurigheid van 0.001.

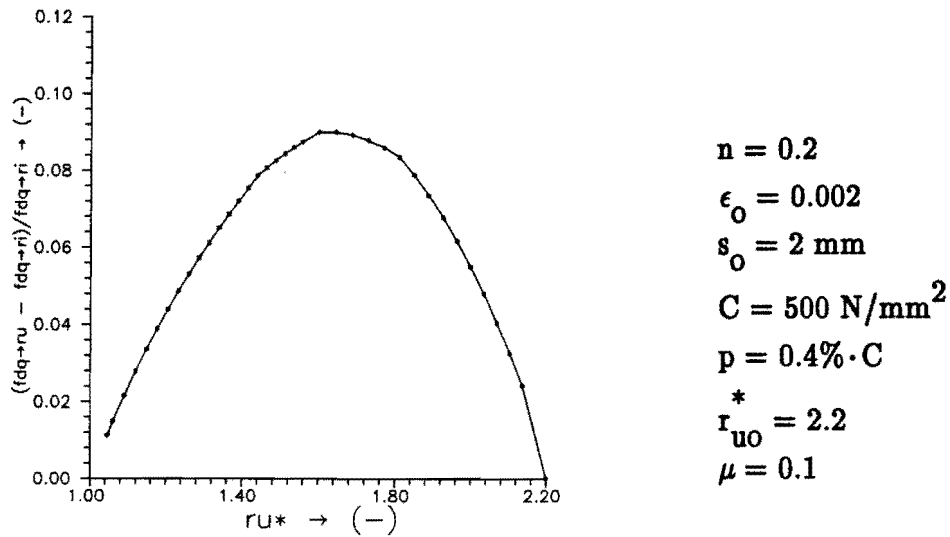


Fig 4.6 De relatieve afwijking ten opzichte van planparalleliteit als functie van de buitenrand van de flens. Het deformatievermogen is bepaald met een nauwkeurigheid van 0.001. Er is aangenomen dat  $\nu = 1$

In figuur 4.7 is het te dissiperen vermogen uitgezet tegen de positie van de buitenrand van flens. Uit de figuur blijkt dat het te bepalen vermogen weinig wordt beïnvloed door de gebruikte nauwkeurigheid.

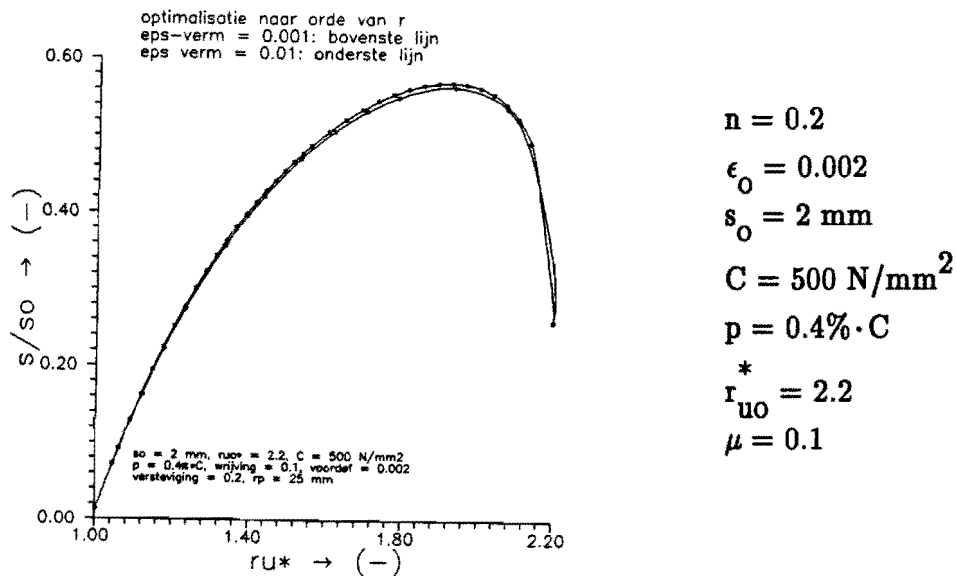


Fig 4.7 Het te dissiperen vermogen uitgezet tegen de positie van de buitenrand van de flens.

De invloed van de nauwkeurigheid waar het vermogen mee bepaald wordt heeft een grotere invloed op de bepaalde geometrie. De grootste invloed echter, met betrekking tot de geometrie, heeft de aanname van de waarde van  $\nu$ , zie figuur 4.8 en 4.9. Men kan concluderen dat het model waar  $\nu$  als te optimaliseren parameter wordt meegenomen de beste oplossing levert. De voorspelde geometrie ligt dan het dichtst bij de geometrie in de buurt die met de aanname lijnspanning aan de rand van de flens wordt bepaald. Er wordt nog opgemerkt dat de invloed van de wrijving in dit model met betrekking tot de geometrie te verwaarlozen is.

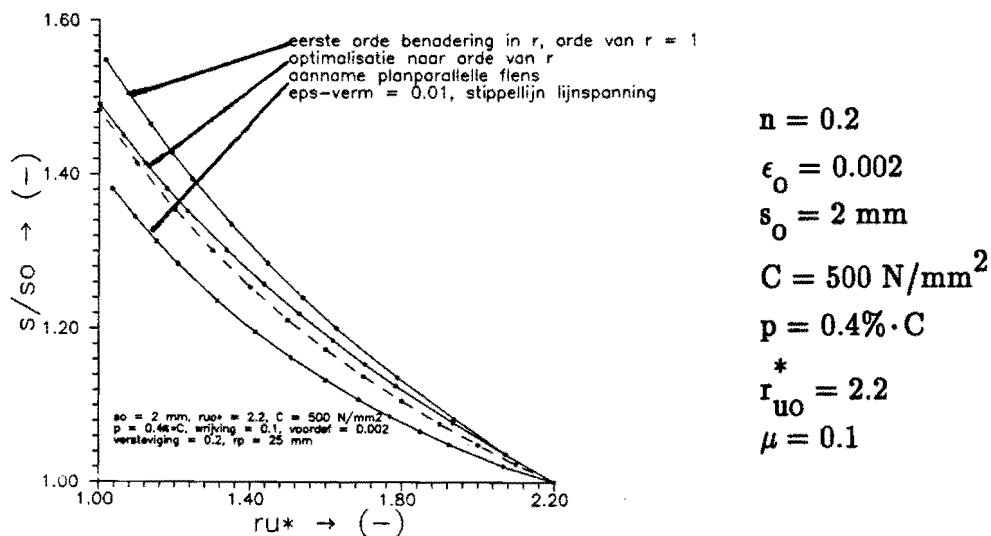


Fig 4.8 De flensdiktecoëfficiënt voor drie verschillende aannamen m.b.t. de waarde van  $\nu$  t.o.v. de lijnspanningswaarde.

In figuur 4.10 is als laatste figuur nog het verloop van de flensdikte over de flens weergegeven zoals bepaald met het model waarbij de orde van de straal  $r$  uit het axiale snelheidsveld,  $\nu$ , als te optimaliseren parameter wordt meegenomen.



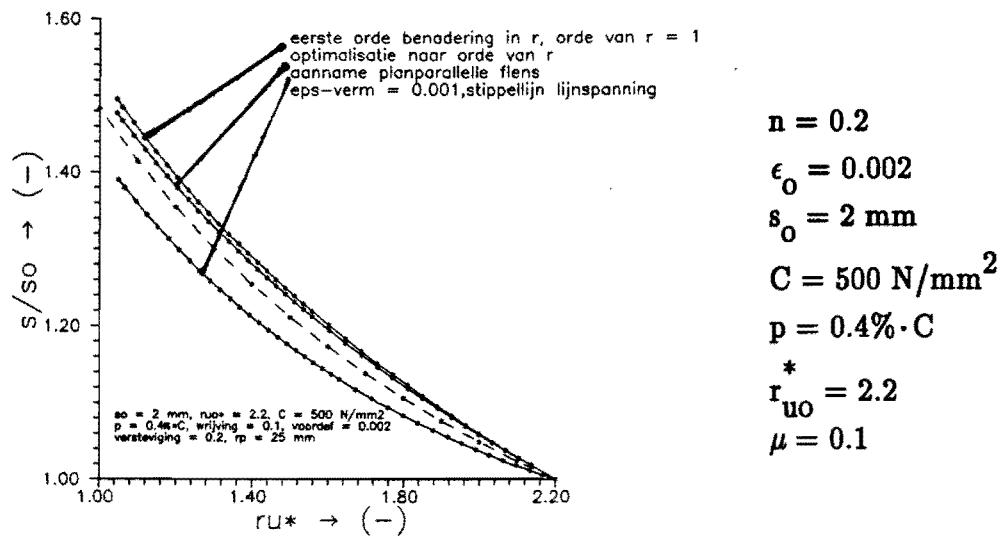


Fig 4.9 De flensdiktecoefficient voor drie verschillende aannamen m.b.t. de waarde van  $\nu$  t.o.v. de lijnspanningswaarde.

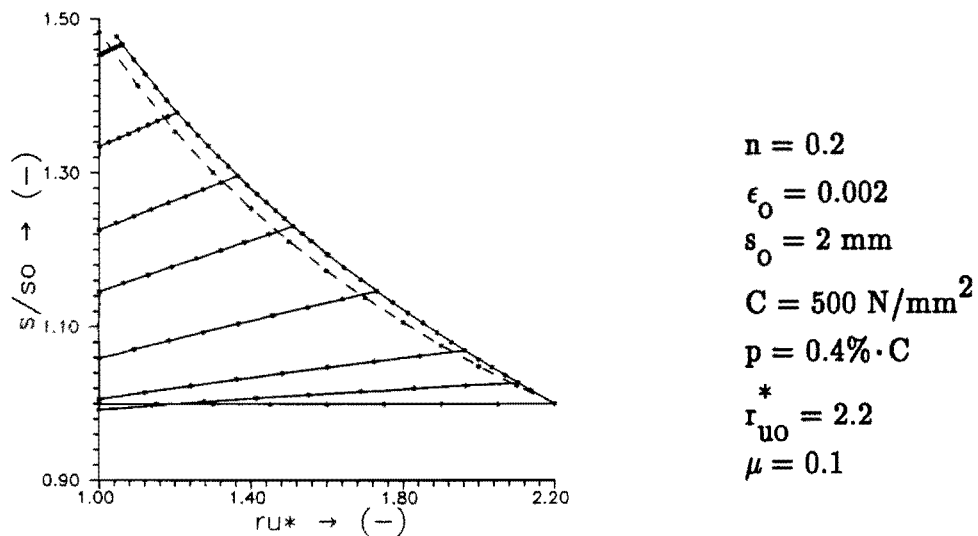


Fig 4.10 Het verloop van de dikte over de flens op verschillende posities van de buitenrand.

## LITERATUUR

- [1] Du Bois, J.M.A. Vergelijking tussen bovengrens- en schillen- methode voor een planparallele flens  
TUE, WPA nr. 1059, IOPM-Dieptrekken 009, Eindhoven (1991).
- [2] Hoogenboom, S.M. Engineering Plasticity. Department of Mechanical Engineering, Eindhoven University of Technology. EUT Report 87-WPA-0560, ISSN 0167-9708.  
Ook:  
Technische Plasticiteitsleer. Diktaat 4406. TUE, Eindhoven (1987).
- [3] Hill, R. The Mathematical Theory of Plasticity. Oxford University Press. Oxford (1988).
- [4] Sniekers, R.J.J.M. Hoogenboom, S.M. IOPM-Dieptrekken, modellen en analyse: de flens  
TUE, WPA nr.1062, IOPM-Dieptrekken nr.007 Eindhoven (1991).

Dieptrekken : de flens  
Listing bij rapport IOPM–dieptrekken 010

ir. J.M.A. Du Bois

juni '91

IOPM–Dieptrekken 011

WPA 1119

```

{*****}
{*                                           *}
{* UNIT      : Prvarpp                       *}
{* PURPOSE   : Globale variabelen behorende bij p_bg(k) *}
{* VERSION   : 1.0                           *}
{* DATE      : 10-01-1991                     *}
{* LANGUAGE  : Turbopascal 5.5 (Borland)     *}
{* SYSTEM    : Cirp-AT met EGA kaart en Numerieke Co-processor *}
{* PROGRAMMER : ir. J.M.A. Du Bois           *}
{* INFORMATION : Technische Universiteit Eindhoven *}
{*           : Faculteit Werktuigbouwkunde   *}
{*           : Vakgroep Produktietechnologie en - automatisering *}
{*           : Laboratorium voor Omvormtechnologie *}
{*                                           *}
{*****}

```

UNIT Prvarpp;

INTERFACE

USES

TYPES;

CONST pi = 3.141592654;

VAR

```

Effektieve_Rek           : ARRAY [0..125] OF Real;
Effektieve_Reksnelheid_Ster : ARRAY [0..125] OF Real;
ru_Ster                  : ARRAY [0..125] OF Real;
r_Ster                   : ARRAY [0..125] OF Real;
s_Ster                   : ARRAY [0..125] OF Real;
K                         : ARRAY [0..125] OF Real;
Lp                       : ARRAY [0..125] OF Real;
orde_n                   : ARRAY [0..125] OF Real;
uz_punt                  : ARRAY [0..125] OF Real;
Delta_Tijd_Ster         : ARRAY [0..125] OF Real;
Minimaal_Vermogen       : ARRAY [0..125] OF Real;
Deformatie_Vermogen     : ARRAY [0..125] OF Real;
Wrijving_Vermogen       : ARRAY [0..125] OF Real;
K_Var,orde_n_var,Lp_Var  : Real;
aantal_te_optimaliseren_parameters : Integer;
eps_eff_punt             : Real;
Voordeformatie           : Real;
Versteving               : Real;
eps,epsint,epstijdstap  : Real;
ruo                       : Real;
ruo_Ster                 : Real;
ru                       : Real;
r_Var_Ster               : Real;
r_Ster_Nulpunt           : Real;
ri                       : Real;
rd                       : Real;
rp                       : Real;
so                       : Real;
so_Ster                  : Real;
u_punt_z                 : Real;
uz_punt_ster             : Real;
min_ver                  : Real;
reksnelheid_phi_Ster,
reksnelheid_z_Ster,
reksnelheid_r_Ster      : Real;

```

```
t,j,l,np_true,o,s,eps_verm : Integer;
data1,data2,data3,data4,
data5,data6                : Text;
Stringvar1,Stringvar2,
Stringvar3                  : String;
karakteristieke_deformatieweerstand: Real;
mu                           : Real;
p_ster,C_waarde,p          : Real;
```

IMPLEMENTATION

END.

```

{*****}
(*                                     *)
(* UNIT      : Initdata                *)
(* PURPOSE   : Lezen van de initiele data met betrekking tot *)
(*            materiaal- en geometrieparameters en de nauwkeurig-*)
(*            heid waar geometrie mee bepaald wordt. Met defini- *)
(*            eren van de files waar de data naartoe wordt *)
(*            geschreven *)
(* VERSION   : 1.0 *)
(* DATE      : 26-06-1991 *)
(* LANGUAGE  : Turbopascal 5.5 (Borland) *)
(* SYSTEM    : Cirp-AT met EGA kaart en Numerieke Co-processor *)
(* PROGRAMMER : ir. J.M.A. Du Bois *)
(* INFORMATION : Technische Universiteit Eindhoven *)
(*            Faculteit Werktuigbouwkunde *)
(*            Vakgroep Produktietechnologie en - automatisering *)
(*            Laboratorium voor Omvormtechnologie *)
(*                                     *)
{*****}

```

```
UNIT Initdata;
```

```
INTERFACE
```

```
USES
Types,
Prvarpp,
Crt,
Printer;
```

```
PROCEDURE Lees_Initiele_Data;
```

```
IMPLEMENTATION
```

```

{*****}
{
{ Procedure voor het inlezen van materiaal en geometrieparameters. }
}

```

```
PROCEDURE Lees_Initiele_Data;
BEGIN
```

```

    Voordeformatie := 0.00;
    versteviging := 0.2;
    C_waarde := 500;
    p := 0.004*C_waarde;
    p_ster := p/C_waarde;
    mu := 0.1;
    so := 2.0;
    rp := 25;
    rd := rp + 1.3*so;
    ri := (rp + rd)/2;
    ruo := 2.2*ri;
    so_Ster := so/ri;
    ruo_Ster := ruo/ri;
    Effektieve_Rek[0] := 0;
    Str(eps_verm,Stringvar3);
    Str(aantal_te_optimaliseren_parameters,Stringvar2);
    Assign(data1,'d:\tp\iopm\datan'+stringvar2+'e'+stringvar3+'\minfun.dat');
    Rewrite(data1);
    Assign(data4,'d:\tp\iopm\datan'+stringvar2+'e'+stringvar3+'\P_min.dat');
    Rewrite(data4);

```

```
Assign(data3,'d:\tp\iopm\datan'+stringvar2+'e'+stringvar3+'\uz_ru.dat');
Rewrite(data3);
Assign(data5,'d:\tp\iopm\datan'+stringvar2+'e'+stringvar3+'\afwpp.dat');
Rewrite(data5);
t := 0;
epstijdstap := eps/10;
epsint := eps/10;
END;( einde Lees_Initiele_Data)
END.
```

```

PROGRAM P_minimaal_vermogen;
{*****}
(*                                     *)
(* PROGRAM       : P_minimaal_vermogen *)
(* PURPOSE      : Bepaling van minimaal vermogen en geometrie m.b.v. *)
(*               de procedure minifun uit de numerieke bibliotheek. *)
(* VERSION      : 1.0 *)
(* DATE         : 12-11-1990 *)
(* LANGUAGE     : Turbopascal 5.5 (Borland) *)
(* SYSTEM       : Cirp-AT met EGA kaart en Numerieke Co-processor *)
(* PROGRAMMER   : ir. J.M.A. Du Bois *)
(* INFORMATION  : Technische Universiteit Eindhoven *)
(*               Faculteit Werktuigbouwkunde *)
(*               Vakgroep Produktietechnologie en - automatisering *)
(*               Laboratorium voor Omvormtechnologie *)
(*                                     *)
{*****}
USES
    Types,
    Prvarpp,
    Minifun,
    Crt,
    U_bgminf,
    Initdata;
{*****}

VAR
    x      : array1dr;
    xtype  : array1di;
    gtype  : array1di;
    lin    : array1db;
    info   : INTEGER;
    method : INTEGER;
    n      : INTEGER;
    m      : INTEGER;
    estima : REAL;
    stop   : INTEGER;
    conv   : INTEGER;
    coc    : INTEGER;
    imax   : INTEGER;

{** *****}

PROCEDURE initialiseer_numerieke_data;

BEGIN
    WriteLn(' Kies het aantal te optimaliseren parameters n ');
    WriteLn;
    WriteLn(' n = 1 : planparallele flens. ');
    WriteLn(' n = 2 : de flensdikte is een functie van de straal r, ');
    WriteLn(' eerste orde benadering. ');
    WriteLn(' n = 3 : de flensdikte is een functie van de straal r, ');
    WriteLn(' de orde van r wordt als te optimaliseren parameter ');
    WriteLn(' bepaald. ');
    WriteLn;
    Write(' n = ');
    ReadLn(n);
    WriteLn;

    aantal_te_optimaliseren_parameters := n;

```



( Deze waarde wordt toegekend aan een stringvariabele. )

```
Writeln(' Geef de nauwkeurigheid waarmee het vermogen wordt bepaald. ');
Writeln;
Writeln(' Let opl. Voor de reketijden gelden globaal de volgende waarden:');
Writeln('     eps_vermogen = 0.01   : 3 min. ');
Writeln('     eps_vermogen = 0.001  : 42 min. ');
Writeln('     eps_vermogen = 0.0001 : 23 uur 11 min. ');
Writeln;
Write(' eps_vermogen = ');
ReadLn(eps);
```

( M.b.v. deze constructie weet het programma naar welke subdirectorie )  
( de data gestuurd moet worden; de waarde van eps\_verm wordt toegekend )  
( aan een stringvariabele )

```
IF eps < 0.011 THEN eps_verm := 2;
IF eps < 0.0011 THEN eps_verm := 3;
IF eps < 0.00011 THEN eps_verm := 4;
```

```
x[1]      := 0.01;
x[2]      := -0.01;
X[3]      := 0.01;
xtype[1]  := 1;
xtype[2]  := 1;
XTYPE[3]  := 1;
gtype[1]  := 1;
lin[1]    := FALSE;
info      := 0;
method    := 0;
m         := 1;
stop      := 2;
coc       := 2;
imax      := 200;
estima    := 0.2;
```

END;

```
{*****}
{ Procedures m.b.t. de procedure minifun      }
{                                           }
```

```
PROCEDURE dist ( VAR x : array1dr;
```

```
VAR g : REAL;
i     : INTEGER );
```

BEGIN

```
K[t] := x[1];
IF n = 1 THEN
  BEGIN
    Lp[t] := 0;
    orde_n[t] := 0;
  END;
IF n = 2 THEN
  BEGIN
    Lp[t] := x[2];
    orde_n[t] := 1;
  END;
IF n = 3 THEN
  BEGIN
    Lp[t] := x[2];
    orde_n[t] := x[3];
  END;
```

```

Totaal_Vermogen(x[1],x[2],x[3]);
min_ver := Minimaal_Vermogen[t];
g := min_ver;
END;

(*****)

PROCEDURE ddist ( VAR x : array1dr;
                  VAR dg : array1dr;
                  i : INTEGER );

BEGIN
END;

(*****)

PROCEDURE dddist ( VAR x : array1dr;
                  VAR ddg : array2dr;
                  i : INTEGER );

BEGIN
END;

(*****)

PROCEDURE optimaliseer_vermogen;
BEGIN
    minfun( x, dist, xtype, gtype, lin, info, ddist, dddist, method, n, m,
           eps, eps, estima, stop, conv, coc, imax, data1 );

    K[t] := x[1];
    IF n = 1 THEN
    BEGIN
        ( de flens blijft planparallel )
        Lp[t] := 0;
        orde_n[t] := 0;
    END;
    IF n = 2 THEN
    BEGIN
        ( de axiale snelheid in de flens is een functie van de straal r )
        ( eerste orde benadering. )
        Lp[t] := x[2];
        orde_n[t] := 1;
    END;
    IF n = 3 THEN
    BEGIN
        ( de axiale snelheid in de flens is een functie van de straal r )
        ( de orde van r wordt bepaald middels optimalisatie. )
        Lp[t] := x[2];
        orde_n[t] := x[3];
    END;
    Totaal_Vermogen(x[1],x[2],x[3]);
END;

(*****)

(*****)
( Het hoofdprogramma. )

BEGIN
    clrScr;

```

```

Initialiseer_numerieke_data;
Lees_Initiele_Data;
REPEAT
    optimaliseer_Vermogen;

    WriteLn(' ru_Ster[' ,t:2,'] = ',ru_Ster[t]:7:5,
            ' Pmin = ',Minimaal_Vermogen[t]:7:5,
            ' K = ',K[t]:7:5,
            ' Lp = ',Lp[t]:7:5,
            ' N = ',orde_n[t]:7:5);
    WriteLn(data4, ' ru_Ster = ',ru_Ster[t]:7:5,
            ' Pmin = ',Minimaal_Vermogen[t]:7:5,
            ' K = ',K[t]:7:5,
            ' Lp = ',Lp[t]:7:5,
            ' N = ',orde_n[t]:7:5);

    Bepaal_flensdikte_en_uz_punt;
    t := t + 1;
    ( De grootte van de tijdstap wordt bepaald )
    Bepaal_Delta_Tijd_Ster;
UNTIL ( ru_ster[t-1] - 1 ) <= 0.05;
Close(data4);
Close(data3);
Close(data1);
Close(data5);
END.

```

```

{*****}
(*
(* UNIT      : Gegeneraliseerde bovengrensooplossing flensdikte(r) *)
(* PURPOSE   : Bepaal geometrie flens voor verschillende aannamen *)
(*            m.b.t. de snelheid in dikterichting,uz=(K.r^o_n+Lp).z*)
(*            van de flens, o_n bepaald de orde van r. *)
(*            De flens blijft planparallel voor n=0. *)
(* VERSION   : 4.0 *)
(* DATE      : 26-06-1991 *)
(* LANGUAGE  : Turbopascal 5.5 (Borland) *)
(* SYSTEM    : Cirp-AT met EGA kaart en Numerieke Co-processor *)
(* PROGRAMMER : ir. J.M.A. Du Bois *)
(* INFORMATION: Technische Universiteit Eindhoven *)
(*            Faculteit Werktuigbouwkunde *)
(*            Vakgroep Produktietechnologie en - automatisering *)
(*            Laboratorium voor Omvormtechnologie *)
(* *)
{*****}

```

```
UNIT U_bgminf;
```

```
INTERFACE
```

```
USES
```

```
Types,
Inttypes,
Calcutil,
Funcanal,
Dualint,
Prvarpp,
Crt,
Printer,
Initdata;
```

```

PROCEDURE Bepaal_Initiele_Geometrie;
FUNCTION Radiale_Snelheid_Ster( r_Var_Ster : Real ) : Real;
FUNCTION Nulpunt_Functie ( r_Ster_Nulpunt : Real ) : Real;
PROCEDURE Bepaal_Nulpunt;
PROCEDURE Bepaal_flensdikte;
PROCEDURE Bepaal_ru_Ster;
PROCEDURE Bepaal_Delta_Tijd_Ster;
FUNCTION Functie_f ( r_Var_Ster : Real ) : Real;
FUNCTION Int_Func_Def_Vermogen_Ster( r_Ster_Var : Real ) : Real;
PROCEDURE Totaal_Vermogen( Var K_Var,Lp_var,orde_n_var : Real ) ;
PROCEDURE Bepaal_flensdikte_en_uz_punt;

```

```
IMPLEMENTATION
```

```

{*****}
{
{ Bepaling van de flensdikte en de flensstraal op t = 0 ( t = 0 ) }
}

```

```

PROCEDURE Bepaal_Initiele_Geometrie;
BEGIN
    ru_Ster[0] := ruo_Ster;
    s_Ster[0] := so_Ster;
END;{Bepaal_Initiele_Geometrie}

```

```
{*****}
```

```

(
( De radiale snelheid van een materieel deeltje als functie v.d. positie)
( in de flens.
)

```

```

FUNCTION Radiale_Snelheid_Ster( r_Var_Ster : Real ) : Real;
VAR tus1,tus2,tus3,tus4 : Real;
BEGIN
  tus1 := -Exp((orde_n[j-1] + 1)*Ln(r_Var_Ster)) + 1/r_Var_Ster;
  tus2 := -r_Var_Ster + 1/r_Var_Ster;
  tus3 := - 1/r_Var_Ster;
  tus4 := (K[j-1]/(orde_n[j-1] + 2))*tus1 + (Lp[j-1]/2)*tus2 + tus3;
  Radiale_Snelheid_Ster := tus4;
END;(Radiale_Snelheid_Ster)

```

```

(*****)
(
( Functie voor het bepalen van de positie van een materieel deeltje )
( op de tijd t = j-1 , r*(j-1), als de positie op de tijd t = j, r*(j), )
( bekend is. De waarde van j loopt van t tot 1. De functie wordt aan- )
( geroepen in de procedure bepaal_nulpunt )
)

```

```

FUNCTION Nulpunt_Functie ( r_Ster_Nulpunt : Real ) : Real;
VAR tus1,tus2: Real;
BEGIN
  tus1 := Delta_Tijd_Ster[j]*Radiale_Snelheid_Ster(r_Ster_nulpunt);
  tus2 := -r_Ster[j] + r_Ster_Nulpunt;
  Nulpunt_Functie := tus1 + tus2;
END;(Nulpunt_Functie)

```

```

(*****)
(
( Procedure voor het bepalen van die waarde van r_Ster_Nulpunt uit de )
( Nulpunt_Functie waarvoor geldt dat de waarde van deze functie gelijk )
( is aan nul. Hiertoe wordt gebruik gemaakt van de procedure zeroinab )
( uit de numerieke bibliotheek.
)

```

```

PROCEDURE Bepaal_Nulpunt;
BEGIN
  Zeroinab(Nulpunt_Functie,r_Ster_Nulpunt,0.99,2.3,1E-6,1E-6,np_true);
  r_Ster[j-1] := r_Ster_Nulpunt;
  ( De waarde van j loopt van t tot 1 )
END;(Bepaal_Nulpunt)

```

```

(*****)
(
( Procedure voor het bepalen van de momentane flensdikte s*(r*(t)) als )
( functie van de straal r*(t).
)

```

```

PROCEDURE Bepaal_flensdikte;
VAR tus1 : Real;
BEGIN
  FOR j := 1 TO t DO
  BEGIN
    tus1 :=( K[j-1]*Exp(orde_n[j-1]*Ln(r_ster[j-1])) + Lp[j-1] );
    tus1 := Delta_Tijd_Ster[j]*tus1;
    s_Ster[j] := s_Ster[j-1]*( tus1 + 1 );
    ( r_ster[j-1] is met de procedure bepaal_nulpunt bepaald )
  END;
END;

```

```
END;  
END;(Bepaal_Flensdikte)
```

```
{*****}  
{  
{ Procedure voor het bepalen van de positie van de buitenrand van de }  
{ flens,  $r^*(t) = ru^*(t)$ , op de tijd t. }  
{
```

```
PROCEDURE Bepaal_ru_Ster;  
VAR tus1 : Real;  
BEGIN  
    j := t;  
    tus1 := ru_Ster[t-1];  
    tus1 := tus1 + Delta_Tijd_Ster[t]*Radiale_Snelheid_Ster(ru_Ster[t-1]);  
    ru_Ster[t] := tus1;  
END;(Bepaal_ru_Ster)
```

```
{*****}  
{  
{ Procedure voor het bepalen van het tijdstapje dt }  
{
```

```
PROCEDURE Bepaal_Delta_Tijd_Ster;  
VAR tus1,tus2,tus3,tus4,dt_begin,dt_eind : Real;  
    mm,oo,ll : integer;  
BEGIN  
    Dt_begin := 1;  
    oo := 2;  
    mm := 2;  
    Repeat  
        delta_tijd_ster[t] := dt_begin;  
        Bepaal_ru_Ster;  
        tus1 := ru_ster[t];  
        tus4 := ru_ster[t-1];  
        delta_tijd_ster[t] := dt_begin/oo;  
        for ll := 1 to oo do  
            begin  
                Bepaal_ru_Ster;  
                ru_Ster[t-1] := ru_Ster[t];  
            end;  
            tus2 := ru_ster[t];  
            tus3 := Abs((tus1-tus2)/tus2);  
            dt_eind := dt_begin;  
            dt_begin := dt_begin/mm;  
            ru_ster[t-1] := tus4;  
        Until ( (tus3) <= (epstijdstap));  
        delta_tijd_ster[t] := dt_eind;  
        writeln('dt= ',delta_tijd_ster[t]:7:5);  
END;
```

```
{*****}  
{  
{ De functie  $f(r^*)$  uit de term voor de effectieve reksnelheid: }  
{
```

```
FUNCTION Funktie_f ( r_Var_Ster : Real ) : Real;  
VAR tus1,tus2,tus3 : Real;  
BEGIN
```

```

tus1 := -Exp(orde_n[t]*Ln(r_Var_Ster)) + Sqr(1/r_Var_Ster);
tus2 := -1 + Sqr(1/r_Var_Ster);
tus1 := (K[t]/(orde_n[t] + 2))*tus1 + (Lp[t]/2)*tus2;
reksnelheid_phi_Ster := tus1 - Sqr(1/r_Var_Ster);
reksnelheid_z_Ster := K[t]*Exp( orde_n[t]*Ln(r_Var_Ster) ) + Lp[t];
reksnelheid_r_Ster := - ( reksnelheid_phi_Ster + reksnelheid_z_Ster );
tus2 := Sqr(reksnelheid_phi_Ster) + Sqr(reksnelheid_r_Ster) + Sqr(reksnelheid_z_Ster);
tus3 := Sqr(orde_n[t])*Sqr(K[t])*Exp( (2*orde_n[t] - 2)*Ln(r_Var_Ster) );
Funktie_f := (2*tus2)/tus3;
END;(Funktie_f)

(*****)
(
( De integraalfunctie van het te dissiperen deformatievermogen als )
( funktie van de momentane positie van een materieel deeltje r_Ster(i) )

FUNCTION Int_Func_Def_Vermogen_Ster( r_Ster_Var : Real ) : Real;
VAR tus1,tus2,tus3,tus4,f_r,s_r,tus5,tus6,tus7,tus8,tus9,tus10 : Real;
    eps_z_ster,eps_r_ster,eps_phi_ster : Real;
BEGIN
    IF orde_n[t] = 0 THEN
        BEGIN
            tus1 := -1 + Sqr(1/r_Ster_Var);
            tus2 := (K[t]/ 2)*tus1 + (Lp[t]/2)*tus1;
            reksnelheid_phi_Ster := tus2 - Sqr(1/r_Ster_Var);
            reksnelheid_z_Ster := K[t] + Lp[t];
            reksnelheid_r_Ster := - ( reksnelheid_phi_Ster + reksnelheid_z_Ster );
            tus2:=Sqr(reksnelheid_phi_Ster)+Sqr(reksnelheid_r_Ster)+Sqr(reksnelheid_z_Ster);
            tus3 := Sqrt( (2/3)*tus2 ) ;
            tus4 := s_Ster[t]*tus3;
                ( de reksnelheid vermenigvuldigd met de momentane plaatdikte )
                ( bij de aanname : planparallele flens. )
        END( Einde IF orde_n[t] = 0 THEN )
    ELSE ( orde_n <> 0 )
        BEGIN
            IF t >= 1 THEN
                BEGIN
                    r_Ster[t] := r_Ster_Var;
                    FOR j := t DOWNTO 1 DO
                        BEGIN
                            Bepaal_nulpunt;
                                ( Hiermee wordt de voorgeschiedenis van het )
                                ( materieel deeltje op de positie r_Ster(t) bepaald )
                                ( tot aan de tijd t = 0. )
                        END;
                            Bepaal_flensdikte;
                                ( Hiermee wordt de flensdikte op de positie )
                                ( r_Ster(t) bepaald op het tijdstip t. )
                END;( Einde IF t >= 1 THEN )
            f_r := Functie_f( r_Ster_Var );
            s_r := s_Ster[t];
                ( s_Ster[t] is bepaald met de procedure Bepaal_Flensdikte )
            tus1 := (f_r/2)*Ln(( Sqrt(f_r + s_r*s_r) + s_r ) / Sqrt(f_r) );
            tus2 := (s_r/2)*Sqrt(f_r + s_r*s_r);
                ( tus1 + tus2 is de waarde van de analytische oplossing )
                ( van de integraal over z. )
            tus3 :=Abs(Sqrt(1/3)*orde_n[t]*K[t]*Exp((orde_n[t]-1)*Ln(r_Ster_Var)));

```

```

tus4 := tus3*( tus1 + tus2 );
                                ( De waarde van de reksnelheid van een materieel deeltje. )
END;( einde if orde_n <> 0 )
IF versterking > 0 THEN
BEGIN
  IF t >= 1 THEN
  BEGIN
    IF orde_n[t] = 0 THEN
    BEGIN
      r_Ster[t] := r_Ster_Var;
      FOR j := t DOWNTO 1 DO
      BEGIN
        Bepaal_nulpunt;
                                ( Hiermee wordt de voorgeschiedenis van het      )
                                ( materieel deeltje op de positie r_Ster(t) bepaald )
                                ( tot aan de tijd t = 0.                          )
      END;
    END;( einde if orde_n = 0, d.w.z. planparallele flens )
    FOR l := 1 TO t DO
    BEGIN
      ( Bepaling van de effectieve rek )
      eps_z_Ster := K[l-1]*Exp( orde_n[l-1]*Ln(r_Ster[l-1])) + Lp[l-1];
      tus5 := -Exp(orde_n[l-1]*Ln(r_Ster[l-1])) + Sqr(1/r_Ster[l-1]);
      tus6 := -1 + Sqr(1/r_Ster[l-1]);
      tus7 := (K[l-1]/(orde_n[l-1] + 2))*tus5 + (Lp[l-1]/2)*tus6;
      eps_phi_Ster := tus7 - Sqr(1/r_Ster[l-1]);
      eps_r_Ster := - ( eps_phi_Ster + eps_z_Ster );
      tus8 := Sqr(eps_phi_Ster) + Sqr(eps_r_Ster) + Sqr(eps_z_Ster);
      Effectieve_Reksnelheid_Ster[l-1] := Sqrt((2/3)*tus8);
      tus9 := Effectieve_Reksnelheid_Ster[l-1];
      Effectieve_Rek[l] := Effectieve_Rek[l-1] + Delta_Tijd_Ster[l]*tus9;
    END;( for l = 1 to t )
  END;( if t >= 1 )
  IF voordeformatie = 0 THEN
  BEGIN
    tus10 := Exp(versterking*Ln(Effectieve_rek[t] + 0.000001));
                                ( Deze constructie wordt toegepast om de waarde v.d.      )
                                ( te optimaliseren parameters te benaderen. Als de voor- )
                                ( deformatie 0 wordt gekozen kan minifun geen waarden    )
                                ( voor de te optimaliseren parameters bepalen omdat het  )
                                ( vermogen voor elke combinatie van de te optimaliseren  )
                                ( parameters gelijk aan 0 zou zijn.                    )
  END
  ELSE ( if voordeformatie > 0 )
    tus10 := Exp(versterking*Ln(effectieve_rek[t] + voordeformatie));
  END( if versterking > 0 )
  ELSE( if versterking = 0 )
  BEGIN
    tus10 := 1;
  END;
  Int_Func_Def_Vermogen_Ster := tus4*tus10*r_Ster_Var;
END;( einde Int_Func_Def_Vermogen_Ster)

```

```

( ***** )
( )
( Met deze procedure wordt het totaal te dissiperen vermogen voor de )
( flens bepaald. Deze procedure wordt aangeroepen door de procedure )
( minifun uit de numerieke bibliotheek. Binnen deze procedure wordt )
( de procedure trapezium_integratie aangeroepen uit de numerieke biblio- )

```



( theek, hiermee kan de integratie over r numeriek worden uitgevoerd. )

PROCEDURE Totaal\_Vermogen( Var K\_Var,Lp\_var,orde\_n\_var : Real ) ;

VAR tus1,tus2,tus3,tus4 : Real;

BEGIN

IF K\_Var = 0 THEN

( Voor het geval de waarde van K gelijk aan 0 zou worden is de vol- )

( gende veiligheidsconstructie ingebouwd, zie analyse: IOPM-Dieptrekken )

( rapportnr. 010, formule (2.18) )

K\_Var := K\_Var + 0.000000001;

K[t] := K\_Var;

IF aantal\_te\_optimaliseren\_parameters = 1 THEN

BEGIN

Lp[t] := 0;

orde\_n[t] := 0;

END;

IF aantal\_te\_optimaliseren\_parameters = 2 THEN

BEGIN

Lp[t] := Lp\_var;

orde\_n[t] := 1;

END;

IF aantal\_te\_optimaliseren\_parameters = 3 THEN

BEGIN

Lp[t] := Lp\_var;

orde\_n[t] := orde\_n\_var;

END;

IF t >= 1 THEN

BEGIN

Bepaal\_ru\_Ster; ( Bepaling van de bovenste integratiegrens )

IF orde\_n[t] = 0 THEN

BEGIN

tus1 := (K[t-1]\*Exp(orde\_n[t]\*Ln(ru\_ster[t-1]))+Lp[t-1]);

tus1 := Delta\_tijd\_Ster[t]\*tus1;

s\_Ster[t] := s\_Ster[t-1]\*( tus1 + 1 );

END;( einde if orde\_n[t] = 0 )

END ( einde if t >= 1 )

ELSE ( t = 0 )

Bepaal\_Initiele\_Geometrie; ( Als t = 0 wordt s\_Ster(0) en ru\_Ster(0) )

tus1:=Trapezium\_Integratie(Int\_Func\_Def\_Vermogen\_Ster,ru\_Ster[t],1,epsint);

Deformatie\_Vermogen[t] := (1/so\_ster)\*tus1;

( Bepaling van het wrijvingsvermogen )

tus2 := (mu/2)\*p\_ster\*(Sqr(ruo\_ster) - 1)/so\_ster;

tus3 := K[t]\*(-Exp((orde\_n[t]+1)\*Ln(ru\_ster[t]))+1/ru\_ster[t])/(orde\_n[t]+2);

tus4 := Lp[t]\*(-ru\_ster[t]+1/ru\_ster[t])/2 - 1/ru\_ster[t];

Wrijving\_Vermogen[t] := tus2\*Abs(tus3+tus4);

Minimaal\_Vermogen[t] := Deformatie\_Vermogen[t] + Wrijving\_vermogen[t];

END;(PROCEDURE Def\_Vermogen)

(\*\*\*\*\*)

( )

( Bepaling van de flensdikte-quotient als functie van de positie van )

( een materieel deeltje in de flens op de tijd t, r\_Ster(t). )

( Dit gebeurt op 9 posities in de momentane flens. )

( Bepaling van de snelheid in dikterichting aan de rand van de flens )

( op de tijd t = i, r\_Ster(t) = ru\_Ster(t). )

PROCEDURE Bepaal\_flensdikte\_en\_uz\_punt;

( bepaald

```

VAR helling,fdq_ri,fdq_ru : Real;
BEGIN
  IF orde_n[t] = 0 THEN
  BEGIN
    u_punt_z := ( K[t]*Exp(orde_n[t]*Ln(ru_Ster[t])) + Lp[t] ) *s_Ster[t];
      ( Bepaal de snelheid in dikterichting aan de buitenrand v.d. flens )
      ( s_Ster(t) is gelijk aan de dikte van de flens )

    WriteLn(' ru_Ster[,t:2,'] = ',ru_Ster[t]:7:5,
            ' uz(ru) = ',u_punt_z:7:5,
            ' fdq(ru) = ',(s_ster[t]/so_Ster):7:5);
    WriteLn(data3,' ru_Ster = ',ru_Ster[t]:7:5,
            ' fdq(ru) = ',(s_ster[t]/so_Ster):7:5,
            ' uz(ru) = ',u_punt_z:7:5);

  END( Einde THEN statement orde_n[i] = 0 )
  ELSE
  BEGIN ( Begin ELSE statement orde_n <> 0 )
    Str(t,Stringvar1);
    Assign(data2,'d:\tp\iopm\datan'+Stringvar2+'e'+Stringvar3+'\fdq'+Stringvar1+'.dat');
    rewrite(data2);
    FOR o := 0 TO 8 DO
    BEGIN
      r_Ster[t] := ((ru_Ster[t]-1)/8)*o + 1;
      ( Bepaling van de radiale positie van een deeltje in de flens )
      ( afhankelijk van de waarde van o op de tijd t. )
      IF t >= 1 THEN
      BEGIN
        FOR j := t DOWNT0 1 DO
        BEGIN
          Bepaal_nulpunt;
          ( Bepaal de voorgeschiedenis van r_Ster(t) )
        END;( einde FOR j := t DOWNT0 1 DO Statement )
        Bepaal_flensdikte;
          ( Bepaal de bij r_Ster(t) Behorende flensdikte )
        END;( Einde statement t>=1 )
        u_punt_z := ( K[t]*Exp(orde_n[t]*Ln(r_Ster[t])) + Lp[t] ) *s_Ster[t];
          ( Bepaal de axiale snelheid )

        IF o = 0 THEN
        fdq_ri := s_Ster[t]*ri/so;
        IF o = 8 THEN
        BEGIN
          fdq_ru := s_Ster[t]*ri/so;
          helling := (fdq_ru - fdq_ri)/fdq_ri;
        END;( einde if o = 8 )
        WriteLn(' t = ',t:3,
            ' r_Ster = ',r_Ster[t]:7:5,
            ' fdq = ',(s_Ster[t]*ri/so):7:5,
            ' uz(r) = ',u_Punt_z:7:5);

        WriteLn(data2,' r_Ster = ',r_Ster[t]:7:5,
            ' fdq = ',(s_Ster[t]*ri/so):7:5,
            ' uz(r) = ',u_Punt_z:7:5);

      END;( einde FOR o := 0 TO 8 DO Statement)
      WriteLn(data5,' ru_ster = ',ru_Ster[t]:7:5,
            ' afwpp = ',helling:7:5);

      WriteLn(' afwpp = ',helling:7:5);
      close(data2);
      u_punt_z := ( K[t]*Exp(orde_n[t]*Ln(ru_Ster[t])) + Lp[t] ) *s_Ster[t];
      ( s_Ster(t) is gelijk aan de dikte van de flens aan de buitenrand )

```

```

      ( op de positie r_Ster(t) = ru_Ster(t) ,dus o = 8 )
WriteLn(' ru_Ster[,t:2,'] = ',ru_Ster[t]:7:5,
        ' uz(ru) = ',u_punt_z:7:5,
        ' fdq(ru) = ',(s_ster[t]/so_Ster):7:5);
WriteLn(data3,' ru_Ster = ',ru_Ster[t]:7:5,
        ' fdq(ru) = ',(s_ster[t]/so_Ster):7:5,
        ' uz(ru) = ',u_punt_z:7:5);

      END;( Einde ELSE Statement orde_n > 0 )
END;( Bepaal_flensdikte_en_uz_punt )

END.

```