MASTER

Ontwikkeling van short-cut rekenmethoden voor het berekenen van de droogtijd van deeltjes

van Schaik-van Hoek, Antoinette

Award date:
1986

Technische Hogeschool Eindhoven
Afd. der Scheikundige Technologie
Vakgroep Fysische Technologie

ONTWIKKELING VAN SHORT-CUT REKENMETHODEN VOOR HET BEREKENEN VAN DE DROOGTIJD VAN DEELTJES

met : - holle en massieve hoofd-geometrieën
- concentratieafhankelijke diffusiecoëfficiënt
- constante grensvlakconcentratie
- krimpende en niet-krimpende systemen

Afstudeerhoogleraar : Prof. Dr. Ir. H. A. C. Thijssen
Afstudeercoach : Ir. W. J. Coumans

Antoinette van Schaik-van Hoek, juni 1986

Samenvatting

Het drooggedrag van deeltjes met een variabele diffusiecoëfficiënt kan worden beschreven met een differentiaalvergelijking die i.h.a. numeriek moet worden opgelost. Dit leidt in de regel tot gecompliceerde rekenmethoden die in de praktijk voor de procesingenieur onhanteerbaar zijn. Voor praktisch gebruik zijn short-cut-rekenmethoden gewenst.

Op basis van voorgaand werk wordt in dit afstudeeronderzoek een bijdrage geleverd aan de ontwikkeling van een kortsluit-rekenmethode voor de berekening van de droogtijd van isotherm (niet-)krimpende holle of massieve deeltjes (bol, cylinder, vlakke laag) met een diffusiecoëfficiënt die volgens een machtsrelatie afhangt van de concentratie.

De short-cut methode kan worden geverifiëerd aan de hand van de "exacte" numerieke oplossing.

INHOUD

Bijlagen :

Appendices :

1. Inleiding

Het berekenen van de droogtijd van een (niet-)krimpend systeem met een concentratieafhankelijke diffusiecoëfficiënt geschiedt door het oplossen van de diffusievergelijking met bijbehorende begin- en randvoorwaarden. De diffusievergelijking is voor constante diffusiecoëfficiënt (Bosch [9]) analytisch oplosbaar, echter meestal moet dit probleem numeriek worden aangepakt. Een bijkomend probleem ontstaat doordat producteigenschappen, zoals diffusiecoëfficiënten en sorptieisothermen meestal niet bekend zijn en alleen via moeizame en moeilijke experimenten kunnen worden bepaald. Op grond van het bovenstaande is het ontwikkelen van eenvoudige en praktisch hanteerbare rekenmethoden (zogenaamde kortsluit-rekenmethoden) voor de beschrijving van het drooggedrag gewenst.
De strategie bij het ontwikkelen van kortsluit-rekenmethoden is:
- zoek de oplossing van het probleem voor extreme situaties (zeer korte respectievelijk zeer lange droogtijden).
- verbind beide extrema op een verantwoord vloeiende wijze met elkaar.
- verifiëer de verkregen kortsluit-rekenmethode m.b.v. de "exacte" numerieke oplossing.

In dit afstudeeronderzoek is een bijdrage geleverd aan de ontwikkeling van een kortsluit-rekenmethode m.b.t. het drooggedrag van holle en massieve (niet-)krimpende systemen met een diffusiecoëfficiënt die volgens een machtsrelatie ($D_r = m^a$) afhangt van de concentratie.
Bij het ontwikkelen van deze methode wordt er gebruik gemaakt van bijzondere eigenschappen van de droogstadia :
- Penetratie Periode (PP) : gedurende deze periode is de concentratie in het centrum van het deeltje nog niet veranderd. Bij voldoend kleine droogtijden ($\tau \rightarrow 0$) speelt het diffusiegedrag zich af in een zeer dunne laag aan de buitenkant van het deeltje. Het drooggedrag gedraagt zich onafhankelijk van de geometrie en de holte en wordt gekenmerkt door een G-parameter (Hfdst. 4).
- Regular Regime (RR) : bij niet-krimpende systemen waarvoor geldt $D_r = m^a$ verandert na voldoend lange tijd de vorm van de concentratieprofielen niet meer.
  Het kengetal van Sherwood voor de disperse fase wordt constant d.w.z. wordt onafhankelijk van de efficiëncy en is dus een karakteristieke grootheid voor deze periode.

Wanneer men de beschikking heeft over een kortsluit-rekenmethode die resultaten levert die binnen enkele procenten liggen van de "exacte" numerieke output kan de procesingenieur voor elke droogsituatie snel een analyse maken voor o.a. de droogtijd van zijn product.

Alhoewel in het onderhavige werk wordt uitgegaan van een droogproces waarbij water de te verwijderen component is, zijn de rekenregels in principe toepasbaar voor elk diffusieproces.

2. Diffusievergelijking.

De droogsnelheid en droogtijd van (niet-)krimpende deeltjes kan als functie van de fractie verwijderd water worden berekend door het numeriek oplossen van een niet-lineaire diffusievergelijking waarbij rekening moet worden gehouden met de sterke afhankelijkheid van de diffusiecoëfficiënt met de waterconcentratie [8].
Analoog aan het werk van Schoeber [1] en van Liou en Bruin [3] is er uitgegaan van een machtsrelatie tussen de diffusiecoëfficiënt en de waterconcentratie waarbij de waarde van de diffusiecoëfficiënt tot nul nadert bij de evenwichtsconcentratie van water in het product.
Aangenomen is dat het grensvlak van het deeltje (in geval van krimp) zich homogeen en gelijkmatig terugtrekt en dat de concentrisch ingesloten gasbel bij holle geometrieën een constante straal heeft.
Er zijn dan vier droogstadia te onderscheiden met elk hun specifieke eigenschappen en randvoorwaarden :

- periode I : met constante wateractiviteit aan het oppervlak en dalende m_i
- periode II : met afnemende wateractiviteit aan het oppervlak en $m_i > 0.1$
- periode III : Penetratieperiode (waterconcentratie aan het oppervlak $m_i < 0.1$ en in het centrum $m_{centr} > 0.9$)
- periode IV : Regular Regime waar $m_i < 0.1$ en de $m_{centr} < 0.9$.

2.1. Gegeneraliseerde diffusievergelijking.

Het desorptiegedrag van massieve en holle deeltjes (vlakke laag, cylinder, bol) kan zowel voor volledige, gedeeltelijke als voor niet-krimpende systemen worden beschreven d.m.v. één algemene diffusievergelijking:

$$\frac{\partial m}{\partial \tau} = \frac{\partial}{\partial \phi}\left(D_r X^2 \frac{\partial m}{\partial \phi}\right) \qquad (2.1)$$

met beginvoorwaarde :

$$m = 1 \quad | \quad (\tau = 0 \wedge 0 \leq \phi \leq 1) \qquad (2.2)$$

en randvoorwaarden :

$$X \frac{\partial m}{\partial \phi} = 0 \quad | \quad (\tau > 0 \wedge \phi = 0) \qquad (2.3)$$

$$m = m_i \quad | \quad (\tau > 0 \wedge \phi = 1) \qquad (2.4.a)$$

of

$$F = - D_r X_i \frac{\partial m}{\partial \phi} \quad | \quad (\tau > 0 \wedge \phi = 1) \qquad (2.4.b)$$

De dimensieloze grootheden m, ϕ, τ, D_r , X en F zijn hierna gedefinieerd :

F = fluxparameter (zie Hfdst. 2.2)

$$m = \frac{v - v^*}{v^0 - v^*} \qquad (2.5)$$

$$\text{met } v = \frac{\text{volumefractie water}}{\text{volumefractie vloeistof}} = \frac{d_s \ \rho_w}{d_w \ \rho_s} \qquad (2.6)$$

In deze vergelijking zijn d_s resp. d_w de partiële dichtheid van vaste stof respectievelijk water. Aangenomen is dat er geen volumeverandering optreedt door menging zodat de dichtheid van de vaste stof gelijk is aan de dichtheid van de zuivere vaste stof ($d_s = d_{s,p}$). De grootheden ρ_w en ρ_s zijn de concentraties van water resp. vaste stof in kg m^{-3}

v^0 = beginwaarde van v

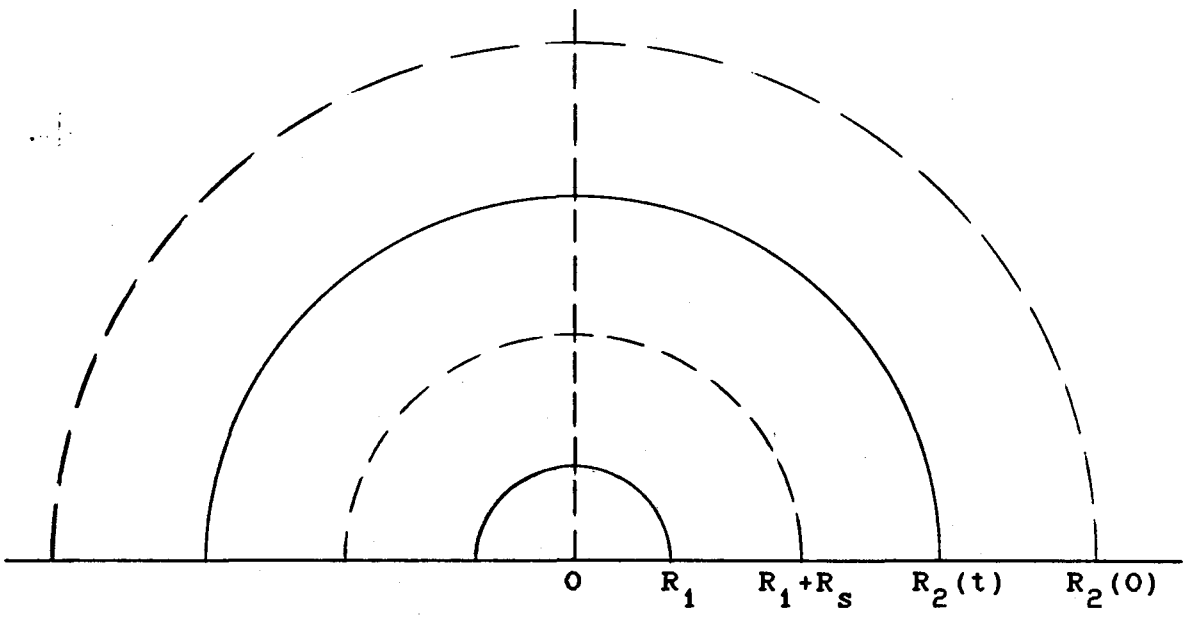
v^* = evenwichtswaarde van v

De dimensieloze plaatscoordinaat :

$$\phi = \frac{R_1\int^{r} \rho_s \, r^{\nu} \, dr}{R_1\int^{R_2(t)} \rho_s \, r^{\nu} \, dr} \qquad (2.7)$$

met r = plaatscoördinaat [m]
R_1 = gefixeerde interne straal van een holle cylinder of bol [m]. Voor een vlakke laag geldt $R_1 = 0$)
$R_2(t)$ = (dynamische) externe straal van het (krimpende) deeltje op tijd t in [m]

ν = geometrieparameter
- $\nu = 0$ voor een vlakke laag
- $\nu = 1$ voor een (holle en massieve) cylinder
- $\nu = 2$ voor een (holle en massieve) bol



bol/cylinder
illustratie plaatscoördinaat en grensvlakken voor een krimpend systeem

fig. 2.1.

De dimensieloze diffusiecoëfficiënt D_r is gedefiniëerd als :

$$D_r = \frac{D\,\rho_s^2}{D_0 \rho_{s0}^2} \qquad (2.8)$$

met D = actuele diffusiecoëfficiënt [$m^2\ s^{-1}$]
D_0 = waarde van de diffusiecoëfficient bij ρ_{w0}

De relatie tussen D_r en m wordt beschreven met een machtsafhankelijkheid :

$$D_r = m^a \qquad (2.9)$$

Deze relatie impliceert dat de diffusiecoëfficiënt tot nul nadert als v nadert tot de evenwichtswaarde v^* .
D_0 en a zijn fysische grootheden die experimenteel moeten worden bepaald.

De dimensieloze tijd τ is gedefinieerd als :

$$\tau = \frac{D_0\,\rho_{s,0}^2\,t}{(d_{s,ap} R_s)^2} \qquad (2.10)$$

met R_s = vaste stof straal [m]; halve dikte van een vlakke laag of dikte van de wanddikte van een hol deeltje indien alle water is verwijderd, dus als de gemiddelde waterinhoud $\bar{v} = 0$

$d_{s,ap}$ = schijnbare dichtheid van de vaste stof na volledige droging [kg m^{-3}].

Indien er gedeeltelijke of zelfs geen krimp optreedt zal de droge vaste stof poreus zijn. In het algemeen kan worden gesteld dat :

$$\rho_{s,0} <= d_{s,ap} <= d_{s,p}$$

$$R_{s,p} <= R_s <= (R_{2,0} - R_i)$$

De holte van het deeltje is gedefiniëerd als :

$$\lambda = \frac{R_i}{R_i + R_s} \qquad (2.11)$$

De dimensieloze parameter X is gedefiniëerd als :

$$X = X_{i,\sigma=0} \left[\lambda^{\nu+1} + (1 - \lambda^{\nu+1}) \int_0^{\phi} (1 + \sigma v)\, d\phi \right]^{\frac{\nu}{\nu+1}} \qquad (2.12)$$

waarin

$$X_{i,\sigma=0} = (\nu + 1)\frac{(1 - \lambda)}{1 - \lambda^{\nu+1}} \qquad 2.13)$$

en σ = volumetrische krimpcoëfficiënt, welke is gedefiniëerd als

$$\frac{V}{V_{droog}} = 1 + \sigma \bar{v} \qquad (2.14)$$

met

V = volume deeltje bij gemiddelde waterconcentratie $\bar{v}$

V_{droog} = volume van het droge deeltje ($\bar{v} = 0$).

Aangenomen is dat σ onafhankelijk is van de waterconcentratie;
$\sigma = 0$ voor niet-krimpende systemen
$\sigma = 1$ voor volledig krimpende systemen
dus $0 \leq \sigma \leq 1$

Ten behoeve van efficiënter rekenwerk is voor het geval van niet-krimpende systemen de diffusievergelijking vereenvoudigd en herschreven in onderstaande vorm zoals deze ook door Reniers [11] is gebruikt.

$$\frac{\partial u}{\partial \tau} = \frac{(1-\lambda)^2}{r^{\nu}} \frac{\partial}{\partial r}\left(r^{\nu} D_r \frac{\partial u}{\partial r}\right) \qquad (2.15)$$

met beginvoorwaarde $u = 1 \quad | \tau = 0 \ \wedge \ \lambda < r < 1$ (2.16)

en randvoorwaarden $\frac{\partial u}{\partial r} = 0 \quad | \tau > 0 \ \wedge \ r = \lambda$ (2.17)

$u = u^{\ddagger}(\tau) \quad | \tau > 0 \ \wedge \ r = 1$ (2.18)

met als aanname de parameterfunctie $D_r = u^a$

waarbij

$$u = \frac{\rho_w - \rho^{\#}_w}{\rho^{0}_w - \rho^{\#}_w}$$

ρ_w : waterconcentratie [kg m^{-3}];

ρ^{0}_w: beginconcentratie;

$\rho^{\ddagger}_w$: concentratie op het interface;

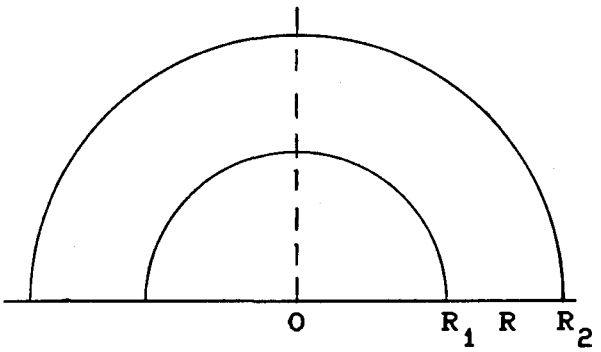
ρ^{*}_w: concentratie in evenwicht met de externe condities in het drooggas;

$\rho^{\#}_w$: referentieconcentratie voor de definitie van een concentratie afhankelijke diffusiecoëfficiënt;

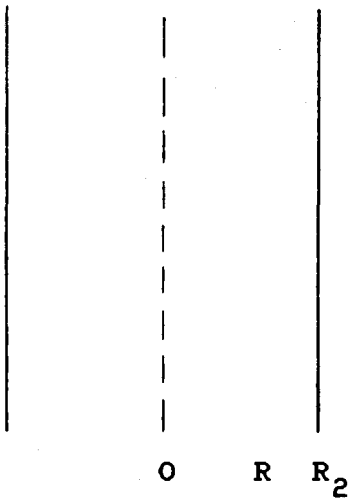
$\bar{\rho}_w$: gemiddelde concentratie;

In een normale droogsituatie geldt: $\rho^{0}_w \geq \bar{\rho}_w \geq \rho^{\ddagger}_w \geq \rho^{*}_w \geq \rho^{\#}_w$

$r = \frac{R}{R_2}$ = afstandscoördinaat gebaseerd op de externe straal van het deeltje (zie fig. 2.2 en 2.3)



bol/ cylinder



vlakke laag

illustratie plaatscoördinaat en grensvlakken voor een niet-krimpend systeem

fig 2.2 fig 2.3

R = plaatscoördinaat, afstand vanaf het centrum van een cylinder, bol of (tweezijdig drogende) vlakke laag [m]

R_1 = positie van de interne grens voor holle bol of cylinder (R_1 = 0 voor een vlakke laag)

R_2 = positie van de externe grens van het deeltje

Door de plaatscoördinaat te normaliseren m. b. v.
$y = \frac{R - R_1}{R_2(t) - R_1}$ is onderstaande vorm van de diffusievergelijking verkregen voor het niet-krimpende systeem :

$$\frac{\partial u}{\partial \tau} = \frac{1}{(\lambda + y(1-\lambda))^{\nu}} \frac{\partial}{\partial y}\left((\lambda + y(1-\lambda))^{\nu} D_r \frac{\partial u}{\partial y} \right) \qquad (2.15a)$$

met beginvoorwaarde $u = 1 \quad | \tau = 0 \ \wedge \ 0 < y < 1$ (2.16a)

en randvoorwaarden $\frac{\partial u}{\partial y} = 0 \quad | \tau > 0 \ \wedge \ y = 0$ (2.17a)

$u = u^{\ddagger}(\tau) \quad | \tau > 0 \ \wedge \ y = 1$ (2.18a)

met als aanname de parameterfunctie $D_r = u^a$

2.2. De massabalans en de droogtijd.

Op het interface (i) (fig 2.1) is de parameter X gedefiniëerd als :

$$X_i = \frac{A\, d_{s,ap}\, R_s}{V\, \bar{\rho}_s} \qquad (2.19)$$

met A = uitwendig oppervlak van het deeltje

Uit 2.12 volgt door substitutie van $\phi = 1$:

$$X_i = X_{i,\sigma=0}\, [\, 1 + (1-\lambda^{\nu+1})\, \sigma\bar{v}\,]^{\frac{\nu}{\nu+1}} \qquad (2.20)$$

De fluxparameter F is gedefiniëerd als :

$$F = \frac{j_{wi}\, d_{s,ap}\, R_s}{D_0\, \rho_{s0}^2 \left(\frac{\rho_{w0} - \rho_w^*}{\rho_{s0} \quad \rho_s^*} \right)} \qquad (2.21)$$

met j_{wi} = waterflux door het dynamische oppervlak [kg $m^{-2}s^{-1}$]

De gemiddelde droog-efficiency E is :

$$E = \frac{v^0 - \bar{v}}{v^0 - v^*} = 1 - \bar{m} \qquad (2.22)$$

De dimensieloze massabalans is nu te schrijven als :

$$F\, X_i\, d\tau = dE \qquad (2.23)$$

waaruit na integratie volgt :

$$\tau = \int_0^E \frac{dE}{F\, X_i} \qquad (2.24)$$

Hieruit blijkt duidelijk dat de droogtijd kan worden berekend indien zowel F als X_i als functie van E bekend zijn.

2.3. Sh_d-concept

Indien tijdens drogen de centrumconcentratie kleiner wordt dan 0.9 spreekt men van de Regular Regime periode. Gedurende deze periode kan de flux door het interface worden beschreven met behulp van het Sherwood-getal voor de disperse fase, gedefiniëerd als :

$$F = \frac{1}{2} \frac{Sh_d}{(a+1)} \left[(1-E)^{a+1} - (1-E_i')^{a+1} \right] \qquad (2.25)$$

waarin $Sh_d = Sh_d\ (a,\ \nu, RVW)$

met F = flux door het grensvlak

a = parameter uit de machtsrelatie : $D_r = m^a$

E = gemiddelde droogefficiency

E_i' = locale droogefficiency op de plaats $r = R_2(t)$

2.4. Periode met constante wateractiviteit aan het oppervlak (periode I).

Gedurende de periode van constante wateractiviteit aan het oppervlak wordt de droogflux volledig gecontroleerd door externe condities maar zal niet constant blijven voor krimpende bollen en cylinders wegens het feit dat de stofoverdrachtscoëfficiënt in de gasfase een functie is van de dynamische externe straal van het krimpende deeltje.
Gebaseerd op massa- en volumebalansen kan de fluxparameter van de constante activiteitsperiode (F_{ca}) worden gerelateerd aan zijn initiële waarde F_{caO} volgens :

$$F = \frac{F_{caO}}{(1-sE)^{\frac{\alpha}{\nu+1}}} \qquad (2.27)$$

Feitelijk wordt zo de randvoorwaarde op $\phi = 1$ (verg. 2.4.b) van de diffusievergelijking voor krimpende systemen vervangen door bovenstaande vergelijking 2.27 met

$$s = \frac{(1-\lambda^{\nu+1})\ \sigma(v^{o} - v^{*})}{1 + (1-\lambda^{\nu+1})\ \sigma v^{o}} = \text{gemodificeerde krimpcoëfficiënt} \qquad (2.28)$$

q = 0 voor een vlakke laag ($\nu = 0$)
q = 1 indien het Sherwoodgetal voor de gasfase (Sh') constant is
q = 1-n indien Sh' afhangt van het Reynoldsgetal volgens $Sh' \approx Re^{n}$

Voor niet-krimpende deeltjes laat vergelijking 2.27 zich vereenvoudigen tot $F_{ca} = F_{ca0}$ en er kan worden gesproken van een constante flux periode.

3. Numerieke berekeningen

3.0. Inleiding

Aan de hand van de beschikbare procedures, ontwikkeld door Reniers [11] zijn de volgende programma's ontwikkeld voor de berekening van het drooggedrag van deeltjes :

-NIETKRIMP : niet-krimp met m_i = constant
-KRIMP : krimp met m_i = constant
-NKCF : niet-krimp met constante flux en
-KCA : krimp met constante wateractiviteit.

Deze programma's berekenen o.a. concentratieprofielen, efficiency's E, droogfluxen F, het kengetal van Sherwood voor de disperse fase Sh_d en de parameter G (zie hfdst. 4.3).

Bovengenoemde programma's worden elk afzonderlijk besproken evenals de hieronder genoemde aangeroepen rekenprocedures.
De resultaten van de berekeningen worden in datafiles op DISK en TAPE bewaard.

AANGEROEPEN REKEN-PROCEDURES		
PROGRAMMA	vlakke laag	cylinder/bol
NIETKRIMP	STARTSLAB (y-coörd.) PDE (y-coörd.)	STARTSLAB (y-coörd.) STARTNONSLAB (y-coörd.) PDE (y-coörd.)
KRIMP	STARTSLAB (y-coörd.) PDE (y-coörd.)	INITIALIZEFI (ϕ-coörd.) STARTNONSLABFI (ϕ-coörd.) PDEFI (ϕ-coörd.)
NKCF	CONSTANTFLUX (y-coörd.)	CONSTANTFLUX (y-coörd.)
KCA	CONSTANTFLUX (y-coörd.)	CONSTANTFLUXFI (ϕ-coörd.)

Tabel 3.1. overzicht gebruikte reken-procedures

Verder zijn er nog :

Procedures voor lay-out

KOPJETABEL	voor constante grensvlakconcentratie
KOPJETAPE	voor constante grensvlakconcentratie
KOPJETABELCF	voor constante flux
KOPJETAPECF	voor constante flux

Programma's voor inlezen van data-files

LEESNIETKRIMP	zie App. E
LEESNKPROFIEL	zie App. F
LEESKRIMPFILE	zie App. G
LEESNKCF	zie App. H
LEESKCA	zie App. I

Tabel 3.2. overzicht gebruikte procedures

(Pas op : Alhoewel in sommige datafiles wordt gesuggereerd dat r-coördinaten zijn gebruikt is dit niet het geval. In al deze situaties is ook in y-coördinaten gewerkt.)

3.1. Korte beschrijving van beschikbare reken-procedures

3.1.1. STARTSLAB (Reniers)

Deze procedure is toepasbaar voor een niet-krimpend systeem ($\sigma=0$) en m_i=constant (verg. 5.6 Reniers[11] pag. 28).

Voor een vlakke laag met constante grensvlakconcentratie m_i retourneert deze procedure het concentratieprofiel en de retentie op het einde van de Penetratie-periode.
Voor een bol en cylinder met constante grensvlakconcentratie m_i retourneert deze procedure het concentratieprofiel en de retentie op $\tau = 0$.

procedure heading :

```
BOOLEAN PROCEDURE STARTSLAB(N, TAU, A, BV, U, RET, ACC, ITMAX);
VALUE N, TAU, A, BV, ACC, ITMAX; INTEGER N, ITMAX;
REAL TAU, A, BV, ACC, RET; REAL ARRAY U[*];
```

formele parameters :

N : aantal intervallen in de y-coördinaat (enkele honderden);

TAU : (geschatte) einde van de penetratie-periode τ_{pp} ;
zo gekozen dat de concentratie in het centrum nog niet merkbaar gaat dalen.
Voor a $\geq$ 0 levert τ_{pp} = 0.01 goede resultaten;

A : exponent in de machtsrelatie $D_r = u^a$;

BV : constante grensvlakconcentratie $u^{\ddagger}$;

U : array [0:N] dat het concentratieprofiel ($u-u^{\ddagger}$) bevat voor een vlakke laag aan het einde van de penetratieperiode.
U[i] bevat de oplossing op de plaats y = i/N;
De concentratieprofielen voor $\tau \leq \tau_{pp}$ zijn vrijwel identiek alleen lineair gecomprimeerd. Hierdoor kan U[i] worden beschouwd als de oplossing op de plaats
$y = 1 - \frac{\sqrt{\tau}}{\sqrt{\tau_{pp}}} \frac{(N-i)}{N}$;

RET : retentie ($1-E^*$) voor een vlakke laag aan het einde van de penetratieperiode. Voor $\tau \leq \tau_{pp}$ kan de retentie worden berekend m.b.v. een Taylorreeksontwikkeling in $\sqrt{\tau}$ omdat gedurende de penetratie E^* evenredig is met $\sqrt{\tau}$;

ACC : nauwkeurigheid die moet worden bereikt voordat de iteratie (successieve substitutie) kan worden beëindigd, (meestal 10^{-6});

ITMAX: maximale aantal toegestane iteraties (meestal 20).

De boolean procedure STARTSLAB krijgt de waarde FALSE indien de vereiste nauwkeurigheid zelfs met het maximale aantal iteratieslagen niet kan worden bereikt. Het aldus verkregen profiel is echter niet nauwkeurig genoeg en en het is dan zinvol om het programma af te breken en die situatie of met grotere ITMAX of een fijner plaatsrooster (N groter) of met kleinere TAU of grotere waarde voor ACC te proberen.

3.1.2. STARTNONSLAB (Reniers)

Deze procedure is toepasbaar voor de penetratieperiode van niet-krimpende bollen en cylinders (σ=0 en m_i=constant), (verg. 5.2 Reniers[11] pag. 28).
De procedure retourneert concentratieprofielen en retenties op diverse tijdsintervallen voor $\tau \leq \tau_{pp}$.
Deze procedure kan alleen worden gebruikt na een voorafgaande aanroep van STARTSLAB die U[0,*] initialiseert. De variabelen N, TAU, A en BV moeten zowel voor STARTSLAB als voor STARTNONSLAB dezelfde waarde hebben.

procedure heading :

```
BOOLEAN PROCEDURE STARTNONSLAB(N, TAU, INTVALS, NU,
                                LAMBDA, A, BV, U, RET, ACC, ITMAX);
VALUE N, TAU, INTVALS, NU, LAMBDA, A, BV, ACC, ITMAX;
INTEGER N, INTVALS, NU, ITMAX; REAL TAU, LAMBDA, A, BV, ACC;
REAL ARRAY U[*,*], RET[*];
```

formele parameters :

N : aantal intervallen in de y-coördinaat (enkele honderden);

TAU : (geschatte) einde van de penetratie-periode; Voor a > 0 levert τ_{pp} = 0.01 goede resultaten;

INTVALS : aantal subintervallen waarin τ_{pp} is verdeeld;

NU : geometriefactor ν, ν = 1 voor cylinders, ν = 2 voor bollen;

LAMBDA : holte λ;

A : exponent in de machtsrelatie $D_r = u^a$;

BV : constante grensvlakconcentratie $u^{\ddagger}$;

U : 2-dimensionaal array [0:INTVALS, 0:N].
Bij aanroep bevat de nulde rij U[0,*]het profiel dat door STARTSLAB is afgeleverd.
Na afloop bevat U[j,i] de oplossing $(u-u^{\ddagger})$ op $\tau = \tau_{pp} * \frac{i}{INTVALS}$ en $y = 1 - \frac{\sqrt{\tau}}{\sqrt{\tau_{pp}}} \frac{(N-i)}{N}$

RET : array[0:INTVALS]; RET[j] bevat de retentie $(1-E^*)$ op $\tau_{pp} * \frac{j}{INTVALS}$

ACC : nauwkeurigheid die moet worden bereikt voordat de iteratie (successieve substitutie) kan worden beëindigd (meestal 10^{-6});

ITMAX : maximale aantal toegestane iteraties (meestal 20).

3.1.3. PDE (Reniers)

Deze procedure is toepasbaar voor een niet-krimpend systeem ($\sigma=0$) en m_i=constant, mits $\tau > \tau_{pp}$ (verg. 5.19 van Reniers[11] pag. 34).
De procedure kan alleen worden gebruikt nadat STARTSLAB en eventueel STARTNONSLAB de oplossing voor de penetratieperiode hebben geleverd.
De procedure retourneert concentratieprofielen en retenties voor tijden groter dan τ_{pp}.
De variabelen N, NU, LAMBDA, A en BV moeten dezelfde waarde hebben als in de aanroep van STARTSLAB en STARTNONSLAB.
De procedure wordt in een lus herhaaldelijk aangeroepen totdat een voldoend groot tijdsdomein is doorgerekend.

procedure heading ;

```
BOOLEAN PROCEDURE PDE(N, DTAU, STEPS, NU, LAMBDA, A,
                      BV, U, RET, ACC, ITMAX);
VALUE N, DTAU, STEPS, NU, LAMBDA, A, BV, ACC, ITMAX;
REAL ARRAY U[*];
INTEGER N, STEPS, NU, ITMAX;
REAL DTAU, LAMBDA, A, BV, ACC, RET;
```

formele parameters :

N : aantal intervallen in de y-coördinaat (enkele honderden);
DTAU : interval $\Delta\tau$ gebruikt in de discretisatie;

STEPS : aantal stappen $\Delta\tau$ dat in 1 aanroep wordt genomen;
NU : geometriefactor ν, ν = 0 voor vlakke lagen,
ν = 1 voor cylinders,
ν = 2 voor bollen;
LAMBDA : holte λ;
A : exponent in de machtsrelatie $D_r = u^a$;

BV : constante grensvlakconcentratie $u^\ddagger$;
U : array [0:N].
Bij aanroep bevat dit array het concentratieprofiel $(u-u^\ddagger)$ door STARTSLAB of STARTNONSLAB afgeleverd.
Na afloop bevat U[*] een nieuw concentratieprofiel dat STEPS*DTAU verder weg ligt in het τ-domein;

RET : de retentie $(1-E^*)$ op het nieuwe τ-niveau;
ACC : nauwkeurigheid die moet worden bereikt voordat de iteratie (successieve substitutie) kan worden beëindigd (meestal 10^{-5});
ITMAX : maximale aantal toegestane iteraties (meestal 20).

Omdat de parameter U een input/output-parameter is, en de waarde die U heeft bij het ingaan van de procedure wordt overschreven

door de uitgaande waarde van U, dient er extern voor te worden gezorgd dat de profielen tijdelijk worden bewaard in een array ("doorgeven van profielen") zodat later tijdens het maken van plotplaatjes en berekeningen van kengetallen alle gegevens nog voorhanden zijn.

3.1.4. INITIALIZEFI (Reniers)

Deze procedure is toepasbaar voor een krimpend systeem ($0<\sigma\leq 0$) en m_i=constant (verg. 6.7 Reniers[11] pag. 40).

INITIALIZEFI initialiseert de oplossing van de diffusievergelijking op $\tau = 0$ voor een constante grensvlakconcentratie.
Voor krimpende vlakke lagen zijn de oplossingen identiek aan die voor niet-krimpende vlakke lagen zodat voor vlakke lagen gebruik wordt gemaakt van de procedures voor niet-krimp.

procedure heading :

```
BOOLEAN PROCEDURE INITIALIZEFI(N, TAU, NU, LAMBDA, A,
                               SIGMA, VREF, VINTF, VO,
                               U, ACC, ITMAX);
VALUE N, TAU, NU, LAMBDA, A, SIGMA, VREF, VINTF, VO, ACC, ITMAX;
INTEGER N, NU, ITMAX; REAL ARRAY U[*];
REAL TAU, LAMBDA, A, SIGMA, VREF, VINTF, VO, ACC;
```

formele parameters :

N : aantal intervallen in het ϕ-domein (enkele honderden);

TAU : (geschatte) einde van de penetratie-periode τ_{pp} ; d.i. wanneer de concentratie in het centrum nog niet merkbaar gaat dalen.

NU : geometriefactor ν, (ν = 0 wordt niet geaccepteerd)
ν = 1 voor cylinders,
ν = 2 voor bollen;

LAMBDA : holte λ;

A : exponent in de machtsrelatie $D_r = u^a$;

SIGMA : krimpcoëfficiënt σ, (verg. 2.14);

VREF, VINTF, VO : volumeverhoudingen, (verg. 2.6)
Voor het deeltje homogeen op resp.
- referentie conditie (u=0)
- grensvlakconcentratie ($u=u^{\ddagger}$)
- beginvoorwaarde (u=1)

U : array [0:N] dat het initialisatieprofiel ($u-u^{\ddagger}$) bevat aan het einde van de penetratieperiode.

ACC : nauwkeurigheid die moet worden bereikt voordat de iteratie (successieve substitutie) kan worden beëindigd (meestal 10^{-6});

ITMAX : maximale aantal toegestane iteraties (ca. 20).

Indien zelfs met het maximale aantal toegestane iteraties de vereiste nauwkeurigheid niet wordt gehaald krijgt de boolean procedure de waarde FALSE. Het programma kan, indien hiervoor wordt geöpteerd, worden afgebroken en er kan eventueel met een kleinere TAU of grotere waarde voor ACC of met een grotere ITMAX of N worden getracht tot een betrouwbare oplossing te komen.

3.1.5. STARTNONSLABFI (Reniers)

Deze procedure is toepasbaar voor krimpende cylinders en bollen ($0<\sigma\leq 0$) en m_i=constant (verg. 6.2 Reniers[11] pag. 39).

De procedure retourneert concentratieprofielen en retenties op diverse tijdsintervallen voor $\tau \leq \tau_{pp}$.

Deze procedure kan alleen worden gebruikt na een voorafgaande aanroep van INITIALIZEFI waarmee U[0,*] wordt geïnitialiseerd. De variabelen N, NU, LAMBDA, A, SIGMA, VREF, VINTF en VO moeten dezelfde waarde hebben als in de aanroep van INITIALIZEFI.

procedure heading :

```
BOOLEAN PROCEDURE STARTNONSLABFI(N, TAU, INTVALS, NU, LAMBDA,
                                 A, SIGMA, VREF, VINTF, VO,
                                 U, RET, ACC, ITMAX);
VALUE N, TAU, INTVALS, NU, LAMBDA, A, SIGMA, VREF, VINTF, VO, ACC, ITMAX;
INTEGER N, INTVALS, NU, ITMAX;
REAL TAU, LAMBDA, A, SIGMA, VREF, VINTF, VO, ACC;
REAL ARRAY U[*,*], RET[*];
```

formele parameters :

N	:	aantal intervallen in het ϕ-domein (enkele honderden);
TAU	:	(geschatte) einde van de penetratie-periode τ_{pp};
INTVALS	:	aantal subintervallen waarin τ_{pp} is verdeeld;
NU	:	geometriefactor ν, (ν = 0 wordt niet geaccepteerd) ν = 1 voor cylinders, ν = 2 voor bollen;
LAMBDA	:	holte λ;
A	:	exponent in de machtsrelatie $D_r = u^a$;
SIGMA	:	krimpcoëfficiënt σ, (verg. 2.14);
VREF, VINTF, VO	:	volumeverhoudingen, (verg. 2.6) Voor het deeltje homogeen op resp. - referentie conditie (u=0) - grensvlakconcentratie ($u=u^{\ddagger}$) - beginvoorwaarde (u=1)

U : 2-dimensionaal array [0:INTVALS, 0:N].
Bij aanroep bevat de nulde rij U[0, *]het profiel dat door INITIALIZEFI is afgeleverd.

Na afloop bevat U[j, i] de oplossing $(u-u^{\ddagger})$ op $\tau = \tau_{pp} * \frac{i}{INTVALS}$ en $\phi = 1 - \frac{\sqrt{\tau}}{\sqrt{\tau_{pp}}} \frac{(N-i)}{N}$

RET : array[0:INTVALS]; RET[j] bevat de retentie $(1-E^*)$ op $\tau_{pp} * \frac{j}{INTVALS}$

ACC : nauwkeurigheid die moet worden bereikt voordat de iteratie (successieve substitutie) kan worden beëindigd (meestal 10^{-6});

ITMAX : maximale aantal toegestane iteraties (meestal 20).

Indien zelfs met het maximale aantal toegestane iteraties de vereiste nauwkeurigheid niet wordt gehaald krijgt de boolean procedure de waarde FALSE. Als gevolg hiervan is uiteraard de output onbetrouwbaar bovendien zou de aanroep van INITIALIZEFI al eerder problemen hebben gegeven.

3.1.6. PDEFI (Reniers)

Deze procedure is toepasbaar voor een krimpend systeem ($0<\sigma\leq 0$) en m_i=constant (verg. 6.17 Reniers[11] pag. 43).

De procedure retourneert concentratieprofielen en retenties voor tijden groter dan τ_{pp}.

PDEFI kan alleen worden aangeroepen indien de procedures INITIALIZEFI en STARTNONSLABFI de penetratieperiode voor hun rekening hebben genomen.
De variabelen N, NU, LAMBDA, A, SIGMA, VREF, VINTF en VO moeten weer dezelfde waarde hebben als in de aanroep van INITIALIZEFI en STARTNONSLABFI.

procedure heading :

```
BOOLEAN PROCEDURE PDEFI(N, DTAU, STEPS, NU, LAMBDA, A, SIGMA, VREF,
                        VINTF, VO, U, RET, ACC, ITMAX);
VALUE N, DTAU, STEPS, NU, LAMBDA, A, SIGMA, VREF, VINTF, VO, ACC, ITMAX;
INTEGER N, STEPS, NU, ITMAX;
REAL DTAU, LAMBDA, A, SIGMA, VREF, VINTF, VO, ACC;
REAL ARRAY U[*];
```

formele parameters :

N : aantal intervallen in het ϕ-domein (enkele honderden);

DTAU : interval $\Delta\tau$ gebruikt in de discretisatie;

STEPS : aantal stappen, $\Delta\tau$, dat in een enkele aanroep wordt genomen;

NU : geometriefactor ν, (ν = 0 wordt niet geaccepteerd)
ν = 1 voor cylinders,
ν = 2 voor bollen;

LAMBDA : holte λ;

A : exponent in de machtsrelatie $D_r = u^a$;

SIGMA : krimpcoëfficiënt σ, (verg. 2.14);

VREF,
VINTF, VO : volumeverhoudingen, (verg. 2.6)
Voor het deeltje homogeen op resp.
- referentie conditie (u=0)
- grensvlakconcentratie ($u=u^{\ddagger}$)
- beginvoorwaarde (u=1)

U : array [0:N].
Bij aanroep bevat het array een concentratieprofiel ($u-u^{\ddagger}$).
Na afloop bevat het een nieuw concentratieprofiel dat STEPS*DTAU verder in het τ-domein ligt;

RET : de retentie ($1-E^*$) op het nieuwe τ-niveau;

ACC : nauwkeurigheid die moet worden bereikt voordat de iteratie (successieve substitutie) kan worden beëindigd (meestal 10^{-5});

ITMAX : maximale aantal toegestane iteraties (meestal 20).

Daar de procedure PDEFI veel robuuster is dan INITIALIZEFI en STARTNONSLABFI zal de booleanwaarde vrijwel nooit FALSE worden. Omdat de parameter U een input/output-parameter is, en de waarde die U heeft bij het ingaan van de procedure wordt overschreven door de uitgaande waarde van U, dient er extern voor te worden gezorgd dat de profielen tijdelijk worden bewaard in een array ("doorgeven van profielen") zodat later tijdens het maken van plotplaatjes en berekeningen van kengetallen alle gegevens nog voorhanden zijn.

3.1.7. CONSTANTFLUX (Reniers)

Deze procedure is toepasbaar voor niet-krimpende systemen met constante flux (verg. 5.19 Reniers[11] pag. 34).
De procedure retourneert het concentratieprofiel en de retentie voor vlakke lagen, cylinders en bollen.

procedure heading :

```
BOOLEAN PROCEDURE CONSTANTFLUX(N, DTAU, NU, LAMBDA, A,
                               FLUX, U, RET, ACC, ITMAX);
VALUE N, DTAU, NU, LAMBDA, A, FLUX, ACC, ITMAX;
REAL ARRAY U[*];
INTEGER N, NU, ITMAX;
REAL DTAU, LAMBDA, A, FLUX, RET, ACC;
```

formele parameters :

N : aantal intervallen in de y-coordinaat (enkele honderden);
DTAU : interval $\Delta\tau$ gebruikt in de discretisatie;
NU : geometriefactor ν, ν = 0 voor vlakke lagen,
ν = 1 voor cylinders,
ν = 2 voor bollen;
LAMBDA : holte λ;
A : exponent in de machtsrelatie $D_r = u^a$;
FLUX : constante flux $F^{\#}$;
U : array [0:N].
Bij aanroep bevat U[*] het concentratieprofiel (u)
Na afloop bevat U[*] een nieuw concentratieprofiel dat DTAU verder weg ligt in het τ-domein;
RET : de retentie $(1-E^{\#})$ op het nieuwe τ-niveau;
ACC : nauwkeurigheid die moet worden bereikt voordat de iteratie (successieve substitutie) kan worden beëindigd (meestal 10^{-5});
ITMAX : maximale aantal toegestane iteraties (meestal 20).

Deze boolean procedure krijgt de waarde FALSE indien zelfs met het maximale aantal toegestane iteraties de vereiste nauwkeurigheid niet kan worden behaald of wanneer de voorgeschreven flux niet langer kan blijven gehandhaafd (bij grote flux vrij snel het geval) omdat de concentratie aan het grensvlak U[N] <= 0 wordt. Omdat de parameter U een input/output-parameter is, en de waarde die U heeft bij het ingaan van de procedure wordt overschreven door de uitgaande waarde van U, dient er extern voor te worden gezorgd dat de profielen tijdelijk worden bewaard in een array ("doorgeven van profielen") zodat later tijdens het maken van plotplaatjes en berekeningen van kengetallen alle gegevens nog voorhanden zijn.

3.1.8. CONSTANTFLUXFI (Reniers)

Deze procedure is toepasbaar voor krimpende systemen met constante flux (verg. 6.2 Reniers[11] pag. 39), ($0 < \sigma \leq 1$).
De procedure retourneert het concentratieprofiel en de retentie voor vlakke lagen, cylinders en bollen.

procedure heading :

```
BOOLEAN PROCEDURE CONSTANTFLUXFI(N, DTAU, NU, LAMBDA, A,
                                 SIGMA, VREF, VO, FLUX,
                                 U, RET, ACC, ITMAX);
VALUE N, DTAU, NU, LAMBDA, A, SIGMA, VREF, VO, FLUX, ACC, ITMAX;
REAL ARRAY U[*];
INTEGER N, NU, ITMAX;
REAL DTAU, LAMBDA, A, SIGMA, VREF, VO, FLUX, RET, ACC;
```

formele parameters :

N : aantal intervallen in het ϕ-domein (enkele honderden);
DTAU : interval $\Delta\tau$ gebruikt in de discretisatie;
NU : geometriefactor ν, (ν = 0 wordt niet geaccepteerd)
ν = 1 voor cylinders,
ν = 2 voor bollen;
LAMBDA : holte λ;
A : exponent in de machtsrelatie $D_r = u^a$;
SIGMA : krimpcoëfficiënt σ, (verg. 2.14);
VREF, VO : volumeverhoudingen, (verg. 2.6)
Voor het deeltje homogeen op resp.
- referentie conditie (u=0)
- beginvoorwaarde (u=1)
FLUX : constante flux $F^{\#}$;
U : array [0:N].
Bij aanroep bevat U[*] het concentratieprofiel (u)
Na afloop bevat U[*] het nieuwe concentratieprofiel dat DTAU verder weg ligt in het τ-domein;
RET : de retentie ($1-E^{\#}$) op het nieuwe τ-niveau;
ACC : nauwkeurigheid die moet worden bereikt voordat de iteratie (successieve substitutie) kan worden beëindigd (meestal 10^{-5});
ITMAX : maximale aantal toegestane iteraties (meestal 20).

Deze boolean procedure krijgt de waarde FALSE indien zelfs met het maximale aantal toegestane iteraties de vereiste nauwkeurigheid niet kan worden behaald of wanneer de voorgeschreven flux niet langer kan blijven gehandhaafd (bij grote flux vrij snel het geval) omdat de concentratie aan het grensvlak U[N] <= 0 wordt.
De procedure wordt herhaaldelijk aangeroepen totdat U[N]=0.
Omdat de parameter U een input/output-parameter is, en de waarde die U heeft bij het ingaan van de procedure wordt overschreven door de uitgaande waarde van U, dient er extern voor te worden

gezorgd dat de profielen tijdelijk worden bewaard in een array ("doorgeven van profielen") zodat later tijdens het maken van plotplaatjes en berekeningen van kengetallen alle gegevens nog voorhanden zijn.

3.1.9. VARIFLUX (van Schaik)

Deze procedure is toepasbaar voor krimpende en niet-krimpende systemen ($0 \leq \sigma \leq 1$) met constante grensvlakwateractiviteit . De procedure retourneert het concentratieprofiel en de retentie voor vlakke lagen, cylinders en bollen.

procedure heading :

```
BOOLEAN PROCEDURE VARIFLUX(N, DTAU, NU, LAMBDA, A,
                           SIGMA, VREF, VO, Q, FLUX,
                           U, RET, ACC, ITMAX);
VALUE N, DTAU, NU, LAMBDA, A, SIGMA, VREF, VO, Q, FLUX, ACC, ITMAX;
REAL ARRAY U[*];
INTEGER N, NU, ITMAX;
REAL DTAU, LAMBDA, A, SIGMA, VREF, VO, Q, FLUX, RET, ACC;
```

formele parameters :

N : aantal intervallen in het ϕ resp. y-domein (enkele honderden);

DTAU : interval $\Delta\tau$ gebruikt in de discretisatie;

NU : geometriefactor ν, $\nu = 0$ voor vlakke lagen, $\nu = 1$ voor cylinders, $\nu = 2$ voor bollen;

LAMBDA : holte λ;

A : exponent in de machtsrelatie $D_r = u^a$;

SIGMA : krimpcoëfficiënt σ, (verg. 2.14);

VREF, VO : volumeverhoudingen, (verg. 2.6)
Voor het deeltje homogeen op resp.
- referentie conditie (u=0)
- beginvoorwaarde (u=1)

Q : parameter, $q = 0$ voor een vlakke laag
$q = 1$ indien Sh' constant
$q = 1 - n$ indien $Sh' \approx Re^n$

FLUX : beginflux F_{ca0};

U : array [0:N].
Bij aanroep bevat U[*] het concentratieprofiel (u)
Na afloop bevat U[*] een nieuw concentratieprofiel dat DTAU verder weg ligt in het τ-domein;

RET : de retentie $(1-E^{\#})$ op het nieuwe τ-niveau;

ACC : nauwkeurigheid die moet worden bereikt voordat de iteratie (successieve substitutie) kan worden beëindigd (meestal 10^{-5});

ITMAX : maximale aantal toegestane iteraties (meestal 20).

Deze boolean procedure krijgt de waarde FALSE indien zelfs met het maximale aantal toegestane iteraties de vereiste nauwkeurigheid niet kan worden behaald of wanneer de voorgeschreven flux niet langer kan blijven gehandhaafd (bij grote flux vrij snel het geval) omdat de concentratie aan het grensvlak U[N] <= 0 wordt.
De procedure wordt herhaaldelijk aangeroepen totdat U[N]=0.
Omdat de parameter U een input/output-parameter is, en de waarde die U heeft bij het ingaan van de procedure wordt overschreven door de uitgaande waarde van U, dient er extern voor te worden gezorgd dat de profielen tijdelijk worden bewaard in een array ("doorgeven van profielen") zodat later tijdens het maken van plotplaatjes en berekeningen van kengetallen alle gegevens nog voorhanden zijn.

3.2. Korte beschrijving van de reken-programma's.

3.2.1. NIETKRIMP - m_i = constant.

NIETKRIMP maakt gebruik van de procedures :

a) STARTSLAB
b) STARTNONSLAB (indien $\nu \neq 0$)
c) PDE

die de oplossing van de diffusievergelijking in y-coördinaten beschrijven (Reniers [11]).

Interpolatieformule voor de penetratieperiode

De efficiency E^* tijdens de penetratieperiode kan zeer goed worden benaderd met een Taylorreeksontwikkeling in $\sqrt{\tau}$ waarbij alleen de eerste drie termen worden meegenomen :

$$E^* = c_1 \frac{\sqrt{\tau}}{\sqrt{\tau_{pp}}} + c_2 \frac{\tau}{\tau_{pp}} + c_3 \frac{\tau \sqrt{\tau}}{\tau_{pp}\sqrt{\tau_{pp}}} \qquad (3.1)$$

Voor de fluxparameter F krijgen we dan :

$$F = \frac{\partial E/\partial \tau}{X_i} = \frac{1}{(X_i * \tau_{pp})} \left(\frac{c_1}{2 \frac{\sqrt{\tau}}{\sqrt{\tau_{pp}}}} + c_2 + 1.5\, c_3 \frac{\sqrt{\tau}}{\sqrt{\tau_{pp}}} \right) \qquad (3.2)$$

waarbij $\tau = \tau_{pp} * \frac{i}{INTVALS}$.

Voor $\nu = 0$ geldt $c_2 = c_3 = 0$.
De procedure STARTSLAB levert de retentie (RET_{ss}) op τ_{pp} zodat c_1 kan worden bepaald m.b.v. $c_1 = E_{ppss} = (1-RET_{ss})$
(waarbij pp = penetratieperiode en ss=startslab)
Voor een bol en cylinder is voor korte tijden alleen de eerste term uit (3.1) dominant, hierbij dient met de geometrie rekening te worden gehouden :

$$c_1 = X_1 E_{ppss} = X_1 (1-RET_{ss}) \qquad (3.3)$$

Voor een bol en cylinder worden de volgende gegevens uit STARTNONSLAB gebruikt om c_2 en c_3 te bepalen :

$\tau_1 = 1/2\ \tau_{pp} \rightarrow E_1 = 1 - RET[INTVALS/2]$
$\tau_2 = \tau_{pp} \rightarrow E_2 = 1 - RET[INTVALS]$

Na invullen in bovenstaande $\sqrt{\tau}$-polynoom volgt :

$$c_3 = \frac{E_2 - 2 * E_1 + c_1(\sqrt{2} - 1)}{1 - \frac{\sqrt{2}}{2}} \qquad \text{en } c_2 = E_2 - c_1 - c_3 \qquad (3.4)$$

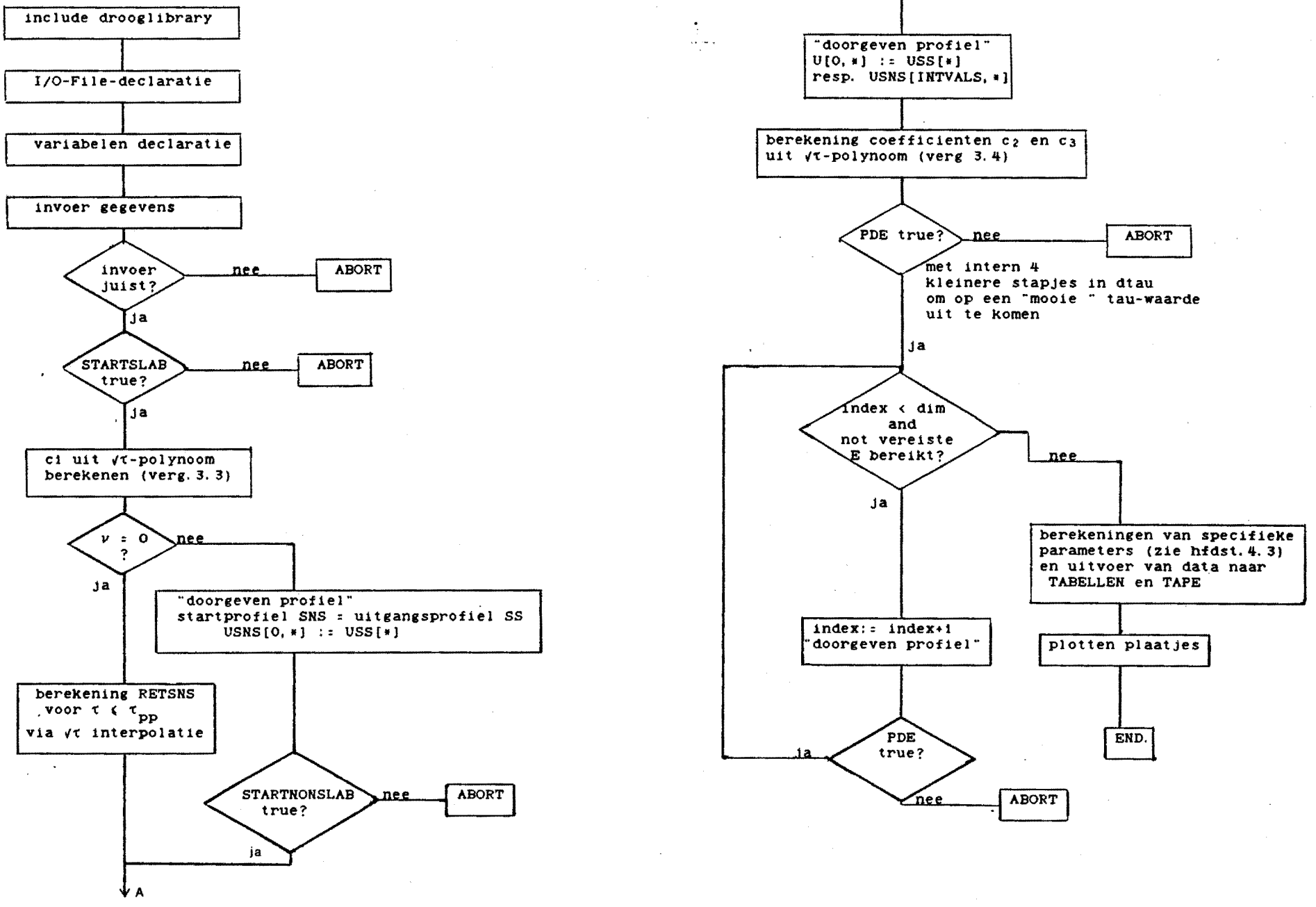
Deze berekeningen geschieden buiten de procedures.

Er moet gelden $c_2 >> c_3$ wat na afloop van het programma kan worden geverifiëerd.

3.2.2. Verklaring symbolen in stromingsdiagram voor NIETKRIMP.

SS	= STARTSLAB
SNS	= STARTNONSLAB
RETSS	= retentie uit SS
RETSNS	= retentie uit SNS
USS	= profiel uit SS
USNS	= profiel uit SNS
U	= profiel uit PDE
τ_{pp}	= tijdsduur penetratieperiode
index	= teller voor het aantal iteraties
dim	= maximale aantal toegestane iteraties

3. 2. 3. Stromingsdiagram voor NIETKRIMP



Daar dit programma voor a > 1 toch nog vrij veel rekentijd vergt om tot een efficiency E van ca. 0.96 te komen is er voor grotere a-waarden tot een veel lagere eindwaarde voor de efficiency doorgerekend. De profielen bij deze eindefficiency zijn op TAPE gezet zodat er eventueel later nog bepaalde situaties verder kunnen worden doorgerekend door dit laatste profiel als opstart-profiel aan PDE aan te bieden.
Met het programma LEESNIETKRIMP kunnen de op TAPE staande databestanden met berekeningen worden ingelezen (zie App. E).
Met het programma LEESNKPROFIEL kunnen de op TAPE staande profielen worden ingelezen (zie App. F).

3.2.4. KRIMP - m_i = constant

Voor krimpende vlakke lagen is de numerieke oplossing identiek aan die voor niet-krimpende vlakke lagen, zodat ook krimpende vlakke lagen worden uitgerekend met NIETKRIMP.
KRIMP maakt gebruik van de volgende procedures :

a) INITIALIZEFI
b) STARTNONSLABFI
c) PDEFI .

N.B. De efficiency E^* gedurende de penetratieperiode kan ook nu weer goed worden benaderd met de eerste drie termen van een Taylorreeksontwikkeling in $\frac{\sqrt{\tau}}{\sqrt{\tau_{pp}}}$

$$E^* = c_1 \frac{\sqrt{\tau}}{\sqrt{\tau_{pp}}} + c_2 \frac{\tau}{\tau_{pp}} + c_3 \frac{\tau \sqrt{\tau}}{\tau_{pp}\sqrt{\tau_{pp}}} \qquad (3.1)$$

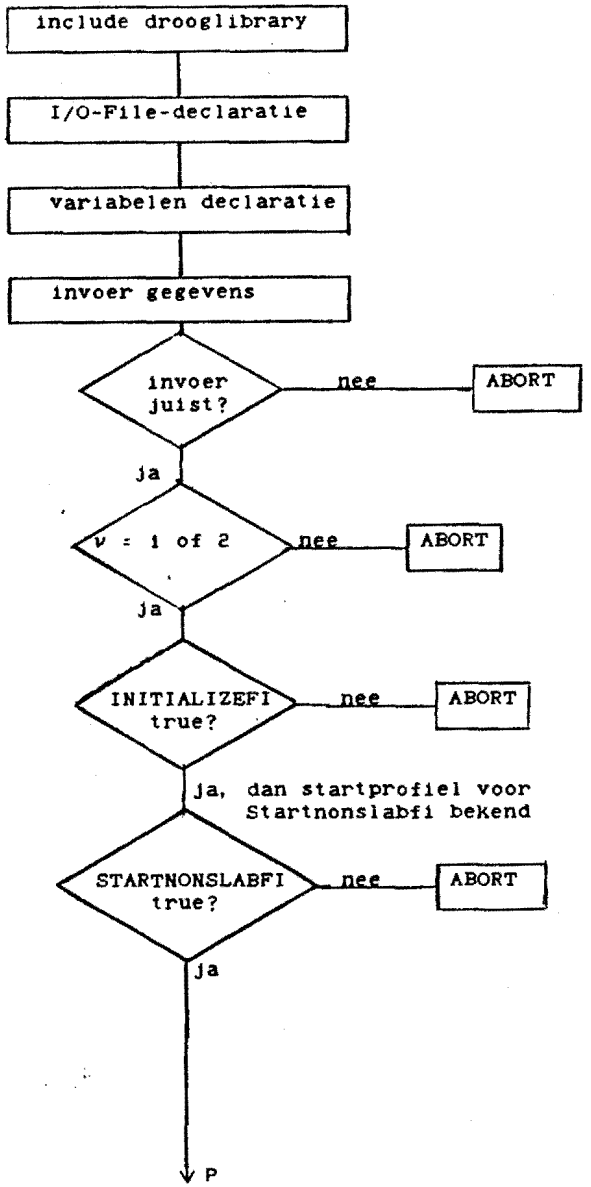
De constanten kunnen ook nu weer worden gevonden door het fitten van RET[1], RET[INTVALS/2] en RET[INTVALS] of door het toepassen van regressietechnieken op alle data uit array RET[0:DIM].

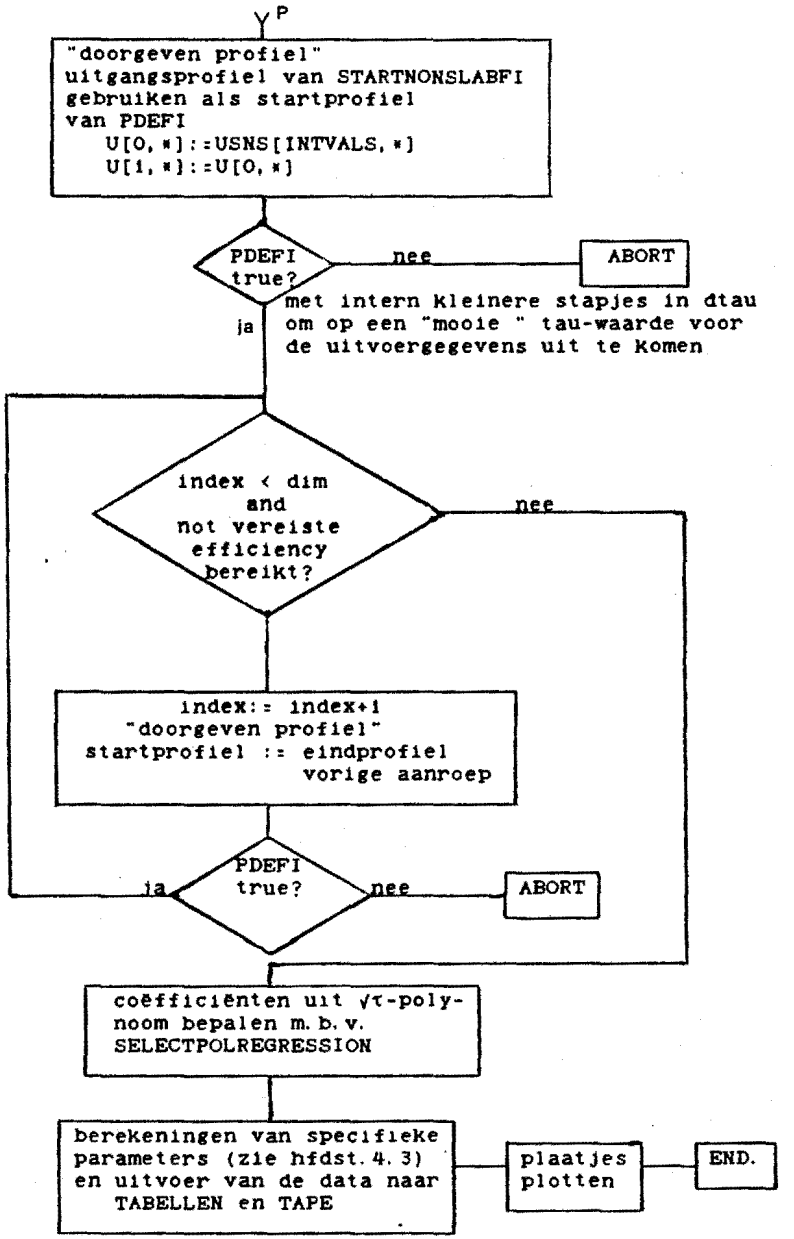
De procedure PDEFI wordt herhaaldelijk aangeroepen totdat een voldoende groot τ-domein is doorgerekend.
Bij waarden van V0 > 1 moet een kleinere DTAU-stap worden genomen dan bij waarden van V0 <= 1 daar anders de output onbetrouwbaar is (zie conclusies hfst. 5).

3.2.5. Verklaring symbolen in stromingsdiagram KRIMP

USNS	=	profiel uit STARTNONSLABFI
U	=	profiel uit PDEFI
τ_{pp}	=	tijdsduur penetratieperiode
index	=	teller voor het aantal iteraties
dim	=	maximale aantal toegestane iteraties

3.2.6. Stromingsdiagram voor KRIMP



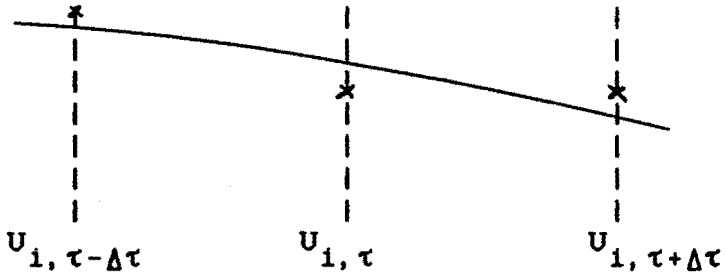


Met het programma LEESKRIMPFILE kunnen de op TAPE staande datafiles worden ingelezen (zie App. G).

3.2.7. NKCF

NKCF maakt gebruik van de procedure
- CONSTANTFLUX

Omdat de profielen uit de procedure CONSTANTFLUX oscillaties vertonen zowel in de plaats als in de tijd worden deze gladgestreken door een lineair gemiddelde te nemen van drie opeenvolgende profielen (zie fig. 3.1). Deze gladstrijkmethode is zeer effectief en succesvol, want zowel oscillaties in tijd als in plaats worden genivelleerd.



$$U_{i,\text{gladgestreken}} = \frac{\frac{1}{2}U_{i,\tau-\Delta\tau} + U_{i,\tau} + \frac{1}{2}U_{i,\tau+\Delta\tau}}{2}$$

concentratie U op één bepaalde roosterplaats
(i = 0, 1, 2, .. N)

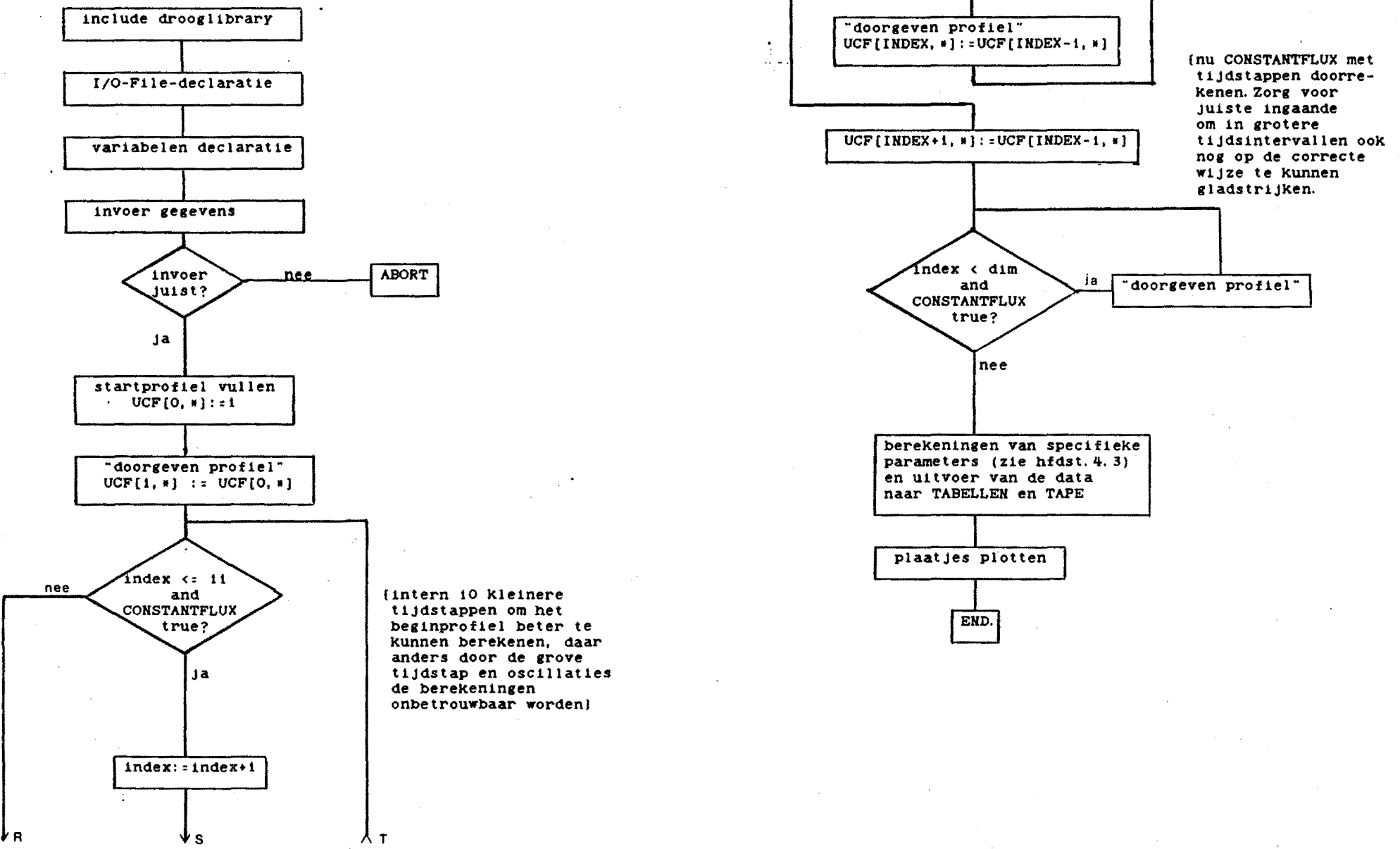
fig. 3.1.

Deze procedure wordt in een lus herhaaldelijk aangeroepen totdat een voldoend groot tijdsdomein is doorgerekend of tot de vereiste flux niet meer kan worden gehandhaafd doordat U[N] = 0

3.2.8. Verklaring symbolen in stromingsdiagram voor NKCF

UCF	=	profiel uit CONSTANTFLUX
index	=	teller voor het aantal iteraties
dim	=	maximale aantal toegestane iteraties

3. 2. 9. Stromingsdiagram voor NKCF



3.2.10. KCA

KCA maakt gebruik van de procedure
- VARIFLUX, waarin weer worden aangeroepen
  - CONSTANTFLUX en
  - CONSTANTFLUXFI.

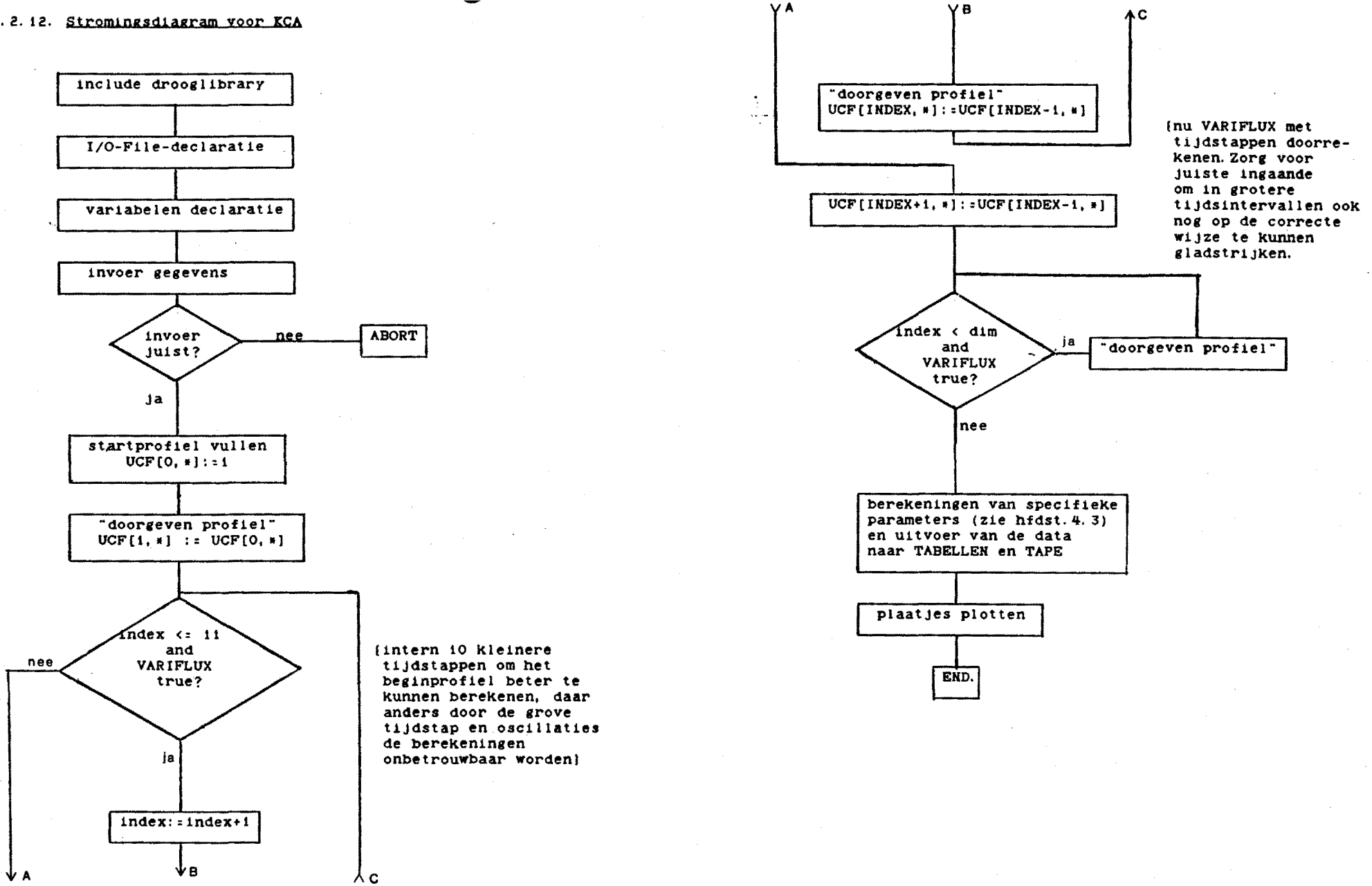
De procedure VARIFLUX roept CONSTANTFLUXFI aan met een over het tijdsinterval gemiddelde flux $F^{\#}$, die wordt berekend volgens vergelijking 2.27.
Voor $\nu = 0$ is de oplossing identiek aan de oplossing in y-coördinaten en wordt er gebruik gemaakt van de procedure CONSTANTFLUX. Evenals CONSTANTFLUX heeft ook CONSTANTFLUXFI last van oscillaties welke op analoge wijze als bij NKCF worden gladgestreken.

Met het programma LEESKCA kunnen de gegevens van de berekeningen die op TAPE staan weer worden ingelezen.

3.2.11. Verklaring symbolen in stromingsdiagram voor KCA

UCF = profiel uit VARIFLUX
index = teller voor het aantal iteraties
dim = maximale aantal toegestane iteraties

2.12. Stromingsdiagram voor KCA



4. Kortsluit-rekenmethoden

4.0. Inleiding

De droogtijd van een deeltje kan m.b.v. vergelijking 2.24 worden berekend indien zowel F als X_i als funktie van E bekend zijn.
Het verband X_i versus E kan uit vergelijking 2.20 worden gevonden. Uit numerieke resultaten kunnen relatief eenvoudige correlaties voor F versus E worden afgeleid.
Voor de ontwikkeling van deze correlaties zijn een aantal specifieke parameters van belang zoals :

- F fluxparameter
- E overall efficiency
- E_i' locale efficiency op het interface (r=R)
- E_{centre} locale efficiency in het centrum ($r=R_c$)
- G hulpparameter
- Sh_d kengetal van Sherwood voor de disperse fase.

De wijze waarop deze parameters worden berekend wordt in hfdst. 4.3 besproken.

4.1. Niet-krimpende systemen met $m_i=0$

PENETRATIEPERIODE

Het blijkt dat het penetratiegedrag goed kan worden beschreven met de G-parameter :

$$G = \frac{F*E}{X_i} \qquad (4.1)$$

Bij voldoend kleine droogtijden ($\tau \to 0$) speelt het diffusieproces zich af in een zeer dunne laag aan de buitenkant van het product. Het systeem gedraagt zich als een vlakke laag, m.a.w. het drooggedrag is onafhankelijk van de hoofdgeometrie en onafhankelijk van de holte (ingesloten gasbel) :

$$\lim_{\tau \to 0} G = G_0 \qquad (4.2)$$

waarbij G_0 onafhankelijk is van de geometrie.

Analytisch kan worden aangetoond dat bij constante diffusiecoëfficiënt $G_0 = \frac{2}{\pi}$ (4.3)

Als a ≠ 0 dan blijkt dat G_0 goed wordt beschreven met de correlatie :

$$G_0 = \frac{2}{\pi}\left(\frac{1.42}{a+1.42}\right)^{1.98} \qquad (4.4)$$

Voor $\tau \geq 0$ blijkt in de penetratieperiode :

$$G = G_0(1-\beta E) \qquad (4.5)$$

zodat de flux F_{pp} gedurende de penetratieperiode goed kan worden beschreven met :

$$F_{pp} = X_{i,\sigma=0}\, G_0\left(\frac{1}{E} - \beta\right) \qquad (4.6)$$

waarbij $\beta = \beta_{a=0}(1.25)^a$

met voor $\nu = 0$: $\beta_{a=0} = 0$

$\nu = 1$: $\beta_{a=0} = 0.71 - 0.13\,\lambda - 0.58\,\lambda^2$

$\nu = 2$: $\beta_{a=0} = 0.88 - 0.08\,\lambda - 0.08\,\lambda^2 - 0.72\,\lambda^3$

REGULAR REGIME

Gedurende een regular regime proces, dat zijn aanvang neemt als de centrumconcentratie lager dan 0.9 wordt, blijkt het kengetal van Sherwood voor de disperse fase (Sh_d) een zinvolle parameter te zijn voor de beschrijving van de flux F versus de efficiency E

$$F = \frac{1}{2}\,\frac{Sh_d}{(a+1)}\left((1-E)^{a+1} - (1-E_i')^{a+1}\right) \qquad (2.25)$$

Voortbouwend op het werk van Schoeber [1] is vastgesteld ([8], [10]) dat in het Regular Regime het getal van Sherwood voor de disperse fase (Sh_d) enkel en alleen een functie is van a, ν en λ in geval van niet-krimpende systemen, m.a.w. voor één bepaalde geometrie (ν, λ) en één bepaalde machtsafhankelijkheid (a) van D_r zal de waarde van Sh_d tijdens het R.R. constant worden, d.w.z. onafhankelijk van E.
Bij constante grensvlakconcentratie ($m_i = 0 \rightarrow E_i' = 1$) wordt de flux gedurende de Regular Regime periode gedefiniëerd als :

$$F_{rr} = \frac{1}{2}\,\frac{Sh_{d,rr}}{a+1}\,(1-E)^{a+1} \qquad (4.7)$$

waarbij de volgende correlaties gelden voor $Sh_{d,rr}$:

$$Sh_{d,rr} = Sh_{d,a=\infty} - \Delta Sh_d \frac{2}{a+2} \qquad (4.8)$$

waarin

$$Sh_{d,a=\infty} = 7.391 + (3.516\nu - 0.034)\left(\frac{X_{i,\sigma=0} - 1}{\nu}\right)^{(1.535 - 0.075\nu)} \qquad (4.9)$$

en

$$\Delta Sh_d = 2.456 + (2.720\nu - 0.087)\left(\frac{X_{i,\sigma=0} - 1}{\nu}\right)^{1.04} \qquad (4.10)$$

OVERGANGSCRITERIUM

Het einde van de penetratieperiode en het begin van de regular regime periode wordt beschreven in [8] door

$$E_{pp,rr} = \frac{1+0.1\nu(5-\nu)(1-\lambda)}{(a+2)} \qquad (4.11)$$

Aldus is de gehele F versus E-curve bekend en kan de droogtijd door integratie worden verkregen.

4.2. Krimpende systemen met $m_i = 0$

Om een verband te kunnen leggen tussen het drooggedrag van niet-krimpende en krimpende systemen wordt een krimpfactor gedefiniëerd als zijnde de verhouding van de fluxparameter van het krimpende systeem (F_σ) en de fluxparameter van het niet-krimpende systeem ($F_{\sigma=0}$) bij dezelfde waarde van de droogefficiency :

$$H = \frac{F_\sigma}{F_{\sigma=0}} \qquad (4.12)$$

Met behulp van vergelijking 4.1 en 2.20 is dit ook te schrijven als :

$$H = \left(1+(1-\lambda^{\nu+1})\ \sigma v^0\right)^{\frac{\nu}{\nu+1}} (1-sE)^{\frac{\nu}{\nu+1}} \frac{G_\sigma}{G_{\sigma=0}} \qquad (4.13)$$

Het vinden van F als functie van E is nu gereduceerd tot het vinden van H als functie van E.

Voor krimpende vlakke lagen is de oplossing van de diffusievergelijking identiek aan die voor niet-krimpende vlakke lagen, zodat voor een vlakke laag H = 1 en onafhankelijk van E.

Voor de krimpfactor van de penetratieperiode (H_{pp}) kan analytisch worden afgeleid dat

$$\lim_{E\to 0} H_{pp} = H_0 = \left\{ 1+(1-\lambda^{\nu+1})\sigma v^0 \right\}^{\frac{\nu}{\nu+1}} \qquad (4.14)$$

Bij benadering kan deze waarde gedurende de gehele penetratieperiode worden gehandhaafd.

Analoog aan 4.14 moet voor de limietsituatie (E→0) voor verg. 4.13 gelden dat $\frac{G_0\ \sigma}{G_0\ \sigma=0} = 1$, d.w.z. dat voor krimpende en niet-krimpende systemen de waarde van G_0 identiek is en dus onafhankelijk van σ.

In geval van σ=1 en λ=0 wordt voor vergelijking 4.14 de extreme waarde $H_0 = (1+v^0)^{\frac{\nu}{\nu+1}}$ bereikt.

Uit tabel 4.1. blijkt dat de H_0-waarden voor steeds grotere holten in het systeem tot 1 naderen.

tabel 4.1. H_0-waarden voor σ=1 , v^0=1 en v^*=0 berekend uit de numerieke oplossing van de differentiaalvergelijking en met de theoretische verg. 4.14 voor de verschillende geometrieën.

v^0 = 1	λ =0		λ =0.3		λ =0.7	
σ = 1	exp.	theor.	exp.	theor.	exp.	theor.
a=0 , ν=1	1.435	1.414	1.400	1.382	1.236	1.229
a=0.5, ν=1	1.414		1.396		1.235	
a=0 , ν=2	1.587	1.587	1.573	1.573	1.400	1.400
a=0.5, ν=2	1.587		1.573		1.400	

Voor de krimpfactor van het Regular Regime is (zie [8]) de volgende correlatie gevonden :

$$H_{rr} = 1 + \frac{Sh_{d,rr,a\to\infty}}{Sh_{d,rr,a}} \left[H_0(1-sE)^{\frac{\nu}{\nu+1}} - 1\right] \qquad (4.15)$$

waaruit blijkt dat H_{rr} afneemt bij toename van E;

als $\bar{v} = v^* = 0$ bereikt de krimpfactor H_{rr} de minimale waarde nl. $H_{rr} = 1$.

De numerieke output van KRIMP is niet altijd betrouwbaar gebleken en er kunnen maar een beperkt aantal overeenkomstige situaties met elkaar worden vergeleken.

Een correlatie over het gehele efficiency-bereik kan misschien worden gevonden m.b.v. $G_\sigma = G_0(1-\beta' E)$

Bij overstap van kleine naar grotere tijdstappen treedt een verstoring op in de berekende waarden zowel bij krimpende als niet-krimpende systemen echter bij verschillende waarden van de efficiency. Dit betekent dat de krimpfactor over een vrij groot traject van efficiency-waarden niet nauwkeurig kan worden berekend. Enkele grafieken van G versus E en H versus E zijn gegeven in bijlage O.

4.3. Berekening specifiek parameters

Berekening specifieke parameters in het programma NIETKRIMP

Voor de berekening van de verschillende parameters staan de gegevens uit de penetratieperiode (na aanroep van STARTSLAB en eventueel STARTNONSLAB) en de tijd volgende op de penetratie-periode (door herhaald aanroepen van PDE) ter beschikking.

De efficiency gedurende de penetratieperiode wordt berekend via interpolatie met verg. (3.1) en de flux met verg. (3.2) :

$$E_{pp} = c_1 \frac{\sqrt{\tau}}{\sqrt{\tau_{pp}}} + c_2 \frac{\tau}{\tau_{pp}} + c_3 \frac{\tau \sqrt{\tau}}{\tau_{pp}\sqrt{\tau_{pp}}} \qquad (3.1)$$

$$F = \frac{1}{X_{i,\sigma=0} * \tau_{pp}} \left\{ \frac{c_1}{2 \frac{\sqrt{\tau}}{\sqrt{\tau_{pp}}}} + c_2 + 1.5 * c_3 \frac{\sqrt{\tau}}{\sqrt{\tau_{pp}}} \right\} \qquad (3.2)$$

Voor de daarop volgende periode wordt de efficiency berekend via integratie van het concentratieprofiel volgens :

$$E_{rest} := 1 - \int_0^1 u \, dr = 1 - RET(i)$$

Uit bovenstaande vergelijkingen blijkt dat op $\tau = 0$ weliswaar $E=0$ en $F \to \infty$, maar dat het produkt $F*E$ wel een eindige waarde ($\neq 0$) heeft

zodat voor $\tau = 0$

$E = E_i' := 0$

F en Sh_d niet te bepalen zijn en

$$G_0 := \frac{(1-RETSS)^2}{2 * \tau_{pp}}$$

voor $0 < \tau \leqslant \tau_{pp}$

E(i) := 1 - RETSNS[i]
F m.b.v. de interpolatieformule (3.2)

$$G := \frac{F*E}{X_{i,\sigma=0}}$$

$$E_i' := 0$$

$$Sh_d := \frac{2*(a+1)*F}{\bar{m}^{a+1} - m_i^{a+1}} = \frac{(1-BV)*2*(a+1)*F}{((1-E)*(1-BV)+BV)^{a+1} - BV^{a+1}}$$

Indien $\tau > \tau_{pp}$ worden de parameters berekend uit de gegevens van het 2-dimensionele U-array, dat met profielen uit PDE-aanroepen is gevuld, en het 1-dimensionele array RET, dat met retentiewaarden is gevuld.
Voor de i^{de} rij :

E(i) := 1 - RET[i]

$$F := \frac{RET[i-1] - RET[i+1]}{X_{i,\sigma=0} * 2 * \Delta\tau} = \frac{\frac{E(i+1) - E(i-1)}{2}}{X_{i,\sigma=0} * \Delta\tau}$$

$$G := \frac{F * E}{X_{i,\sigma=0}}$$

$$E'_i := \frac{1 - U[i,0]}{1 - BV}$$

$$Sh_d := \frac{(1-BV) * 2 * (a+1) * F}{((1-E(i))(1-BV)+BV)^{a+1} - BV^{a+1}}$$

Berekening specifieke parameters in het programma KRIMP

Door de aanroepen van INITIALIZEFI, STARTNONSLABFI en PDEFI zijn er array's met profielen en met retenties voor de verschillende tijdsniveau's gevuld.
Analoog aan de handelwijze bij NIETKRIMP worden de parameters berekend waarbij alleen als extra berekening tijdens elke tijdstap de waarde de waarde van X_i moet worden berekend voordat F en G kunnen worden bepaald.

Berekening specifieke parameters voor het programma NKCF

Voor de berekening van de verschillende parameters is na herhaaldelijk aanroepen van CONSTANTFLUX de beschikking verkregen over een twee-dimensionaal array UCF (met concentratieprofielen)en een een-dimensionaal array RETCF (met retenties) op verschillende tijdsniveau's.

Voor $\tau = 0$ zijn $\frac{F_n}{E'_i}$ en Sh_d oneindig groot; E, E_c en $E'_i = 0$ en is er geen gebruik gemaakt van een $\sqrt{\tau}$-polynoom aanpassing voor E'_i om G_0 te berekenen.

E_c en E'_i kunnen pas bij de tweede tijdstap worden berekend via de gladstrijk-methode.

Vanaf de 2^{de} tijdstap geldt :

E(i) := 1 - RETCF[i] met i= tijdstap-indexnummer
en voor E_c en E'_i wegens het gladstrijken van de profielen over de tijdsniveau's

$$E_c(i) := 1 - \frac{1}{4}\,(UCF[i-1,0] + 2*UCF[i,0] + UCF[i+1,0])$$

$$E'_i(i) := 1 - \frac{1}{4}\,(UCF[i-1,N] + 2*UCF[i,N] + UCF[i+1,N])$$

$$G := \frac{\frac{F}{E'_i}\,\frac{E}{E'_i}}{X_i}$$

en

$$Sh_d := \frac{2*(a+1)*F_{ca0}}{(1-E)^{a+1} - (1-E'_i)^{a+1}}$$

Hierbij dient erop te worden gelet dat de 11^{de} rij niet wordt afgedrukt (deze is voor een goede overstap naar grotere tijdstappen nodig) en dat voor de 12^{de} rij E_c en E'_i worden gladgestreken tussen i=10, 12 en 13.

Berekening specifieke parameters voor het programma KCA

De berekeningen zijn identiek aan die voor NKCF, alleen moet bij elke tijdstap $X_{i,\sigma}$ worden aangepast om G te kunnen berekenen;

Voor het drogen met een constante flux door het grensvlak is de numerieke computer-output nog niet verwerkt in correlaties.

Door het overstappen van kleine tijdstap naar grotere tijdstap treedt er een discontinuïteit in de berekende waarden op, hetgeen problematisch is voor de gladstrijk-methode, die bovendien uitgaat van equidistante tijdsintervallen. Deze moeilijkheden kunnen wellicht worden vermeden door de tijdstap geleidelijk te laten toenemen en niet sprongsgewijs.

5. CONCLUSIES EN AANBEVELINGEN

- De procedures van Reniers vergen voor waarden van a $\geq$ 1.5 grote rekentijden. Het is derhalve wenselijk dat de efficiëntie van de numerieke berekeningen wordt verbeterd.
Voor niet-krimpende systemen bleek namelijk dat voor a = 1.5 een rekentijd van ca. 1600 processing units was benodigd om tot een eind-efficiency van 0.96 door te kunnen rekenen.
Voor krimpende systemen lag de rekentijd in dezelfde grootte-orde.
Voor nietkrimpende systemen met een constante grensvlakwateractiviteit bedroeg de rekentijd voor a = 1 ca. 500 en voor a = 0.5 zelfs ca. 900 processing units.

- Voor systemen met constante grensvlakwaterconcentratie wordt $Sh_{d,rr}$ constant (indien $D_r = m^a$) als $m_i = 0$ en gaat $Sh_{d,rr}$ als funktie van de efficiency E door een minimum als $m_i \neq 0$.
Voor systemen met constante grensvlakwateractivteit wordt $Sh_{d,rr}$ alleen constant als de diffusiecoëfficiënt constant is (a=0) en gaat $Sh_{d,rr}$ bij variabele diffusiecoëfficiënt als funktie van de efficiency door een minimum.

- Voor <u>alle</u> berekende droogsituaties is de G-parameter gedurende de penetratieperiode bij goede benadering lineair afhankelijk van E/E_i'.

- Voor krimpende en niet-krimpende systemen is de waarde van G_0 identiek, d.w.z. onafhankelijk van σ.

- Voor krimpende vlakke lagen is de oplossing van de diffusievergelijking identiek aan die voor niet-krimpende vlakke lagen, zodat de krimpfactor H de waarde 1 heeft en onafhankelijk is van de efficiency E.

-Het is zinvol de krimpfactor zowel te definiëren op basis van de verhouding $\frac{F_\sigma}{F_{\sigma=0}}$ als op basis van de verhouding $\frac{G_\sigma}{G_{\sigma=0}}$ waarbij deze laatste definitie als voordeel heeft dat situaties met gelijke waarde voor de krimpcoëfficiënt s wellicht hetzelfde verband voor $\frac{G_\sigma}{G_{\sigma=0}}$ vertonen. Zo zal voor $\lambda \geq 0.7$ en/of $\sigma \ll 1$ hetzelfde resultaat worden gevonden omdat dan de verhouding van de G-parameters tot 1 nadert.

- De berekende $Sh_{d,rr}$-waarden van holle en massieve niet-krimpende systemen met $m_i=0$ vertonen een zeer goede overeenkomst met de waarden van van Schaik [14] (afwijkingen < 0.1%).

- Voor constante grensvlakwateractiviteit (niet-krimp, constante diffusiecoëfficiënt ($a=0, \sigma=0$)) komen de numeriek berekende efficiency's zeer goed overeen met de analytische oplossingen van Bosch [9].

- Het programma KRIMP heeft bij waarden voor $v^0>1$ een zeer kleine tijdstap nodig om redelijk betrouwbare resultaten te krijgen; dit valt te verifiëren aan de hand van de waarde van G_0 welke gelijk moet zijn aan de waarde voor de nietkrimpende situatie.

- Bij overstap van een kleine tijdstap in het begin van het proces naar een grotere tijdstap voor de rest van het droogproces ontstaat er in de berekeningen van de specifieke parameters een discontinuïteit . Het is aan te bevelen dit probleem te ondervangen door met een continue toenemende tijdsstap tot een maximale waarde deze beginfase van het droogproces door te rekenen.

- Bij het programma NKCF en KCA moet er rekening worden gehouden met het optreden van oscillerende concentratieprofielen (zowel in plaats als in tijd); deze oscillaties kunnen echter door gladstrijk-methoden worden geminimaliseerd.

- Het zou raadzaam zijn om voor de situaties met constante grensvlakwateractiviteit gedurende de penetratieperiode gebruik te maken van een $\sqrt{\tau}$-polynoom-aanpassing voor E_i' waardoor het mogelijk wordt de waarde voor G_0 te berekenen.

- Bij vergelijking van de berekende krimpfactoren (bijlage O) met de analytische oplossing volgens vergelijking 4.10 blijkt dat de H_0-waarden redelijk overeen komen met de berekende waarden volgens de analytische oplossing.

LITERATUUR

1. Schoeber, W. J. A. H. , Regular Regimes in sorption processes, Ph. D. thesis, Technical University Eindhoven, 1976.
2. Liou, J. K. , An approximate method for nonlinear diffusion applied to enzyme inactivation during drying, Ph. D. thesis, Agricultural University Wageningen, 1982.
3. Liou, J. K. and Bruin, S. , An approximate method for the non-linear diffusion problem with a power relation between diffusion coefficient and concentration, Int. J. Heat mass Transfer, vol. 25, pp. 1209-1229, 1982.
4. Thijssen, H. A. C. and Coumans, W. J. , Short-cut calculation for non-isothermal drying of shrinking and non-shrinking particles and of hollow spheres containing an expanding central gas core, in DRYING '85, eds. R. Toei and A. S. Mujumdar, pp. 11-20, 1985.
5. Gregory, D. J. , The extended Algol Primer, Gregory publishing Co. , Sonoma, California, 1983.
6. Lijn, J. v. d. , Simulation of heat and mass transfer in spray-drying, Ph. D. thesis, Agricultural University Wageningen, 1976.
7. Thijssen, H. A. C. and Coumans, W. J. , Concise procedure for the calculation of isothermal drying rates of non-shrinking solid and hollow particles, Proc. World Congress III of Chemical Engineering, Tokyo, 1986.
8. Coumans, W. J. and Thijssen, H. A. C. , A simplified method for the isothermal drying of solid and hollow systems with any degree of shrinkage, Paper presented at 5th International Drying Symposium, Cambridge, Mass. , USA, Aug. 13-16, 1986.
9. Bosch, M. L. , Diffusie in niet krimpende systemen met een constante diffusiecoëfficiënt (CAP), praktikumverslag TF, THE, 1985.
10. Coumans, W. J. , Drogen, collegedictaat, THE, 1983-1986.
11. Reniers, P. , Numerical method for solving the diffusion equation, afstudeerverslag, THE, 1985.
12. Veltkamp, G. W. , Geurts, A. J. , collegedictaat Numerieke Methoden, THE, dictaatnr. : 2. 211.
13. Rekencentrum informatie.
14. Schaik-van Hoek van, A. L. , Berekening van Shd-waarden voor het Regular Regime van holle en massieve niet-krimpende systemen, praktikumverslag TF, THE, 1985.

BIJLAGE O

BEPALING KRIMPFACTOREN

BIJLAGE 0

Bepaling krimpfactor H voor

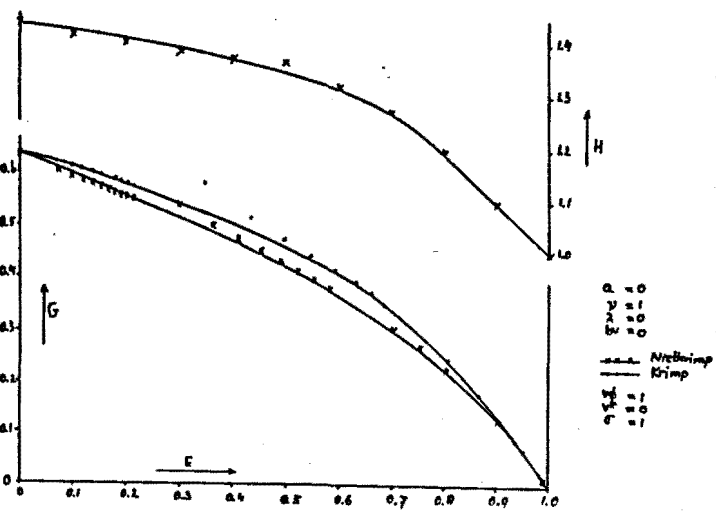
a : 0	bv : 0
v : 1	v0 : 1
λ : 0	vs : 0
d : 1	

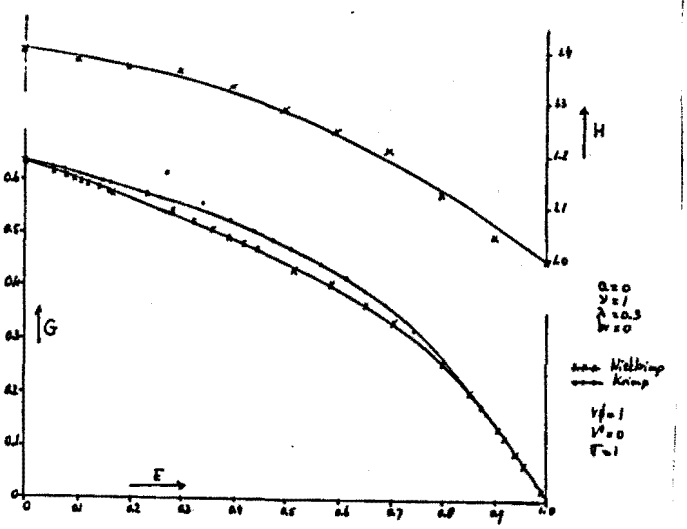
E	G_d	$G_{d:0}$	$\frac{G_d}{G_{d:0}}$	H
0.0	0.646	0.636	1.015	1.435
0.1	0.613	0.598	1.025	1.413
0.2	0.582	0.558	1.043	1.399
0.3	0.550	0.518	1.062	1.384
0.4	0.515	0.476	1.082	1.369
0.5	0.473	0.425	1.113	1.363
0.6	0.410	0.368	1.114	1.318
0.7	0.340	0.305	1.115	1.271
0.8	0.253	0.232	1.091	1.195
0.9	0.136	0.130	1.046	1.097
1.0	0.001	0.001	1.000	1.000

Bepaling krimpfactor H voor

a : 0	bv : 0
v : 1	v0 : 1
λ : 0.3	vs : 0
d : 1	

E	G_d	$G_{d:0}$	$\frac{G_d}{G_{d:0}}$	H
0.0	0.645	0.636	1.013	1.400
0.1	0.615	0.601	1.023	1.380
0.2	0.587	0.564	1.041	1.368
0.3	0.560	0.527	1.063	1.360
0.4	0.524	0.490	1.069	1.330
0.5	0.480	0.450	1.067	1.287
0.6	0.426	0.399	1.068	1.247
0.7	0.360	0.336	1.071	1.209
0.8	0.267	0.258	1.035	1.125
0.9	0.146	0.146	1.000	1.045
1.0	0.005	0.005	1.000	1.000

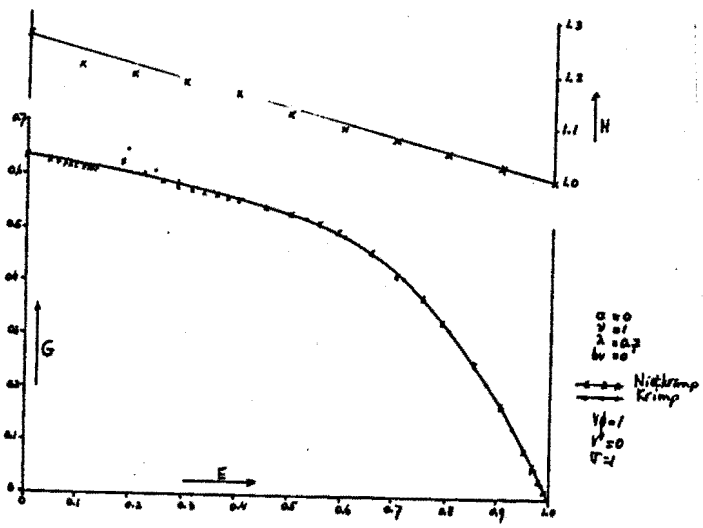




Bepaling krimpfactor H voor

a : 0	bv : 0
ν : 1	ν0 : 1
λ : 0.7	νs : 0
d : 1	

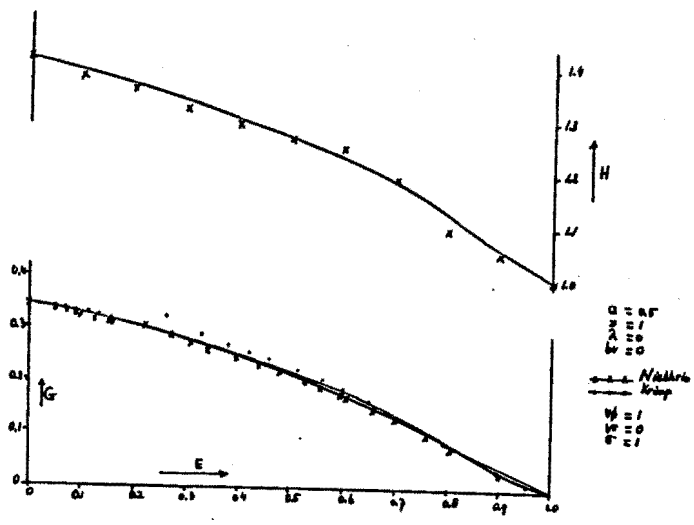
E	G_d	$G_{d:0}$	$\frac{G_d}{G_{d:0}}$	H
0.0	0.640	0.636	1.006	1.236
0.1	0.617	0.617	1.000	1.208
0.2	0.598	0.596	1.003	1.190
0.3	0.580	0.575	1.009	1.175
0.4	0.560	0.554	1.011	1.155
0.5	0.534	0.534	1.000	1.120
0.6	0.495	0.495	1.000	1.097
0.7	0.432	0.432	1.000	1.074
0.8	0.328	0.328	1.000	1.050
0.9	0.185	0.185	1.000	1.025
1.0				1.000



Bepaling krimpfactor H voor

a : 0.5	bv : 0
ν : 1	ν0 : 1
λ : 0	νs : 0
d : 1	

E	G_d	$G_{d:0}$	$\frac{G_d}{G_{d:0}}$	H
0.0	0.348	0.348	1.000	1.414
0.1	0.330	0.330	1.000	1.378
0.2	0.308	0.305	1.010	1.355
0.3	0.283	0.280	1.011	1.318
0.4	0.255	0.250	1.020	1.290
0.5	0.224	0.217	1.032	1.264
0.6	0.187	0.177	1.056	1.249
0.7	0.140	0.134	1.045	1.191
0.8	0.085	0.085	1.000	1.095
0.9	0.030	0.030	1.000	1.049
1.0				1.000



Bepaling krimpfactor H voor

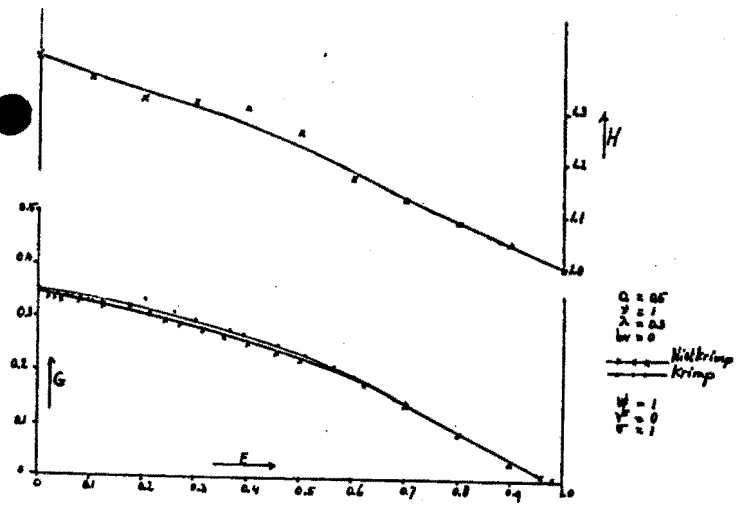
a : 0.5	bv : 0
v : 1	v0 : 1
λ : 0.3	vs : 0
d : 1	

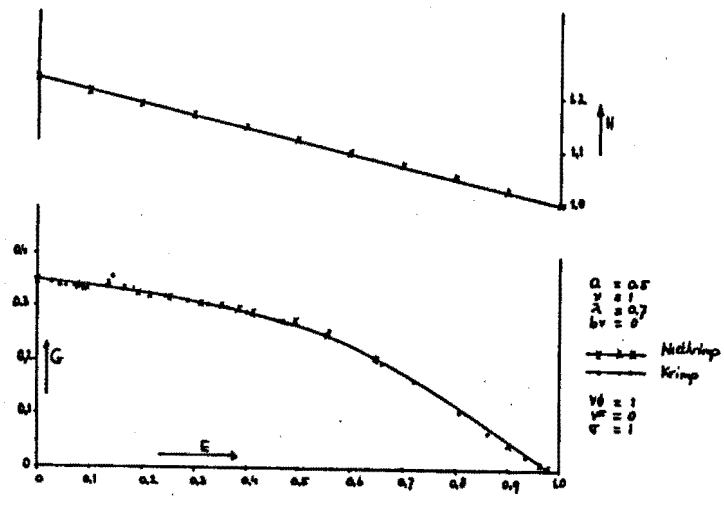
E	G_d	$G_{d:0}$	$\frac{G_d}{G_{d:0}}$	H
0.0	0.352	0.349	1.010	1.396
0.1	0.330	0.329	1.003	1.353
0.2	0.310	0.310	1.000	1.315
0.3	0.290	0.283	1.025	1.311
0.4	0.265	0.253	1.047	1.302
0.5	0.233	0.224	1.040	1.254
0.6	0.190	0.190	1.000	1.168
0.7	0.147	0.147	1.000	1.128
0.8	0.093	0.093	1.000	1.087
0.9	0.040	0.040	1.000	1.045
1.0				1.000

Bepaling krimpfactor H voor

a : 0.5	bv : 0
v : 1	v0 : 1
λ : 0.7	vs : 0
d : 1	

E	G_d	$G_{d:0}$	$\frac{G_d}{G_{d:0}}$	H
0.0	0.350	0.349	1.000	1.235
0.1	0.340	0.340	1.000	1.208
0.2	0.327	0.327	1.000	1.187
0.3	0.312	0.312	1.000	1.165
0.4	0.294	0.294	1.000	1.143
0.5	0.268	0.268	1.000	1.120
0.6	0.230	0.230	1.000	1.097
0.7	0.175	0.175	1.000	1.074
0.8	0.110	0.110	1.000	1.050
0.9	0.057	0.057	1.000	1.025
1.0				1.000





Bepaling krimpfactor H voor

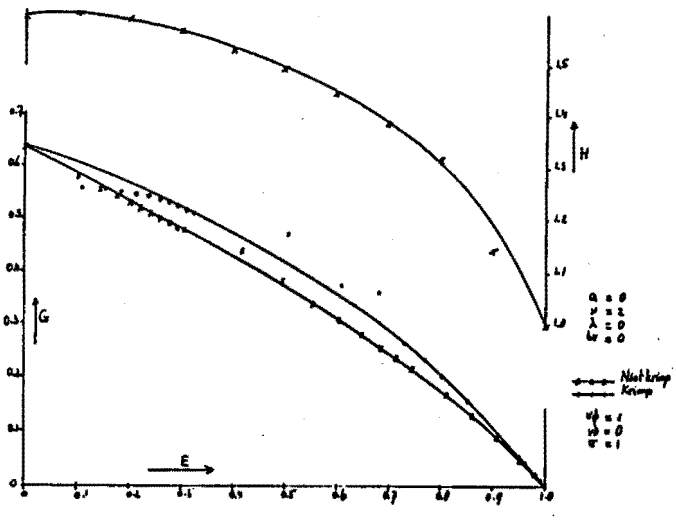
a : 0	bv : 0
v : 2	v0 : 1
λ : 0	vs : 0
σ : 1	

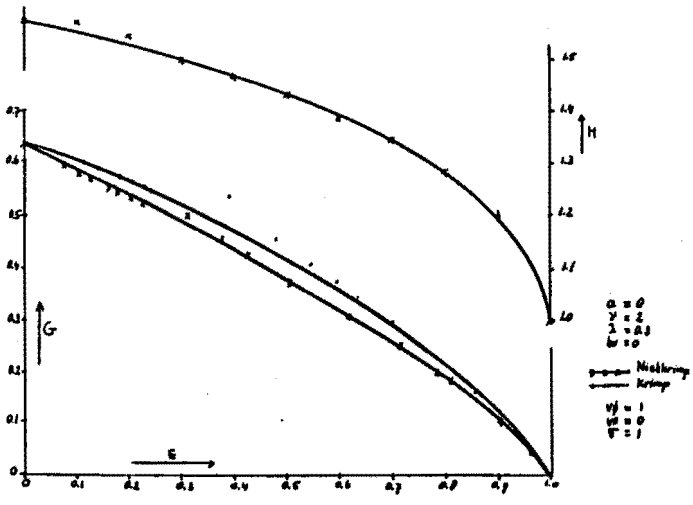
E	G_σ	$G_{\sigma:0}$	$\frac{G_\sigma}{G_{\sigma:0}}$	H
0.0	0.636	0.636	1.000	1.587
0.1	0.605	0.582	1.040	1.595
0.2	0.565	0.527	1.072	1.586
0.3	0.525	0.478	1.098	1.564
0.4	0.475	0.426	1.115	1.525
0.5	0.423	0.372	1.137	1.490
0.6	0.363	0.315	1.152	1.442
0.7	0.293	0.252	1.163	1.385
0.8	0.210	0.180	1.167	1.318
0.9	0.106	0.099	1.071	1.141
1.0	0.001	0.001	1.000	1.000

Bepaling krimpfactor H voor

a : 0	bv : 0
v : 2	v0 : 1
λ : 0.3	vs : 0
σ : 1	

E	G_σ	$G_{\sigma:0}$	$\frac{G_\sigma}{G_{\sigma:0}}$	H
0.0	0.635	0.635	1.000	1.573
0.1	0.605	0.587	1.031	1.568
0.2	0.567	0.540	1.050	1.541
0.3	0.521	0.492	1.059	1.497
0.4	0.472	0.438	1.078	1.465
0.5	0.417	0.380	1.097	1.429
0.6	0.358	0.322	1.112	1.384
0.7	0.295	0.260	1.135	1.346
0.8	0.220	0.193	1.140	1.283
0.9	0.134	0.112	1.134	1.206
1.0				1.000





Bepaling krimpfactor H voor

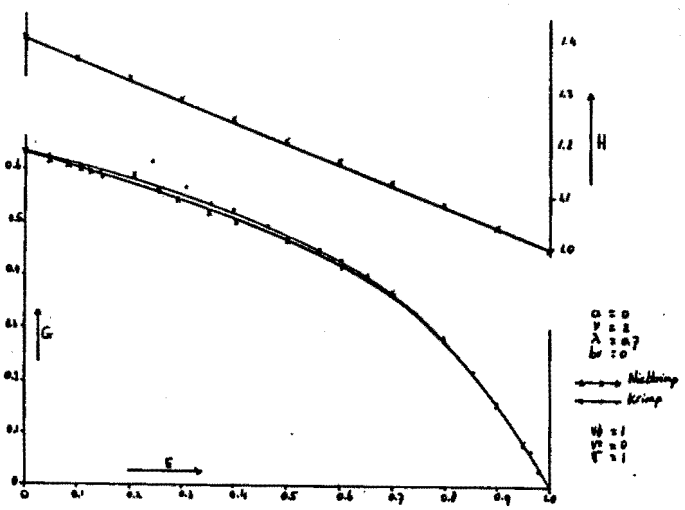
a = 0	bv = 0
v = 2	v0 = 1
λ = 0.7	vs = 0
d = 1	

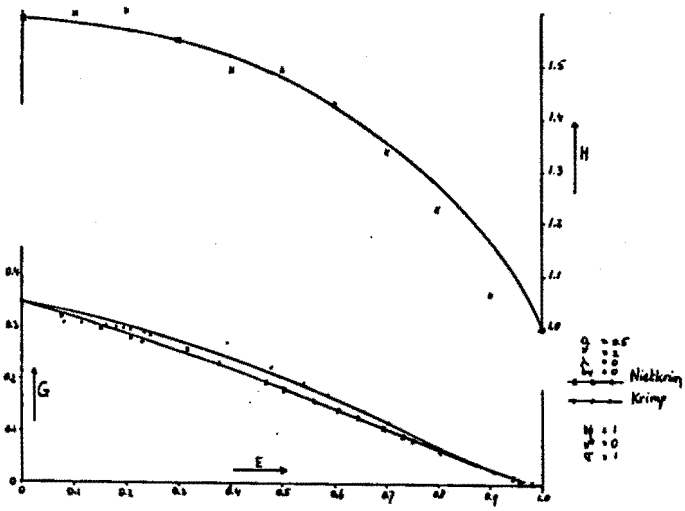
E	G_d	$G_{d=0}$	$\frac{G_d}{G_{d=0}}$	H
0.0	0.635	0.635	1.000	1.400
0.1	0.613	0.613	1.000	1.363
0.2	0.584	0.584	1.000	1.325
0.3	0.553	0.553	1.000	1.287
0.4	0.517	0.517	1.000	1.248
0.5	0.476	0.476	1.000	1.208
0.6	0.430	0.430	1.000	1.168
0.7	0.367	0.367	1.000	1.127
0.8	0.275	0.275	1.000	1.086
0.9	0.155	0.155	1.000	1.043
1.0				1.000

Bepaling krimpfactor H voor

a = 0.5	bv = 0
v = 2	v0 = 1
λ = 0	vs = 0
d = 1	

E	G_d	$G_{d=0}$	$\frac{G_d}{G_{d=0}}$	H
0.0	0.349	0.349	1.000	1.587
0.1	0.328	0.315	1.041	1.597
0.2	0.305	0.282	1.082	1.601
0.3	0.274	0.252	1.087	1.548
0.4	0.240	0.220	1.091	1.492
0.5	0.205	0.180	1.139	1.493
0.6	0.163	0.143	1.140	1.427
0.7	0.118	0.105	1.124	1.339
0.8	0.073	0.067	1.090	1.231
0.9	0.028	0.028	1.000	1.066
1.0				1.000





Bepaling krimpfactor H voor

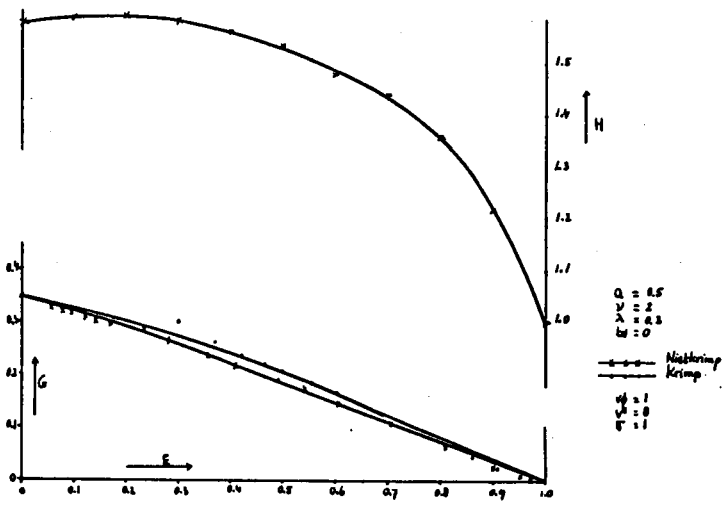
a : 0.5	bv : 0
v : 2	v0 : 1
λ : 0.3	vs : 0
d : 1	

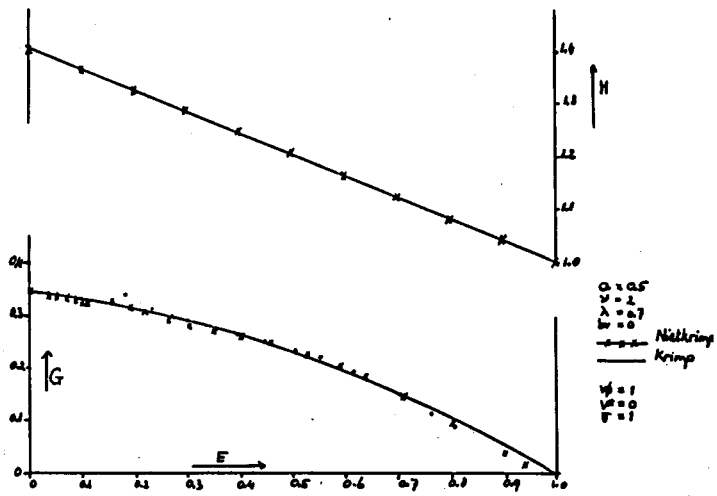
E	G_d	$G_{d:0}$	$\frac{G_d}{G_{d:0}}$	H
0.0	0.348	0.348	1.000	1.573
0.1	0.326	0.313	1.042	1.585
0.2	0.301	0.278	1.083	1.590
0.3	0.273	0.244	1.119	1.582
0.4	0.240	0.209	1.148	1.560
0.5	0.205	0.174	1.178	1.534
0.6	0.165	0.139	1.187	1.478
0.7	0.125	0.103	1.214	1.440
0.8	0.082	0.068	1.206	1.358
0.9	0.040	0.035	1.143	1.216
1.0				1.000

Bepaling krimpfactor H voor

a : 0.5	bv : 0
v : 2	v0 : 1
λ : 0.7	vs : 0
d : 1	

E	G_d	$G_{d:0}$	$\frac{G_d}{G_{d:0}}$	H
0.0	0.345	0.345	1.000	1.400
0.1	0.333	0.333	1.000	1.363
0.2	0.315	0.315	1.000	1.325
0.3	0.290	0.290	1.000	1.287
0.4	0.263	0.263	1.000	1.248
0.5	0.231	0.231	1.000	1.208
0.6	0.193	0.193	1.000	1.168
0.7	0.150	0.150	1.000	1.127
0.8	0.105	0.105	1.000	1.086
0.9	0.057	0.057	1.000	1.043
1.0				1.000





BIJLAGE 1

Overzicht van G0 en Shd -waarden voor NIETKRIMP

BIJLAGE 1

Niet-krimpende systemen met constante grensvlakconcentratie

Overzicht van G_0- en Sh_d- waarden :

	G_0 als funktie van a en m_i voor alle waarden van ν en λ			
m_i	a = 0	a = 0.5	a = 1.0	a = 1.5
0	0.63639	0.34868	0.22063	0.15215
0.025		0.36232	0.23153	0.16024
0.050		0.37456	0.24241	0.16862
0.075		0.38583	0.25325	0.17729
0.100		0.39638	0.26404	0.18625

	Sh_d waarden voor niet-krimpende en krimpende vlakke lagen voor $0 \leq \lambda \leq 1$			
m_i	a = 0	a = 0.5	a = 1.0	a = 1.5
0	4.936	5.442	5.772	6.004
0.025		5.175	5.694	5.991
0.050		5.087	5.552	5.933
0.075		5.046	5.411	5.841
0.100		5.019	5.291	5.732

Voor $m_i = 0$ zijn de waarden voor Sh_d constant in het Regular Regime.

Sh_d-waarden voor ν = 1

		uit [14]		bij		bij		bij		bij	
		m_i		m_i	E_{eind}	m_i	E_{eind}	m_i	E_{eind}	m_i	E_{eind}
λ	a	0		0.025		0.005		0.075		0.1	
0	0	5.788	5.783								
0.3		4.945	4.943								
0.7		4.831	4.830								
0	0.5	6.839	6.835	6.316	0.980	6.139	0.980	6.045	0.980	5.991	0.980
0.3		5.726	5.726	5.329	0.980	5.194	0.980	5.126	0.980	5.089	0.980
0.7		5.149	5.417	5.114	0.980	5.013	0.979	4.959	0.980	4.931	0.980
0	1	7.526	7.525	6.996	0.969	6.637	0.970	6.447	0.970	6.320	0.970
0.3		6.240	6.240	6.007	0.944	5.679	0.958	5.422	0.970	5.328	0.970
0.7		5.804	5.804	5.690	0.917	5.498	0.933	5.321	0.946	5.180	0.958
0	1.5	8.019x	0.555	8.020x	0.565	8.001x	0.569	7.977x	0.557	6.793x	0.955
0.3		6.614x	0.495	6.610x	0.526	6.597x	0.536	6.583x	0.517	5.948x	0.920
0.7		6.079x	0.466	6.080x	0.474	6.073x	0.477	6.060x	0.475	5.706x	0.884

Sh_d-waarden voor ν = 2

		uit [14]		bij		bij		bij		bij	
		m_i		m_i	E_{eind}	m_i	E_{eind}	m_i	E_{eind}	m_i	E_{eind}
λ	a	0		0.025		0.005		0.075		0.1	
0	0	6.596	6.580								
0.3		5.131	5.127								
0.7		4.753	4.752								
1		4.936	4.935								
0	0.5	8.213	8.208	7.442	0.980	7.166	0.980	7.016	0.981	6.925	0.981
0.3		6.254	6.251	5.705	0.980	5.517	0.980	5.415	0.980	5.357	0.980
0.7		5.430	5.418	5.080	0.980	4.961	0.980	4.904	0.980	4.873	0.980
1		5.442	5.440	5.175	0.980	5.067	0.980	5.046	0.980	5.019	0.980
0	1	9.271	9.270	8.461	0.970	7.969	0.970	7.669	0.970	7.469	0.970
0.3		6.994	6.994	6.437	0.969	6.061	0.970	5.861	0.970	5.725	0.970
0.7		5.877	5.877	5.714	0.931	5.461	0.946	5.246	0.959	5.085	0.969
1		5.772	5.772	5.694	0.900	5.552	0.917	5.411	0.931	5.291	0.944
0	1.5	10.046x	0.597	10.044x	0.613	10.014x	0.614	9.942x	0.629	9.867x	0.629
0.3		7.533x	0.574	7.547x	0.559	7.521x	0.577	7.507x	0.560	7.459x	0.568
0.7		6.198x	0.496	6.202x	0.486	6.193x	0.498	6.180x	0.487	6.155x	0.495
1		6.004		5.991	0.806	5.933	0.826	5.841	0.844	5.732	0.862

BIJLAGE 2

Overzicht van G0 en Shd -waarden voor KRIMP en NKCF

BIJLAGE 2

Krimpende systemen met constante grensvlakconcentratie

Overzicht van G_0- en Sh_d- waarden :

	$G_{0,d:i}$ als funktie van a voor $\nu = 1$, $\lambda = 0$ en $m_i = 0$			
$\bar{v}^0$	a = 0	a = 0.5	a = 1.0	a = 1.5
1	0.64560	0.35245	0.22293	0.15387
2	0.63405	0.34718	0.21991	0.15195
4	0.57093	0.31788	0.20297	0.14092
$G_{0,d:0}$	0.63639	0.34868	0.22063	0.15215

	G_0 als funktie van a voor $\nu = 1$, $\lambda = 0.3$ en $m_i = 0$			
$\bar{v}^0$	a = 0	a = 0.5	a = 1.0	a = 1.5
1	0.64457	0.35211	0.22268	0.15364
2	0.64678	0.35297	0.22318	0.15400
4	0.63959	0.34963	0.22129	0.15262
$G_{0,d:0}$	0.63639	0.34868	0.22063	0.15215

G_0 als funktie van a
voor $\nu = 1$, $\lambda = 0.7$ en $m_i = 0$

ν_-^0	a = 0	a = 0.5	a = 1.0	a = 1.5
1	0.64028	0.35044	0.22164	0.15280
2	0.64195	0.35106	0.22201	0.15310
4	0.64447	0.35203	0.22260	0.15353
$G_{0,d=0}$	0.63639	0.34868	0.22063	0.15215

G_0 als funktie van a
voor $\nu = 2$, $\lambda = 0$ en $m_i = 0$

ν_-^0	a = 0	a = 0.5	a = 1.0	a = 1.5
1	0.54730	0.30670	0.19647	0.13657
2	0.40978	0.23782	0.15511	0.10902
4	##	##	##	##
$G_{0,d=0}$	0.63639	0.34868	0.22063	0.15215

: totaal onbetrouwbaar

G_0 als funktie van a
voor $\nu = 2$, $\lambda = 0.3$ en $m_i = 0$

ν^0	a = 0	a = 0.5	a = 1.0	a = 1.5
1	0.63985	0.34987	0.22148	0.15285
2	0.57850	0.32142	0.20502	0.14214
4	##	##	##	##
$G_{0,d=0}$	0.63639	0.34868	0.22063	0.15215

= totaal onbetrouwbaar

G_0 als funktie van a
voor $\nu = 2$, $\lambda = 0.7$ en $m_i = 0$

ν^0	a = 0	a = 0.5	a = 1.0	a = 1.5
1	0.64457	0.35218	0.22269	0.15361
2	0.64817	0.35364	0.22355	0.15422
4	0.64516	0.35216	0.22271	0.15371
$G_{0,d=0}$	0.63639	0.34868	0.22063	0.15215

Sh_d waarden voor krimpende cylinders

voor m_i : 0 , λ :0 en σ : 1

v^0	a : 0	E(eind)	a : 0.5	E(eind)	a : 1.0	E(eind)	a : 1.5	E(eind)
1	5.848	0.991	6.953	0.980	7.769	0.956	8.554	0.898
2	5.902	0.991	7.281	0.980	7.964	0.956	8.978	0.907
4	6.019	0.991	7.281	0.980	8.279	0.965	9.651	0.917
$Sh_{d,\sigma=0}$	5.788		6.839		7.526		8.019	0.555

Sh_d waarden voor krimpende cylinders

voor m_i : 0 , λ :0.3 en σ : 1

v^0	a : 0	E(eind)	a : 0.5	E(eind)	a : 1.0	E(eind)	a : 1.5	E(eind)
1	4.980	0.990	5.796	0.980	6.469	0.934	7.065	0.865
2	5.017	0.990	5.867	0.980	6.650	0.940	7.424	0.877
4	5.081	0.991	6.002	0.980	6.938	0.948	7.990	0.891
$Sh_{d,\sigma=0}$	4.945		5.726		6.240		6.614	0.495

Sh_d waarden voor krimpende cylinders

voor m_i : 0 , λ :0.7 en σ : 1

v^0	a : 0	E(eind)	a : 0.5	E(eind)	a : 1.0	E(eind)	a : 1.5	E(eind)
1	4.843	0.990	5.446	0.978	5.921	0.905	6.297	0.823
2	4.854	0.990	5.469	0.979	6.021	0.912	6.485	0.834
4	4.877	0.990	5.516	0.780	6.188	0.920	6.799	0.849
$Sh_{d,\sigma=0}$	4.831		5.419		5.804		6.079	0.466

Sh_d waarden voor krimpende bollen
voor m_i : 0 , λ :0 en d : 1

v^0	a : 0	E(eind)	a : 0.5	E(eind)	a : 1.0	E(eind)	a : 1.5	E(eind)
1	6.709	0.992	8.318	0.990	9.575	0.970	10.678	0.932
2	6.826	0.991	8.423	0.990	9.671	0.970	11.204	0.938
4	**		**		**		**	
$Sh_{d, d:0}$	6.596		8.213		9.217		10.046	0.597

** : totaal onbetrouwbaar

Sh_d waarden voor krimpende bollen
voor m_i : 0 , λ :0.3 en d : 1

v^0	a : 0	E(eind)	a : 0.5	E(eind)	a : 1.0	E(eind)	a : 1.5	E(eind)
1	5.203	0.991	6.391	0.980	7.277	0.958	8.153	0.902
2	5.273	0.991	6.525	0.980	7.497	0.962	8.640	0.912
4	**		**		**		**	
$Sh_{d, d:0}$	5.131		6.254		6.994		7.533	0.574

** : totaal onbetrouwbaar

Sh_d waarden voor krimpende bollen
voor m_i : 0 , λ :0.7 en d : 1

v^0	a : 0	E(eind)	a : 0.5	E(eind)	a : 1.0	E(eind)	a : 1.5	E(eind)
1	4.775	0.991	5.478	0.980	6.065	0.922	6.563	0.848
2	4.799	0.990	5.524	0.980	6.214	0.929	6.859	0.861
4	4.843	0.990	5.620	0.980	6.456	0.939	7.336	0.879
$Sh_{d, d:0}$	4.753		5.430		5.877		6.198	0.496

** : totaal onbetrouwbaar

APPENDIX A

listing van het programma
NIETKRIMP

NIETKRIMP

```
BEGIN
 $INCLUDE "(TGTFDR)DROOGLIBRARY ON USER4."
 FILE IN(KIND=DISK,FILETYPE=7),
      TAPE(KIND=DISK,PROTECTION=SAVE,MAXRECSIZE=22,UNITS=WORDS),
      PROFIEL(KIND=DISK,PROTECTION=SAVE,MAXRECSIZE=22,UNITS=WORDS),
      PRINT(KIND=PRINTER);

 STRING SITUATIE;
 REAL LAMBDA,A,BV,STOPE,TAU,ACC,DTAU,RETSS,XI,XH,SIGMA,
      E1,E2,E,F,G,SHD;
 INTEGER NU,ITMAX,INTVALS,STEPS,N,DIM,INDEX,I,FACTOR,SIT;
 REAL ARRAY COEF[1:3];
```

```
% heading printerfile-tabellen constante grensvlakconcentratie
PROCEDURE KOPJETABEL(F, A, BV, NU, N, LAMBDA, TAU, DTAU,
                     SIGMA, VSTER, VREF, VO, COEF,
                     INTVALS, ITMAX, ACC, RET, XI);
VALUE A, BV, NU, N, LAMBDA, TAU, DTAU,
      SIGMA, VSTER, VREF, VO, INTVALS, ITMAX, ACC, RET;
REAL LAMBDA, A, BV, TAU, DTAU, SIGMA, VSTER, VREF, VO, ACC, RET, XI;
INTEGER NU, ITMAX, INTVALS, N, WF;
REAL ARRAY COEF[*];
FILE F;
BEGIN
 STRING S, SA, SBV, SNU, SLAMBDA, SN, STAU, SINTVALS, SDTAU;
 INTEGER NUP;
 SA:= STRING(A, *) CAT "          ";
 IF FIRST(SA)="." THEN SA:="0" CAT SA;
 SBV:= STRING(BV, *) CAT "          ";
 IF FIRST(SBV)="." THEN SBV:="0" CAT SBV;
 SNU:= STRING(NU, *) CAT "          ";
 SN:= STRING(N, *) CAT "          ";
 SLAMBDA:= STRING(LAMBDA, *) CAT "          ";
 IF FIRST(SLAMBDA)="." THEN SLAMBDA:="0" CAT SLAMBDA;
 SSIGMA:= STRING(SIGMA, *) CAT "          ";
 IF FIRST(SSIGMA)="." THEN SSIGMA:="0" CAT SSIGMA;
 SVSTER:= STRING(VSTER, *) CAT "          ";
 IF FIRST(SVSTER)="." THEN SVSTER:="0" CAT SVSTER;
 SVREF:= STRING(VREF, *) CAT "          ";
 IF FIRST(SVREF)="." THEN SVREF:="0" CAT SVREF;
 SVO:= STRING(VO, *) CAT "          ";
 IF FIRST(SVO)="." THEN SVO:="0" CAT SVO;
 STAU:= STRING(TAU, *) CAT "          ";
 IF FIRST(STAU)="." THEN STAU:="0" CAT STAU;
 SDTAU:= STRING(DTAU, *) CAT "          ";
 IF FIRST(SDTAU)="." THEN SDTAU:="0" CAT SDTAU;
 SINTVALS:= STRING(INTVALS, *) CAT "          ";
 S:=REPEAT("-", 65);
 WRITE(F[SKIP(1)]);
 WRITE(F, <X7, A65>, S);
 WRITE(F, <X7, "1", X31, "1", X31, "1">);
 IF SIGMA NEQ 0 THEN
 BEGIN
  WRITE(F, <X7, "1", X2, "Physical properties", X10, "1", X2,
   "SHRINKAGE parameters", X9, "1", /, X7, "1", X31, "1", X31, "1",
   /, X7, "1", X2, "A        =", X5, A10, X5, "1", X7, "SIGMA =  ", X2, A10, X4, "1",
   /, X7, "1", X2, "BV       =", X5, A10, X5, "1", X7, "V#    =  ", X2, A10, X4, "1",
   /, X7, "1", X2, "NU       =", X5, A10, X5, "1", X7, "V*    =  ", X2, A10, X4, "1",
   /, X7, "1", X2, "LAMBDA   =", X5, A10, X5, "1", X7, "VO    =  ", X2, A10, X4, "1",
   /, X7, "1", X31, "1", X31, "1">, TAKE(SA, 10), TAKE(SSIGMA, 10),
   TAKE(SBV, 10), TAKE(SVREF, 10), TAKE(SNU, 10), TAKE(SVSTER, 10),
   TAKE(SLAMBDA, 10), TAKE(SVO, 10));
  WRITE(F, <X7, A65>, S);
  WRITE(F, <X7, "1", X31, "1", X31, "1">);
  WRITE(F, <X7, "1", X2, "Discretization", X15, "1", X2, "Accuracy",
   X21, "1", /, X7, "1", X31, "1", X31, "1", /, X7, "1", X2,
   "N        =", X5, A10, X5, "1", X2, "ACC    =", E8.1, X14, "1", /, X7, "1", X2,
   "TAU pp   =", X5, A10, X5, "1", X2, "ITMAX(INITIALIZEFI)    =", I4, X2, "1", /,
   X7, "1", X2, "INTVALS =",
   X5, A10, X5, "1", X2, "ITMAX                    =", I4, X2, "1", /, X7,
```

```
  TAKE(SN,10),ACC,TAKE(STAU,10),5*ITMAX,TAKE(SINTVALS,10),ITMAX,
  TAKE(SDTAU,10));
  WRITE(F,<X7,A65>,S);
  %
 WRITE(F,</,X8,"Penetration">);
 WRITE(F,</,X9,"TAU",X7,"E",X12,"F",X11,"G",X8,"E(r=Rc)",
                X8,"Shd",/>);
 NUP:=NU+1;
 XI:=XH*(LAMBDA**NUP+(1-LAMBDA**NUP)*(1+SIGMA*V0))**(NU/NUP);
 WRITE(F,<X7,"0.0000",X7,"0",X10,"--",X3,F12.5,X7,"0",
                X12,"--">,(COEF[1]/XI)**2/(2*TAU));
END
ELSE
BEGIN
 WRITE(F,<X7,"1",X2,"Physical properties",X10,"1",X2,
  "NON-SHRINKING SYSTEMS",X8,"1",/,X7,"1",X31,"1",X31,"1",
  /,X7,"1",X2,"A        =",X5,A10,X5,"1",X31,"1",
  /,X7,"1",X2,"BV       =",X5,A10,X5,"1",X31,"1",
  /,X7,"1",X2,"NU       =",X5,A10,X5,"1",X31,"1",
  /,X7,"1",X2,"LAMBDA   =",X5,A10,X5,"1",X31,"1",
  /,X7,"1",X31,"1",X31,"1">,TAKE(SA,10),TAKE(SBV,10),
  TAKE(SNU,10),TAKE(SLAMBDA,10));
 WRITE(F,<X7,A65>,S);
 WRITE(F,<X7,"1",X31,"1",X31,"1">);
 WRITE(F,<X7,"1",X2,"Discretization",X15,"1",X2,"Accuracy",
  X21,"1",/,X7,"1",X31,"1",X31,"1",/,X7,"1",X2,
  "N        =",X5,A10,X5,"1",X2,"ACC    =",E8.1,X14,"1",/,X7,"1",X2,
  "TAU pp   =",X5,A10,X5,"1",X2,"ITMAX(STARTSLAB)       =",I4,X2,"1",/,
  X7,"1",X2,"INTVALS =",
  X5,A10,X5,"1",X2,"ITMAX                    =",I4,X2,"1",/,X7,
  "1",X2,"DTAU     =",X5,A10,X5,"1",X31,"1",/,X7,"1",X31,"1",X31,"1">,
  TAKE(SN,10),ACC,TAKE(STAU,10),5*ITMAX,TAKE(SINTVALS,10),ITMAX,
  TAKE(SDTAU,10));
 WRITE(F,<X7,A65>,S);
 %
 WRITE(F,</,X8,"Penetration">);
 WRITE(F,</,X9,"TAU",X7,"E",X12,"F",X11,"G",X8,"E(r=Rc)",
                X8,"Shd",/>);
 WRITE(F,<X8,"0.000",X7,"0",X10,"--",X3,F12.5,X7,"0",
                X12,"--">,(1-RET)**2/(2*TAU));
END;
```

```
%heading tape- en diskfiles voor constante grensvlakconcentratie
PROCEDURE KOPJETAPE(F, WF, SIT, SIGMA, A, BV, NU, LAMBDA,
                    N, TAU, INTVALS, DTAU, INDEX,
                    ITMAX, DIM, VREF, VO, COEF, XI);
VALUE F, WF, SIT, SIGMA, A, BV, NU, LAMBDA, N, TAU, INTVALS,
      DTAU, INDEX, ITMAX, DIM, VREF, VO, XI;
INTEGER WF, SIT, NU, N, INTVALS, INDEX, ITMAX, DIM;
REAL TAU, DTAU, SIGMA, A, BV, LAMBDA, VREF, VO;
REAL ARRAY COEF[*];
FILE F;
BEGIN
 STRING SITUATIE;
 CASE SIT OF
  BEGIN
   1 : SITUATIE:="%rvw constant,    r coordinates";
   2 : SITUATIE:="%rvw constant,   fi coordinates";
   3 : SITUATIE:="%flux constant,   r coordinates";
   4 : SITUATIE:="%flux constant,  fi coordinates";
  END;
 %
 WRITE(F, <I2, ", ", A30, ", ">, SIT, SITUATIE);
 WRITE(F, <5(E21.9, ", "), "% SIGMA, A, BV, NU, LAMBDA">, SIGMA,
           A, BV, NU, LAMBDA);
 IF SIGMA =0 THEN
 BEGIN
  WRITE(F, <I5, ", ", F10.5, ", ", I5, ", ", F10.5, ", ", I5, ", ", I5, ", ",
           "%N, TAU, INTVALS, DTAU, INDEX, ITMAX">,
           N, TAU, INTVALS, DTAU, INDEX, ITMAX);
  WRITE(F, <"%TAU, E, F, G, E(r=Rc), Shd">);
  WRITE(F, <X8, "0.000, ", X7, "0, ", X10, "--, ", X10, "--, ", X10, "0, ",
           X12, "--, ">);
 END
 ELSE
 BEGIN
  %als WF=1 dan algolversie, als WF=2 dan pascalversie
  %als WF=3 dan PC-versie
  XI:=XH*(LAMBDA**NUP+(1-LAMBDA**NUP)*(1+SIGMA*VO))**(NU/NUP);
  IF WF=1 THEN
  BEGIN
   WRITE(F, <I5, ", ", F10.5, ", ", I5, ", ", F10.5, ", ", I5, ", ", I5, ", ",
            "%N, TAU, INTVALS, DTAU, INDEX, DIM">,
            N, TAU, INTVALS, DTAU, [INDEX+INTVALS], DIM);
   WRITE(F, <I5, ", ", F6.3, ", ", F6.3, ", ", "% Vref, VO">, VREF, VO);
   WRITE(F, <"%TAU, E, F, G, E(r=Rc), Shd">);
   WRITE(F, <X7, "0.0000, ", X7, "0, ", X10, "--, ", X3, F12.5, X7, ", 0, ",
            X12, "--, ">, (COEF[1]/XI)**2/(2*TAU));
  END
  ELSE
  BEGIN
   IF WF=2 THEN
   BEGIN
    WRITE(F, <I6, F11.5, I6, F11.5, 2(I6),
             "{N, TAU, INTVALS, DTAU, INDEX, DIM}">,
             N, TAU, INTVALS, DTAU, [INDEX+INTVALS], DIM);
    WRITE(F, <I5, ", ", F6.3, ", ", F6.3, ", ", "% Vref, VO">, VREF, VO);
    WRITE(F, <"{TAU, E, F, G, E(r=Rc), Shd}">);
    WRITE(F, <X7, "0.0000 ", X7, "0 ", X10, "-- ", X3, F12.5, X7, " 0 ",
```

```
  END
  ELSE
  IF WF=3 THEN
  BEGIN
   WRITE(F, <I2, ", ", A30, ", ">, SIT, SITUATIE);
   WRITE(F, <5(E20.9, ", "), "%SIGMA, A, BV, NU, LAMBDA">, SIGMA,
             A, BV, NU, LAMBDA);
   WRITE(F, <I5, ", ", F10.5, ", ", I5, ", ", F10.5, ", ", I5, ", ", I5, ", ",
             "%N, TAU, INTVALS, DTAU, INDEX, DIM">,
             N, TAU, INTVALS, DTAU,
     INTVALS+22+ENTIER((INDEX-1-MIN(INDEX-1, 19)-FACTOR)/FACTOR),
              ITMAX);
   WRITE(F, <I5, ", ", F6.3, ", ", F6.3, ", ", "% Vref, V0">, VREF, V0);
   WRITE(F, <"%       TAU            , E             , F                  ,
          G       , E(r=Rc)       , Shd">);
   WRITE(F, <X7, "0.0000, ", X7, "0, ", X10, "--, ", X3, F12.5, X7, ", 0, ",
             X12, "--, ">, (COEF[1]/XI)**2/(2*TAU));
  END;
 END;
END;
```

```
 %
 READ(IN, /, NU, LAMBDA, A, BV, STOPE, DIM, SIGMA);
TAU:=DTAU:=0.01;ACC:=@-6;ITMAX:=20;INTVALS:=10;SIT:=1;
VREF:=VSTER:=V0:=0;
%Check input, ABORT (and display error message) if inconsistent
IF A<0 OR A>2 THEN ABORT("NIET-KRIMP : only A>=0 or A<=2 allowed");
IF NU<0 OR NU>2 THEN ABORT("NIET-KRIMP : only NU=0, 1 or 2 allowed");
IF NU>0 AND (LAMBDA<0 OR LAMBDA>=1) THEN
 ABORT("NIET-KRIMP : LAMBDA<0 or LAMBDA>=1");
IF BV<0 OR BV>=1 THEN
 ABORT("NIET-KRIMP : boundary value out of range, BV<0 or BV>=1");
IF STOPE<=0 OR STOPE>0.99 THEN
 ABORT("NIET-KRIMP : efficiency-stop out of range, STOPE<=0 or
      STOPE>0.99");
IF DIM>1000 THEN ABORT("NIETKRIMP : try DIM=500");
IF DIM<20 THEN DIM:=20;
N:=200;
IF A>=0.5 THEN N:=400;
IF A>1 THEN N:=800;
IF A>1.5 THEN N:=1600;
IF NU=0 THEN LAMBDA:=0;
BEGIN
 %
 REAL ARRAY USS[0:N], USNS[0:INTVALS, 0:N], U[0:DIM, 0:N],
            RETSNS[0:INTVALS], RET[0:DIM];
 %
 IF NOT STARTSLAB(N, TAU, A, BV, USS, RETSS, ACC, 5*ITMAX)
  THEN ABORT("convergentieproblemen in STARTSLAB");
 %
 CASE NU OF
 BEGIN
  0 : XH:= NU+1;
  1 : XH:=(NU+1)/(1+LAMBDA);
  2 : XH:=(NU+1)/(1+LAMBDA+LAMBDA*LAMBDA)
 END;
 XI:=XH;
 COEF[1]:=(1-RETSS)*XI;
 IF NU=0 THEN
  FOR I:= 0 TO INTVALS DO
        RETSNS[I]:=1-C1*SQRT(I/INTVALS)
  ELSE
   BEGIN FOR I:=0 TO N DO USNS[0, I]:=USS[I];
    IF NOT
     STARTNONSLAB(N, TAU, INTVALS, NU, LAMBDA, A, BV, USNS, RETSNS, ACC, ITMAX)
    THEN ABORT("convergentieproblemen in STARTNONSLAB")
   END;
 IF NU=0 THEN
  BEGIN FOR I:=0 TO N DO U[1, I]:=U[0, I]:=USS[I];RET[0]:=RETSS END
   ELSE
    BEGIN FOR I:=0 TO N DO U[1, I]:=U[0, I]:=USNS[INTVALS, I];
          RET[0]:=RETSNS[INTVALS];
    END;
 IF NU=0 THEN COEF[2]:=COEF[3]:=0 ELSE
  BEGIN
   E1:=1-RETSNS[INTVALS/2];
   E2:=1-RETSNS[INTVALS];
   COEF[3]:=(E2-2*E1+COEF[1]*(SQRT(2)-1))/(1-(1/SQRT(2)));
```

```
 END;
 %
IF NOT PDE(N, DTAU/4, 4, NU, LAMBDA, A, BV, U[1, *], RET[1], ACC, ITMAX)
   THEN ABORT("convergentieproblemen in PDE ");
%
INDEX:=1;STOP1:=STOP:=FALSE;
WHILE INDEX<DIM AND NOT STOP DO
 BEGIN
  IF STOP1 THEN STOP:=TRUE;
  IF 1-RET[INDEX]>STOPE THEN STOP1:=TRUE;
  INDEX:=INDEX+1;
  FOR I:=0 TO N DO U[INDEX, I]:=U[INDEX-1, I];
  %
  IF NOT PDE(N, DTAU, 1, NU, LAMBDA, A, BV, U[INDEX, *], RET[INDEX], ACC, ITMAX)
   THEN ABORT("convergentieproblemen in PDE while loop");
 END;
%
KOPJETABEL(PRINT, A, BV, NU, N, LAMBDA, TAU, DTAU,
                    SIGMA, VSTER, VREF, V0, COEF,
                    INTVALS, ITMAX, ACC, RETSS, XI);
% berekening van efficiency's, flux, G en Shd
FOR I:=1 TO INTVALS DO
 BEGIN
  E:= 1- RETSNS[I];
  F:=(C1/(2*SQRT(I/INTVALS))+C2+1.5*C3*SQRT(I/INTVALS))/(XI*TAU);
  G:=E*F/XI;
  SHD:=(1-BV)*2*(A+1)*F/((RETSNS[I]*(1-BV)+BV)**(A+1)-BV**(A+1));
  WRITE(PRINT, <F13.3, F11.6, F12.5, F12.5, X7, "0", F16.3>,
                TAU*I/INTVALS, E, F, G, SHD);
 END;
%
%
WRITE(PRINT, </, /, X7, "Period after penetration", />);
WRITE(PRINT, <X9, "TAU", X7, "E", X12, "F", X11, "G", X8, "E(r=Rc)",
             X8, "Shd", />);
%
WRITE(PRINT, <F13.2, F11.6, 3(F12.5), F12.3>, TAU, E, F, G, 0, SHD);
FOR I:=1 TO MIN(INDEX, 19) DO
WRITE(PRINT, <F13.2, F11.6, 3(F12.5), F12.3>, TAU+I*DTAU, 1-RET[I],
            (RET[I-1]-RET[I+1])/(XI*2*DTAU),
            (RET[I-1]-RET[I+1])*(1-RET[I])/(XI*XI*2*DTAU),
            1-U[I, 0]/(1-BV),
            (1-BV)*(A+1)*(RET[I-1]-RET[I+1])/(XI*DTAU)/
            ((RET[I]*(1-BV)+BV)**(A+1)-BV**(A+1)));
%
IF INDEX>19 THEN
 BEGIN
  WRITE(PRINT, <X9, "TAU", X7, "E", X12, "F", X11, "G", X8, "E(r=Rc)",
               X8, "Shd", />);
  IF (INDEX-19) MOD 60 =0
   THEN FACTOR:= (INDEX-19)/60
    ELSE FACTOR:= 1+ ENTIER((INDEX-19)/60);
  %
  FOR I:= 19 STEP FACTOR UNTIL INDEX-1 DO
   WRITE(PRINT, <F13.2, F11.6, 3(F12.5), F12.3>, TAU+I*DTAU, 1-RET[I],
               (RET[I-1]-RET[I+1])/(XI*2*DTAU),
               (RET[I-1]-RET[I+1])*(1-RET[I])/(XI*XI*2*DTAU),
```

```
                (1-BV)*(A+1)*(RET[I-1]-RET[I+1])/(XI*DTAU)/
                ((RET[I]*(1-BV)+BV)**(A+1)-BV**(A+1)));
 END;
WRITE(PRINT[SKIP(1)]);
WRITE(PRINT,</,"C1= ",E14.6,"  E1= ",E14.6,"  E2= ",E14.6>,
            COEF[1],E1,E2);
WRITE(PRINT,</,"C2= ",E14.6," C3= ",E14.6,"C3/C2=",E14.6>,
            COEF[2],COEF[3],
            IF NU=0 THEN [0] ELSE [COEF[3]/COEF[2]]);

%
%als WF=1 dan algolversie, als WF=2 dan pascalversie
%als WF=3 dan PC-versie
KOPJETAPE(TAPE,1,SIT,SIGMA,A,BV,NU,LAMBDA,
                    N,TAU,INTVALS,DTAU,INDEX,
                    ITMAX,DIM,VREF,V0,COEF,XI);
FOR I:=1 TO INTVALS DO
 BEGIN
  E:= 1- RETSNS[I];
  F:=(C1/(2*SQRT(I/INTVALS))+C2+1.5*C3*SQRT(I/INTVALS))/(XI*TAU);
  G:=E*F/XI;
  SHD:=2*(1-BV)*(A+1)*F/((RETSNS[I]*(1-BV)+BV)**(A+1)-BV**(A+1));
  WRITE(TAPE,<6(E21.9,",")>,
             TAU*I/INTVALS,E,F,G,0,SHD);
 END;
%
FOR I:=1 TO INDEX-1 DO
WRITE(TAPE,<6(E21.9,",")>,TAU+I*DTAU,1-RET[I],
          (RET[I-1]-RET[I+1])/(XI*2*DTAU),
          (RET[I-1]-RET[I+1])*(1-RET[I])/(XI*XI*2*DTAU),
          1-U[I,0]/(1-BV),
          (1-BV)*(A+1)*(RET[I-1]-RET[I+1])/(XI*DTAU)/
          ((RET[I]*(1-BV)+BV)**(A+1)-BV**(A+1)));
KOPJETAPE(PROFIEL,SIT,SIGMA,A,BV,NU,LAMBDA,
                    N,TAU,INTVALS,DTAU,INDEX,ITMAX );
WRITE(PROFIEL,<"%PROFIELEN BIJ ESTOP=",F12.8>,STOPE);
FOR I:= 1 TO N DO
WRITE(PROFIEL,<E24.17,",">,U[INDEX,I]);
WRITE(PROFIEL,<"% RET[INDEX]= ",E24.17>,RET[INDEX]);
LOCK(TAPE,CRUNCH);
LOCK(PROFIEL,CRUNCH);
%

PROFILES(N,TAU,INTVALS,DTAU,INDEX,NU,LAMBDA,A,BV,USNS,U,RETSNS,
         RET,2);
PROFILES(N,TAU,INTVALS,DTAU,INDEX,NU,LAMBDA,A,BV,USNS,U,RETSNS,
         RET,5);
 END
END.
```

APPENDIX B

listing van het programma

KRIMP

KRIMP

```
BEGIN
 $INCLUDE "STATLIB/ALGOL/DECLARATION          ON APPL"
 $INCLUDE "STATLIB/ALGOL/SELECTPOLREGRESSION ON APPL"
 $INCLUDE "(TGTFDR)DROOGLIBRARY ON USER4."
 FILE IN(KIND=DISK,FILETYPE=7),
      TAPE(KIND=DISK,PROTECTION=SAVE,MAXRECSIZE=22,UNITS=WORDS),
      TAPPASC(KIND=DISK,PROTECTION=SAVE,MAXRECSIZE=22,UNITS=WORDS),
      B(KIND=DISK,PROTECTION=SAVE,MAXRECSIZE=22,UNITS=WORDS),
      PRINT(KIND=PRINTER);

 STRING S,SA,SBV,SNU,SLAMBDA,SN,STAU,SINTVALS,SDTAU,SITUATIE,
        SSIGMA,SVREF,SVO,SVSTER;
 REAL LAMBDA,A,BV,STOPE,TAU,ACC,DTAU,RETSS,XI,SIGMA,
      E,F,G,SHD,XH,VO,VREF,VSTER,DETERM,UGEM,VGEM;
 INTEGER NU,NUP,ITMAX,INTVALS,STEPS,N,DIM,INDEX,I,FACTOR,SIT;
```

```
% heading printerfile-tabellen constante grensvlakconcentratie
PROCEDURE KOPJETABEL(F, A, BV, NU, N, LAMBDA, TAU, DTAU,
                     SIGMA, VSTER, VREF, VO, COEF,
                     INTVALS, ITMAX, ACC, RET, XI);
VALUE A, BV, NU, N, LAMBDA, TAU, DTAU,
      SIGMA, VSTER, VREF, VO, INTVALS, ITMAX, ACC, RET;
REAL LAMBDA, A, BV, TAU, DTAU, SIGMA, VSTER, VREF, VO, ACC, RET, XI;
INTEGER NU, ITMAX, INTVALS, N, WF;
REAL ARRAY COEF[*];
FILE F;
BEGIN
 STRING S, SA, SBV, SNU, SLAMBDA, SN, STAU, SINTVALS, SDTAU;
 INTEGER NUP;
 SA:= STRING(A, *) CAT "          ";
 IF FIRST(SA)="." THEN SA:="0" CAT SA;
 SBV:= STRING(BV, *) CAT "          ";
 IF FIRST(SBV)="." THEN SBV:="0" CAT SBV;
 SNU:= STRING(NU, *) CAT "          ";
 SN:= STRING(N, *) CAT "          ";
 SLAMBDA:= STRING(LAMBDA, *) CAT "          ";
 IF FIRST(SLAMBDA)="." THEN SLAMBDA:="0" CAT SLAMBDA;
 SSIGMA:= STRING(SIGMA, *) CAT "          ";
 IF FIRST(SSIGMA)="." THEN SSIGMA:="0" CAT SSIGMA;
 SVSTER:= STRING(VSTER, *) CAT "          ";
 IF FIRST(SVSTER)="." THEN SVSTER:="0" CAT SVSTER;
 SVREF:= STRING(VREF, *) CAT "          ";
 IF FIRST(SVREF)="." THEN SVREF:="0" CAT SVREF;
 SVO:= STRING(VO, *) CAT "          ";
 IF FIRST(SVO)="." THEN SVO:="0" CAT SVO;
 STAU:= STRING(TAU, *) CAT "          ";
 IF FIRST(STAU)="." THEN STAU:="0" CAT STAU;
 SDTAU:= STRING(DTAU, *) CAT "          ";
 IF FIRST(SDTAU)="." THEN SDTAU:="0" CAT SDTAU;
 SINTVALS:= STRING(INTVALS, *) CAT "          ";
 S:=REPEAT("-", 65);
 WRITE(F[SKIP(1)]);
 WRITE(F, <X7, A65>, S);
 WRITE(F, <X7, "1", X31, "1", X31, "1">);
 IF SIGMA NEQ 0 THEN
 BEGIN
  WRITE(F, <X7, "1", X2, "Physical properties", X10, "1", X2,
   "SHRINKAGE parameters", X9, "1", /, X7, "1", X31, "1", X31, "1",
   /, X7, "1", X2, "A        =", X5, A10, X5, "1", X7, "SIGMA =  ", X2, A10, X4, "1",
   /, X7, "1", X2, "BV       =", X5, A10, X5, "1", X7, "V#    =  ", X2, A10, X4, "1",
   /, X7, "1", X2, "NU       =", X5, A10, X5, "1", X7, "V*    =  ", X2, A10, X4, "1",
   /, X7, "1", X2, "LAMBDA   =", X5, A10, X5, "1", X7, "VO    =  ", X2, A10, X4, "1",
   /, X7, "1", X31, "1", X31, "1">, TAKE(SA, 10), TAKE(SSIGMA, 10),
   TAKE(SBV, 10), TAKE(SVREF, 10), TAKE(SNU, 10), TAKE(SVSTER, 10),
   TAKE(SLAMBDA, 10), TAKE(SVO, 10));
  WRITE(F, <X7, A65>, S);
  WRITE(F, <X7, "1", X31, "1", X31, "1">);
  WRITE(F, <X7, "1", X2, "Discretization", X15, "1", X2, "Accuracy",
   X21, "1", /, X7, "1", X31, "1", X31, "1", /, X7, "1", X2,
   "N        =", X5, A10, X5, "1", X2, "ACC    =", E8.1, X14, "1", /, X7, "1", X2,
   "TAU pp   =", X5, A10, X5, "1", X2, "ITMAX(INITIALIZEFI)    =", I4, X2, "1", /,
   X7, "1", X2, "INTVALS =",
   X5, A10, X5, "1", X2, "ITMAX                    =", I4, X2, "1", /, X7,
```

```
  TAKE(SN,10),ACC,TAKE(STAU,10),5*ITMAX,TAKE(SINTVALS,10),ITMAX,
  TAKE(SDTAU,10));
  WRITE(F,<X7,A65>,S);
  %
 WRITE(F,</,X8,"Penetration">);
 WRITE(F,</,X9,"TAU",X7,"E",X12,"F",X11,"G",X8,"E(r=Rc)",
               X8,"Shd",/>);
 NUP:=NU+1;
 XI:=XH*(LAMBDA**NUP+(1-LAMBDA**NUP)*(1+SIGMA*V0))**(NU/NUP);
 WRITE(F,<X7,"0.0000",X7,"0",X10,"--",X3,F12.5,X7,"0",
               X12,"--">,(COEF[1]/XI)**2/(2*TAU));
END
ELSE
BEGIN
 WRITE(F,<X7,"1",X2,"Physical properties",X10,"1",X2,
  "NON-SHRINKING SYSTEMS",X8,"1",/,X7,"1",X31,"1",X31,"1",
  /,X7,"1",X2,"A         =",X5,A10,X5,"1",X31,"1",
  /,X7,"1",X2,"BV        =",X5,A10,X5,"1",X31,"1",
  /,X7,"1",X2,"NU        =",X5,A10,X5,"1",X31,"1",
  /,X7,"1",X2,"LAMBDA    =",X5,A10,X5,"1",X31,"1",
  /,X7,"1",X31,"1",X31,"1">,TAKE(SA,10),TAKE(SBV,10),
  TAKE(SNU,10),TAKE(SLAMBDA,10));
 WRITE(F,<X7,A65>,S);
 WRITE(F,<X7,"1",X31,"1",X31,"1">);
 WRITE(F,<X7,"1",X2,"Discretization",X15,"1",X2,"Accuracy",
  X21,"1",/,X7,"1",X31,"1",X31,"1",/,X7,"1",X2,
  "N         =",X5,A10,X5,"1",X2,"ACC     =",E8.1,X14,"1",/,X7,"1",X2,
  "TAU pp    =",X5,A10,X5,"1",X2,"ITMAX(STARTSLAB)        =",I4,X2,"1",/,
  X7,"1",X2,"INTVALS =",
  X5,A10,X5,"1",X2,"ITMAX                        =",I4,X2,"1",/,X7,
  "1",X2,"DTAU      =",X5,A10,X5,"1",X31,"1",/,X7,"1",X31,"1",X31,"1">,
  TAKE(SN,10),ACC,TAKE(STAU,10),5*ITMAX,TAKE(SINTVALS,10),ITMAX,
  TAKE(SDTAU,10));
 WRITE(F,<X7,A65>,S);
 %
 WRITE(F,</,X8,"Penetration">);
 WRITE(F,</,X9,"TAU",X7,"E",X12,"F",X11,"G",X8,"E(r=Rc)",
               X8,"Shd",/>);
 WRITE(F,<X8,"0.000",X7,"0",X10,"--",X3,F12.5,X7,"0",
               X12,"--">,(1-RET)**2/(2*TAU));
END;
```

```
%heading tape- en diskfiles voor constante grensvlakconcentratie
PROCEDURE KOPJETAPE(F, WF, SIT, SIGMA, A, BV, NU, LAMBDA,
                    N, TAU, INTVALS, DTAU, INDEX,
                    ITMAX, DIM, VREF, V0, COEF, XI);
VALUE F, WF, SIT, SIGMA, A, BV, NU, LAMBDA, N, TAU, INTVALS,
      DTAU, INDEX, ITMAX, DIM, VREF, V0;
INTEGER WF, SIT, NU, N, INTVALS, INDEX, ITMAX, DIM;
REAL TAU, DTAU, SIGMA, A, BV, LAMBDA, VREF, V0, XI;
REAL ARRAY COEF[*];
FILE F;
BEGIN
 STRING SITUATIE;
 CASE SIT OF
  BEGIN
   1 : SITUATIE:="%rvw constant,    r coordinates";
   2 : SITUATIE:="%rvw constant,   fi coordinates";
   3 : SITUATIE:="%flux constant,   r coordinates";
   4 : SITUATIE:="%flux constant,  fi coordinates";
  END;
 %
 WRITE(F, <I2, ", ", A30, ", ">, SIT, SITUATIE);
 WRITE(F, <5(E21.9, ", "), "% SIGMA, A, BV, NU, LAMBDA">, SIGMA,
             A, BV, NU, LAMBDA);
 IF SIGMA =0 THEN
 BEGIN
  WRITE(F, <I5, ", ", F10.5, ", ", I5, ", ", F10.5, ", ", I5, ", ", I5, ", ",
              "%N, TAU, INTVALS, DTAU, INDEX, ITMAX">,
             N, TAU, INTVALS, DTAU, INDEX, ITMAX);
  WRITE(F, <"%TAU, E, F, G, E(r=Rc), Shd">);
  WRITE(F, <X8, "0.000, ", X7, "0, ", X10, "--, ", X10, "--, ", X10, "0, ",
             X12, "--, ">);
 END
 ELSE
 BEGIN
  %als WF=1 dan algolversie, als WF=2 dan pascalversie
  %als WF=3 dan PC-versie
  XI:=XH*(LAMBDA**NUP+(1-LAMBDA**NUP)*(1+SIGMA*V0))**(NU/NUP);
  IF WF=1 THEN
  BEGIN
   WRITE(F, <I5, ", ", F10.5, ", ", I5, ", ", F10.5, ", ", I5, ", ", I5, ", ",
              "%N, TAU, INTVALS, DTAU, INDEX, DIM">,
             N, TAU, INTVALS, DTAU, [INDEX+INTVALS], DIM);
   WRITE(F, <I5, ", ", F6.3, ", ", F6.3, ", ", "% Vref, V0">, VREF, V0);
   WRITE(F, <"%TAU, E, F, G, E(r=Rc), Shd">);
   WRITE(F, <X7, "0.0000, ", X7, "0, ", X10, "--, ", X3, F12.5, X7, ", 0, ",
             X12, "--, ">, (COEF[1]/XI)**2/(2*TAU));
  END
  ELSE
  BEGIN
   IF WF=2 THEN
   BEGIN
    WRITE(F, <I6, F11.5, I6, F11.5, 2(I6),
               "{N, TAU, INTVALS, DTAU, INDEX, DIM}">,
              N, TAU, INTVALS, DTAU, [INDEX+INTVALS], DIM);
    WRITE(F, <I5, ", ", F6.3, ", ", F6.3, ", ", "% Vref, V0">, VREF, V0);
    WRITE(F, <"{TAU, E, F, G, E(r=Rc), Shd}">);
    WRITE(F, <X7, "0.0000 ", X7, "0 ", X10, "-- ", X3, F12.5, X7, " 0 ",
```

```
  END
  ELSE
  IF WF=3 THEN
  BEGIN
   WRITE(F, <I2, ", ", A30, ", ">, SIT, SITUATIE);
   WRITE(F, <5(E20.9, ", "), "%SIGMA, A, BV, NU, LAMBDA">, SIGMA,
             A, BV, NU, LAMBDA);
   WRITE(F, <I5, ", ", F10.5, ", ", I5, ", ", F10.5, ", ", I5, ", ", I5, ", ",
             "%N, TAU, INTVALS, DTAU, INDEX, DIM">,
             N, TAU, INTVALS, DTAU,
     INTVALS+22+ENTIER((INDEX-1-MIN(INDEX-1, 19)-FACTOR)/FACTOR),
              ITMAX);
   WRITE(F, <I5, ", ", F6.3, ", ", F6.3, ", ", "% Vref, V0">, VREF, V0);
   WRITE(F, <"%       TAU            , E             , F                  ,
          G       , E(r=Rc)       , Shd">);
   WRITE(F, <X7, "0.0000, ", X7, "0, ", X10, "--, ", X3, F12.5, X7, ", 0, ",
             X12, "--, ">, (COEF[1]/XI)**2/(2*TAU));
  END;
 END;
END;
```

```
READ(IN, /, NU, LAMBDA, A, BV, VREF, V0, STOPE, DIM, SIGMA); DTAU:=0.01;
TAU:=0.005; ACC:=@-6; ITMAX:=20; INTVALS:=10; SIT:=2;
%Check input, ABORT (and display error message) if inconsistent
IF A<0 OR A>2 THEN ABORT("KRIMP : only A>=0 or A<=2 allowed");
IF SIGMA<0 OR SIGMA>1 THEN ABORT("KRIMP:only 0<SIGMA<1 allowed");
IF NU=0 THEN ABORT("KRIMP : if NU=0 then use STARTSLAB etc. ");
IF NU<0 OR NU>2 THEN ABORT("KRIMP : only NU=0, 1 or 2 allowed");
IF NU>0 AND (LAMBDA<0 OR LAMBDA>=1) THEN
 ABORT("KRIMP : LAMBDA<0 or LAMBDA>=1");
IF VREF<0 THEN ABORT("KRIMP : VREF<0");
IF VREF>V0 THEN ABORT("KRIMP : VREF>V0");
IF BV<0 OR BV>=1 THEN
 ABORT("KRIMP : boundary value out of range, BV<0 or BV>=1");
IF STOPE<=0 OR STOPE>0.99 THEN
 ABORT("KRIMP : efficiency-stop out of range, STOPE<=0 or
      STOPE>0.99");
IF DIM>1000 THEN ABORT("KRIMP : try DIM=500");
IF DIM<20 THEN DIM:=20;
N:=200;
IF A>=0.5 THEN N:=400;
IF A>1 THEN N:=800;
IF A>1.5 THEN N:=1600;
IF NU=0 THEN LAMBDA:=0;
BEGIN
 %
 REAL ARRAY X, Y, RESIDUAL[1:INTVALS], COEF, POWER[1:3],
           USNS[0:INTVALS, 0:N], U[0:DIM, 0:N],
           RETSNS[0:INTVALS], RET[0:DIM];
 %
 VSTER:= VREF+BV*(V0-VREF);
 NUP:=NU+1;
 IF NOT INITIALIZEFI(N, TAU, NU, LAMBDA, A, SIGMA, VREF, VSTER, V0,
                     USNS[0, *], ACC, 5*ITMAX)

  THEN ABORT("problems in INITIALIZEFI");

 %
 %
 IF NOT
  STARTNONSLABFI(N, TAU, INTVALS, NU, LAMBDA, A, SIGMA, VREF, VSTER, V0,
                 USNS, RETSNS, ACC, ITMAX)
  THEN ABORT("convergencyproblems in STARTNONSLABFI") ;
 FOR I:=0 TO N DO U[1, I]:=U[0, I]:=USNS[INTVALS, I];
 RET[0]:=RETSNS[INTVALS];
 %
 IF NOT PDEFI(N, DTAU/5, 5, NU, LAMBDA, A, SIGMA, VREF, VSTER, V0,
              U[1, *], RET[1], @-5, ITMAX)
    THEN ABORT("convergencyproblems in PDEFI");
 %
 INDEX:=1; STOP1:=STOP:=FALSE;
 WHILE INDEX<DIM AND NOT STOP DO
  BEGIN
   IF STOP1 THEN STOP:=TRUE;
   IF 1-RET[INDEX]>STOPE THEN STOP1:=TRUE;
   INDEX:=INDEX+1;
   FOR I:=0 TO N DO U[INDEX, I]:=U[INDEX-1, I];
   %
```

```
                U[INDEX,*],RET[INDEX],@-5,ITMAX)
   THEN ABORT("convergencyproblems in PDE while loop");
 END;
%
CASE NU OF
BEGIN
 0 : XH:= NU+1;
 1 : XH:=(NU+1)/(1+LAMBDA);
 2 : XH:=(NU+1)/(1+LAMBDA+LAMBDA*LAMBDA)
END;
XI:=XH*(LAMBDA**NUP+(1-LAMBDA**NUP)*(1+SIGMA*VO))**(NU/NUP);
%
% bepaling van coefficienten in Taylorreeks van sqrt(TAU/TAUpp)
FOR I:=1 TO INTVALS DO
 BEGIN X[I]:=SQRT(I/INTVALS);
       Y[I]:=1-RETSNS[I];
 END;
FOR I:=1 TO 3 DO POWER[I]:=I;

%
SELECTPOLREGRESSION(X,Y,INTVALS,POWER[*],
                    3,COEF,RESIDUAL,DETERM,SINGULAR);
IF SINGULAR OR DETERM<0.99999 THEN ABORT("KRIMP : SELECTPOL mis");
%
%als WF=1 dan algolversie, als WF=2 dan pascalversie
%als WF=3 dan PC-versie
KOPJETABEL(PRINT,A,BV,NU,N,LAMBDA,TAU,DTAU,
                    SIGMA,VSTER,VREF,VO,COEF,
                    INTVALS,ITMAX,ACC,RET,XI);
KOPJETAPE(TABEL,1,SIT,SIGMA,A,BV,NU,LAMBDA,
                    N,TAU,INTVALS,DTAU,INDEX,
                    ITMAX,DIM,VREF,VO,COEF,XI);
KOPJETAPE(TAPPASC,2,SIT,SIGMA,A,BV,NU,LAMBDA,
                    N,TAU,INTVALS,DTAU,INDEX,
                    ITMAX,DIM,VREF,VO,COEF,XI);
KOPJETAPE(B,3,SIT,SIGMA,A,BV,NU,LAMBDA,
                    N,TAU,INTVALS,DTAU,INDEX,
                    ITMAX,DIM,VREF,VO,COEF,XI);
%
FOR I:=1 TO INTVALS DO
BEGIN
 E:= 1- RETSNS[I];  UGEM:=BV+(1-BV)*RETSNS[I];
 VGEM:=VREF+(VO-VREF)*UGEM;
 XI:=XH*(LAMBDA**NUP+(1-LAMBDA**NUP)*(1+SIGMA*VGEM))**(NU/NUP);
 F:=(COEF[1]/(2*SQRT(I/INTVALS))+COEF[2]+1.5*COEF[3]*SQRT(I/INTVALS)
    )/(XI*TAU);
 G:=F*E/XI;
 SHD:=(1-BV)*2*(A+1)*F/((UGEM)**(A+1)-BV**(A+1));
 WRITE(PRINT,<F13.4,F11.6,F12.5,F12.5,X7,"0",F16.3>,
               TAU*I/INTVALS,E,F,G,SHD);
 WRITE(TAPE,<6(E21.9,",")>,
            TAU*I/INTVALS,E,F,G,0,SHD);
 WRITE(TAPPASC,<6(E22.9)>,
                 TAU*I/INTVALS,E,F,G,0,SHD);
 WRITE(B,<F14.4,F11.6,F12.5,F12.5,X7,"0",F16.3>,
           TAU*I/INTVALS,E,F,G,SHD);
```

```
WRITE(PRINT,</,/,X7,"Period after penetration",/>);
WRITE(PRINT,<X9,"TAU",X7,"E",X12,"F",X11,"G",X8,"E(r=Rc)",
             X8,"Shd",/>);
%
WRITE(PRINT,<F14.3,F11.6,3(F12.5),F12.3>,TAU,E,F,G,0,SHD);
FOR I:=1 TO MIN(INDEX-1,19) DO
BEGIN
 E:= 1- RET[I];  UGEM:=BV+(1-BV)*RET[I];
 VGEM:=VREF+(VO-VREF)*UGEM;
 XI:=XH*(LAMBDA**NUP+(1-LAMBDA**NUP)*(1+SIGMA*VGEM))**(NU/NUP);
 F:=(RET[I-1]-RET[I+1])/(2*XI*DTAU);
 SHD:=(1-BV)*2*(A+1)*F/(UGEM**(A+1)-BV**(A+1));
 WRITE(PRINT,<F14.3,F11.6,3(F12.5),F12.3>,TAU+I*DTAU,E,F,F*E/XI,
          1-U[I,0]/(1-BV),SHD);
 WRITE(B,<F14.3,F11.6,3(F12.5),F12.3>,TAU+I*DTAU,E,F,F*E/XI,
         1-U[I,0]/(1-BV),SHD);
END;
%
IF INDEX>19 THEN
BEGIN
 WRITE(PRINT,<X9,"TAU",X7,"E",X12,"F",X11,"G",X8,"E(r=Rc)",
              X8,"Shd",/>);
 IF (INDEX-19) MOD 60 =0
  THEN FACTOR:= (INDEX-19)/60
    ELSE FACTOR:= 1+ ENTIER((INDEX-19)/60);
 %
 FOR I:= 19 STEP FACTOR UNTIL INDEX-1 DO
 BEGIN
  E:= 1- RET[I];  UGEM:=BV+(1-BV)*RET[I];
  VGEM:=VREF+(VO-VREF)*UGEM;
  XI:=XH*(LAMBDA**NUP+(1-LAMBDA**NUP)*(1+SIGMA*VGEM))**(NU/NUP);
  F:=(RET[I-1]-RET[I+1])/(2*XI*DTAU);
  G:=F*E/XI;
  SHD:=(1-BV)*2*(A+1)*F/((UGEM)**(A+1)-BV**(A+1));
  IF I=19 THEN
  WRITE(PRINT,<F14.3,F11.6,3(F12.5),F12.3>,
               TAU+I*DTAU,E,F,G,1-U[I,0]/(1-BV),SHD)
  ELSE
  BEGIN
   WRITE(PRINT,<F14.3,F11.6,3(F12.5),F12.3>,
                TAU+I*DTAU,E,F,G,1-U[I,0]/(1-BV),SHD);
   WRITE(B,<F14.3,F11.6,3(F12.5),F12.3>,
            TAU+I*DTAU,E,F,G,1-U[I,0]/(1-BV),SHD);
  END;
 END;
END;
   %
WRITE(PRINT[SKIP(1)]);
WRITE(PRINT,*/,COEF);
WRITE(PRINT,*/,DETERM,SINGULAR);
WRITE(PRINT,*/,RESIDUAL);
%
FOR I:=1 TO INDEX-1 DO
BEGIN
 E:= 1- RET[I];  UGEM:=BV+(1-BV)*RET[I];
 VGEM:=VREF+(VO-VREF)*UGEM;
 XI:=XH*(LAMBDA**NUP+(1-LAMBDA**NUP)*(1+SIGMA*VGEM))**(NU/NUP);
```

```
    SHD:=(1-BV)*(A+1)*2*F/(UGEM**(A+1)-BV**(A+1));
    WRITE(TAPE, <6(E21.9, ", ")>, TAU+I*DTAU, E, F, F*E/XI,
              1-U[I,0]/(1-BV), SHD);
    WRITE(TAPPASC, <6(E22.9)>, TAU+I*DTAU, E, F, F*E/XI,
                      1-U[I,0]/(1-BV), SHD);
  END;

  LOCK(TAPE, CRUNCH);
  LOCK(TAPPASC, CRUNCH);
  LOCK(B, CRUNCH);

  PROFILESFI(N, TAU, INTVALS, DTAU, INDEX, NU, LAMBDA, A, SIGMA, VREF, VSTER,
                  VO, USNS, U, RETSNS, RET, 5);
 END
END.
```

APPENDIX C

listing van het programma NKCF

NKCF

```
BEGIN
 $INCLUDE "(TGTFDR)DROOGLIBRARY ON USER4."

 FILE INVOER(KIND=DISK, FILETYPE=7),
      TAPE(KIND=DISK, PROTECTION=SAVE, MAXRECSIZE=22, UNITS=WORDS),
      TAPPASC(KIND=DISK, PROTECTION=SAVE, MAXRECSIZE=22, UNITS=WORDS),
      B(KIND=DISK, PROTECTION=SAVE, MAXRECSIZE=22, UNITS=WORDS),
      PRINT(KIND=PRINTER),
      TABEL(KIND=PRINTER);


 STRING S, SA, SF, SNU, SLAMBDA, SN, SACC, SITMAX, SDTAU, SITUATIE;
 REAL LAMBDA, A, ACC, DTAU, SIGMA, CFLUX, XI, TAU, E, G, ERC, ER, FN, ED, SHD;
 INTEGER NU, ITMAX, N, DIM, INDEX, I, SIT, FACTOR;
```

```
%heading printerfile-tabellen constante flux
PROCEDURE KOPJECFTABEL(F, WF, A, FLUX, NU, N, LAMBDA, TAU, DTAU,
                       SIGMA, VSTER, VREF, V0, Q, COEF,
                       ITMAX, ACC, RET, FLUXARR);
VALUE WF, A, FLUX, NU, N, LAMBDA, TAU, DTAU,
      SIGMA, VSTER, VREF, V0, Q, ITMAX, ACC;
REAL LAMBDA, A, FLUX, TAU, DTAU, SIGMA, VSTER, VREF, V0, Q, ACC, RET;
INTEGER NU, ITMAX, N, WF;
REAL ARRAY RET, FLUXARR, COEF[*];
FILE F;
BEGIN
 STRING S, SA, SF, SSIGMA, SVREF, SV0, SQ, SNU, SLAMBDA,
        SN, SACC, SDTAU, STAU, SITMAX;
 INTEGER NUP;
 SA:= STRING(A, *) CAT "          ";
 IF FIRST(SA)="." THEN SA:="0" CAT SA;
 SF:= STRING(FLUX, *) CAT "          ";
 IF FIRST(SF)="." THEN SF:="0" CAT SF;
 SSIGMA:= STRING(SIGMA, *) CAT "          ";
 IF FIRST(SSIGMA)="." THEN SSIGMA:="0" CAT SSIGMA;
 SVREF:= STRING(VREF, *) CAT "          ";
 IF FIRST(SVREF)="." THEN SVREF:="0" CAT SVREF;
 SV0:= STRING(V0, *) CAT "          ";
 IF FIRST(SV0)="." THEN SV0:="0" CAT SV0;
 SQ:= STRING(Q, *) CAT "          ";
 IF FIRST(SQ)="." THEN SQ:="0" CAT SQ;
 SNU:= STRING(NU, *) CAT "          ";
 SN:= STRING(N, *) CAT "          ";
 SLAMBDA:= STRING(LAMBDA, *) CAT "          ";
 IF FIRST(SLAMBDA)="." THEN SLAMBDA:="0" CAT SLAMBDA;
 SACC:= STRING(ACC, *) CAT "          ";
 IF FIRST(SACC)="." THEN SACC:="0" CAT SACC;
 SDTAU:= STRING(DTAU, *) CAT "          ";
 IF FIRST(SDTAU)="." THEN SDTAU:="0" CAT SDTAU;
 SITMAX:= STRING(ITMAX, *) CAT "          ";
 S:=REPEAT("-", 65);
 WRITE(F[SKIP(1)]);
 WRITE(F, <X12, A65>, S);
 WRITE(F, <X12, "1", X26, "1", X36, "1">);
 IF SIGMA NEQ 0 THEN
 BEGIN
  WRITE(F, <X12, "1", X2, "Physical properties", X5, "1", X2,
  "SHRINKAGE parameters ", X13, "1", /, X12, "1", X26, "1", X36, "1",
  /, X12, "1", X2, "A        =", X3, A10, X2, "1", X7, "SIGMA =  ", X2, A10, X9, "1",
  /, X12, "1", X2, "Fca0     =", X3, A10, X2, "1", X7, "V#    =  ", X2, A10, X9, "1",
  /, X12, "1", X2, "NU       =", X3, A10, X2, "1", X7, "V0    =  ", X2, A10, X9, "1",
  /, X12, "1", X2, "LAMBDA   =", X3, A10, X2, "1", X7, "Q     =  ", X2, A10, X9, "1",
  /, X12, "1", X26, "1", X36, "1">, TAKE(SA, 10), TAKE(SSIGMA, 10), TAKE(SF, 10),
  TAKE(SVREF, 10), TAKE(SNU, 10), TAKE(SV0, 10), TAKE(SLAMBDA, 10),
  TAKE(SQ, 10));
  WRITE(F, <X12, A65>, S);
  WRITE(F, <X12, "1", X26, "1", X36, "1">);
  WRITE(F, <X12, "1", X2, "Discretization", X10, "1", X2, "Accuracy",
  X26, "1", /, X12, "1", X26, "1", X36, "1", /, X12, "1", X2,
  "N        =", X3, A10, X2, "1", X17, "ACC    =", X1, A10, X1, "1", /, X12, "1", X2,
  "DTAU     =", X3, A10, X2, "1", X2, "ITMAX(CONSTANTFLUXFI)=", X1, A10, X1,
  "1">, TAKE(SN, 10), TAKE(SACC, 10), TAKE(SDTAU, 10), TAKE(SITMAX, 10));
```

```
END
ELSE
BEGIN
 WRITE(F,<X12,"1",X2,"Physical properties",X5,"1",X2,
 "NON-SHRINKING SYSTEMS ",X12,"1",/,X12,"1",X26,"1",X36,"1",
 /,X12,"1",X2,"A         =",X3,A10,X2,"1",X36,"1",
 /,X12,"1",X2,"FcaO      =",X3,A10,X2,"1",X36,"1",
 /,X12,"1",X2,"NU        =",X3,A10,X2,"1",X36,"1",
 /,X12,"1",X2,"LAMBDA    =",X3,A10,X2,"1",X36,"1",
 /,X12,"1",X26,"1",X36,"1">,TAKE(SA,10),TAKE(SF,10),
 TAKE(SNU,10),TAKE(SLAMBDA,10));
 WRITE(F,<X12,A65>,S);
 WRITE(F,<X12,"1",X26,"1",X36,"1">);
 WRITE(F,<X12,"1",X2,"Discretization",X10,"1",X2,"Accuracy",
 X26,"1",/,X12,"1",X26,"1",X36,"1",/,X12,"1",X2,
 "N         =",X3,A10,X2,"1",X17,"ACC     =",X1,A10,X1,"1",/,X12,"1",X2,
 "DTAU      =",X3,A10,X2,"1",X2,"ITMAX(CONSTANTFLUX)   =",X1,A10,X1,
 "1">,TAKE(SN,10),TAKE(SACC,10),TAKE(SDTAU,10),TAKE(SITMAX,10));
 WRITE(F,<X12,A65>,S);
END;
% als F=PRINT dan WF=1 anders als F=TABEL dan WF=2 invoeren
IF WF=1 THEN
BEGIN
 WRITE(F,</,X13,"TAU",X11,"E",X14,"E(r=Rc)",X7,
                  "E(r=R)",X7,"Fca",/>);
 WRITE(F,<X10,"0.000000",X9,"0",X14,"0",X13,"0",F14.6>,FLUXARR[0]);
 WRITE(F,<F18.6,F14.6,X31,F14.6>,DTAU/10,1-RET[1],FLUXARR[1]);
END
ELSE
BEGIN
 IF WF=2 THEN
 BEGIN
  WRITE(F,</,X13,"TAU",X10,"Fn/Ei'",X8,
                   "E/Ei'",X9,"Shd",X12,"G",/>);
  WRITE(F,<X10,"0.000000",X9,"-",X13,"0",X13,"-",X13,"-">);
  WRITE(F,<F18.6>,DTAU/10);
 END;
END;
```

```
PROCEDURE KOPJETAPECF(F, WF, SIT, SIGMA, A, CFLUX, NU, LAMBDA,
                          N, DTAU, INDEX, ITMAX, DIM);
VALUE WF, SIT, SIGMA, A, CFLUX, NU, LAMBDA,
      N, DTAU, INDEX, ITMAX, DIM;
INTEGER WF, SIT, NU, N, INDEX, ITMAX, DIM;
REAL SIGMA, A, CFLUX, LAMBDA, DTAU;
FILE F;
BEGIN
 STRING SITUATIE;
 CASE SIT OF
 BEGIN
  1 : SITUATIE:="%rvw constant,    r coordinates";
  2 : SITUATIE:="%rvw constant,   fi coordinates";
  3 : SITUATIE:="%flux constant,   r coordinates";
  4 : SITUATIE:="%flux constant,  fi coordinates";
 END;
 % als WF=1 dan naar TAPE,  als WF=2 dan naar TAPPASC,
 % als WF=3 dan naar B
 WRITE(F, <I2, ", ", A30, ", ">, SIT, SITUATIE);
 WRITE(F, <8(F8.3, ", "), "% SIGMA, Q, VREF, V0, A, CFLUX, NU, LAMBDA">,
       SIGMA, Q, VREF, V0, A, CFLUX, NU, LAMBDA);
 WRITE(F, <I5, ", ", F10.5, ", ", I5, ", ", I5, ", ", I5, ", ", X2,
             "%N, DTAU, INDEX, ITMAX, DIM, ">,
             N, DTAU, [INDEX+1], ITMAX, DIM);
 IF WF=1 THEN
 BEGIN
  WRITE(F, <"%", X4, "TAU, ", X12, "E, ", X13, "E(r=Rc), ", X7,
      "E(r=R), ", X10, "Fn/Ei', ", X7, "E/Ei', ", X12, "Shd, ", X10, "G, ">);
  WRITE(F, <X8, "0.000, ", X7, "0, ", X14, "0, ", X14, "0, ", X15, "  , ",
               X15, "0, ", X15, "  , ", X15, "  , ">);
 END
 ELSE
 BEGIN
  IF WF=2 THEN
  BEGIN
   WRITE(F, <"{", X4, "TAU, ", X12, "E, ", X13, "E(r=Rc), ", X7,
      "E(r=R), ", X10, "Fn/Ei', ", X7, "E/Ei', ", X12, "Shd, ", X10, "G, }">);
   WRITE(F, <X8, "0.000", X8, "0", X15, "0", X15, "0", X16, "-",
               X16, "0", X16, "-", X16, "-">);
  END
  ELSE
  IF WF=3 THEN
  BEGIN
   WRITE(F, <I5, ", ", F10.5, ", ", I5, ", ", I5, ", ", I5, ", ", X2,
    "%N, DTAU, INDEX, ITMAX, DIM, ">, N, DTAU,
    [IF INDEX-1<=40 THEN INDEX+1 ELSE
    43+ENTIER((INDEX-1-MIN(INDEX-1,40)-FACTOR)/FACTOR)], ITMAX, DIM);
   WRITE(F, <"%", X4, "TAU, ", X12, "E, ", X13, "E(r=Rc), ", X7,
      "E(r=R), ", X10, "Fn/Ei', ", X7, "E/Ei', ", X12, "Shd, ", X10, "G, ">);
   WRITE(F, <X8, "0.000, ", X7, "0, ", X14, "0, ", X14, "0, ", X15, "  , ",
               X15, "0, ", X15, "  , ", X15, "  , ">);
  END;
 END;
```

```
READ(INVOER, /, NU, LAMBDA, A, CFLUX, DTAU, DIM, PLOT);
ACC:=@-5; ITMAX:=20; SIGMA:=0; VREF:=VSTER:=V0:=0; SIT:=3;
%Check input, ABORT (and display error message) if inconsistent
IF A<0 OR A>2 THEN ABORT("NKCF : only A>=0 or A<=2 allowed");
IF NU<0 OR NU>2 THEN ABORT("NKCF : only NU=0, 1 or 2 allowed");
IF NU>0 AND (LAMBDA<0 OR LAMBDA>=1) THEN
 ABORT("NKCF : LAMBDA<0 or LAMBDA>=1");
IF CFLUX<=0 OR CFLUX>20 THEN
 ABORT("NKCF : FLUX not positive or too large");
IF DIM>600 THEN ABORT("NKCF : try DIM=400");
IF DIM<1   THEN ABORT("NKCF : DIM not positive");
N:=200;
IF A>=0.5 THEN N:=400;
IF A>1 THEN N:=800;
IF A>1.5 THEN N:=1600;
IF NU=0 THEN LAMBDA:=0;
BEGIN
 %
 REAL ARRAY UCF[0:DIM, 0:N], RETCF[0:DIM];
REAL ARRAY TAUH, EH, ERCH, ERH, FNH, EDH, GH, SHDH[0:DIM];
 % UCF[0, 0:N] initialiseren op homogeen beginprofiel
 INDEX:=0;
 FOR I:= 0 TO N DO UCF[0, I]:=UCF[1, I]:=1; RETCF[0]:=1;
 WHILE INDEX:=INDEX+1<=11 AND CONSTANTFLUX(N, DTAU/10, NU, LAMBDA, A,
      CFLUX, UCF[INDEX, *], RETCF[INDEX], ACC, ITMAX) DO
 BEGIN
  FOR I:=0 TO N DO UCF[INDEX+1, I]:=UCF[INDEX, I];
 END; INDEX:=INDEX-1;
 FOR I:=0 TO N DO UCF[INDEX+1, I]:=UCF[INDEX-1, I];
 WHILE INDEX:=INDEX+1<DIM AND CONSTANTFLUX(N, DTAU, NU, LAMBDA, A,
      CFLUX, UCF[INDEX, *], RETCF[INDEX], ACC, ITMAX) DO
 BEGIN
  FOR I:=0 TO N DO UCF[INDEX+1, I]:=UCF[INDEX, I];
 END;
 INDEX:=INDEX-1;
 % index heeft nu een waarde 1 kleiner dan de opgegeven dim
 % de profielen zijn berekend t/m dim-1=index
 %
 CASE NU OF BEGIN
  0 : XI:= NU+1;
  1 : XI:=(NU+1)/(1+LAMBDA);
  2 : XI:=(NU+1)/(1+LAMBDA+LAMBDA*LAMBDA)   END;
 %
 KOPJECFTABEL(PRINT, 1, A, FLUX, NU, N, LAMBDA, TAU, DTAU,
                     SIGMA, VSTER, VREF, V0, Q, COEF,
                     ITMAX, ACC, RET, FLUXARR);
 KOPJECFTABEL(TABEL, 2, A, FLUX, NU, N, LAMBDA, TAU, DTAU,
                     SIGMA, VSTER, VREF, V0, Q, COEF,
                     ITMAX, ACC, RET, FLUXARR);
 I  :=1;
 TAU:= I*DTAU/10;
   E:= 1-RETCF[I];
 WRITE(PRINT, <F18.6, F14.6>, TAU, E);
 WRITE(TABEL, <F18.6>, TAU);

 %
 FOR I:=2 TO 10 DO
```

```
   TAU:=I*DTAU/10;
     E:= 1-RETCF[I];
   ERC:= 1-((UCF[I-1,0]+2*UCF[I,0]+UCF[I+1,0])/4);
    ER:= 1-((UCF[I-1,N]+2*UCF[I,N]+UCF[I+1,N])/4);
    FN:= CFLUX/ER;
    ED:= E/ER;
     G:= FN*ED/XI;
   SHD:= 2*(A+1)*CFLUX/(RETCF[I]**(A+1)-(1-ER)**(A+1));
   WRITE(PRINT,<F18.6,3(F14.6)>,TAU,E,ERC,ER);
   WRITE(TABEL,<F18.6,4(F14.6)>,TAU,FN,ED,SHD,G);
 END;
I   :=12;
TAU:= (I-10)*DTAU;
  E:= 1-RETCF[I];
ERC:= 1-((UCF[I-2,0]+2*UCF[I,0]+UCF[I+1,0])/4);
 ER:= 1-((UCF[I-2,N]+2*UCF[I,N]+UCF[I+1,N])/4);
 FN:= CFLUX/ER;
 ED:= E/ER;
  G:= FN*ED/XI;
SHD:= 2*(A+1)*CFLUX/(RETCF[I]**(A+1)-(1-ER)**(A+1));
WRITE(PRINT,<F18.6,3(F14.6)>,TAU,E,ERC,ER);
WRITE(TABEL,<F18.6,4(F14.6)>,TAU,FN,ED,SHD,G);
FOR I:=13 TO MIN(INDEX-1,40) DO
 BEGIN
  TAU:= (I-10)*DTAU;
    E:= 1-RETCF[I];
  ERC:= 1-((UCF[I-1,0]+2*UCF[I,0]+UCF[I+1,0])/4);
   ER:= 1-((UCF[I-1,N]+2*UCF[I,N]+UCF[I+1,N])/4);
   FN:= CFLUX/ER;
   ED:= E/ER;
    G:= FN*ED/XI;
  SHD:= 2*(A+1)*CFLUX/(RETCF[I]**(A+1)-(1-ER)**(A+1));
  WRITE(PRINT,<F18.6,3(F14.6)>,TAU,E,ERC,ER);
  WRITE(TABEL,<F18.6,4(F14.6)>,TAU,FN,ED,SHD,G);
 END;
WRITE(PRINT[SKIP(1)]);
WRITE(TABEL[SKIP(1)]);
IF INDEX-1>40
 THEN
  BEGIN
   WRITE(PRINT,<X13,"TAU",X11,"E",X11,
                "E(r=Rc)",X7,"E(r=R)",/>);
   WRITE(TABEL,<X13,"TAU",X10,"Fn/Ei'",X8,
                "E/Ei'",X9,"Shd",X12,"G",/>);
   IF (INDEX-41) MOD 60 =0
    THEN FACTOR:=(INDEX-41)/60
     ELSE FACTOR:=1+ENTIER((INDEX-41)/60);
  END;
IF (INDEX-1)>40 THEN
BEGIN
FOR I:=MIN(INDEX-1,40) TO INDEX-1 DO
 BEGIN
  TAUH[I]:= (I-10)*DTAU;
    EH[I]:= 1-RETCF[I];
  ERCH[I]:= 1-((UCF[I-1,0]+2*UCF[I,0]+UCF[I+1,0])/4);
   ERH[I]:= 1-((UCF[I-1,N]+2*UCF[I,N]+UCF[I+1,N])/4);
   FNH[I]:= CFLUX/ERH[I];
```

```
    GH[I]:= FNH[I]*EDH[I]/XI;
  SHDH[I]:= 2*(A+1)*CFLUX/(RETCF[I]**(A+1)-(1-ERH[I])**(A+1));
 END;
%waarde index-1 is 1 lager dan tot waar profielen zijn berekend
FOR I:=MIN(INDEX-1,40) STEP FACTOR UNTIL INDEX-1 DO
 BEGIN
  WRITE(PRINT,<F18.6,3(F14.6)>,TAUH[I],EH[I],ERCH[I],ERH[I]);
  WRITE(TABEL,<F18.6,4(F14.6)>,TAUH[I],FNH[I],EDH[I],SHDH[I],GH[I]);
 END;
END;
IF PLOT THEN
 BEGIN
  CFPROFILES(N,DTAU,INDEX,NU,LAMBDA,A,CFLUX,UCF,4,TRUE);
  CFPROFILES(N,DTAU,INDEX,NU,LAMBDA,A,CFLUX,UCF,6,TRUE);
 END;
%
I   := 1;
TAU:= I*DTAU/10;
  E:= 1-RETCF[I];
WRITE(TAPE,<2(E15.8,","),6(X12,"--,")>,TAU,E);
WRITE(B,<2(E15.8,","),6(X12,"--,")>,TAU,E);
WRITE(TAPPASC,<2(E16.8),6(X13,"--")>,TAU,E);
FOR I:= 2 TO 10 DO
 BEGIN
  TAU:=I*DTAU/10;
    E:= 1-RETCF[I];
  ERC:= 1-((UCF[I-1,0]+2*UCF[I,0]+UCF[I+1,0])/4);
   ER:= 1-((UCF[I-1,N]+2*UCF[I,N]+UCF[I+1,N])/4);
   FN:= CFLUX/ER;
   ED:= E/ER;
    G:= FN*ED/XI;
  SHD:= 2*(A+1)*CFLUX/(RETCF[I]**(A+1)-(1-ER)**(A+1));
  WRITE(TAPE,<8(E15.8,",")>,TAU,E,ERC,ER,FN,ED,SHD,G);
  WRITE(B,<8(E15.8,",")>,TAU,E,ERC,ER,FN,ED,SHD,G);
  WRITE(TAPPASC,<8(E16.8)>,TAU,E,ERC,ER,FN,ED,SHD,G);
 END;
I   := 12;
TAU:= (I-10)*DTAU;
  E:= 1-RETCF[I];
ERC:= 1-((UCF[I-2,0]+2*UCF[I,0]+UCF[I+1,0])/4);
 ER:= 1-((UCF[I-2,N]+2*UCF[I,N]+UCF[I+1,N])/4);
 FN:= CFLUX/ER;
 ED:= E/ER;
  G:= FN*ED/XI;
SHD:= 2*(A+1)*CFLUX/(RETCF[I]**(A+1)-(1-ER)**(A+1));
WRITE(TAPE,<8(E15.8,",")>,TAU,E,ERC,ER,FN,ED,SHD,G);
WRITE(TAPPASC,<8(E16.8)>,TAU,E,ERC,ER,FN,ED,SHD,G);
WRITE(B,<8(E15.8,",")>,TAU,E,ERC,ER,FN,ED,SHD,G);
FOR I:= 13 TO INDEX-1 DO
 BEGIN
  TAU:= (I-10)*DTAU;
    E:= 1-RETCF[I];
  ERC:= 1-((UCF[I-1,0]+2*UCF[I,0]+UCF[I+1,0])/4);
   ER:= 1-((UCF[I-1,N]+2*UCF[I,N]+UCF[I+1,N])/4);
   FN:= CFLUX/ER;
   ED:= E/ER;
    G:= FN*ED/XI;
```

```
    WRITE(TAPE, <8(E15.8, ", ")>, TAU, E, ERC, ER, FN, ED, SHD, G);
    WRITE(TAPPASC, <8(E16.8)>, TAU, E, ERC, ER, FN, ED, SHD, G);
   END;
  FOR I:=13 TO MIN(INDEX-1,40) DO
   BEGIN
    TAU:= (I-10)*DTAU;
      E:= 1-RETCF[I];
    ERC:= 1-((UCF[I-1,0]+2*UCF[I,0]+UCF[I+1,0])/4);
     ER:= 1-((UCF[I-1,N]+2*UCF[I,N]+UCF[I+1,N])/4);
     FN:= CFLUX/ER;
     ED:= E/ER;
      G:= FN*ED/XI;
    SHD:= 2*(A+1)*CFLUX/(RETCF[I]**(A+1)-(1-ER)**(A+1));
    WRITE(B, <8(E15.8, ", ")>, TAU, E, ERC, ER, FN, ED, SHD, G);
   END;
  IF INDEX-1>40
   THEN
    BEGIN
     IF (INDEX-41) MOD 60 =0
      THEN FACTOR:=(INDEX-41)/60
       ELSE FACTOR:=1+ENTIER((INDEX-41)/60);
    END;
 IF INDEX-1>40  THEN
 BEGIN
  FOR I:=MIN(INDEX-1,40) TO INDEX-1 DO
   BEGIN
    TAUH[I]:= (I-10)*DTAU;
      EH[I]:= 1-RETCF[I];
    ERCH[I]:= 1-((UCF[I-1,0]+2*UCF[I,0]+UCF[I+1,0])/4);
     ERH[I]:= 1-((UCF[I-1,N]+2*UCF[I,N]+UCF[I+1,N])/4);
     FNH[I]:= CFLUX/ERH[I];
     EDH[I]:= EH[I]/ERH[I];
      GH[I]:= FNH[I]*EDH[I]/XI;
    SHDH[I]:= 2*(A+1)*CFLUX/(RETCF[I]**(A+1)-(1-ERH[I])**(A+1));
   END;
  %waarde index-1 is 1 lager dan tot waar profielen zijn berekend
  FOR I:=MIN(INDEX-1,40)+FACTOR STEP FACTOR UNTIL INDEX-1 DO
   BEGIN
    WRITE(B, <8(E15.8, ", ")>, TAUH[I], EH[I], ERCH[I], ERH[I], FNH[I],
                               EDH[I], SHDH[I], GH[I]);
   END;
 END;
  LOCK(TAPE, CRUNCH);
  LOCK(B, CRUNCH);
  LOCK(TAPPASC, CRUNCH);
  %
 END
END.
```

APPENDIX D

listing van het programma KCA

KCA

```
BEGIN  %programma om voor alle sigma's VARIFLUX te berekenen
 $INCLUDE "(TGTFDR)DROOGLIBRARY ON USER4."

 FILE  INVOER(KIND=DISK, FILETYPE=7),
       TAPE(KIND=DISK, PROTECTION=SAVE, MAXRECSIZE=22, UNITS=WORDS),
       B(KIND=DISK, PROTECTION=SAVE, MAXRECSIZE=22, UNITS=WORDS),
       OUT(KIND=REMOTE),
       TABEL(KIND=PRINTER),
       PRINT(KIND=PRINTER);

 STRING  S, SA, SF, SNU, SLAMBDA, SN, SACC, SITMAX, SDTAU, SITUATIE,
         SSIGMA, SVREF, SVO, SQ;
 REAL  LAMBDA, A, ACC, DTAU, SIGMA, CFLUX, XI, XH, VREF, VO, G, ERC, ER,
       E, VGEM, NUP, TAU, FN, ED, SHD, SFAC, FCAO, Q, RETH;
 INTEGER  NU, ITMAX, N, DIM, INDEX, I, SIT, FACTOR;
```

```
 %predictor-corrector formule toegepast omdat flux na elke stap
 %aangepast wordt bij krimp
BOOLEAN PROCEDURE VARIFLUX(N, DTAU, NU, LAMBDA, A, SIGMA, VREF, VO,
                  Q, FCAO, U, RET, ACC, ITMAX);
VALUE N, DTAU, NU, LAMBDA, A, SIGMA, VREF, VO, Q, FCAO, ACC, ITMAX;
REAL ARRAY U[*];
INTEGER N, NU, ITMAX;
REAL DTAU, LAMBDA, A, SIGMA, VREF, VO, Q, FCAO, RET, ACC;
BEGIN
 BOOLEAN GELUKT;
 IF NU=0 AND (Q NEQ 0 OR SIGMA NEQ 0 )
 THEN ABORT("VCONSTANTFLUX :
  als NU=0 heeft varieren van Q en SIGMA geen zin");
 IF SIGMA=0 AND Q NEQ 0 THEN
 ABORT("VCONSTANTFLUX : als SIGMA=0 heeft Q>0 geen invloed");
 IF Q=0 AND (SIGMA NEQ 0 OR NU NEQ 0 ) THEN
 ABORT("VCONSTANTFLUX : Q=0 impliceert vlakke laag dus NU=0 en
 heeft varieren van SIGMA geen zin");

 IF Q=0 OR SIGMA=0 OR NU=0
 THEN
 % rekenen alsof we werken met een niet-krimpend systeem
 GELUKT:=CONSTANTFLUX(N, DTAU, NU, LAMBDA, A, FCAO, U, RET, ACC, ITMAX)
 ELSE
 % rekenen alsof we werken met krimp
 BEGIN
  REAL S, RETH, E, FLUX;
  INTEGER I;
  REAL ARRAY UH[0:N];
  S:=(1-LAMBDA**(NU+1))*SIGMA*(VO-VREF)/(1+(1-LAMBDA**(NU+1)*SIGMA*VO);
  % initialiseren
  FOR I:=0 TO N DO UH[I]:=U[I];
  % retentie berekenen
  RET:=U[0]+U[N];
  FOR I:=1 STEP 2 UNTIL N-1 DO RET:=RET+4*U[I];
  FOR I:=2 STEP 2 UNTIL N-2 DO RET:=RET+2*U[I];
  RETH:=RET:=RET/3/N;
  %
  E:=1-RET;
  FLUX:=FCAO/(1-S*E)**(Q/(NU+1));
  GELUKT:= CONSTANTFLUXFI(N, DTAU, NU, LAMBDA, A, SIGMA, VREF, VO, FLUX, U,
                  RET, ACC, ITMAX);
  % tweede initialisatie
  E:=1-(RETH+RET)/2;
  FLUX:=FCAO/(1-S*E)**(Q/(NU+1));
  IF GELUKT THEN
   GELUKT:= CONSTANTFLUXFI(N, DTAU, NU, LAMBDA, A, SIGMA, VREF, VO, FLUX, UH,
                  RET, ACC, ITMAX);
 FOR I:=0 TO N DO U[I]:=UH[I];
 END;
 VCONSTANTFLUX:=GELUKT;
```

```
%heading printerfile-tabellen constante flux
PROCEDURE KOPJECFTABEL(F, WF, A, FLUX, NU, N, LAMBDA, TAU, DTAU,
                       SIGMA, VSTER, VREF, VO, Q, COEF,
                       ITMAX, ACC, RET, FLUXARR);
VALUE WF, A, FLUX, NU, N, LAMBDA, TAU, DTAU,
      SIGMA, VSTER, VREF, VO, Q, ITMAX, ACC;
REAL LAMBDA, A, FLUX, TAU, DTAU, SIGMA, VSTER, VREF, VO, Q, ACC, RET;
INTEGER NU, ITMAX, N, WF;
REAL ARRAY RET, FLUXARR, COEF[*];
FILE F;
BEGIN
 STRING S, SA, SF, SSIGMA, SVREF, SVO, SQ, SNU, SLAMBDA,
        SN, SACC, SDTAU, STAU, SITMAX;
 INTEGER NUP;
 SA:= STRING(A, *) CAT "          ";
 IF FIRST(SA)="." THEN SA:="0" CAT SA;
 SF:= STRING(FLUX, *) CAT "          ";
 IF FIRST(SF)="." THEN SF:="0" CAT SF;
 SSIGMA:= STRING(SIGMA, *) CAT "          ";
 IF FIRST(SSIGMA)="." THEN SSIGMA:="0" CAT SSIGMA;
 SVREF:= STRING(VREF, *) CAT "          ";
 IF FIRST(SVREF)="." THEN SVREF:="0" CAT SVREF;
 SVO:= STRING(VO, *) CAT "          ";
 IF FIRST(SVO)="." THEN SVO:="0" CAT SVO;
 SQ:= STRING(Q, *) CAT "          ";
 IF FIRST(SQ)="." THEN SQ:="0" CAT SQ;
 SNU:= STRING(NU, *) CAT "          ";
 SN:= STRING(N, *) CAT "          ";
 SLAMBDA:= STRING(LAMBDA, *) CAT "          ";
 IF FIRST(SLAMBDA)="." THEN SLAMBDA:="0" CAT SLAMBDA;
 SACC:= STRING(ACC, *) CAT "          ";
 IF FIRST(SACC)="." THEN SACC:="0" CAT SACC;
 SDTAU:= STRING(DTAU, *) CAT "          ";
 IF FIRST(SDTAU)="." THEN SDTAU:="0" CAT SDTAU;
 SITMAX:= STRING(ITMAX, *) CAT "          ";
 S:=REPEAT("-",65);
 WRITE(F[SKIP(1)]);
 WRITE(F, <X12,A65>, S);
 WRITE(F, <X12,"1",X26,"1",X36,"1">);
 IF SIGMA NEQ 0 THEN
 BEGIN
  WRITE(F, <X12,"1",X2,"Physical properties",X5,"1",X2,
  "SHRINKAGE parameters ",X13,"1",/,X12,"1",X26,"1",X36,"1",
  /,X12,"1",X2,"A        =",X3,A10,X2,"1",X7,"SIGMA =  ",X2,A10,X9,"1",
  /,X12,"1",X2,"Fca0     =",X3,A10,X2,"1",X7,"V#     =  ",X2,A10,X9,"1",
  /,X12,"1",X2,"NU       =",X3,A10,X2,"1",X7,"VO     =  ",X2,A10,X9,"1",
  /,X12,"1",X2,"LAMBDA   =",X3,A10,X2,"1",X7,"Q      =  ",X2,A10,X9,"1",
  /,X12,"1",X26,"1",X36,"1">,TAKE(SA,10),TAKE(SSIGMA,10),TAKE(SF,10),
  TAKE(SVREF,10),TAKE(SNU,10),TAKE(SVO,10),TAKE(SLAMBDA,10),
```

```
 WRITE(F,<X12,A65>,S);
 WRITE(F,<X12,"1",X26,"1",X36,"1">);
 WRITE(F,<X12,"1",X2,"Discretization",X10,"1",X2,"Accuracy",
 X26,"1",/,X12,"1",X26,"1",X36,"1",/,X12,"1",X2,
 "N        =",X3,A10,X2,"1",X17,"ACC    =",X1,A10,X1,"1",/,X12,"1",X2,
 "DTAU     =",X3,A10,X2,"1",X2,"ITMAX(CONSTANTFLUXFI)=",X1,A10,X1,
 "1">,TAKE(SN,10),TAKE(SACC,10),TAKE(SDTAU,10),TAKE(SITMAX,10));
 WRITE(F,<X12,A65>,S);
END
ELSE
BEGIN
 WRITE(F,<X12,"1",X2,"Physical properties",X5,"1",X2,
 "NON-SHRINKING SYSTEMS ",X12,"1",/,X12,"1",X26,"1",X36,"1",
 /,X12,"1",X2,"A          =",X3,A10,X2,"1",X36,"1",
 /,X12,"1",X2,"FcaO       =",X3,A10,X2,"1",X36,"1",
 /,X12,"1",X2,"NU         =",X3,A10,X2,"1",X36,"1",
 /,X12,"1",X2,"LAMBDA     =",X3,A10,X2,"1",X36,"1",
 /,X12,"1",X26,"1",X36,"1">,TAKE(SA,10),TAKE(SF,10),
 TAKE(SNU,10),TAKE(SLAMBDA,10));
 WRITE(F,<X12,A65>,S);
 WRITE(F,<X12,"1",X26,"1",X36,"1">);
 WRITE(F,<X12,"1",X2,"Discretization",X10,"1",X2,"Accuracy",
 X26,"1",/,X12,"1",X26,"1",X36,"1",/,X12,"1",X2,
 "N        =",X3,A10,X2,"1",X17,"ACC    =",X1,A10,X1,"1",/,X12,"1",X2,
 "DTAU     =",X3,A10,X2,"1",X2,"ITMAX(CONSTANTFLUX)   =",X1,A10,X1,
 "1">,TAKE(SN,10),TAKE(SACC,10),TAKE(SDTAU,10),TAKE(SITMAX,10));
 WRITE(F,<X12,A65>,S);
END;
% als F=PRINT dan WF=1 anders als F=TABEL dan WF=2 invoeren
IF WF=1 THEN
BEGIN
 WRITE(F,</,X13,"TAU",X11,"E",X14,"E(r=Rc)",X7,
                  "E(r=R)",X7,"Fca",/>);
 WRITE(F,<X10,"0.000000",X9,"0",X14,"0",X13,"0",F14.6>,FLUXARR[0]);
 WRITE(F,<F18.6,F14.6,X31,F14.6>,DTAU/10,1-RET[1],FLUXARR[1]);
END
ELSE
BEGIN
 IF WF=2 THEN
 BEGIN
  WRITE(F,</,X13,"TAU",X10,"Fn/Ei'",X8,
                  "E/Ei'",X9,"Shd",X12,"G",/>);
  WRITE(F,<X10,"0.000000",X9,"-",X13,"0",X13,"-",X13,"-">);
  WRITE(F,<F18.6>,DTAU/10);
 END;
END;
```

```
PROCEDURE KOPJETAPECF(F, WF, SIT, SIGMA, A, CFLUX, NU, LAMBDA,
                          N, DTAU, INDEX, ITMAX, DIM);
VALUE WF, SIT, SIGMA, A, CFLUX, NU, LAMBDA,
      N, DTAU, INDEX, ITMAX, DIM;
INTEGER WF, SIT, NU, N, INDEX, ITMAX, DIM;
REAL SIGMA, A, CFLUX, LAMBDA, DTAU;
FILE F;
BEGIN
 STRING SITUATIE;
 CASE SIT OF
 BEGIN
  1 : SITUATIE:="%rvw constant,    r coordinates";
  2 : SITUATIE:="%rvw constant,   fi coordinates";
  3 : SITUATIE:="%flux constant,   r coordinates";
  4 : SITUATIE:="%flux constant,  fi coordinates";
 END;
 % als WF=1 dan naar TAPE,  als WF=2 dan naar TAPPASC,
 % als WF=3 dan naar B
 WRITE(F, <I2, ", ", A30, ", ">, SIT, SITUATIE);
 WRITE(F, <8(F8.3, ", "), "% SIGMA, Q, VREF, V0, A, CFLUX, NU, LAMBDA">,
       SIGMA, Q, VREF, V0, A, CFLUX, NU, LAMBDA);
 WRITE(F, <I5, ", ", F10.5, ", ", I5, ", ", I5, ", ", I5, ", ", X2,
             "%N, DTAU, INDEX, ITMAX, DIM, ">,
             N, DTAU, [INDEX+1], ITMAX, DIM);
 IF WF=1 THEN
 BEGIN
  WRITE(F, <"%", X4, "TAU, ", X12, "E, ", X13, "E(r=Rc), ", X7,
      "E(r=R), ", X10, "Fn/Ei', ", X7, "E/Ei', ", X12, "Shd, ", X10, "G, ">);
  WRITE(F, <X8, "0.000, ", X7, "0, ", X14, "0, ", X14, "0, ", X15, " , ",
              X15, "0, ", X15, " , ", X15, " , ">);
 END
 ELSE
 BEGIN
  IF WF=2 THEN
  BEGIN
   WRITE(F, <"{", X4, "TAU, ", X12, "E, ", X13, "E(r=Rc), ", X7,
       "E(r=R), ", X10, "Fn/Ei', ", X7, "E/Ei', ", X12, "Shd, ", X10, "G, }">);
   WRITE(F, <X8, "0.000", X8, "0", X15, "0", X15, "0", X16, "-",
              X16, "0", X16, "-", X16, "-">);
  END
  ELSE
  IF WF=3 THEN
  BEGIN
   WRITE(F, <I5, ", ", F10.5, ", ", I5, ", ", I5, ", ", I5, ", ", X2,
     "%N, DTAU, INDEX, ITMAX, DIM, ">, N, DTAU,
     [IF INDEX-1<=40 THEN INDEX+1 ELSE
     43+ENTIER((INDEX-1-MIN(INDEX-1, 40)-FACTOR)/FACTOR)], ITMAX, DIM);
   WRITE(F, <"%", X4, "TAU, ", X12, "E, ", X13, "E(r=Rc), ", X7,
      "E(r=R), ", X10, "Fn/Ei', ", X7, "E/Ei', ", X12, "Shd, ", X10, "G, ">);
   WRITE(F, <X8, "0.000, ", X7, "0, ", X14, "0, ", X14, "0, ", X15, " , ",
              X15, "0, ", X15, " , ", X15, " , ">);
  END;
 END;
```

```
%  ********************* MAIN *********************
%
READ(INVOER, /, SIGMA, Q, VREF, V0, NU, LAMBDA, A, CFLUX, DTAU, DIM, PLOT);
ACC:=@-5; ITMAX:=20;
IF SIGMA=0 THEN SIT:=3 ELSE SIT:=4; NUP:=NU+1;
%Check input, ABORT (and display error message) if inconsistent
IF A<0 OR A>25 THEN ABORT("CONSTFLUX : only A>=0 or A<=25 allowed");
IF NU<0 OR NU>2 THEN ABORT("CONSTFLUX : only NU=0, 1 or 2 allowed");
IF NU>0 AND (LAMBDA<0 OR LAMBDA>=1) THEN
 ABORT("CONSTFLUX : LAMBDA<0 or LAMBDA>=1");
IF VREF<0 THEN ABORT("CONSTFLUX : VREF<0");
IF VREF>V0 THEN ABORT("CONSTFLUX : VREF>V0");
IF SIGMA>1 OR SIGMA<0 THEN ABORT("CONSTFLUX : SIGMA out of range");
IF NU=0 AND Q NEQ 0 THEN
 BEGIN
  WRITE(OUT, <"CONSTFLUX: als NU=0 dan moet Q=0">);
  Q:=0;
 END;
IF CFLUX<=0 OR CFLUX>20 THEN
 ABORT("CONSTFLUX : FLUX not positive or too large");
IF DIM>200 THEN ABORT("CONSTFLUX : try DIM=200");
IF DIM<1   THEN ABORT("CONSTFLUX : DIM not positive");
N:=200;
IF A>=0.5 THEN N:=400;
IF A>1 THEN N:=800;
IF A>1.5 THEN N:=1600;
IF NU=0 THEN LAMBDA:=0;


BEGIN
 %
 REAL ARRAY UCF[0:DIM, 0:N],
            RETCF, TAUH, EH, ERCH, ERH, FNH, EDH, GH, XIH, SHDH, FLUXARR[0:DIM];

  SFAC:=(1-LAMBDA**NUP)*SIGMA*(V0-VREF)/
        (1+(1-LAMBDA**NUP)*SIGMA*V0);
  % UCF[0, 0:N] initialiseren op homogeen beginprofiel
  INDEX:=0; FCA0:=CFLUX; FLUXARR[INDEX]:=FCA0;
  FOR I:=0 TO N DO UCF[0, I]:=UCF[1, I]:=1;
  RETCF[0]:=1; RETH:=1;
  WHILE INDEX:=INDEX+1<=11 AND VARIFLUX(N, DTAU/10, NU, LAMBDA, A,
                                        SIGMA, VREF, V0, Q, FCA0, UCF[INDEX, *],
                                        RETCF[INDEX], ACC, ITMAX)
  DO
  BEGIN
   FOR I:=0 TO N DO UCF[INDEX+1, I]:=UCF[INDEX, I];
   FLUXARR[INDEX]:=FCA0/(1-SFAC*(1-RETCF[INDEX]))**(Q/NUP);
  END;
```

```
 FOR I:=0 TO N DO UCF[INDEX+1,I]:=UCF[INDEX-1,I];
 WHILE INDEX:=INDEX+1<DIM AND VARIFLUX(N,DTAU,NU,LAMBDA,A,
                                       SIGMA,VREF,V0,Q,FCA0,UCF[INDEX,*],
                                       RETCF[INDEX],ACC,ITMAX)
 DO
 BEGIN
  FOR I:=0 TO N DO UCF[INDEX+1,I]:=UCF[INDEX,I];
  FLUXARR[INDEX]:=FCA0/(1-SFAC*(1-RETCF[INDEX]))**(Q/NUP);
 END;
 INDEX:=INDEX-1;
END;
%
CASE NU OF BEGIN
 0 : XH:= NU+1;
 1 : XH:=(NU+1)/(1+LAMBDA);
 2 : XH:=(NU+1)/(1+LAMBDA+LAMBDA*LAMBDA)   END;
%
%
%
 CASE SIT OF
  BEGIN
   1 : SITUATIE:="%rvw constant,    r coordinates";
   2 : SITUATIE:="%rvw constant,   fi coordinates";
   3 : SITUATIE:="%flux constant,   r coordinates";
   4 : SITUATIE:="%flux constant,  fi coordinates";
  END;
%
KOPJECFTABEL(PRINT,1,A,FLUX,NU,N,LAMBDA,TAU,DTAU,
                        SIGMA,VSTER,VREF,V0,Q,COEF,
                        ITMAX,ACC,RETCF,FLUXARR);

KOPJECFTABEL(TABEL,2,A,FLUX,NU,N,LAMBDA,TAU,DTAU,
                        SIGMA,VSTER,VREF,V0,Q,COEF,
                        ITMAX,ACC,RETCF,FLUXARR);

KOPJETAPECF(B,3,SIT,SIGMA,A,CFLUX,NU,LAMBDA,
                       N,DTAU,INDEX,ITMAX,DIM);
KOPJETAPECF(TAPE,1,SIT,SIGMA,A,CFLUX,NU,LAMBDA,
                          N,DTAU,INDEX,ITMAX,DIM);
%
%
% berekening uitvoergegevens
% naar TAPE worden alle berekeningen weggeschreven
% naar B,PRINT en TABEL worden maximaal 2 A-4-tjes geschreven
% gegevens in TAPE en B zijn gescheiden door komma's
```

```
 FOR I:=2 TO 10 DO
 BEGIN
  TAU :=I*DTAU/10;
   E  :=1-RETCF[I];
   ERC:=1-((UCF[I-1,0]+2*UCF[I,0]+UCF[I+1,0])/4);   % smoothen
   ER :=1-((UCF[I-1,N]+2*UCF[I,N]+UCF[I+1,N])/4);   % smoothen
   FN :=FLUXARR[I]/ER;
   ED :=E/ER;
   SHD:=2*(A+1)*FLUXARR[I]/(RETCF[I]**(A+1)-(1-ER)**(A+1));
   IF SIGMA = 0 THEN
   BEGIN
    G  :=FN*ED/XH;
   END
   ELSE
   BEGIN
    VGEM:=VREF+(VO-VREF)*RETCF[I];
    XI :=XH*(LAMBDA**NUP+(1-LAMBDA**NUP)*(1+SIGMA*VGEM))**(NU/NUP);
    G  :=FN*ED/XI;
   END;


 WRITE(PRINT, <F18.6,4(F14.6)>, TAU, E, ERC, ER, FLUXARR[I]);
 WRITE(TABEL, <F18.6,4(F14.6)>, TAU, FN, ED, SHD, G);
 WRITE(B, <9(F13.6,",")>, TAU, E, ERC, ER, FN, ED, SHD, G, FLUXARR[I]);
 WRITE(TAPE, <9(F13.6,",")>, TAU, E, ERC, ER, FN, ED, SHD, G, FLUXARR[I]);
END;
I:=12;
 BEGIN
  TAU :=(I-10)*DTAU;
   E  :=1-RETCF[I];
   ERC:=1-((UCF[I-2,0]+2*UCF[I,0]+UCF[I+1,0])/4);
   ER :=1-((UCF[I-2,N]+2*UCF[I,N]+UCF[I+1,N])/4);
   FN :=FLUXARR[I]/ER;
   ED :=E/ER;
   SHD:=2*(A+1)*FLUXARR[I]/(RETCF[I]**(A+1)-(1-ER)**(A+1));
   IF SIGMA = 0 THEN
   BEGIN
    G  :=FN*ED/XH;
   END
   ELSE
   BEGIN
    VGEM:=VREF+(VO-VREF)*RETCF[I];
    XI :=XH*(LAMBDA**NUP+(1-LAMBDA**NUP)*(1+SIGMA*VGEM))**(NU/NUP);
    G  :=FN*ED/XI;
   END;


 WRITE(PRINT, <F18.6,4(F14.6)>, TAU, E, ERC, ER, FLUXARR[I]);
 WRITE(TABEL, <F18.6,4(F14.6)>, TAU, FN, ED, SHD, G);
 WRITE(B, <9(F13.6,",")>, TAU, E, ERC, ER, FN, ED, SHD, G, FLUXARR[I]);
 WRITE(TAPE, <9(F13.6,",")>, TAU, E, ERC, ER, FN, ED, SHD, G, FLUXARR[I]);
```

```
  FOR I:=13 TO MIN(INDEX-1,40) DO
   BEGIN
    TAU :=(I-10)*DTAU;
     E  :=1-RETCF[I];
     ERC:=1-((UCF[I-1,0]+2*UCF[I,0]+UCF[I+1,0])/4);
     ER :=1-((UCF[I-1,N]+2*UCF[I,N]+UCF[I+1,N])/4);
     FN :=FLUXARR[I]/ER;
     ED :=E/ER;
     SHD:=2*(A+1)*FLUXARR[I]/(RETCF[I]**(A+1)-(1-ER)**(A+1));
     IF SIGMA = 0 THEN
     BEGIN
      G  :=FN*ED/XH;
     END
     ELSE
     BEGIN
      VGEM:=VREF+(V0-VREF)*RETCF[I];
      XI :=XH*(LAMBDA**NUP+(1-LAMBDA**NUP)*(1+SIGMA*VGEM))**(NU/NUP);
      G  :=FN*ED/XI;
     END;


    WRITE(PRINT,<F18.6,4(F14.6)>,TAU,E,ERC,ER,FLUXARR[I]);
    WRITE(TABEL,<F18.6,4(F14.6)>,TAU,FN,ED,SHD,G);
    WRITE(B,<9(F13.6,",")>,TAU,E,ERC,ER,FN,ED,SHD,G,FLUXARR[I]);
    WRITE(TAPE,<9(F13.6,",")>,TAU,E,ERC,ER,FN,ED,SHD,G,FLUXARR[I]);
  END;

    WRITE(PRINT[SKIP(1)]);
    WRITE(TABEL[SKIP(1)]);

    IF INDEX-1>40
    THEN
    BEGIN
WRITE(PRINT,</,X13,"TAU",X11,"E",X14,"E(r=Rc)",X7,"E(r=R)",X7,"FCA0",/>);
WRITE(TABEL,</,X13,"TAU",X10,"Fn/Ei'",X8,
                    "E/Ei'",X9,"Shd",X12,"G",/>);
    END;

    IF (INDEX-41) MOD 60 =0
    THEN FACTOR:= (INDEX-41)/60
    ELSE FACTOR:= 1+ ENTIER((INDEX-41)/60);
      %
    FOR I:=MIN(INDEX-1,40) TO INDEX-1 DO
    BEGIN
     TAUH[I] :=(I-10)*DTAU;
     EH[I]  :=1-RETCF[I];
     ERCH[I]:=1-((UCF[I-1,0]+2*UCF[I,0]+UCF[I+1,0])/4);
     ERH[I] :=1-((UCF[I-1,N]+2*UCF[I,N]+UCF[I+1,N])/4);
     FNH[I] :=FLUXARR[I]/ERH[I];
     EDH[I] :=EH[I]/ERH[I];
     SHDH[I]:=2*(A+1)*FLUXARR[I]/(RETCF[I]**(A+1)-(1-ERH[I])**(A+1));
     IF SIGMA = 0 THEN
     BEGIN
      GH[I]  :=FNH[I]*EDH[I]/XH;
     END
     ELSE
     BEGIN
```

```
      XIH[I]:=XH*(LAMBDA**NUP+(1-LAMBDA**NUP)*(1+SIGMA*VGEM))**(NU/NUP);
      GH[I]  :=FNH[I]*EDH[I]/XIH[I];
     END;
    END;
     %waarde index-1 is 1 lager dan tot waar de profielen zijn berekend

     FOR I:=(MIN(INDEX-1,40)+1) TO INDEX-1 DO
     WRITE(TAPE,<9(F13.6,",")>,TAUH[I],EH[I],ERCH[I],ERH[I],
                                FNH[I],EDH[I],SHDH[I],GH[I],FLUXARR[I]);

   IF FACTOR >0 THEN
    BEGIN
     FOR I:=MIN(INDEX-1,40) STEP FACTOR UNTIL INDEX-1 DO
      BEGIN
WRITE(PRINT,<F18.6,4(F14.6)>,TAUH[I],EH[I],ERCH[I],ERH[I],FLUXARR[I]);
WRITE(TABEL,<F18.6,4(F14.6)>,TAUH[I],FNH[I],
                                EDH[I],SHDH[I],GH[I]);
WRITE(B,<9(F13.6,",")>,TAUH[I],EH[I],ERCH[I],ERH[I],
                          FNH[I],EDH[I],SHDH[I],GH[I],FLUXARR[I]);
      END;
    END;

LOCK(B,CRUNCH);LOCK(TAPE,CRUNCH);

    IF PLOT
    THEN
    BEGIN
     IF SIGMA=0 THEN
     BEGIN
      CFPROFILES(N,DTAU,INDEX,NU,LAMBDA,A,CFLUX,UCF,4,TRUE);
      CFPROFILES(N,DTAU,INDEX,NU,LAMBDA,A,CFLUX,UCF,6,TRUE);
     END
     ELSE
      CFPROFILESFI(N,DTAU,INDEX,NU,LAMBDA,A,SIGMA,VREF,V0,CFLUX,
                   UCF,RETCF,6,TRUE);
     END;
    END;
    %
 END.
```

APPENDIX E

listing van het programma
LEESNIETKRIMP

LEESNIETKRIMP

Programma om data van niet-krimpende situaties
met constante grensvlakconcentratie van TAPE
te kunnen lezen.

```
BEGIN
 % PROGRAMMA OM TAPE-FILES IN ALGOL-VERSIE TE LEZEN

 FILE IN(KIND=DISK, FILETYPE=7),
      TAPE(KIND=DISK, PROTECTION=SAVE, MAXRECSIZE=22, UNITS=WORDS),
      TEEPPASC(KIND=DISK, PROTECTION=SAVE, MAXRECSIZE=22, UNITS=WORDS);

 REAL LAMBDA, A, BV, SIGMA, DTAU, TAUNUL;
 INTEGER NU, ITMAX, INTVALS, N, INDEX, SIT, I;
 STRING SITUATIE;

 READ(IN, /, SIT);
 READ(IN, /, SIGMA, A, BV, NU, LAMBDA);
 READ(IN, /, N, TAUNUL, INTVALS, DTAU, INDEX, ITMAX);
 BEGIN
  %
  REAL ARRAY TAU, E, G, F, ERC, SHD[0:INDEX];
  %
  READ(IN, /, );READ(IN, /, );
  FOR I:=0 STEP 1 UNTIL INDEX DO
  READ(IN, /, TAU[I], E[I], F[I], G[I], ERC[I], SHD[I]);
  %
  %controle of er goed is ingelezen
  CASE SIT OF
  BEGIN
   1 : SITUATIE:="%rvw constant,    r coordinates";
   2 : SITUATIE:="%rvw constant,   fi coordinates";
   3 : SITUATIE:="%flux constant,   r coordinates";
   4 : SITUATIE:="%flux constant,  fi coordinates";
  END;
  %
  WRITE(TAPE, <I2, ", ", A30, ", ">, SIT, SITUATIE);
  WRITE(TAPE, <5(E21.9, ", "), "% SIGMA, A, BV, NU, LAMBDA">, SIGMA,
              A, BV, NU, LAMBDA);
  WRITE(TAPE, <I5, ", ", F10.5, ", ", I5, ", ", F10.5, ", ", I5, ", ", I5, ", ",
              "%N, TAU, INTVALS, DTAU, INDEX, ITMAX">,
               N, TAUNUL, INTVALS, DTAU, INDEX, ITMAX);
  WRITE(TAPE, <"%TAU, E, F, G, E(r=Rc), Shd">);
  FOR I:=0 STEP 1 UNTIL (INDEX+INTVALS) DO
   WRITE(TAPE, <6(E21.9, ", ")>, TAU[I], E[I], F[I], G[I], ERC[I], SHD[I]);
  LOCK(TAPE, CRUNCH);
  %
 END
END.
```

APPENDIX F

listing van het programma

LEESNKPROFIEL

LEESNKPROFIEL

Programma om het concentratieprofiel
voor situaties met nietkrimp met constante
grensvlakconcentratie en a=1.5van TAPE te
kunnen inlezen.

```
BEGIN
 % PROGRAMMA OM TAPE-FILES IN ALGOL-VERSIE TE LEZEN

 FILE IN(KIND=DISK, FILETYPE=7);

 REAL LAMBDA, A, BV, SIGMA, DTAU, TAUNUL;
 INTEGER NU, ITMAX, INTVALS, N, INDEX, SIT, I;
 STRING SITUATIE, ESTRING;

 READ(IN, /, SIT);
 READ(IN, /, SIGMA, A, BV, NU, LAMBDA);
 READ(IN, /, N, TAUNUL, INTVALS, DTAU, INDEX, ITMAX);
 READ(IN, /, ESTRING);
 BEGIN
  %
  REAL ARRAY U[0:N];
  %
  FOR I:=0 STEP 1 UNTIL N DO
  READ(IN, /, U[I]);
  %
  CASE SIT OF
  BEGIN
   1 : SITUATIE:="%rvw constant,    r coordinates";
   2 : SITUATIE:="%rvw constant,   fi coordinates";
   3 : SITUATIE:="%flux constant,   r coordinates";
   4 : SITUATIE:="%flux constant,  fi coordinates";
  END;
  %
 END
END.
```

APPENDIX G

listing van het programma LEESKRIMPFILE

LEESKRIMPFILE

Programma om data voor situaties met krimp en constante grensvlakconcentratie van TAPE te kunnen lezen.

```
BEGIN
 % PROGRAMMA OM TAPE-FILES IN ALGOL-VERSIE TE LEZEN

 FILE IN(KIND=DISK, FILETYPE=7),
      TAPE(KIND=DISK, PROTECTION=SAVE, MAXRECSIZE=22, UNITS=WORDS),
      TEEPPASC(KIND=DISK, PROTECTION=SAVE, MAXRECSIZE=22, UNITS=WORDS);

 REAL LAMBDA, A, BV, SIGMA, DTAU, TAUNUL, VREF, V0;
 INTEGER NU, DIM, INTVALS, N, INDEX, SIT, I;
 STRING SITUATIE;

 READ(IN, /, SIT);
 READ(IN, /, SIGMA, A, BV, NU, LAMBDA);
 READ(IN, /, N, TAUNUL, INTVALS, DTAU, INDEX, DIM);
 READ(IN, /, VREF, V0);

 BEGIN
  %
  REAL ARRAY TAU, E, G, F, ERC, SHD[0:INDEX];
  %
  READ(IN, /, );
  FOR I:=0 STEP 1 UNTIL INDEX DO
  READ(IN, /, TAU[I], E[I], F[I], G[I], ERC[I], SHD[I]);
  %
  %controle of er goed is ingelezen
  CASE SIT OF
  BEGIN
   1 : SITUATIE:="%rvw constant,    r coordinates";
   2 : SITUATIE:="%rvw constant,   fi coordinates";
   3 : SITUATIE:="%flux constant,   r coordinates";
   4 : SITUATIE:="%flux constant,  fi coordinates";
  END;
  %
  WRITE(TAPE, <I2, ", ", A30, ", ">, SIT, SITUATIE);
  WRITE(TAPE, <5(E21.9, ", "), "% SIGMA, A, BV, NU, LAMBDA">, SIGMA,
              A, BV, NU, LAMBDA);
  WRITE(TAPE, <I5, ", ", F10.5, ", ", I5, ", ", F10.5, ", ", I5, ", ", I5, ", ",
              "%N, TAU, INTVALS, DTAU, INDEX, ITMAX">,
               N, TAUNUL, INTVALS, DTAU, INDEX, ITMAX);
  WRITE(TAPE, <"%TAU, E, F, G, E(r=Rc), Shd">);
  FOR I:=0 STEP 1 UNTIL (INDEX+INTVALS) DO
   WRITE(TAPE, <6(E21.9, ", ")>, TAU[I], E[I], F[I], G[I], ER[I], SHD[I]);
  LOCK(TAPE, CRUNCH);
  %
 END
END.
```

APPENDIX H

listing van het programma

LEESNKCF

LEESNKCF

programma om data voor situaties met niet-krimp constante flux van TAPE te lezen

```
BEGIN
 % PROGRAMMA OM TAPE-FILES IN ALGOL-VERSIE TE LEZEN

 FILE IN(KIND=DISK, FILETYPE=7);

 REAL LAMBDA, A, SIGMA, Q, VREF, VO, CFLUX, DTAU;
 INTEGER NU, ITMAX, N, INDEX, SIT, I;
 STRING SITUATIE;

 READ(IN, /, SIT);
 READ(IN, /, SIGMA, Q, VREF, VO, A, CFLUX, NU, LAMBDA);
 READ(IN, /, N, DTAU, INDEX, ITMAX);
 BEGIN
  %
  REAL ARRAY TAU, E, ERC, ER, FNE, EEI, SHD, G, FLUX[0:INDEX];
  %
  READ(IN, /, );
  FOR I:=0 STEP 1 UNTIL INDEX DO
  READ(IN, /, TAU[I], E[I], ERC[I], ER[I], FNE[I], EEI[I], SHD[I],
               G[I], FLUX[I]);
  %
  CASE SIT OF
  BEGIN
   1 : SITUATIE:="%rvw constant,    r coordinates";
   2 : SITUATIE:="%rvw constant,   fi coordinates";
   3 : SITUATIE:="%flux constant,   r coordinates";
   4 : SITUATIE:="%flux constant,  fi coordinates";
  END;
  %
 END
END.
```

APPENDIX I

listing van het programma

LEESKCA

LEESKCA

programma om data voor situaties met (niet-)krimp
en constante flux van TAPE te lezen

```
BEGIN
 % PROGRAMMA OM TAPE-FILES IN ALGOL-VERSIE TE LEZEN

 FILE IN(KIND=DISK, FILETYPE=7);

 REAL LAMBDA, A, SIGMA, Q, VREF, VO, CFLUX, DTAU;
 INTEGER NU, ITMAX, N, INDEX, SIT, I;
 STRING SITUATIE;

 READ(IN, /, SIT);
 READ(IN, /, SIGMA, Q, VREF, VO, A, CFLUX, NU, LAMBDA);
 READ(IN, /, N, DTAU, INDEX, ITMAX);
 BEGIN
  %
  REAL ARRAY TAU, E, ERC, ER, FNE, EEI, SHD, G, FLUX[0:INDEX];
  %
  READ(IN, /, );
  FOR I:=0 STEP 1 UNTIL INDEX DO
  READ(IN, /, TAU[I], E[I], ERC[I], ER[I], FNE[I], EEI[I], SHD[I],
               G[I], FLUX[I]);
  %
  CASE SIT OF
  BEGIN
   1 : SITUATIE:="%rvw constant,    r coordinates";
   2 : SITUATIE:="%rvw constant,   fi coordinates";
   3 : SITUATIE:="%flux constant,   r coordinates";
   4 : SITUATIE:="%flux constant,  fi coordinates";
  END;
  %
 END
END.
```

APPENDIX J

Archivering van TAPE-files

De listings van alle datafiles op TAPE bevinden zich in het archief van ir. W. J. Coumans.