

Instituut voor Cultuurtechniek en Waterhuishouding
Wageningen

ASPECTEN VAN INFORMATIEVERWERKING

2

OPTIMALISEREN VAN PARAMETERS

Het gereedmaken van een functie
voor toepassing in NLV

dr. Ph.Th. Stol

Nota's van het Instituut zijn in principe interne communicatiemiddelen, dus geen officiële publikaties.

Hun inhoud varieert sterk en kan zowel betrekking hebben op een eenvoudige weergave van cijferreeksen, als op een concluderende discussie van onderzoeksresultaten. In de meeste gevallen zullen de conclusies echter van voorlopige aard zijn omdat het onderzoek nog niet is afgesloten.

Bepaalde nota's komen niet voor verspreiding buiten het Instituut in aanmerking

YSN 184107 - 01

A S P E C T E N V A N I N F O R M A T I E V E R W E R K I N G

Deel	Titel	Auteur	Nota	Datum
1	Computerverwerking van lange reeksen getallen	J.B.H.M. van Gils	935	nov. 1976
2	Optimaliseren van parameters: Het gereedmaken van een functie voor toepassing in NLV	Ph.Th. Stol	943	febr. 1977
3	Registratieverwerking voor automatische pF-bepalingen	J.B.H.M. van Gils	949	febr. 1977
4	Het systematisch bepalen van de afgeleiden van een functie ten behoeve van hun programmering	Ph.Th. Stol	948	febr. 1977

De nota's handelende over Aspecten van Informatieverwerking bevatten inlichtingen over de ontwikkeling van de informatieverwerking binnen het Instituut. Naast meer concluderende en toelichtende beschouwingen zal aandacht worden besteed aan het gebruik van programma's en programmapakketten en zullen zakelijke inlichtingen over praktijkervaring met en toepassing van de informatieverwerking worden gegeven

I N H O U D

	Blz.
I. INLEIDING	1
II. TERMINOLOGIE	2
III. APPARATUUR	3
IV. PROGRAMMA-OPBOUW	3
V. TOELICHTING	5
VI. SAMENHANG TUSSEN DE SUBROUTINES	5
VII. BENODIGDE INFORMATIE	6
VIII. CHECK-LIST	6
IX. TOELICHTING	7
1. Projectidentificatie	7
2. Onderwerpnaam	8
3. Verkorte onderwerpnaam	8
4. COMMON-code	8
4.1. Het aantal besturingsparameters	9
4.2. Het aantal te optimaliseren parameters	11
4.3. Het aantal voorkomende variabelen	11
4.4. Het aantal gegevens	12
4.5. Het gebruik van tweede afgeleiden	12
5. De functie	13
6. Indicering van variabelen en parameters	13
7. De afgeleiden	14
8. Betekenis van de variabelen	15
9. Betekenis van de parameters	15
10. Naamgeving parameters	15

	Blz.
11. Grenzen parameterwaarden	11
12. Parameter beginschattingen	16
13. Identificatie van de gegevens	16
14. Toe te passen gegevens	
14.1. Orde van grootte	17
14.2. Aantal decimalen	17
X. SLOTOPMERKINGEN	18
XI. REFERENTIES	19
BIJLAGE 1. ENKELE AANWIJZINGEN TEN BEHOEVE VAN DE PROGRAMMERING	20
1. Algemeen	20
2. Projectidentificatie	20
3. COMMON-code	20
4. Indicering van variabelen en parameters	20
5. Naamgeving parameters	22
6. Identificatie van de gegevens	22
BIJLAGE 2. HET VASTSTELLEN VAN DE COMMON-CODE	24

I. INLEIDING

Voor het optimaliseren van de parameters in een gegeven functie kan gebruik worden gemaakt van het door STOL (1975) ontworpen programma NLV (Niet-Lineaire Vereffening).

Het programma NLV bestaat uit een aantal subroutines waarmee de parameter-optimalisatie plaats vindt. De subroutines worden geactiveerd door het hoofdprogramma. Dit laatste biedt de mogelijkheid op eenvoudige wijze alternatieve rekentechnieken toe te passen en het rekenproces te modificeren door variaties aan te brengen in de wijze waarop de subroutines worden bestuurd.

Om een nieuwe functie, of een nieuw mathematisch model, zo gemakkelijk mogelijk op de juiste plaats in de subroutines op te nemen is een systeem ontwikkeld waarbij het grootste deel van het werk door de computer zelf wordt verricht. De noodzakelijke voorbereidende en met de hand uit te voeren werkzaamheden zijn tot een minimum gereduceerd.

In deze nota zal in korte toelichtingen worden aangegeven welke informatie de Afdeling Wiskunde ter beschikking moet hebben om een nieuwe functie in de subroutines van NLV te kunnen inbouwen.

Een belangrijk deel van deze informatie is vereist voor de organisatie en administratie van het werk. Voor de organisatie van de programma-opbouw met een nieuwe functie is het nodig dat de omvang van het probleem duidelijk is vastgelegd. Ten einde in de output het aangeboden probleem van het juiste commentaar te kunnen voorzien dienen enige administratieve voorzieningen te worden getroffen. Ook hieraan zal in het volgende aandacht worden besteed.

In een Bijlage worden enkele praktische aanwijzingen gegeven die samenhangen met het programmeren van de hierbeschreven onderdelen.

II. TERMINOLOGIE

Degene die een nieuwe functie ter verwerking aanbiedt zal met 'de gebruiker van NLV' worden aangeduid of kortweg met 'de gebruiker'.

Waar in het volgende gesproken wordt over 'interne organisatie' wordt bedoeld op de wijze waarop door de computer de aangeboden informatie wordt verwerkt en de wijze waarop de computerjobs hiervoor moeten worden samengesteld.

De functie welke in behandeling zal worden genomen zal, afhankelijk van het zinsverband worden aangeduid met 'nieuwe functie', 'gegeven of specifieke functie' respectievelijk met 'aangeboden functie'.

Het pakket van subroutines van NLV dat de eigenlijke optimaliseringsalgoritmen bevat en waarin nog geen specifieke functie is opgenomen, wordt genoemd het basisprogramma NLV.

Het opnemen of inbouwen van een nieuwe functie en zijn administratie in de reeds aanwezige subroutines van NLV (het basisprogramma) en het verzorgen van de interne organisatie zal worden aangeduid met 'het samenstellen van een nieuw programma NLV'. De job-opbouw hiervoor zal te zijner tijd in een afzonderlijke nota worden toegelicht.

Van een gegeven functie kan gezegd worden dat de parameters moeten worden geoptimaliseerd, respectievelijk dat de functie aan waarnemingsuitkomsten moet worden aangepast. Ook kan worden gezegd dat afwijkingen van gemeten functiewaarden ten opzichte van berekende waarden onderling moeten worden vereffend. In de terminologie over de met het programma NLV uit te voeren rekentechnieken zal hier geen onderscheid tussen deze drie aspecten worden gemaakt. Afhankelijk van het zinsverband zal de uit te voeren bewerking daarom worden aangeduid met 'optimalisatie', 'aanpassing' respectievelijk 'vereffening', of meer algemeen met 'rekenproces'.

Waar in het volgende gesproken wordt over eerste en tweede afgeleiden, worden de partiële afgeleiden van de gegeven functie naar elk van de parameters bedoeld. Worden in de functie k te optimaliseren

parameters onderscheiden, dan zijn er k eerste afgeleiden. Ieder van de afgeleiden kan weer naar elk van de parameters worden gedifferentieerd zodat er in het algemeen $\frac{(k+1)k}{2}$ verschillende tweede afgeleiden zijn.

Er worden verschillende karaktersets of tekensets onderscheiden. Te weten sets bevattende:

Alfanumerieke tekens: A t/m Z, 0, 1, ..., 9

Bijzondere tekens : + - * / () \$ =

Hollerith tekens : alle op de regeldrukker voorkomende tekens

III. APPARATUUR

Het programma NLV wordt gedraaid op computers van Control Data te weten de Cyber van IWIS-TNO, den Haag, bereikbaar via de terminal van IWIS-TNO, Wageningen en de CDC-6600 van het ECN, Petten, bereikbaar via de terminal van het Technisch en Wetenschappelijk Rekencentrum van de CD, Utrecht.

IV. PROGRAMMA-OPBOUW

Ten einde de gebruiker van NLV enig inzicht in de totale programma-opbouw te geven zal in het kort worden geschetst welke subroutines er zijn en wat hun belangrijkste taak in het gehele rekenproces is. Voor alle details wordt verwezen naar STOL (1975, pag. 136 e.v.). De subroutines zijn in drie groepen ingedeeld.

Subroutines van de eerste groep genaamd FUNCTIE

INITL	Het definieren van beginwaarden voor de besturing van NLV. Het printen van administratieve toelichtingen
READ	Het inlezen van de benodigde gegevens (data)
FNCTN	Het berekenen van de functievoorwaarden

DFDA Het berekenen van de eerste afgeleiden

D2FDA Het berekenen van de tweede afgeleiden

Subroutines van de tweede groep genaamd ROUTINE

NRMEQ Het opstellen van de normaalvergelijkingen

SOLVE Het oplossen van de normaalvergelijkingen

HOWA Het onderzoek naar het verloop van het rekenproces aan het begin van een nieuwe cyclus

MIN Het bepalen van de minimum kwadraatsom (in een punt genaamd B) in een gegeven zoekrichting

AISB Het overzetten van de resultaten in punt B, het minimum in een gegeven richting, naar punt A als beginpunt van een nieuwe cyclus

LISTING Het printen van de eindresultaten

Subroutines van de derde groep genaamd SPECIAL

BLOCK Het indelen van de parameters in alternatieve groepen

BACK Het toepassen van de methode van het terugprojecteren

TRACK Het onderzoek naar de afgelegde paden op het vereffeningsoppervlak

LIHYPEX Het toepassen van lineaire, hyperbolische en exponentiële extrapolatie van tussenresultaten

CORGRAF Een correlatieberekening langs de grafiek van de functie

V. TOELICHTING

Alle aan een nieuwe functie gebonden programma-onderdelen vinden hun plaats in de subroutines van de eerste groep, genaamd FUNCTIE. In feite vormen ze daar het, vaak relatief kleine, gedeelte van dat probleemgebonden is en afzonderlijk geprogrammeerd moet worden. Deze probleemgebonden programma-onderdelen worden door de computer automatisch op de juiste plaats in de onderscheiden subroutines ingevoegd met behulp van het voor de CDC-computers beschikbare UPDATE-systeem dat mede voor een dergelijke manipulatie met programma-onderdelen ontworpen is.

De meest gebruikelijke optimalisatietechnieken met enkele nieuwe modificaties zijn opgenomen in de subroutines van de tweede groep genaamd ROUTINE. Aangezien het rekenproces niet direct van de vorm van de aangeboden functie afhangt, behoeven deze subroutines niet aan een nieuwe functie te worden aangepast.

Enkele bijzondere technieken zijn verzameld in de subroutines van de derde groep genaamd SPECIAL. Ook deze subroutines behoeven niet aan een nieuwe functie te worden aangepast. Bij omvangrijke problemen, bijvoorbeeld functies met meer dan 10 variabelen en tevens meer dan 10 parameters en meer dan 500 gegevens, kan men besluiten de subroutines uit deze laatste groep niet te laden om op geheugenruimte te bezuinigen.

VI. SAMENHANG TUSSEN DE SUBROUTINES

De overdracht van berekende waarden van onderdelen van het rekenproces van de ene subroutine naar de andere en naar het hoofdprogramma vindt plaats door middel van variabelen die in zogenoemde COMMON-statements zijn opgenomen. Met de COMMON-statements wordt de ruimte, gereserveerd voor deze variabelen, georganiseerd en een vaste plaats gegeven in het geheugen van de computer. In elk van de subroutines moet hiervoor dezelfde ruimte worden gereserveerd. Aangezien deze ruimte mede afhangt van het aantal variabelen en het aantal parameters

in de aangeboden functie en van het aantal ter beschikking staande gegevens, is het duidelijk dat in elk van de subroutines de te reserveren ruimte bepaald wordt door de omvang van het te verwerken probleem. Alle subroutines moeten hieraan worden aangepast. Deze aanpassing gebeurt geheel automatisch met het reeds genoemde UPDATE-systeem wanneer de later te bespreken COMMON-code eenmaal is vastgesteld.

VII. BENODIGDE INFORMATIE

In deze nota zal niet verder op de technische aspecten van het samenstellen van een nieuw programma NLV voor een gegeven functie worden ingegaan. Daarentegen zal een check-list worden gegeven aan de hand waarvan een ieder die parameters in een functie of mathematisch model wil laten optimaliseren kan nagaan of de informatie waarover de Afdeling Wiskunde moet beschikken om dit werk ter hand te nemen, compleet is. Hierbij moet ook aan de naamgeving van de diverse onderdelen aandacht worden besteed. Het is namelijk wenselijk dat de gebruiker hieraan zelf een bijdrage levert teneinde in de output de relevante onderdelen gemakkelijk te kunnen herkennen.

VIII. CHECK-LIST

- . Projectidentificatie
- . Onderwerpnaam
- . Verkorte onderwerpnaam
- . Aantal besturingsparameters
- . Aantal parameters
- . Aantal variabelen
- . Aantal gegevens
- . Functie
- . Indicering
- . Betekenis variabelen
- . Betekenis parameters
- . Namen van parameters

- . Grenzen van parameterwaarden
- . Beginschattingen van de parameters
- . Identificatie van de gegevens
- . Orde van grootte van de gegevens
- . Aantal decimalen van de gegevens

IX. TOELICHTING

In deze paragraaf wordt een toelichting op de check-list gegeven. Volledigheidshalve zullen enkele onderwerpen behandeld worden die niet op de lijst voorkomen maar toch met de samenstelling van een nieuw programma NLV samenhangen. Veelal hoeft de gebruiker zelf hier geen actie te ondernemen.

In de toelichting op de check-list worden enkele verzonden voorbeelden gegeven om, waar nodig, de gang van zaken te verduidelijken.

1. P r o j e c t i d e n t i f i c a t i e

Voor het identificeren van het onderwerp waarvoor de berekeningen zullen worden uitgevoerd dienen de volgende gegevens beschikbaar te zijn.

- Een korte omschrijving van het doel waarvoor de berekeningen worden uitgevoerd. Deze omschrijving dient als algemene informatie over het onderwerp en wordt niet in het programma opgenomen.
- Een korte omschrijving ter herkenning van het onderwerp. Deze omschrijving komt op de eerste regel output. De gebruiker van NLV heeft maximaal 50 Hollerith posities beschikbaar.
- Het ICW-projectnummer, eveneens op te nemen in de 50 beschikbare posities.

Voorbeeld (47 posities):

REKENMODEL NO 3 (VERSIE 12/12/76) PROJECT 99.99

2. O n d e r w e r p n a a m

De voor een nieuwe functie te programmeren programma-onderdelen worden opgenomen in een apart kaartdek waaraan voor de interne organisatie een naam moet worden gegeven. Een onderwerpnaam mag uit maximaal 9 alfanumerieke en bijzondere tekens bestaan. De naam mag met een van de bijzondere tekens beginnen.

De naam mag vrij worden gekozen. Een voorbeeld is:

MY-MODEL3

Het gebruik van minder algemene namen, respectievelijk afkortingen verdient overigens de voorkeur

CUMVOCHT LUV0/SF6

voor respectievelijk 'cumulatieve vochtgehalten' en luchtverontreinigingsproef met SF₆' (LASEUR en POODT, 1975).

3. V e r k o r t e o n d e r w e r p n a a m

Voor de interne file organisatie van de job mogen slechts namen worden gebruikt die uit zeven tekens of minder bestaan. Ten einde enige aansluiting te verkrijgen bij de algemene wijze waarop de namen van de files van NLV worden gekozen is het wenselijk een naam van vier alfanumerieke tekens te verzinnen. Voorbeelden:

MOD3 CUVO LSF6

4. C O M M O N - c o d e

De omvang van een probleem dient in code opgegeven te worden ten einde de vereiste ruimte in de computer te kunnen reserveren. De hiervoor benodigde COMMON statements worden met een aangepaste versie van een door VAN GILS (1974) ontworpen programma automatisch uit de code verkregen. De code wordt door de afdeling Wiskunde samengesteld op grond van door de gebruiker opgegeven gegevens (zie Bijlage 2). De code bevat informatie over de volgende onderdelen.

4.1. Het aantal besturingsparameters

Besturingsparameters dienen om vanuit het hoofdprogramma alternatieve mogelijkheden in de mathematische formulering te kunnen kiezen. Veelal zullen deze besturingsparameters niet worden gebruikt, doch bij weinig verschillende alternatieve formuleringen van een gesteld probleem kan het efficiënt zijn deze tegelijk te programmeren en met besturingsparameters achteraf een keuze te doen. Vanuit het hoofdprogramma kan dan met de integers IPAR(I) de gewenste keuze worden gedaan.

Met een aantal voorbeelden zal het gebruik van de besturingsparameters worden verduidelijkt.

a. toevoeging van variabelen

Voorbeeld:

- Indien IPAR(1)=0 moet worden gebruikt

$$y = ae^{-bx} + \sin ct$$

- Indien IPAR(1)=1 moet worden gebruikt

$$y = ae^{-bx} + x \sin ct$$

Deze beide mogelijkheden kunnen tegelijk worden ingebouwd en wel als volgt.

- Indien IPAR(1)=0 definieer $z = 1$
- Indien IPAR(1)=1 definieer $z = x$
- Schrijf de formule als volgt:

$$y = ae^{-bx} + z \sin ct$$

b. Afhankelijkheid tussen parameters

Een ander voorbeeld van het gebruik van besturingsparameters is wanneer men een afhankelijkheid tussen de parameters in het model wil inbouwen. Een voorbeeld is:

Model:

$$y = ae^{-bx} + \sin ct$$

Alternatief:

$$y = ae^{-bx} + \sin 2at$$

waarin gesteld is $c=2a$. De eerste afgeleiden naar de parameters a en c zijn nu:

Model:

$$\frac{\partial y}{\partial a} = e^{-bx}$$

$$\frac{\partial y}{\partial c} = t \cos ct$$

Alternatief:

$$\frac{\partial y}{\partial a} = e^{-bx} + 2t \cos 2at$$

c ontbreekt

Vat men echter c op als een functie van a dan worden de afgeleiden naar a en c als volgt:

Model en alternatief:

$$\frac{\partial y}{\partial c} = t \cos ct$$

$$\frac{\partial y}{\partial a} = e^{-bx} + \frac{\partial y}{\partial c} \cdot \frac{dc}{da}$$

Tussen beide mogelijkheden kan nu als volgt onderscheid worden gemaakt. Bijvoorbeeld:

- Indien $IPAR(2)=0$ definieer $\frac{dc}{da} = 0$

- Indien $IPAR(2)=1$ definieer $c = 2a$

en $\frac{dc}{da} = 2$

en optimaliseer slechts a en b

c. Output tussenresultaten

Hoewel in NLV zelf reeds is voorzien in het al of niet uitschrijven van tussenresultaten ten einde het gehele rekenproces indien ge-

wenst te kunnen volgen kan het ook wenselijk zijn in de testfase tussenresultaten van de nieuwe functie te laten uitprinten. Ten einde de mogelijkheid te hebben dit al of niet laten uitprinten vanuit het hoofdprogramma te besturen kan een besturingsparameter worden gedefinieerd. Bijvoorbeeld:

- Indien $IPAR(3)=0$ sla uitschrijven tussenresultaten over
- Indien $IPAR(3)=1$ schrijf tussenresultaten uit

Worden alle drie de bovenbeschreven situaties toegepast, dan kunnen hiervoor drie besturingsparameters worden gedefinieerd.

In de regel zullen deze besturingsparameters niet worden gebruikt. Hun aantal wordt dan gelijk aan 1 gesteld.

4.2. Het aantal te optimaliseren parameters

Het verdient aanbeveling de mathematische formulering van een probleem zo algemeen mogelijk te houden. Zou men besluiten achteraf een parameter aan het model toe te voegen dan moet een deel van de programmeerarbeid opnieuw worden gedaan. Daarentegen is het in het hoofdprogramma op eenvoudige wijze mogelijk elke parameter of elke combinatie van parameters constant te houden, zodat een algemene eerste opzet de voorkeur heeft.

Aangezien dus uitbreiding van het model met meer parameters het toepassen van een meer algemene functie betekent en daarmee het programmeren van een geheel nieuw probleem, behoeft geen reserve ruimte voor het aantal parameters in rekening te worden gebracht. Er kan mee worden volstaan het exacte aantal parameters op te geven.

4.3. Het aantal voorkomende variabelen

Het aantal variabelen dat in een model voorkomt is medebepalend voor de omvang van een probleem. Bij de bepaling van het aantal variabelen worden zowel afhankelijke als onafhankelijke variabelen meegeteld.

Aangezien men niet zonder meer de gegeven functie met nieuwe variabelen kan uitbreiden zonder een geheel nieuw programma te moeten samenstellen behoeft ook hier geen reserve ruimte in rekening gebracht te worden. Er kan dus mee volstaan worden het exacte aantal variabelen op te geven.

4.4. Het aantal gegevens

Wat betreft het aantal gegevens waarmee de parameters zullen worden geoptimaliseerd geldt dat men hier wél rekening moet houden met het aantal aanvullende gegevens dat eventueel later zal worden toegevoegd; met het aantal gegevens dat ontstaat wanneer men verschillende reeksen gegevens wil samenvoegen, en dergelijke. Overschrijdt het aantal te gebruiken gegevens het opgegeven aantal waarmee het programma NLV is samengesteld, dan zullen nieuwe COMMON-statements moeten worden samengesteld en moeten alle subroutines opnieuw worden vertaald. Een veilige, overigens reële, reserve verdient dus aanbeveling bij het vaststellen van het aantal gegevens.

Opgemerkt wordt nog dat tijdens het toepassen van het programma NLV het werkelijke aantal aangeboden respectievelijk te gebruiken gegevens op twee manieren achteraf kan worden vastgesteld.

. De eerste manier, welke standaard is ingebouwd, is dat het werkelijke aanbod van gegevens (aantal kaarten), ook wanneer dit minder is dan werd opgegeven in de COMMON-code, automatisch wordt geteld in subroutine READ.

. De tweede manier is dat het te gebruiken aantal gegevens wordt gedefinieerd in het hoofdprogramma. Bijvoorbeeld:

Stel dat men als maximaal aantal gegevens opgeeft in de COMMON-code $n = 75$. (Dit betekent voor alle variabelen 75 waarnemingen). Worden vooreerst slechts 60 gegevens (kaarten) aangeboden dan wordt dit automatisch geconstateerd in subroutine READ. Wil men van deze 60 gegevens slechts de eerste opeenvolgende 45 gegevens gebruiken, dan wordt in het hoofdprogramma gedefinieerd $NDATA = 45$.

Voor het verwerken van meer dan 75 gegevens moet meer ruimte in het geheugen van de computer worden gereserveerd en moeten alle subroutines opnieuw worden vertaald.

4.5. Het gebruik van tweede afgeleiden

In de COMMON-code moet worden opgegeven of tweede afgeleiden zullen worden gebruikt. Aangezien het gebruik van tweede afgeleiden niet direct noodzakelijk is zal de Afdeling Wiskunde afhankelijk van de

gecompliceerdheid en de omvang van het probleem besluiten over het al of niet opnemen van tweede afgeleiden.

5. D e f u n c t i e

Vanzelfsprekend neemt de nieuwe functie een centrale plaats in in het aangeboden probleem. Toch behoeft bij de samenstelling van een nieuw programma NLV relatief weinig aandacht aan de functie zelf besteed te worden.

De aan te passen functie of functies worden door de gebruiker opgegeven als mathematisch model in zijn onderzoek. Is de nieuwe functie in zijn definitieve vorm beschikbaar gekomen dan kan deze geprogrammeerd worden.

Waar het het samenstellen van een nieuw programma NLV betreft wordt opgemerkt dat de vorm van de functie in principe aan slechts weinig restricties gebonden is (STOL, 1975, Hfdst. 4). De functie moet differentieerbaar zijn naar de te optimaliseren parameters om voldoende inzicht in het verloop van het rekenproces en de betekenis van het eindresultaat te verkrijgen. Alhoewel in NLV procedures beschikbaar zijn die zonder het gebruik van eerste afgeleiden werken, moet toepassing ervan om bovengenoemde reden van essentieel belang worden geacht.

A priori kan niet worden aangegeven of convergentie naar eindwaarden voor de parameters zal optreden. Dit kan zowel van de functie, van de wijze waarop de parameters in de functie voorkomen, als van de aard en het aantal van de gegevens afhangen zodat de steeds bestaande onzekerheid over te bereiken convergentie op zichzelf geen reden behoeft te zijn om een optimaliseringsprocedure niet toe te passen. Eerst na enige 'numerieke ervaring' met een gegeven probleem te hebben opgedaan, ontstaat verder inzicht in het gedrag van de functie bij het vereffenen.

6. I n d i c e r i n g v a n v a r i a b e l e n e n p a r a m e t e r s

Voor de interne organisatie wordt een indicering van de variabelen en van de parameters toegepast. Voor de gebruiker betekent dit

dat hij de herkenbaarheid van de output van NLV voor zijn eigen specifieke probleem kan vergroten door de volgorde waarin de variabelen en parameters moeten worden geïndiceerd zelf aan te geven. Zonder verdere suggesties van de gebruiker ligt het toepassen van een nummering van links naar rechts in de formule het meest voor de hand. De gebruiker zou echter redenen kunnen hebben een andere volgorde te prefereren.

Ter vermijding van het schrijven van indices in de formules zou men een alfabetische schrijfwijze kunnen toepassen om een gewenste volgorde van indiceren aan te geven. De enige restrictie is dat de gemeten waarde van de onafhankelijk variabele het indexcijfer 1 moet krijgen. Voorbeeld:

Model:

$$z = ae^{-kx} + \sin(t+\phi)$$

Dit kan alfabetisch geschreven worden als:

$$y = ae^{-cu} + \sin(v+b)$$

en wordt voor de programmering opgevat als de functie:

$$y = c_1 e^{-c_3 x_2} + \sin(x_3 + c_2)$$

7. D e a f g e l e i d e n

Het is niet strikt noodzakelijk dat de gebruiker van NLV zelf de eerste en eventueel tweede afgeleiden van de gegeven functie naar alle parameters bepaalt. In de meeste gevallen zal het programmeren van de afgeleiden zo nauw samenhangen met het programmeren van de functie zelf dat dit werk het meest efficiënt tegelijkertijd met de voorbereidende werkzaamheden voor het samenstellen van een nieuw programma NLV kan plaatsvinden. Voor het systematisch bepalen van afgeleiden ten behoeve van hun programmering werden suggesties gedaan door STOL (1977).

8. B e t e k e n i s v a n d e v a r i a b e l e n

In de output van het programma NLV kan vermeld worden wat de betekenis van de variabelen is; in welke eenheden gemeten is, enz. Deze vermeldingen worden gegeven als Hollerith tekst en zijn dus niet aan restricties wat betreft de toegestane tekens en de lengte van de mededeling gebonden. Voorbeeld:

X(3)=T, (TIJD IN DAGEN)

waarin de index voor het rangnummer van de gegevens is weggelaten

9. B e t e k e n i s v a n d e p a r a m e t e r s

Op dezelfde wijze als voor de betekenis van de variabelen kan in de output een vermelding omtrent de betekenis van de parameters worden opgenomen. Voorbeeld:

A(2)= PHI = (FASE-VERSCHUIVING)

10. N a a m g e v i n g p a r a m e t e r s

Ten einde in het rekenproces de rol die elk van de parameters daarin speelt duidelijk te kunnen volgen, worden in de output de parameters met een korte naam aangeduid. Een parameternaam mag elk beschikbaar Hollerith teken bevatten met een maximum van 10. Deze restricties inachtnemend kan ook een korte omschrijving als naam worden toegepast. Voorbeelden:

voor a_1 : FACTOR
voor a_2 : PHI
voor a_3 : EXP(-B*U)

11. G r e n z e n p a r a m e t e r w a a r d e n

Parameters in een mathematisch model hebben soms de eigenschap dat hun waardenbereik beperkt is. Zo kan gelden dat alleen positieve waarden mogen (of kunnen) voorkomen of dat de waarde tussen 0 en 1

ligt of tussen 0 en 100 (procent) bijvoorbeeld. Tijdens het rekenproces kan het voorkomen dat parameters tengevolge van toe te passen correcties de beginwaarden (tijdelijk) een niet mogelijke waarde zouden moeten aannemen. Het programma NLV voorziet in deze moeilijkheid door die parameter in de omgeving van de grens van de toegestane waarden tijdelijk constant te houden.

Van de gebruiker wordt gevraagd van elke parameter op te geven of er grenzen aan het waardenbereik zijn, en zo ja wat de uiterste waarden van het toegestane waardenbereik is.

12. P a r a m e t e r b e g i n s c h a t t i n g e n

Van alle parameters die in de gegeven functie voorkomen dient een geschatte beste waarde te worden opgegeven. Deze waarden zijn nodig voor het op gang brengen van het iteratieve rekenproces. Dit proces verloopt het meest effectief wanneer deze zogenoemde beginschattingen in de buurt van de optimale waarden liggen. Een zorgvuldige keuze van de beginschattingen, het liefst gebaseerd op vroegere ervaring of berekeningen is derhalve een eerste voorwaarde voor het slagen van de vereffening.

13. I d e n t i f i c a t i e v a n d e g e g e v e n s

Voor het identificeren van de toegepaste gegevens staan 80 Hollerith tekens ter beschikking. De informatie op deze wijze aan de gegevens meegegeven wordt in de output boven de - eventueel verkorte - lijst van ingelezen gegevens geprint. Voorbeeld (45 tekens):

EERSTE SERIE (AUG 1976) PROEFGEBIED N.HOLLAND

14. T o e t e p a s s e n g e g e v e n s

Hoewel bij de werkelijke uitvoering van het rekenproces voor alle variabelen waarnemingsuitkomsten ter beschikking moeten staan is het niet noodzakelijk bij de voorbereiding van een nieuw programma NLV reeds over alle waarnemingen te beschikken. Wel moeten in een vroeg stadium ten behoeve van de programmering van de input en de output de volgende details ter beschikking staan:

14.1. Orde van grootte

Van elk van de variabelen moet de orde van grootte van de waarnemingsuitkomsten opgegeven worden. Aan de hand hiervan kan een kaartindeling worden ontworpen voor het ponsen en later weer voor het programmeren van het inlezen van de gegevens.

Ook wanneer de gegevens reeds op andere wijze zijn vastgelegd (tape) dient bekend te zijn wat de orde van grootte van de getallen is, en hoe de gegevens naar de verschillende variabelen zijn ingedeeld.

14.2. Aantal decimalen

Nauw verbonden met het voorgaande is de noodzakelijkheid kennis te hebben van het aantal decimalen dat in de waarnemingsuitkomsten voorkomt. De orde van grootte en het aantal decimalen bepalen samen met welk format de gegevens dienen te worden ingelezen.

Opgemerkt wordt nog dat het mogelijk is tijdens het inlezen het juiste aantal decimalen te verwezenlijken ook al stonden deze niet op de ponskaart vermeld. Men kan dus eenvoudigheidshalve besluiten slechts integer getallen te laten ponsen. Wordt opgegeven wat het aantal decimalen is, dan kan hiermede bij het programmeren rekening worden gehouden. Voorbeeld:

Gegeven:

- . De waarnemingen voor 3 variabelen zijn opgegeven als integers.
- . Aantal decimalen respectievelijk 2, 3 en 0.
- . Stel dat op ponskaart vermeld wordt:

351	2718	12
60	3145	1
2	29167	6

Na inlezen in de computer, rekening houdend met de opgegeven kompositie, wordt dit

3.51	2.718	12.0
0.60	3.145	1.0
0.02	29.167	6.0

De kommapositie opgegeven in de leesinstructie in het programma overheerst een eventuele komma op de ponskaart.

Op basis van deze informatie en de onder 5 genoemde volgorde van benoeming van de variabelen kan in gezamenlijk overleg tot een ponskaartindeling worden besloten.

Het overbrengen van waarnemingsuitkomsten op ponsdocument alvorens de gegevens worden gepost verdient over het algemeen aanbeveling. Alleen als het aantal variabelen gering is (niet meer dan drie bijvoorbeeld) en het aantal gegevens beperkt, kan soms met een eenvoudige tabel als aanbod van gegevens worden volstaan. Overleg met de Afdeling Wiskunde over de beste gang van zaken is noodzakelijk om eenheid in gegevensaanbod en programma-instructies te bewerkstelligen.

X. SLOTOPMERKINGEN

De aanwijzingen in deze nota zijn voornamelijk gericht op de toepassing van het rekenproces in het programma NLV. De aanwijzingen zijn gegeven los van het probleem van het programmeren van de onderdelen en het samenstellen van het programma zelf. Hoewel het toepassen van NLV reeds in een vergevorderd definitief stadium is gekomen zullen onvermijdelijk nog aanvullingen en verbeteringen aan het basisprogramma worden aangebracht. De door de gebruiker noodzakelijkerwijs op te geven informatie over zijn te vereffenen functie zal hierdoor overigens geen wijziging ondergaan.

XI. REFERENTIES

- GILS, J.B.H.M. VAN, (1974). ICW-programma 100, PROGRAM PONS: Het maken van het standaarddeck voor het samenstellen van een nieuw programma (NLV, project 410.1).
- LASEUR, J. en J.B.G. POODT, (1975). Verspreidingsonderzoek met de tracer SF_6 . Verslag V-44, Landbouwhogeschool Vakgroep Luchtverontreiniging, Wageningen.
- STOL, Ph.Th., (1975). A Contribution to Theory and Practice of Non-linear Parameter Optimization. Pudoc Wageningen, pp. 197.
- (1977). Het systematisch bepalen van de afgeleiden van een functie ten behoeve van hun programmering. Aspecten van Informatieverwerking 4. ICW-nota 948.

Bijlage 1

ENKELE AANWIJZINGEN TEN BEHOEVE VAN DE PROGRAMMERING

1. Algemeen

Voor het programmeren en inpassen in het basisprogramma NLV van de diverse programma-onderdelen is bij de Afdeling Wiskunde een uitvoerige handleiding beschikbaar. Voor zover echter met betrekking tot hetgeen in deze nota behandeld is de gebruiker reeds met de programmeringsmogelijkheden rekening kan houden, volgen daarvoor hieronder enkele aanwijzingen.

2. Projectidentificatie

Voor de projectidentificatie zijn 50 posities beschikbaar. Deze identificatie (of omschrijving) wordt geprint in een daartoe gereserveerd kader. Voor het verkrijgen van een symmetrische layout is het aantrekkelijk de omschrijving te centreren binnen de beschikbare 50 posities.

3. COMMON-code

De COMMON-code dient op een aparte kaart, beginnend in de eerste kolom, te worden geponst. Voor het samenstellen van de COMMON-code uit gegevens omtrent de omvang van het probleem wordt naar Bijlage 2 verwezen.

4. Indicering van variabelen en parameters

In par. IX.6 is aangegeven hoe de variabelen en parameters geïndiceerd worden. De naamgeving van de variabelen kan nog iets verder aan de mathematische schrijfwijze worden aangepast, wat ook van voordeel is bij het programmeren van de afgeleiden (STOL, 1977). De mogelijkheden en beperkingen zijn de volgende:

In de subroutines is de naam van de dummy variabele voor de parameters C(K) waarin K de parameter-volgorde indiceert. De onafhankelijk variabelen worden in de subroutines aangeduid met de naam X(I, J) waarin I het gegeven-nummer indiceert en J = 2, ... de volgorde van de onafhankelijke variabelen.

De waarnemingsuitkomsten van de afhankelijk variabele worden aangeduid met X(I, 1).

Ten einde nu het veelvuldig gebruik van indices tussen haakjes in de programmering te vermijden kan als volgt worden gehandeld.

Definieer:

$$A1 = C(1) \quad \$ \quad A2 = C(2) \quad \$ \quad A3 = C(3)$$

$$X1 = X(I, 2)$$

$$X2 = X(I, 3)$$

Niet gebruikt kunnen worden de symbolen A en X zonder volgend indexcijfer.

De formule uit het voorbeeld van par. IX.6 kan dan geschreven worden in de volgende vorm die tevens kan dienen voor het bepalen van de afgeleiden (STOL, 1977):

$$Y = A1 \cdot e^{-A3 \cdot X1} + \sin(X2 + A2)$$

(dit is geen FORTRAN, maar de 'gebruikelijke' wiskundige schrijfwijze waarin echter A1 het symbool van één parameter voorstelt, en X1 het symbool van één variabele, enz.).

Na berekening van de functiewaarde moet dan weer aan het basisprogramma worden aangesloten met (in FORTRAN):

$$YCLC(I) = Y$$

voor het vastleggen van de berekende waarden te weten 'y-calculated'.

Als hulpvariabelen bij het programmeren van de functie en zijn afgeleiden kunnen gebruikt worden de namen U1, U2, ..., V1, V2, ..., W1, W2, ... en op dezelfde wijze Q, S en T. Deze namen mogen ook alle zonder indexcijfer worden gebruikt.

Vervolg Bijlage 1

5. Naamgeving parameters

Voor het verkrijgen van een overzichtelijke layout dienen de parameternamen in de beschikbare ruimte van 10 tekens te worden gecentreerd. Voorbeeld (in FORTRAN) (zie ook par. IX.9):

```
1234567890
```

```
NAME(1)=10H FACTOR  
NAME(2)=10H PHI  
NAME(3)=10H EXP(-B*U)
```

6. Identificatie van de gegevens

Bij het kopiëren van lange reeksen gegevens naar een magnetische tape, gaat de file naam waaronder de reeks bekend stond verloren. Via een zogenaamde ITEMIZE instructie kan wel een inhoudsopgave van hetgeen op de tape staat verkregen worden, waarbij de eerste zeven karakters van de file als identificatie wordt vermeld. Bij cijfermateriaal betekent dit bijvoorbeeld dat het eerste getal wordt uitgeprint doch daarvan alleen datgene wat in de eerste zeven posities voorkomt. Dit kunnen overigens 'blanks' zijn waardoor geen enkele identificatie meer wordt verkregen.

Hieraan kan tegemoet gekomen worden door aan de file met gegevens een record (ponskaart) met een identificatie van de cijferreeks te laten voorafgaan en in de eerste 7 kolommen daarvan een verkorte identificatie op te nemen. Voorbeeld (zie IX.13) nu met 54 tekens:

```
HOL1/76 PROEFGEBIED N.HOLLAND, EERSTE SERIE (AUG.1976)
```

Bij de output van NLV wordt deze gehele tekst als identificatie van de gegevens vermeld. Wil men deze tekst centreren in de beschikbare 80 posities, dan moet begonnen worden met 13 blanke kolommen. Dit betekent dat de verkorte identificatie niet op de ITEMIZE output komt. Men moet dus de toelichtende tekst centreren met behoud van de identificatie HOL1/76 in de eerste zeven kolommen van de kaart.

Dus:

HOL1/76

PROEFGEBIED N.HOLLAND, EERSTE SERIE (AUG.1976)

Bij de inhoudsopgave van de tape wordt van deze file dan vermeld
HOL1/76 wat dan tevens de aansluiting geeft aan de omschrijving in
de output van NLV.

