

april 1977

NN31545.0951

Instituut voor Cultuurtechniek en Waterhuishouding
Wageningen

ASPECTEN VAN INFORMATIEVERWERKING

BIBLIOTHEEK DE HAAFF

Droevendaalsesteeg 3a
Postbus 241
6700 AE Wageningen

6

**BIBLIOTHEEK
STARINGGEBOUW**

OVER HET SAMENSTELLEN VAN EEN COMPUTERPROGRAMMA VOOR HET
OPTIMALISEREN VAN PARAMETERS

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

1752646

CENTRALE LANDBOUWCATALOGUS



0000 0941 1386

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
5	Het samenstellen van een input tape voor een elektrisch model	J.B.H.M. van Gils		
6	Over het samenstellen van een computerprogramma voor het optimaliseren van parameters	Ph.Th. Stol	951	apr. 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.
1. INLEIDING	1
2. APPARATUUR	2
3. BEGRIPSOMSCHRIJVING	2
4. ALGEMENE WERKWIJZE	5
5. HET UPDATE-SYSTEEM	7
1. UPDATE creation	7
2. UPDATE correction	8
3. Vertaling (compilation)	8
4. Uitvoering (execution)	8
6. AANPASSING AAN DE OMVANG VAN HET PROBLEEM	9
7. NAAMGEVING VAN FILES	11
8. SAMENHANG TUSSEN DE FILES	13
9. PERMANENT FILES EN LOCAL FILES	16
10. COMPUTERJOBS	19
11. BIJZONDERE AANPASSING VAN HET HOOFDPROGRAMMA	23
12. GEDETAILLEERDE TOELICHTING	24
13. REFERENTIES	27

1. INLEIDING

Voor het bepalen van de parameters die niet-lineair in een gegeven functie voorkomen is een algemeen computerprogramma beschikbaar. Dit programma bestaat uit een hoofdprogramma en een aantal subroutines waarin, in de basisvorm, nog geen bepaalde functie voorkomt.

In deze nota zal nader ingegaan worden op de werkwijze om de verschillende 'lege' subroutines tot een volledig programma samen te stellen.

In het volgende zal het optimaliseringsprogramma worden aangeduid met de afkorting NLV van Niet-Lineaire Vereffening.

Het basisprogramma voor NLV is zodanig samengesteld dat met een minimum aan werk voor de gebruiker met standaardprogramma's een nieuwe functie automatisch kan worden ingebouwd en een zo te noemen 'nieuw programma NLV' kan worden samengesteld.

De onderdelen en alle verdere informatie die nodig zijn om een nieuwe functie in het basisprogramma NLV op te nemen en in te bouwen moeten door de gebruiker worden verstrekt. Een toelichtende beschrijving op hetgeen bekend moet zijn staat beschreven in Nota 943 (STOL, 1977). Aan de hand hiervan kunnen de in te bouwen onderdelen worden geprogrammeerd.

In deze nota zal worden vastgelegd wat het systeem is volgens hetwelk het inbouwen van de voor een gegeven functie specifieke onderdelen in NLV kan worden tot stand gebracht.

Het nieuw samengestelde programma NLV biedt nu alle mogelijkheden zoals beschreven door STOL (1975), om parameters in een gegeven functie te optimaliseren.

2. APPARATUUR

De hier beschreven gang van zaken is momenteel gerealiseerd op de volgende Control Data computers: de CDC-6600 van het Energie-onderzoek Centrum Nederland (ECB) te Petten en de kleinere Cyber van IWIS-TNO, den Haag. De eerste computer wordt gebruikt via de DATA 100 terminal van de Afd. voor Technisch Wetenschappelijk Rekenwerk van de Cultuurtechnische Dienst te Utrecht en de tweede via de als terminal geprogrammeerde PDP-11 computer van IWIS-TNO, Wageningen.

Op beide terminals kunnen, op enkele administratieve details na, voor het uitvoeren van de jobs dezelfde stuurkaarten worden gebruikt. Op beide computers is hetzelfde UPDATE-systeem toepasbaar. In het navolgende wordt de jobbeschrijving voor de CDC-6600 van ECN gegeven.

Er wordt aangenomen dat, evenals voor het basisprogramma geldt, in FORTRAN IV (extended) geprogrammeerd wordt.

3. BEGRIPSOMSCHRIJVING

Hieronder volgt een omschrijving volgens het spraakgebruik van een aantal begrippen die veel in deze nota voorkomen.

N L V

Programmanaam voor Niet Lineaire Vereffening.

H o o f d p r o g r a m m a

Fortranprogramma waarmee de subroutines van NLV worden bestuurd en geactiveerd. Het hoofdprogramma kan worden gemodificeerd waardoor verschillende algorithmen kunnen worden toegepast.

B a s i s p r o g r a m m a v o o r N L V

Subroutines voor niet lineaire vereffening waarin nog geen te vereffenen functie is ingebouwd.

N i e u w e f u n c t i e (nieuw probleem)

Ook: gegeven functie, te vereffenen functie, specifieke functie. Functie waarvan de parameters moeten worden bepaald aan de hand van waarnemingsuitkomsten waarvoor de functie moet worden ingebouwd in het basisprogramma.

P r o g r a m m a o n d e r d e l e n

De in FORTRAN geprogrammeerde specifieke functie en verdere instructies die in de verschillende subroutines moeten worden ingebouwd om daarvan een volledig en uitvoerbaar programma te maken.

N i e w p r o g r a m m a v o o r N L V

Subroutines voor niet lineaire vereffening waarin wél een te vereffenen functie is ingebouwd. Ook: nieuw samengesteld programma NLV.

S t u u r k a a r t , j o b c o n t r o l k a a r t

Ponskaart waarop instructies staan vermeld waarmee files kunnen worden opgevraagd, gecopieerd, verwijderd en waarmee file manipulaties zoals bijwerken (update) en vertalen (Fortran) kunnen worden geregeld.

J o b , c o m p u t e r j o b

Totaal aan stuurkaarten waarmede een bepaald, afgerond werkonderdeel kan worden uitgevoerd.

F i l e n a a m

Naam gegeven aan een eenheid informatie (file) zoals een Fortranprogramma, vertaald Fortranprogramma, serie gegevens, groep subroutines enz. De gebruiker kan de filenamen binnen de restricties van het computersysteem vrij kiezen; de in deze nota voorkomende namen moeten in dit opzicht als voorbeelden worden beschouwd.

U P D A T E, U p d a t e - s y s t e e m

Systeem waarbij het mogelijk is wijzigingen, verbeteringen, aanvullingen enz. aan te brengen aan informatie die vastgelegd is op files in een computersysteem.

U P D A T E - f i l e

File die 'aangemaakt' is met het update-systeem en waarop dan dit systeem kan worden toegepast.

P e r m a n e n t f i l e

File die na het beëindigen van een job in het computersysteem aanwezig blijft en bij volgende jobs weer opgeroepen kan worden.

L o c a l f i l e

File die slechts tijdens het uitvoeren van een bepaalde job in het computersysteem aanwezig blijft en na het beëindigen ervan verloren gaat. Met behulp van de daartoe bestemde jobcontrol- of stuurkaarten kunnen local files permanent gemaakt worden (gecatalogiseerd) of permanent files local (geattached).

N P L

New Program Library: aanduiding voor een met het update systeem bijgewerkte file. Aan een NPL kan een zelf verzonnen filenaam gegeven worden.

O P L

Old Program Library: aanduiding voor een met het update systeem te verbeteren file. Aan een OPL kan een zelf verzonnen filenaam gegeven worden.

I d e n t i f i e r n a m e

Zelf te kiezen naam die aan een update correctie set kan worden gegeven om de verschillende correctie sets van elkaar te onderscheiden.

F U N C T I E

Naam van de eerste groep subroutines van NLV, te weten INITL, READ, FNCTN, DFDA en DZFDA (zie STOL, 1977).

R O U T I N E

Naam van de tweede groep subroutines van NLV, te weten NRMEQ, SOLVE, HOWA, MIN, AISB en LISTING.

S P E C I A L

Naam van de derde groep subroutines van NLV, te weten BLOCK, BACK, TRACK, LIHYPEX en CORGRAF.

4. ALGEMENE WERKWIJZE

De in deze nota geschetste werkwijze kan overal worden toegepast waar men in een computerprogramma bepaalde onderdelen wil vervangen door een geheel ander programma-onderdeel.

Het te vervangen programma-onderdeel behoeft niet een zelfstandig programma of zelfstandige subroutine te vormen. Het kan bestaan uit enkele losse, overigens wel in volgorde aansluitende, instructies.

Het vervangende programma-onderdeel vormt een afzonderlijk kaartdek met een eigen naam en kan apart van het basisprogramma worden bewaard en geadministreerd.

De procedure waarmee door de computer het inbouwen wordt gerealiseerd, verloopt in principe als volgt. Stel dat twee programma-onderdelen 'uitwisselbaar' moeten zijn waaraan men de naam geeft FIRST en SECOND. Het basisprogramma ziet er dan in schema als volgt uit:

<u>Basisprogramma</u>	<u>Toelichting</u>
PROGRAM MYJOB	eerste kaart
-	
- fortran tekst	
-	
*CALL FIRST	in te voeren 1e programma-onder-
-	deel (versie 1 of 2)
-	
- fortran tekst	
-	
-	
*CALL SECOND	in te voeren 2e programma-onder-
-	deel (versie 1 of 2)
-	
- fortran tekst	
-	
-	
END	laatste kaart

Daarnaast kan men dan beschikken over de volgende programma-onderdelen met corresponderende namen:

<u>Programma-onderdelen 1e versie</u>	<u>Programma-onderdelen 2e versie</u>
*COMDECK FIRST	*COMDECK FIRST
-	-
- fortran tekst,	- fortran tekst,
- versie 1	- versie 2
-	-
-	-
*COMDECK SECOND	*COMDECK SECOND
-	-
- fortran tekst,	- fortran tekst,
- versie 1	- versie 2
-	-
-	-

Stel men wil van beide onderdelen de eerste versie inbouwen in het basisprogramma MYJOB - dat overigens ook een subroutine mag zijn - dan worden het incomplete basisprogramma en de te prefereren versie op file gezet met te kiezen namen bijvoorbeeld BASIS en NFILE. Beide files worden in een aparte computerjob samengevoegd tot één zogenaamde update file, bijvoorbeeld AFILE. Een update-systeem zorgt er nu voor dat, alvorens het programma door de computer wordt vertaald, de fortran tekst in de comdeck's op de juiste plaats in het basisprogramma wordt gecopieerd.

Op dezelfde wijze kan men met versie 2 te werk gaan, dan wel van het programma-onderdeel FIRST de eerste versie gebruiken en van programma-onderdeel SECOND de tweede versie en omgekeerd. Uiteindelijk kan men dus in dit voorbeeld op deze wijze 4 nieuwe programma's MYJOB samenstellen.

Uit deze toelichting is het duidelijk dat de gebruiker niet geheel vrij is in het schrijven van fortran teksten voor de genoemde programma-onderdelen. Deze moeten namelijk één geheel gaan vormen met de overige tekst van het basisprogramma. Bij elk comdeck behoort dus een instructie waarin aanwijzingen gegeven worden omtrent het gebruik van namen van variabelen, labelnummers en indicering van variabelen opdat één compleet programma kan worden verkregen.

Voor het gebruik van NLV zijn dergelijke instructies beschikbaar bij de Afdeling Wiskunde.

5. HET UPDATE-SYSTEEM

In deze paragraaf zal in het kort iets over het toegepaste UPDATE-systeem worden gezegd.

In principe komt het erop neer dat een vier-deling in het gehele bewerkingsproces wordt toegepast. Deze vier-deling is:

1. UPDATE creation

Het inlezen van ponskaarten naar een file waarop het UPDATE-systeem kan worden toegepast. In het vorige voorbeeld de update-files BASIS en NFILE.

2. UPDATE correction

Het wijzigen, aanvullen, verbeteren, bijwerken van de programma-onderdelen op de gecreëerde update-files respectievelijk het samenvoegen van de files - zoals BASIS en NFILE - tot één nieuwe file bijvoorbeeld AFILE. Het kopiëren van de fortran tekst in de comdeck's op die plaats in het basisprogramma waar deze onderdelen worden opgeroepen. Het wegschrijven van het samengestelde programma naar een compile-file, bijvoorbeeld te noemen CFL.

3. Vertaling (compilation)

Het vertalen van de compile-file CFL en het kopiëren van het vertaalde programma naar een binaire file bijvoorbeeld BFILE.

4. Uitvoering (execution)

Het uitvoeren van het fortran programma door de BFILE te laden (load and execute).

De laatste twee fasen zullen meer bekend zijn dan de eerste twee. Kleine programma's kan men namelijk steeds efficiënt opnieuw inlezen en vertalen. Met omvangrijke kaartendecks wordt het herhaaldelijk inlezen een bezwaar en men zal er spoedig toe overgaan deze programma's op een file in het computersysteem te plaatsen. De moeilijkheid doet zich dan voor hoe de onderdelen van het programma bereikbaar te maken om wijzigingen aan te kunnen brengen.

Voor dit laatste doel is door Control Data het UPDATE-systeem ontworpen. Alvorens correcties op een bestand te kunnen aanbrengen moet de file waarop het bestand staat gecopieerd, door het systeem als update-file herkend kunnen worden. Hiertoe dient de UPDATE-creation run waarmee de file van de nodige ingangen wordt voorzien zodat op de file alle vroegere ponskaarten als eenheid te onderscheiden en bereikbaar zijn.

Ad 1. U P D A T E - c r e a t i o n. In een UPDATE-creation run krijgt een hoeveelheid informatie de status van update file. Deze hoeveelheid informatie bestaat in de regel uit informatie vastgelegd op ponskaarten. Het behoeft dus niet een fortran programma te zijn, elke

vorm van informatie (zoals hydrologische gegevens, een adressenbestand, programma-onderdelen) kan als invoer van een UPDATE-creation run dienen. Een zodanig aangemaakte file heet dan een NPL (New Program Library).

Ad 2. U P D A T E - c o r r e c t i o n. Alle handelingen die op de informatie aanwezig op de NPL moeten worden uitgevoerd kunnen nu, met een gering aantal stuurkaarten door middel van een UPDATE-correction run worden verricht (zoals verbeteren van gegevens, wijzigen van adressen, herstellen van fouten in programma's). Hiertoe wordt de eerder aangemaakte NPL herdoopt tot OPL (Old Program Library), om na te zijn verbeterd weer te worden gerekend een NPL te zijn.

Een volgende updating van de informatie heeft tot gevolg dat de NPL weer gerekend wordt een OPL te zijn, als invoer voor de update-correction run dient, met als uitvoer weer een NPL, enz.

De instructies te gebruiken in een correctie run worden afzonderlijk onder een te verzinnen identifier-name geadministreerd en blijft als zodanig herkenbaar. Bijvoorbeeld *IDENT OKT16 en *IDENT OKT19, zodat men de verschillende correctiesets later steeds kan onderkennen.

6. AANPASSING AAN DE OMVANG VAN HET PROBLEEM

Onder de omvang van een nieuw probleem wordt hier verstaan het aantal parameters, het aantal variabelen, het aantal gegevens en het aantal besturingsparameters waaruit het probleem, te weten een te vereffenen functie, bestaat. Samengevat betekent dit dat de dimensies van de relevante variabelen deze omvang bepalen. Aangezien deze van probleem tot probleem kan variëren is het van voordeel deze dimensies ook in een comdeck op te nemen en uitwisselbaar te maken.

In een Fortranprogramma worden dimensies gedefinieerd met DIMENSION en COMMON statements. Gebruikt men voor de variabele dimensies de naam DVAR, dan kan een programma als volgt hieraan met het update-systeem worden aangepast:

```

SUBROUTINE THIS1
*CALL DVAR
-
- fortran tekst
-
END

*COMDECK DVAR
-
- fortran dimension en common
- statements met werkelijk te
- gebruiken dimensies
-

```

Aangezien bij problemen van verschillende omvang de variabelen zelf in de dimension en common statements toch dezelfde zijn, kan een dergelijk comdeck als output van een fortranprogramma worden verkregen wanneer de input voor dit programma uit de te gebruiken dimensies bestaat. Als toelichting mag dienen een stukje Fortran tekst uit zulk een programma, dat het principe demonstreert volgens hetwelk het programma PONS werkt (VAN GILS, 1974).

Schrijfopdrachten in programma PONS:

```

WRITE(1,20)
20 FORMAT("*COMDECK DVAR")
K=5
L=100
WRITE(1,30)K,K,K,L,L
30 FORMAT(6X"DIMENSION A("I2"),B("I2"),
*X("I2","I3"),Y("I3")")

```

Wat in de output van dit programma tot resultaat heeft:

Output van het voorgaande:

```

*COMDECK DVAR
DIMENSION A( 5), B( 5), X( 5,100),Y(100)

```

De tweede regel van deze output is een Fortran dimension statement die nu dus met een hulpprogramma is verkregen. Door op de plaats waar deze statement in het programma moet worden opgenomen te schrijven *CALL DVAR wordt bereikt dat met behulp van een update correction run de dimensie statement op de juiste plaats in het programma wordt opgenomen.

De totale output kan op een inputfile voor een update creation run geschreven worden. De dan te verkrijgen update file wordt voor het programma NLV genoemd: CFILE (voor het hoofdprogramma) en: KFILE (voor de subroutines).

Opgemerkt wordt nog dat als in a l l e subroutines van het basisprogramma de statement *CALL DVAR voorkomt er wordt bereikt dat automatisch in a l l e subroutines dezelfde omvang van het probleem en dezelfde volgorde van de variabelen wordt gedefinieerd zodat automatisch een correcte overdracht van uitkomsten van de ene subroutine naar de andere en naar het hoofdprogramma is gegarandeerd.

7. NAAMGEVING VAN FILES

Teneinde gemakkelijk in staat te zijn de inhoud van de diverse files uit de naam af te leiden is het nu volgende systeem opgezet. De genoemde files zijn als zogenaamd permanent files in het computersysteem aanwezig en kunnen voor elke job weer opgeroepen en verwerkt worden. Voor details over de inhoud van de eerste drie files wordt verwezen naar Nota 943 (STOL, 1977). Op deze plaats wordt reeds verwezen naar fig. 1 waarin de te bespreken files in logische samenhang voorkomen.

FUNCTIE

Op deze update-file staan de subroutines die aan de omvang van het probleem aangepast moeten worden en waarin de nieuw te programmeren programma-onderdelen een plaats moeten krijgen. Veranderingen in het basisprogramma, aangebracht met een UPDATE-correction run krijgen de NPL-naam FUNCTIE_i, waarin voor _i, voor opeenvolgende correcties, de waarden 1, 2, ... gekozen kan worden.

ROUTINE

De update-file waarop de subroutines staan die de meest gebruikelijke algorithmen en hun modificaties bevatten. De subroutines moeten aan de omvang van het probleem worden aangepast. Eventueel reeds geupdate files worden aangegeven met ROUTINE_i, $i = 1, 2, \dots$

SPECIAL

De update-file waarop de subroutines staan die enkele speciale procedures bevatten. De subroutines moeten aan de omvang van het probleem worden aangepast. Eventueel reeds geupdate files worden aangegeven met SPECIAL_i, $i = 1, 2, \dots$

AFILE

De update-file waarop de files FUNCTIE, ROUTINE, SPECIAL, KFILE en NFILE als één update-file worden samengebracht. Voor omvangrijke problemen kan op geheugenruimte worden bezuinigd door de file SPECIAL niet op te nemen. Geupdate files worden opnieuw AFILE genoemd. Worden geupdate versies van FUNCTIE, enz. gebruikt dan worden FUNCTIE₁, enz. op de AFILE verzameld.

BFILE

De binaire file waarop het vertaalde programma voorkomt, dat wil zeggen alle vertaalde subroutines van AFILE. Geupdate files worden opnieuw BFILE genoemd.

CFILE

De update-file waarop de dimension en common statements voor het hoofdprogramma staan vermeld.

DFILE

De binaire file waarop te verwerken gegevens staan vermeld. Eventueel, bij een gering aantal gegevens, kan men besluiten de

gegevens via de kaartlezer steeds opnieuw bij elke job in te lezen - en eventueel aan te passen en aan te vullen - in plaats van deze file te gebruiken.

KFILE

De update-file waarop de dimension en common statements voor de subroutines staan vermeld.

NFILE

De update-file waarop de nieuw geprogrammeerde programma-onderdelen voor de te vereffenen functie zelf staan vermeld.

PFILE

De door update te lezen file waarop de nieuwe parameterwaarden verkregen na beëindigen van uitvoering van het hoofdprogramma staan vermeld. Dit kunnen tussenresultaten zijn voor zover het eindpunt van de convergentie nog niet werd bereikt. De uitkomsten, geschreven op deze file, kunnen in een volgende ronde van het iteratieproces weer van deze file worden ingelezen en dienen dan als startpunt voor verdere berekeningen.

*FILE i met $i = 1, 2 \dots$

Identificer naam waarmee bij het updaten de correctieset toegepast op NFILE wordt aangeduid. Achtereenvolgende correctiesets worden van elkaar onderscheiden door achtereenvolgens te kiezen $i = 1, 2, \dots$ enz.

8. SAMENHANG TUSSEN DE FILES

In fig. 1 staat de logische samenhang tussen de verschillende files weergegeven. Deze samenhang zal nu in het kort worden besproken.

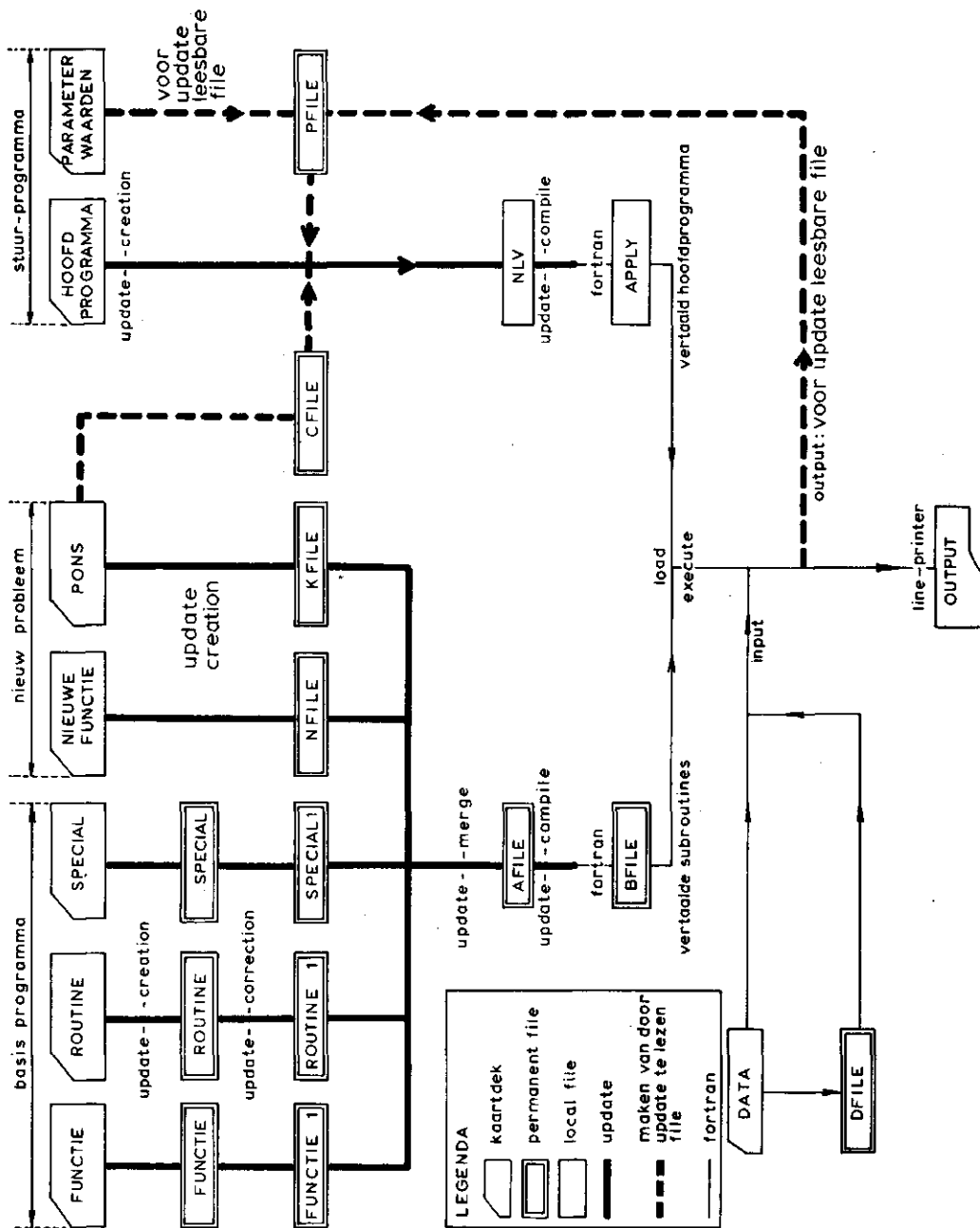


Fig. 1. De logische samenhang tussen de verschillende files voor het samenstellen van een programma NLV

De drie groepen subroutines die het basisprogramma vormen zijn bij de Afdeling Wiskunde aanwezig geponst op ponskaart in 26 code. Met een update creation run zijn deze kaartdeks op permanent file gezet in de vorm van een NPL. Voor zover verbeteringen of veranderingen aan het basisprogramma moeten worden aangebracht kan dit gebeuren vanaf deze NPL, waarna met een update correction run de nieuwe permanent files, met aan de naam een 1 toegevoegd, wordt verkregen. Het is van belang op te merken dat de files FUNCTIE, ROUTINE en SPECIAL als zodanig bewaard blijven en niet gewijzigd worden. Dit houdt in dat indien onverhoopt in een job een file wordt vernield, deze vanuit de bovengenoemde files gereconstrueerd kan worden.

De programma-onderdelen die voor een nieuwe in te bouwen functie geprogrammeerd moeten worden, worden op ponskaart geponst in 29 code. In fig. 1 is het betreffende kaartdeck met 'Nieuw probleem' aangeduid. Met een update creation run wordt dit dek op permanent file gezet. Aan deze file is de naam NFILE gegeven. Eventuele verbeteringen aan deze file aan te brengen kunnen het eenvoudigst worden gerealiseerd door in het kaartdek de gewenste veranderingen aan te brengen en een nieuwe NFILE te maken met een update creation run.

De Fortran statements die de omvang van het probleem regelen (common en dimension statements) worden vervaardigd met het Fortran-programma PONS (VAN GILS, 1974) dat in 29 ponscode op kaart beschikbaar is. De dimension en common-statements benodigd voor de subroutines worden eerst met dit programma vervaardigd en aansluitend hieraan met een update creation run op een file gezet waaraan de naam KFILE is gegeven. Het programma PONS vervaardigt tenslotte ook de dimension en common-statements voor het hoofdprogramma die op een file gezet worden waaraan de naam CFILE werd gegeven.

De volgende stap in het proces van het samenstellen van een nieuw vereffeningsprogramma is het samenbrengen van alle tot nu behandelde files in één update file. Dit kan worden gerealiseerd met een zogenaamde update merge job. De gecombineerde update file wordt AFILE genoemd.

De onderdelen in de AFILE worden nu met een update-compile run tot een compleet Fortran programma samengesteld. Het vertaalde pro-

programma staat nu op een binaire file en wordt BFILE genoemd. De subroutines op deze file worden bestuurd met het hoofdprogramma dat op ponskaart in 29 kaartcode beschikbaar is.

Het hoofdprogramma roept respectievelijk de CFILE op waarop de dimension en common-statements staan en de PFILE waarop parameterwaarden staan vermeld. Hiermede is het hoofdprogramma als Fortran-programma compleet en kan vertaald worden. De binaire file waarop het vertaalde hoofdprogramma staat is APPLY genoemd. Beide binaire files, BFILE en APPLY worden nu door middel van de relevante job-control kaarten in het geheugen van de computer geladen en tot uitvoering gebracht.

9. PERMANENT FILES EN LOCAL FILES

In het voorgaande is erop gewezen dat de belangrijke files beschikbaar zijn als permanent file. In fig. 2 staat schematisch aangegeven hoe na een update-correction run verbeterde permanent files met dezelfde permanent file namen kan worden verkregen. Ter toelichting diene het volgende, waarbij wordt opgemerkt dat alle beschreven handelingen automatisch kunnen worden uitgevoerd met behulp van stuurkaarten of jobcontrol kaarten.

De volgende afkortingen worden gebruikt:

OPLSM = smoothed old program library
NPLSM = smoothed new program library
OBFSM = smoothed old binary file
NBFSM = smoothed new binary file
CSM = smoothed compile file
BSM = smoothed binary file

waarin met 'smooth' wordt aangegeven dat op de file de subroutines van de drie hoofdgroepen FUNCTIE, ROUTINE en SPECIAL zijn samengevoegd.

Stel dat in een van de subroutines een fortran statement moet worden gewijzigd. Dit kan met een update-correction run. Met behulp

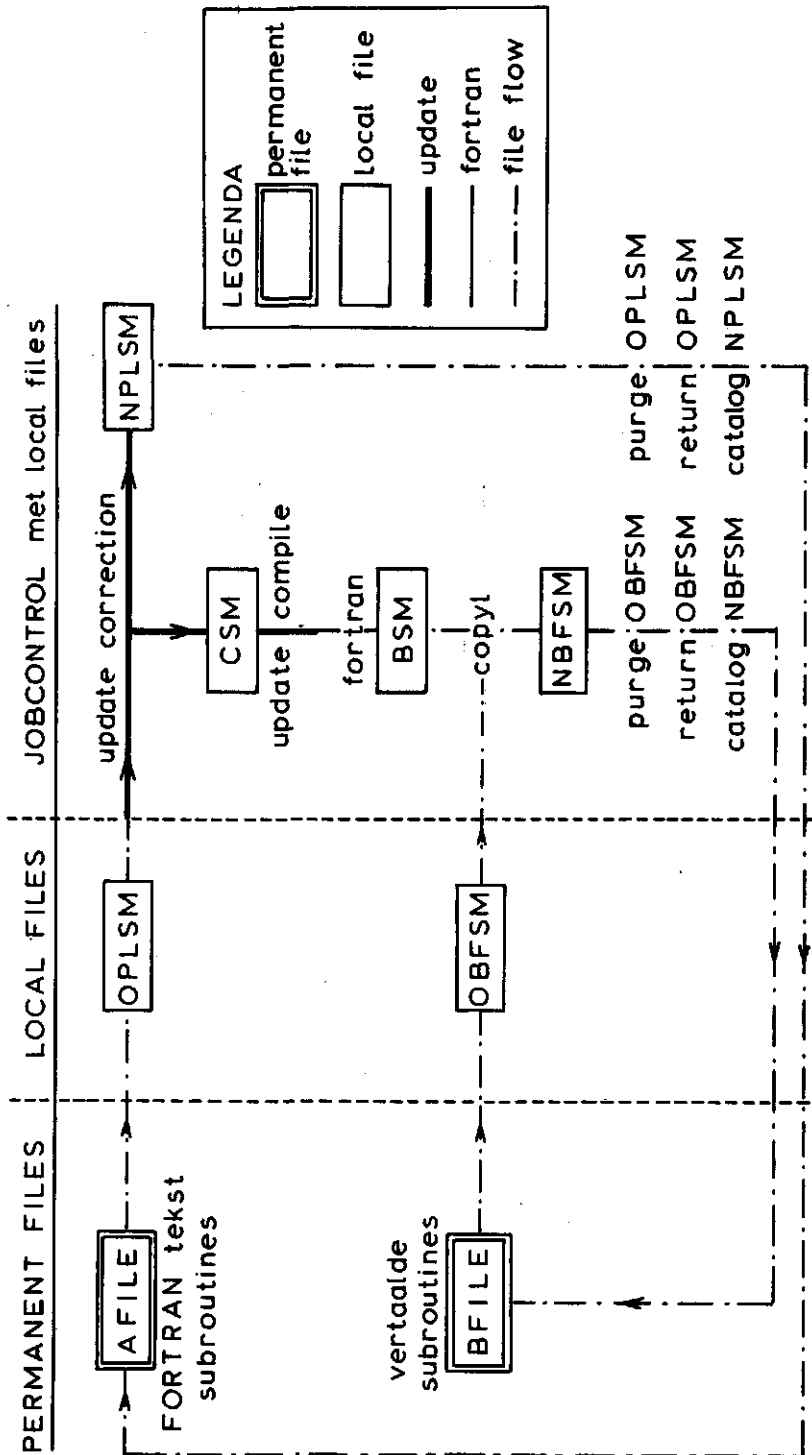


Fig. 2. De logische samenhang tussen permanent en local files

van de daartoe geëigende opdracht (als voorbeeld wordt verwezen naar par. 12) wordt de OPL waarop de FORTRAN tekst van de subroutines staat geschreven - die als permanent file AFILE heet - tot local file gemaakt waaraan de naam OPLSM wordt toegekend. De toevoeging SM is hier gekozen om er aan te herinneren dat het een file betreft waarop alle subroutines zijn samengebracht (een z.g. 'smoothed' file).

De verbeterde OPLSM wordt nu aangeduid met NPLSM, de oude naam OPLSM wordt nu uit het systeem geschrapt (gepurged) en de local file wordt vrijgegeven (gereturned) waarna de local file NPLSM permanent gemaakt (gecatalogiseerd) kan worden onder dezelfde naam AFILE. De voorgaande AFILE is nu vervangen door de verbeterde update file en onder dezelfde naam weer in het systeem opgeslagen.

Het bovenstaande houdt in dat het gehele proces cyclisch is zodat het met dezelfde jobcontrol kaarten en dezelfde file namen kan worden herhaald wanneer men opnieuw verbeteringen in de subroutines wil aanbrengen. De enige voorzorg is dat bij volgende correctieruns een nieuwe identifier naam moet worden gegeven aan het dek correctie opdrachten, aangezien niet tweemaal dezelfde naam mag worden gebruikt.

De hiervoor besproken update correction heeft tot gevolg gehad dat de FORTRAN tekst in de subroutines is verbeterd. Nu moeten de verbeterde subroutines weer opnieuw worden vertaald. Hiertoe worden alleen de verbeterde subroutines naar de compile file gecopieerd, hier aangeduid met de naam CSM. De vertaalde subroutines komen nu op een file genaamd BSM. Met een zogenoemde COPYL opdracht worden alleen de verbeterde subroutines vervangen in de BFILE die hiervoor local gemaakt is onder de naam OBFSM.

De subroutines van de OBFSM file worden met de COPYL opdracht gecopieerd naar de file NBFSM waarbij de te verbeteren subroutines automatisch door de verbeterde subroutines op de BSM file worden vervangen. De herziene FORTRAN versie staat nu op de NBFSM file. Op dezelfde wijze als voor de update file kan met de hiertoe geëigende stuurkaarten de oude BFILE vervangen worden door de nieuwe file onder dezelfde naam.

Ook dit proces is cyclisch en kan met dezelfde jobcontrol- of stuurkaarten en dezelfde file namen worden herhaald.

10. COMPUTERJOBS

Het samenstellen van een nieuw programma NLV gebeurt in de praktijk met een aantal computerjobs die in een standaardvorm beschikbaar zijn. Er is voor gekozen de verschillende onderdelen van de in het vorige toegelichte bewerking met aparte jobs uit te voeren zodat, wanneer onverhoopt iets mis mocht gaan, slechts dat onderdeel behoeft te worden hersteld.

De computerjobs sluiten alle op elkaar aan wat naamgeving van de files betreft, zoals in fig. 1 is weergegeven, zodat de jobs in feite één programmapakket jobcontrol voor NLV vormen.

In deze paragraaf zal een kort overzicht gegeven worden van de achtereenvolgend toe te passen jobs. De complete opbouw van het programmapakket is in teks aanwezig bij de Afdeling Wiskunde.

1. A U D I T

Een systeemroutine waarmee aan het computersysteem wordt opgevraagd welke permanent files nog in het systeem aanwezig zijn. Het is van belang een overzicht over bestaande permanent files te hebben zodat kan worden nagegaan of bij het nieuw samenstellen van een programma NLV permanent files eerstuit het systeem moeten worden verwijderd alvorens deze weer, nu met een nieuwe functie, opnieuw aan te maken.

Input: geen.

2. I N F O

Een routine waarmee de namen van de subroutines van NLV worden getabelleerd die op de update file (AFILE) en de binaire (vertaalde) file (BFILE) voorkomen. Met deze job wordt gecontroleerd of alle subroutines nog op de daartoe bestemde files voorkomen.

Input: geen.

3. N P L / N L V

Job waarmee de drie groepen subroutines van NLV te weten FUNCTIE, ROUTINE en SPECIAL van ponskaart op een update file worden weggezet. De kaartendeks van de drie groepen subroutines worden onder dezelfde naam op permanent file gezet en aldaar bewaard als basisprogramma waaruit alle verdere handelingen plaatsvinden.

Input: op ponskaarten vastgelegde subroutines.

4. F R S

Job waarmee in de subroutines van het basisprogramma veranderingen kunnen worden aangebracht. Dit gebeurt door het UPDATE systeem toe te passen. De verbeterde files worden genoemd FUNCTIE1, ROUTINE1 en SPECIAL1. De oorspronkelijke files FUNCTIE, ROUTINE en SPECIAL worden niet gewijzigd.

Input: op ponskaarten vastgelegde aan te brengen verbeteringen.

5. N F I L E

Job waarmee de programma onderdelen die voor een nieuwe job geprogrammeerd zijn op een update file (NFILE) worden weggezet.

Input: kaartdek met nieuw geprogrammeerde programma-onderdelen voor een nieuwe functie.

6. N E W C O M

Job waarmee de omvang van een nieuw probleem wordt gedefinieerd. Zowel voor het hoofdprogramma als voor de subroutines worden de vereiste dimension en common-statements gemaakt en op permanent file gezet respectievelijk met gekozen namen CFILE en KFILE.

Input: ponskaart met common code.

7. P F I L E

Job waarmee parameterwaarden op een permanent file worden geplaatst. De permanent file kan door het hoofdprogramma worden gelezen.

Input: parameterwaarden gedefinieerd volgens de Fortran statements

A(1) = ... enz.

8. M E R G E

Job waarmede de files FUNCTIE1, ROUTINE1, SPECIAL1, KFILE en NFILE op één update file worden samengebracht, de zogenaamde AFILE.
Input: geen.

9. C O M P

Job waarmede de onderdelen, aanwezig op de AFILE, worden samengesteld tot een volledig Fortranprogramma. Dit Fortranprogramma wordt vertaald en binair vastgelegd op een permanent file welke genoemd is BFILE.
Input: geen.

10. S U B S

Job waarmede een enkele of een aantal subroutines afzonderlijk kunnen worden vertaald voor het nalezen van de tekst respectievelijk ter controle van de programmering. In de lijst die van het programma wordt verkregen staat de update identificatie van alle statements vermeld.
Input: ponskaart waarop vermeld staat welke subroutine moet worden vertaald.

11. C O D Y L

Job waarmede in de subroutines, respectievelijk in de nieuw ge-programmeerde programma-onderdelen veranderingen kunnen worden aangebracht. De oude files AFILE en BFILE werden hierbij vervangen door de verbeterde files die dezelfde naam krijgen.
Input: aan te brengen verbeteringen.

12. R E N A M E

Job waarmede de namen van permanent files kunnen worden veranderd voor het geval meer dan één probleem in bewerking is en de namen AFILE, BFILE enz. dubbel zouden gaan voorkomen, hetgeen niet geoorloofd is.
Input: ponskaarten waarop de nieuwe permanent file namen staan vermeld.

Met het update systeem is het mogelijk programma-onderdelen met een eigen deck-name uit het programma te verwijderen. Dit houdt in dat, om een nieuw probleem te kunnen inbouwen, het niet noodzakelijk is weer met job 5 te beginnen, de AFILE kan rechtstreeks voor de nieuw te vereffenen functie gereed worden gemaakt waarna een nieuwe BFILE kan worden gecreëerd. In het laatste geval, het vervaardigen van een nieuwe vertaling, behoeven alleen de veranderde subroutines te worden vertaald. In overzicht:

Overzicht weergevende de noodzaak tot vertaling van de subroutines				
Subroutines van de groep	Bevat onderdelen van een nieuwe functie	Bevat common en dimension statements	Vertaling nodig bij veranderen van	
			functie	dimensies
FUNCTIE	ja	ja	ja	ja
ROUTINE	nee	ja	nee	ja
SPECIAL	nee	ja	nee	ja
Toe te passen jobs nr			I IA	- en IIA

Met job I worden de programma-onderdelen van de oude functie verwijderd en worden de programma-onderdelen van de nieuwe functie, reeds geplaatst op NFILE, daarvoor in de plaats gezet. Daarna worden de subroutines van de groep FUNCTIE opnieuw vertaald.

Met job IA vindt dezelfde bewerking plaats echter zonder vertaling. Met job IIA worden de oude dimensie en common statements uit alle subroutines verwijderd en worden de nieuwe, reeds geplaatst op KFILE, daarvoor in de plaats gezet. Daarna worden alle subroutines opnieuw vertaald.

Moeten bij een gegeven functie de dimensies (en dus de omvang van het probleem) worden aangepast, dan heeft alleen job IIA te worden uitgevoerd, uiteraard weer nadat de KFILE was vervaardigd.

11. BIJZONDERE AANPASSING VAN HET HOOFDPROGRAMMA

In het hoofdprogramma, dat de subroutines bestuurt, worden de real/integer, de dimensie- en de common-statements via de CFILE opgenomen. Dit betekent dat aparte voorzorgen moeten worden genomen voor het bijzondere geval dat men een nieuwe geïndiceerde variabele wil toepassen in het hoofdprogramma. De dimensie van zulk een variabele moet gedefinieerd worden ná de real/integer definitie en vóór de common-statements. Deze positie in het hoofdprogramma is niet zonder meer bereikbaar zodat te werk moet worden gegaan met een extra update run. De stuurkaarten hiervoor zijn als volgt:

```
JOBNAME,CM77000,T20,IO10.
ACCOUNT,---.
ATTACH,BFILE,BFILE,ID=YYY.
ATTACH,COMMON,CFILE,ID=YYY.
UPDATE,N=NLV,C=0,W,D.
UPDATE,P=NLV,W,D.
FTN,I=COMPILE,B=APPLY,L=0.
LOAD,APPLY.
LOAD,BFILE.
EXECUTE.
EOR
*DECK NLV
    PROGRAM NLV(---)
*READ COMMON
-
-   fortran tekst hoofdprogramma
-
-
    END
EOR
*IDENT  MAIN
*INSERT NLV.4
    DIMENSION AA(5)
EOR
-
-   input data
```

EOR
EOF

Nadat de BFILE en de CFILE van permanent file zijn opgehaald en 'local' gemaakt onder de namen BFILE en COMMON wordt de tekst van het programma NLV, bekend onder de deknaam NLV, samengesteld tot een Fortranprogramma dat overigens nog niet vertaald wordt. Een tweede update run, toegepast op de OPL uit de vorige update, zorgt er nu voor dat de statement DIMENSION AA(5) op de juiste plaats (n.l. na de 4e statement) in het programma wordt ingezet. Hierna volgt de vertaling van het hoofdprogramma (FTN) en het laden (LOAD) en de uitvoering (EXECUTE) er van.

Teneinde een indruk te geven hoe de eerste opdrachten van het hoofdprogramma er nu uitzien volgen hieronder de eerste zes regels van de geupdate Fortran tekst.

*DECK NLV	NLV	1
PROGRAM NLV(INPUT,OUTPUT,	NLV	2
*TAPE2=INPUT,TAPE3=OUTPUT,TAPE7)	NLV	3
REAL M \$ INTEGER END	NLV	4
DIMENSION AA(5)	MAIN	1
COMMON A(4),B(5), enz.	NLV	5

De nummering geheel rechts wordt automatisch met het update systeem gerealiseerd. De 'hoofd'-nummering is volgens de deknaam NLV. De, na de vierde regel, tussengevoegde opdracht wordt genummerd volgens de kaarten voorkomend in de correctieset. Dit gebeurt onder de 'identificer'naam MAIN.

12. GEDETAILLEERDE TOELICHTING

Als voorbeeld van de betekenis van de jobcontrol opdrachten zal een toelichting op de in de vorige paragraaf behandelde job gegeven worden.

De totale job heeft de status van een file, en wordt dan ook afgesloten door een end-of-file kaart.

De file is echter opgebouwd uit 4 records die ieder afgesloten zijn met een end-of-recordkaart. De hoofdingeling is hiermede als volgt:

Jobcontrol kaarten

EOR

Input eerste update (creation); FORTRAN tekst

EOR

Input tweede update (correction); wijzigingen

EOR

Gegevens te lezen door subroutine READ

EOR

EOF

Ter verdere toelichting dient het volgende:

- De jobnaam kan zelf worden gekozen. De jobkaart bevat de aanvraag voor geheugenruimte (Core Memory), computer tijd (Time) en input/output tijd (IO). Geheugenruimte en tijd (in sec) moeten octale getallen zijn.
- Op de ACCOUNT-kaart dienen administratieve gegevens te worden vermeld om toegang tot het computersysteem te krijgen.
- De permanent file BFILE (2e naam) wordt local gemaakt onder dezelfde naam (BFILE, 1e naam). Een identificatie (ID) YYY waaronder de permanent file gecatalogiseerd is dient te worden vermeld.
- De permanent file CFILE wordt local gemaakt onder de naam COMMON.
- Het in de input voorkomende kaartdek *DECK NLV wordt met het update systeem weggezet op een NPL (N) met dezelfde naam. Er wordt geen vertaling uitgevoerd (C = 6). De file wordt aangemaakt met de mogelijkheid tot tabelleren van de file inhoud (W) en het vermelden van de update nummering na de 80e positie (D).
- De update file NLV wordt als OPL (P) ingevoerd voor een update correction run. De uit te voeren verbeteringen komen in de input voor in de derde record met identificatie (*IDENT) genoemd MAIN. Er wordt automatisch een te vertalen file aangemaakt waarop het complete en gewijzigde fortran programma voorkomt (de parameter C is niet genoemd, hiermede is automatisch het programma op een file met naam COMPILE gezet).

- De input file (I) die moet worden vertaald is de file COMPILE, het vertaalde programma wordt weggezet op de binaire file (B) met verzonden naam APPLY. Er wordt geen lijst met de programmatekst geprint (L = 0).
- Het vertaalde hoofdprogramma nu bekend onder de naam APPLY wordt in het geheugen geladen.
- De van permanent file opgehaalde BFILE wordt in het geheugen geladen.
- De programma's worden tot uitvoer gebracht.
- Het eerste record waarop de stuurkaarten voorkomen wordt afgesloten.
- Het tweede record bevat de fortran tekst van het hoofdprogramma. Update herkent dit record als input voor een update creation doordat de eerste kaart een 'sterretje dek-'kaart is. In de fortran tekst komt de opdracht *READ voor waarmee de common-statements op de local file COMMON ter plaatse worden ingelezen en aan het programma worden toegevoegd waarmee het fortranprogramma compleet is.
- Het derde record bevat de set aan te brengen wijzigingen als input voor de tweede update.
- De set verbeteringen is de naam (Identifier) gegeven MAIN.
- Na de 4e kaart van het programma NLV zoals dat is samengesteld tijdens de eerste update, moeten nieuwe kaarten worden toegevoegd (Insert).
- De toe te voegen kaart is een fortran dimensie kaart.
- Het derde record wordt afgesloten met een end-of-record kaart.
- Het vierde record bevat gegevens die door het fortranprogramma tijdens de uitvoering moeten worden gelezen.
- Het vierde record wordt met een end-of-record kaart afgesloten.
- De computerjob wordt met een end-of-file kaart afgesloten.

13. REFERENTIES

GILS, J.B.H.M. VAN, 1974. Fortranprogramma PONS voor het automatisch vervaardigen van dimensie en commonkaarten (ICW-programma no. 100).

STOL, Ph.Th., 1975. A contribution to theory and practice of non-linear parameter optimization. Wageningen.

——— 1977. Optimaliseren van parameters: Het gereedmaken van een functie voor toepassing in NLV. Aspecten van Informatieverwerking Nr 2, ICW-nota 943.