

NN31545.1465

465

Instituut voor Cultuurtechniek en Waterhuishouding
Wageningen

ASPECTEN van INFORMATIEVERWERKING
46

+-----+
| Het subroutine-pakket |
| IMSL, |
| een korte kennismaking |
+-----+

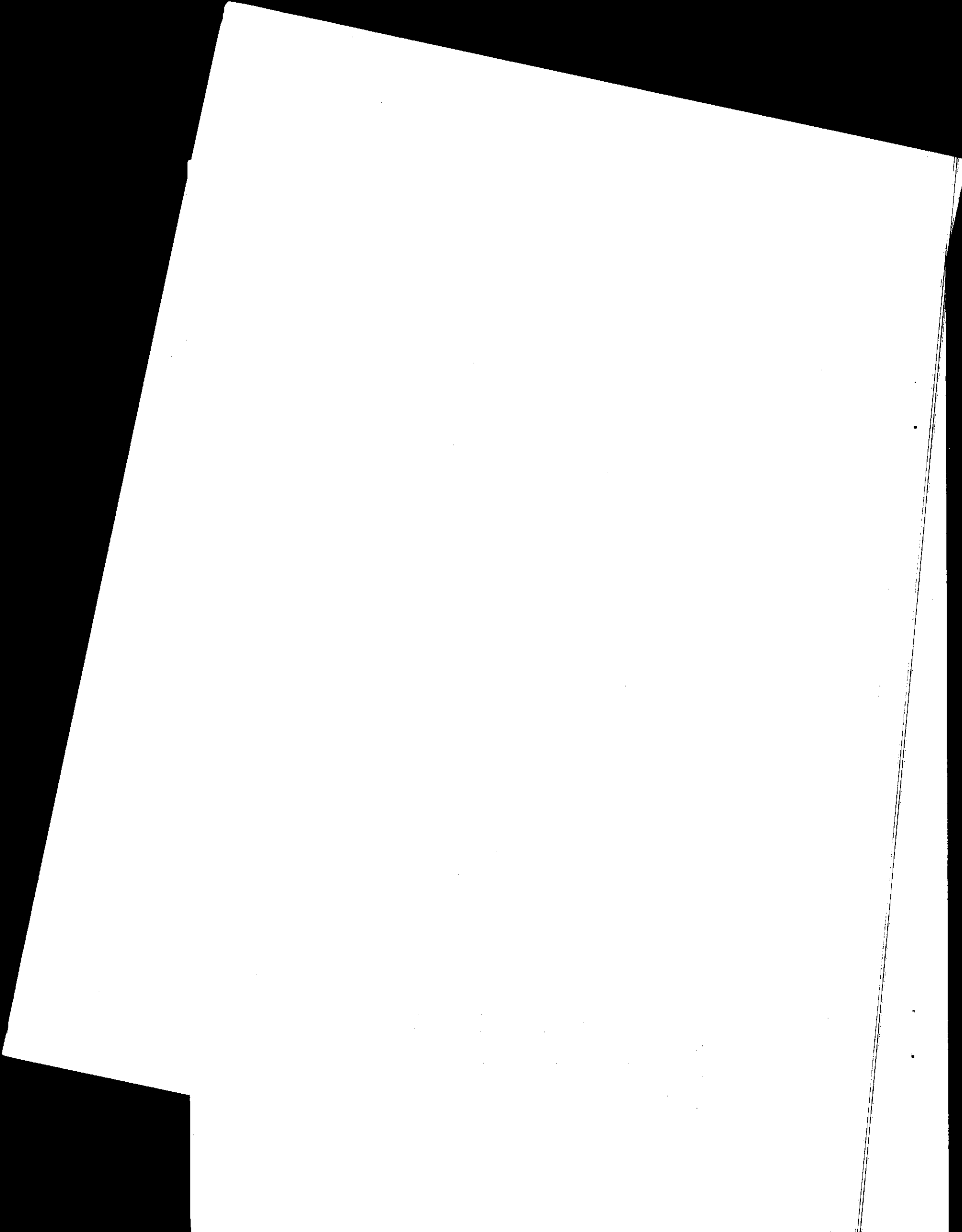
W. van Doorne

De n
bevat
verwe
toel
gebruik
Tevens
en toer

Nota's van het Instituut zijn in principe interne communicatiemidde-
len, dus geen officiële publicaties.
Hun inhoud varieert sterk en kan zowel betrekking hebben op een een-
voudige 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.

17 JAN. 1984

ISBN 90-01-10000-2



I N H O U D

	Pag.
1. Waarom een subroutine-pakket?	1
2. De voorbeelden; hoe zijn ze na te spelen?	2
3. Waaruit bestaat de INSL-bibliotheek?	4

VOORBEELDEN 1 t.m. 5 programmatexten,
uitkomsten,
commentaren.

1. Waarom een subroutine-pakket?

Het ontwerpen en uitwerken van een goed computerprogramma kan veel tijd en vindingsrijkheid vergen. Die moeite moet vaak voor een flink deel worden besteed aan zaken die niet direct het te onderzoeken probleem betreffen maar problemen met de programmeertechniek; het testen van een programma kan zeer tijdrovend zijn. En zelfs na grondig testen komt het voor dat een als 'correct' gekwalificeerd programma bij een bepaalde wijze van toepassing toch nog niet blijkt te werken zoals de bedoeling is.

Een wat ingewikkelder programma vereist in het algemeen een goede documentatie, niet alleen voor derden maar ook voor de ontwerper zelf. Deze immers, is na verloop van tijd vaak ook niet meer zo goed thuis in zijn eigen programma. De documentatie behoort aan hoge eisen te voldoen. Er moet in ieder geval een exacte gebruiksaanwijzing in voorkomen. Daarnaast is het van belang dat alle beperkende voorwaarden ten aanzien van het gebruiken van het programma volledig en nauwkeurig beschreven worden. Een voorbeeld van toepassing (invoer, uitvoer) is meestal niet overbodig.

Betreft het een onderzoek waarvoor een specifiek computer-programma nodig is, dan kunnen bovenaangeduide inspanningen wel interessant of de moeite waardig worden. Anders is het wanneer het te vervaardigen programma geheel of gedeeltelijk kan worden opgebouwd uit bewerkingen waarvan men vermoedt dat ze al eerder geprogrammeerd en wellicht ook gedocumenteerd zijn. In dat geval zou men graag van de voorgeprogrammeerde gedeelten gebruik maken.

Er zijn daarbij twee belangrijke mogelijkheden; of men maakt gebruik van een speciale programmeertaal waarin de gewenste mogelijkheden beschikbaar zijn of men roept in zijn programma bepaalde voorgeprogrammeerde bewerkingen op. Als voorbeeld van een speciale programmeertaal wordt hier genoemd GENSTAT, een taal vooral bedoeld voor statistische bewerkingen. Het voordeel van zo'n taal is haar betrekkelijke eenvoud. Daar staat tegenover dat men vertrouwd moet raken met (alweer) een programmeertaal naast de taal die men al beheerst. De tweede mogelijkheid is, dat men in zijn vertrouwde programmeertaal bepaalde bewerkingen oproept in de vorm van subroutine-aanroepen. De voordelen zijn duidelijk; men heeft geen andere programmeertaal nodig en kan de subroutine-aanroepen naar behoefte inbouwen. We doelen hier met name op het subroutineepakket IMSL (International Mathematical Subroutine Library) waarvan men in een VAX-Fortran programma gebruik kan maken. In een aantal voorbeelden zal in deze Nota worden aangegeven hoe men Fortran (efficiënt, alom bekend) combineert met IMSL-subroutines (uitstekend gedocumenteerd, op eenvoudige en flexibele wijze toepasbaar).

Nu reeds wordt gesteld dat het raadzaam is dat men zich, alvorens over te gaan tot het schrijven van een VAX-Fortran programma, afvraagt of (geheel) zelf programmeren wel nodig is. Het raadplegen van het IMSL-handboek zou in voorkomende gevallen wel eens kunnen uitwijzen dat er kant en klare subroutines beschikbaar zijn voor de statistische of wiskundige bewerkingen waarin men geïnteresseerd is.

2. De voorbeelden; hoe zijn ze na te spelen?

Aan de hand van enkele voorbeelden, gekozen op grond van gebleken belangstelling, zal het inbouwen van een IMSL-routine in een VAX-Fortran programma worden gedemonstreerd. Samenvatting:

VOORBEELD	FILE-NAAM
1. op basis van willekeurige gegeven punten (x,y,z) de waarden Z te bepalen in gegeven, rechthoekig verspreide punten (X,Y);	VBIMSL1.FOR
2. integreren van een differentiaal-vergelijking $y' = f(x,y)$ voor gegeven $x=X$ met randvoorwaarde $(x,y) = (X_0, Y_0)$;	VBIMSL2.FOR
3. plotten van reeks X tegen de reeksen Y1 en Y2;	VBIMSL3.FOR
4. lineaire vereffening $Y = a + b.X$;	VBIMSL4.FOR
5. niet-lineaire vereffening $Y = a - b.c**X$	VBIMSL5.FOR

De voorbeelden zijn als volgt opgebouwd:

- specificatie en definitie van de argumenten
[de waarden van de input-argumenten worden gegeven in de aanroep van de IMSL-routine]
- aanroep van de IMSL-routine d.m.v. CALL prog (...)
['prog' is de naam van de betreffende IMSL-routine]
- het uitschrijven van de uitkomsten
[naar behoefte worden output-argumenten uitgeschreven door Fortran-opdrachten]

Wat betreft de punten a en c wordt het volgende opgemerkt.

In de voorbeelden is een eenvoudige wijze van argument-specificatie gebruikt met toepassing van de Fortran-opdrachten PARAMETER en DATA. Daardoor worden de gegevens opgenomen in de programma-tekst. Dat is bij voorbeelden wel overzichtelijk, gangbare praktijk zal deze werkwijze niet zijn. Eerder zal men een enkel gegeven intikken en de overige informatie van een file laten lezen met een READ-opdracht. In die zin kan men de voorbeelden modificeren voor praktische toepassing.

In de voorbeelden worden de waarden van de output-argumenten zoals geleverd door een IMSL-routine, op het terminal-scherm uitgeschreven met gebruikmaking van de list-directed schrijf-opdracht PRINT *; voorbeeld 3 vormt een uitzondering. Het voordeel is dat men bij het naspelen van de voorbeelden de uitkomsten onmiddellijk setoond krijgt. Bovendien worden de voorbeelden niet onnodig belast met formats, wat de overzichtelijkheid ten goede komt. Ook hier zal de praktijk anders zijn, vooral wanneer de output wat uitgebreider is. In dat geval zal op een file worden geschreven met WRITE, met gebruikmaking van formats als er tabellen of anderzortige overzichten verlangd worden. Voorbeeld 3 is zo'n geval van een uitgebreidere output: een complete X-Y grafiek met bijchriften langs de assen.

Elk van de voorbeelden kan men op de VAX naspelen door de file waar op het voorkomt te kopiëren naar de eigen directory en daarna de systeem-opdrachten FOR (Fortran-vertaling), LIN (aanhaken IMSL) en RUN (uitvoeren van het programma) toe te passen.

Mede omdat voorbeeld n de filenaam VBIMSLn.FOR draagt, verloopt het naspeelen van voorbeeld n als volgt

```
$ COPY DRB0:[CDRN.29010180]VBIMSLn.FOR MYFILE.FOR
$ FOR MYFILE
$ LIN MYFILE, DRA0:[UTIL.IMSL]IMSLIBS/LIBRARY
$ RUN MYFILE
```

Na een keer draaien is het wellicht instructief het program te wijzigen door met de editor in MYFILE.FOR in te grijpen. Te denken valt aan het veranderen van waarnemingsreeksen of -punten, het selecteren van andere te plotten variabelen of aan het inbouwen van een andere niet-lineaire vereffeningscurve.

Hoofdstuk onderwerp in het IMSL-handboek

A	Analysis of Variance
B	Basic Statistics Data Screening; Transgeneration; Elementary Classical Inference
C	Categorized Data Analysis
D	Differential Equations; Quadrature; Differentiation
E	Eigensystem Analysis
F	Forecasting; Econometrics; Time Series; Transforms
G	Generation and Testing of Random Numbers
I	Interpolation; Approximation; Smoothing
L	Linear Algebraic Equations
M	Mathematical and Statistical Special Functions Probability Distribution Functions; Special Functions of Mathematical Physics
N	Non-Parametric Statistics Analysis of Variance; Binomial or Multinomial Bases; Hyper or Multi-hyper Geometric Bases; Kolmogorov-Smirnov Tests; Other Bases; Randomization Bases
O	Observation Structure; Multivariate Statistics Cluster Analysis; Discriminant Analysis; Factor Analysis; Principal Components Analysis
R	Regression Analysis Linear Models; Special Non-Linear Models
S	Sampling
U	Utility Functions Error Detection; Special I/O Routines
V	Vector-Matrix Arithmetic
Z	Zeros and Extrema; Linear Programming

FIG.3;1. Onderwerpen van IMSL-routines.

3. Waaruit bestaat de IMSL-bibliotheek?

De IMSL-bibliotheek is een verzameling van oorspronkelijk in Fortran geschreven programma's voor wiskundige en statistische berekeningen. De programma's zijn van algemene aard en toepasbaar als Fortran subroutine. Het IMSL-handboek speelt in eerste instantie de rol van catalogus en biedt daarbij twee inzichten tot de programma-bibliotheek; de inhoudsopgave (CONTENTS) en de KWIC-index. De inhoudsopgave bevat onder meer een alfabetische lijst van programma-namen met korte omschrijvingen. De lijst is alfabetisch onderverdeeld waarbij de beginletter het hoofdstuk aanduidt. De programma-namen beginnend met A bijv., betreffen Analysis of Variance; de beginletter D duidt op het hoofdstuk Differential Equations, enz. De andere inzig is de KWIC-index. Hierin wordt op trefwoord gezocht. Een indruk van 't soort programmatuur in de IMSL-bibliotheek krijgt men uit het overzicht in Figuur 3;1.

Het handboek geeft verder bij elke IMSL-routine een beschrijving met literatuurverwijzingen naar theoretische achtergronden en een voorbeeld van toepassing. De eigenlijke programma-beschrijving gebeurt op een standaard wijze en bestaat hoofdzakelijk uit een nauwkeurige omschrijving van de argumenten. Ter kennismaking volgt in Figuur 3;2 de letterlijke weergave van een gedeelte van de beschrijving van de in voorbeeld 1 toe te passen IMSL-routine IQHSCV.

Een bijzonder argument van vele IMSL-routines is de zogenoemde 'error parameter', in Fig.3;2 de variabele IER. Deze grootte krijgt een waarde via de betreffende routine. In de programma-beschrijving wordt aangegeven welke waarde van de error parameter overeenkomt met welke fout. Zo'n fout is het gevolg van een minder gelukkig samenstel van gegevens, waardoor de berekening niet vlekkeloos verloopt. Een paar voorbeelden: een iteratieve berekening die niet binnen het gestelde aantal herhalingen afloopt; afhankelijkheid in een stelsel lineaire vergelijkingen; onbepaalde functie-waarden bij integratie; enz. Deze voorbeelden laten zien dat de fouten in categorieën te verdelen zijn. De iteratieve berekeningen die niet snel genoeg afloopt, levert een waarschuwing in de vorm van een zogenoemde 'warning error'. De andere voorbeelden leiden tot het afbreken van de berekeningen vanwege een 'terminal error'. In het geval van een waarschuwing neemt de routine zelf maatregelen die het hem mogelijk maken door te gaan. Genoemde maatregelen worden beschreven onder de betreffende waarde van de error parameter (IER). Of er een fout is opgetreden en welk soort fout, kan men slechts met zekerheid constateren na het uitschrijven van IER na de aanroep CALL ... van de IMSL-routine.

IMSL ROUTINE NAME	- IQHSCV
PURPOSE	- SMOOTH SURFACE FITTING WITH IRREGULARLY DISTRIBUTED DATA POINTS
USAGE	- CALL IQHSCV (XD,YD,ZD,ND,XI,NXI,YI,NYI,ZI,IZI, IWK,WK,IER)
ARGUMENTS	
XD	- VECTOR OF LENGTH ND. (INPUT) 1)
YD	- VECTOR OF LENGTH ND. (INPUT) 1)
ZD	- VECTOR OF LENGTH ND. (INPUT) 1) ZD(I) IS THE FUNCTION VALUE AT THE POINT (XD(I),YD(I)) FOR I=1,...,ND.
ND	- NUMBER OF DATA POINTS. (INPUT) ND MUST BE GREATER THAN OR EQUAL TO 4.
XI	- VECTOR OF LENGTH NXI. (INPUT) 1) THE SEQUENCE XI MUST BE MONOTONE INCREASING.
NXI	- NUMBER OF ELEMENTS IN XI. (INPUT) NXI MUST BE GREATER THAN OR EQUAL TO 1.
YI	- VECTOR OF LENGTH NYI. (INPUT) 1) THE SEQUENCE YI MUST BE MONOTONE INCREASING.
NYI	- NUMBER OF ELEMENTS IN YI. (INPUT) NYI MUST BE GREATER THAN OR EQUAL TO 1.
ZI	- NXI BY NYI MATRIX CONTAINING THE INTERPOLATED VALUES. (OUTPUT) ZI(I,J) IS THE ESTIMATED VALUE AT THE POINT (XI(I),YI(J)) FOR I=1,...,NXI J=1,...,NYI.
IZI	- ROW DIMENSION OF THE MATRIX ZI EXACTLY AS SPECIFIED IN THE DIMENSION STATEMENT IN THE CALLING PROGRAM. (INPUT)
IWK	- INTEGER WORK VECTOR OF LENGTH 31*ND+NXI*NYI. 1)
WK	- REAL WORK VECTOR OF LENGTH 6*ND. 1)
IER	- ERROR PARAMETER. (OUTPUT) TERMINAL ERROR IER = 129, ND IS LESS THAN 4, OR NXI IS LESS THAN 1, OR NYI IS LESS THAN 1. IER = 130, ALL DATA POINTS ARE COLLINEAR IER = 131, SOME DATA POINTS ARE IDENTICAL, THAT IS, XD(I)=XD(J) AND YD(I)=YD(J) FOR SOME I AND J, I NOT EQUAL TO J AND I=1,...,ND, J=1,...,ND.
PRECISION/HARDWARE	- SINGLE AND DOUBLE/H32 - SINGLE/H36,H48,H60

FIG.3;2. Gedeelte van de beschrijvins van een IMSL-routine.

1) 'LENGTH' is hier het aantal elementen van een vector (reeks).

V O O R B E E L D 1

PROGRAM VBINSL1

```

! Het toepassen van de IMSL-routine IQHSCV voor inter/extrapolatie.
! Het berekenen van de functiewaarden Z(Xi,Yj) op basis van willekeurig
! verspreide data-punten (XD,YD,ZD).
! De punten (Xi,Yj) liggen in een 'rechthoekig' patroon.
    
```

```

PARAMETER (NDATA=7, NX=4, NY=2)
DIMENSION XD(NDATA), YD(NDATA), ZD(NDATA), Xi(NX), Yj(NY),
*          Z(NX,NY), IWK(31*NDATA+NX*NY), WK(6*NDATA)
DATA XD / 10, 30, 20, 70, 80, 100, 110 /
DATA YD / 80, 20, 60, 70, 10, 60, 30 /
DATA ZD / 12, 14, 21, 25, 17, 22, 19 /
DATA Xi / 15, 50, 90, 130 /
DATA Yj / 25, 55 /

CALL IQHSCV (XD,YD,ZD,NDATA,Xi,NX,Yj,NY,Z,NX,IWK,WK,IER)

PRINT*
PRINT*, 'D A T A - P U N T E N -----'
PRINT*, '      XD          YD          ZD'
DO 1 L=1,NDATA
1 PRINT*, XD(L),YD(L),ZD(L)
PRINT*
PRINT*, 'B E R E K E N D E   Z - W A A R D E N -----'
PRINT*, '      X          Y          Z(berekend)'
DO 2 i=1,NX
DO 2 j=1,NY
2 PRINT*, Xi(i), Yj(j), Z(i,j)
PRINT*, 'IER = ',IER
END
    
```

```

D A T A - P U N T E N -----
      XD          YD          ZD
10.00000      80.00000      12.00000
30.00000      20.00000      14.00000
20.00000      60.00000      21.00000
70.00000      70.00000      25.00000
80.00000      10.00000      17.00000
100.00000     60.00000      22.00000
110.00000     30.00000      19.00000
    
```

```

B E R E K E N D E   Z - W A A R D E N -----
      X          Y          Z(berekend)
15.00000     25.00000      9.958502
15.00000     55.00000     19.05066
50.00000     25.00000     22.73410
50.00000     55.00000     28.41000
90.00000     25.00000     23.04803
90.00000     55.00000     25.70829
130.00000    25.00000      9.180542
130.00000    55.00000     9.727385
IER =                0
    
```

=====
 Bovenstaand programma levert, onder gebruikmaking van de gegevens zoals opgenomen in de programmatext, de vermelde uitkomsten op het beeldscherm. Het laatste resultaat, IER = 0, geeft aan dat er geen fouten optraden.

PROGRAM VBINSL2

```

.....
! Het integreren van de differentiaal-vergelijking  $y' = f(x,y)$  !
! (randvoorwaarde  $x=X0$ ,  $y=Y0$ ) voor gegeven  $x=XEND$ , gebruik !
! makend van de IMSL - routine DVERK. !
! De functie f verstrekt men in de vorm SUBROUTINE FCN (...) !
.....

```

```

EXTERNAL FCN
PARAMETER (N=1) !--> N=1 curven  $y'=f(x,y)$ 
DIMENSION Y(N),YPRIME(N),C(24),W(N,9)
DATA x,y /1.0, 0.5 / !--> randvoorw: ( $x=1,y=0.5$ )
DATA XEND /2.0 / !--> x waarvoor te integreren
DATA TOL /0.01/ !--> onnauwkeurigheid y

DO 1 J= 1,8
IND = 1 !--> besturings-variabele IND:
! wordt <0 bij nauwk.-probl.
! wordt +3 indien seen probl.

CALL DVERK (N,FCN,x,y,XEND,TOL,IND,C,N,W,IER)
XEND = XEND + 0.1 !--> integratie voor  $x = 2(.1)2.7$ 
DO 1 I= 1,N
1 PRINT*, 'X= ',X, ' Y= ',Y(I), ' IND=',IND !--> na elke integratie wordt IND
END ! weersgeven ter controle

```

```

-----
SUBROUTINE FCN (N,X,Y,YPRIME) !'''SUBROUTINE FCN'''!
REAL Y(N),YPRIME(N) !
DO I=1,N ! De diff.-vgl. is !
YPRIME(I)= -X * Y(I) !  $y' = -x * y$  !
END DO !-----!
END

```

X=	2.000000	Y=	0.1115710	IND=	3
X=	2.100000	Y=	9.0891019E-02	IND=	3
X=	2.200000	Y=	7.3307388E-02	IND=	3
X=	2.300000	Y=	5.8537155E-02	IND=	3
X=	2.400000	Y=	4.6277780E-02	IND=	3
X=	2.500000	Y=	3.6221839E-02	IND=	3
X=	2.599999	Y=	2.8068909E-02	IND=	3
X=	2.699999	Y=	2.1534640E-02	IND=	3

De IMSL-routine DVERK integreert een differentiaal-vergelijking van het type $y'=f(x,y)$ voor gegeven x onder de voorwaarde $(x,y)=(X0,Y0)$. Hier wordt de routine toegepast voor $x=2(.1)3$ ter berekening van y conform de differentiaalvergelijking. Deze definieert men in de subroutine FCN, die als eerste argument in DVERK voorkomt. In DVERK komt naast de error parameter IER de indicator IND voor. Hiermee kan men de rekentechniek beïnvloeden door waarden mee te geven. Aan de andere kant retourneert DVERK een indicatie; in het voorbeeld duidt IND=3 er op dat de berekening goed is verlopen. Het handboek geeft een uitgebreide beschrijving.

```

## Waar het datatype niet wordt vermeld in het handboek, doet men er
## goed aan het standaardtype volgens de besinletter van de naam te
## hanteren. In dit voorbeeld leidt met name de declaratie INTEGER*2
## IER,IND tot onjuiste resultaten omdat voor de betreffende variabelen
## stilzijgend het datatype INTEGER*4 geldt.

```

V O O R B E E L D 3

PROGRAM VBIMSL3

```

.....
! Het plotten van de reeks X tegen de reeksen Y1,Y2,...,Ym
! gebruik makend van de IMSL-routine USPLO.
! Een algemeen opschrift en de labels bij de assen worden
! meeseseven evenals de plot-symbolen voor Y1,Y2,...;
! ook de lensten van de bijschriften moet men verstrekken.
! De plot-resultaten bewaart IMSL op een nieuwe file FOR002.DAT.
!
-----

```

```

PARAMETER (N=7, M=2, INC=1, IOPT=0)
DIMENSION X(N), Y(N,M), RANGE(4)
DATA X / 12, 34, 6, 23, 43, 33, 5/
DATA Y / 15, 40, 9, 30, 58, 31, 8,
*      11, 30, 8, 27, 51, 27, 6/
.....
CALL USPLO (X, Y, N, N, M, INC,
*          'PLOT BY IMSL',12,
*          'INDEPENDENT VARIABLE X', 22,
*          'DEPENDENT VARIABLES 1,2',23,
*          RANGE,'12', IOPT, IER)
.....
END
-----

```

De IMSL-routine USPLO vervaardigt in eenzelfde figuur een plot van een reeks X (lenste N) tegen een aantal Y-reeksen (M stuks ter lenste N). Deze Y-reeksen worden in het programma aangeboden als de kolommen van een N*M matrix Y. Er worden hoogstens tien reeksen geplot. Daarbij kan men de reeksen volledig plotten (INC=1) of steeds een aantal getallen, INC-1 stuks, overslaan. Het eerste X,Y-paar wordt zonder meer geplot. Een andere selectie-mogelijkheid is dat men aangeeft binnen welk interval van X en/of Y het plotten moet plaatsvinden. Hiertoe dient de reeks RANGE waarin men de interval-grenzen van X resp. Y aangeeft. In het voorbeeld werd dat niet gedaan, waardoor ze 0.0 blijven en de routine de uitersten zelf bepaalt.

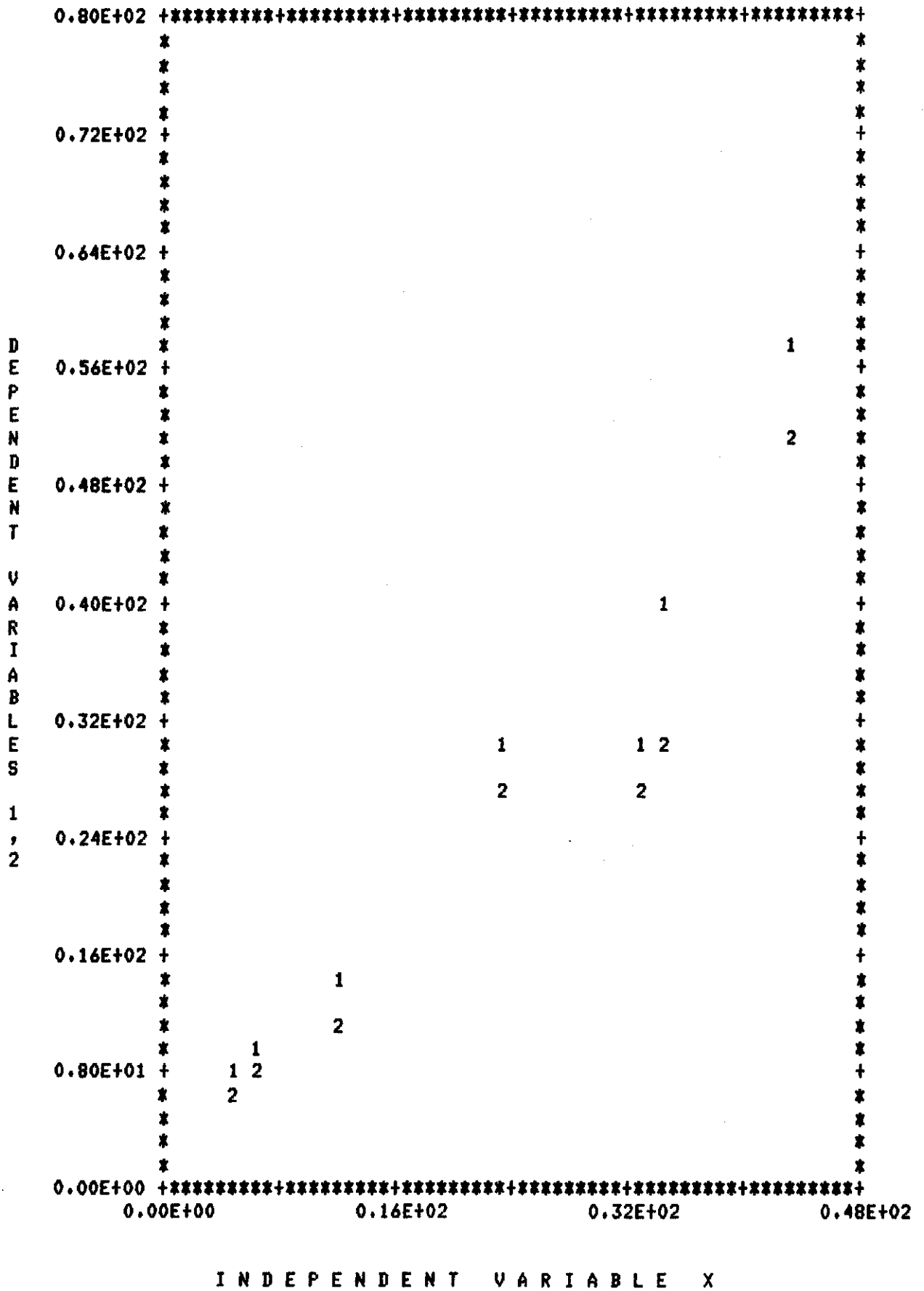
Verder is er de mogelijkheid de plot smal of breed te laten uitvoeren: IOPT=0 levert een breedte van 80 tekens, IOPT=1 levert er 129. Het plot-resultaat wordt door IMSL geplaatst op een nieuwe file FOR002.DAT die men naar verkiezing op het beeldscherm kan brengen of kan uitschrijven met een reseldrukker.

Het is mogelijk een algemeen opschrift, bijschriften en plot-symbolen op te geven.

Helaas is voor VAX-computers het character datatype niet beschikbaar bij IMSL-routines. Daarom moeten tekstregels als hollerith-grootheden worden gehanteerd en kan men niet gebruikmaken van variabele lenste zoals bij character-grootheden. Dus moet bij een hollerith-grootheid die optreedt als argument van een IMSL-routine het aantal tekens worden meeseseven, althans voorzover dit aantal niet uit anderen hoofde bekend is.

Met name in dit voorbeeld wordt in de routine USPLO een viertal hollerith-argumenten toegepast: een algemeen opschrift, bijschriften bij de assen en een reeks plot-symbolen. Onmiddellijk na elk van de op- en bijschriften wordt, eveneens als argument, het betreffende aantal tekens vermeld; het aantal plot-symbolen daarentegen, is bekend want dat moet overeenkomen met het aantal te plotten Y-reeksen.

Elk op- en bijschrift kan men onderdrukken door de lenste ervan 0 te maken.



```

PROGRAM VBINSL4
.....
! Regressie-analyse  $Y=a.X+b$  met de IMSL-routine RLONE. !
!-----!
    
```

```

PARAMETER (N=7, NN=7)
DIMENSION XY(N,2),ALBAP(3),DES(5),ANOVA(14),STAT(9),PRED(NN,7)
DIMENSION Y(N)

DATA XY /1, 3, 4, 5, 6, 9, 10,
*      3, 5, 6, 7, 8, 8, 9/
DATA IPRED /1/
DATA ALBAP /0.10, 0.10, 0.10 /
DATA (PRED(i,1),i=1,NN)/1,3,4,5,6,9,10/

DATA Y /3, 5, 6, 7, 8, 8, 9/
CALL RLONE(XY,N,N,IMOD,IPRED,ALBAP,DES,ANOVA,STAT,PRED,NN,NN,IER)
    
```

```

..... INPUT .....
! sedevens X
! sedevens Y
! gevraagd: BETR.INTERVAL
! risico-niveaus a,b;Yber
! X-waarden waarbij Y en
! betr.int. te berekenen !
!-----!
    
```

```

STAF = SQRT(ANOVA(8))
PRINT*
PRINT*, 'VEREFFENING: Y = aX + b'
PRINT*, 'STANDAARD AFW. = ',STAF
PRINT*, 'VRIJHEIDSGRADEN = ',ANOVA(2)
PRINT*, 'CORRELATIE-COEFF.= ',DES(5)
PRINT*
PRINT*, 'a = ',STAT(1), 'st.afw.',STAT(2)
PRINT*, 'b = ',STAT(5), 'st.afw.',STAT(6)
    
```

```

!***OUTPUT OP TERMINAL***
!
!.Probleem-aanduiding
!.stand.afw. t.o.v. lijn
!.bijbehorende vrijh.gr.
!.correlatie-coefficient
!
!.regr.coeff., stand.afw.
!.intercept , stand.afw.
!.TABEL:X,Y,Yber,Interval
!.IER=0 als vereff. o.k.
!-----!
    
```

```

PRINT*
PRINT*, ' X Y Y(ber) 10% INTERVAL Y(ber) '
PRINT*, '-----|-----|-----|-----|-----|'
DO 1 I=1,NN
1 PRINT9, PRED(I,1), Y(I), PRED(I,2), PRED(I,4),PRED(I,5)
9 FORMAT (F7.3, 2F10.3, 3X, 2F10.3)
PRINT*, 'IER = ',IER
END
    
```

```

VEREFFENING: Y = aX + b
STANDAARD AFW. = 0.7984360
VRIJHEIDSGRADEN = 5.000000
CORRELATIE-COEFF.= 0.9359710

a = 0.6041666 st.afw. 0.1016359
b = 3.291667 st.afw. 0.6288769
    
```

X	Y	Y(ber)	10% INTERVAL Y(ber)	
----- ----- ----- ----- -----				
1.000	3.000	3.896	2.804	4.988
3.000	5.000	5.104	4.319	5.890
4.000	6.000	5.708	5.034	6.383
5.000	7.000	6.313	5.698	6.927
6.000	8.000	6.917	6.297	7.536
9.000	8.000	8.729	7.778	9.680
10.000	9.000	9.333	8.217	10.450
IER = 0				

De routine RLONE voert een regressie-analyse $Y = a.X + b$ uit. Naar wens wordt hierbij het intercept b al dan niet op nul sefixeerd. Naast samenvattende grootheden (gemiddelde en standaard-afwijking van X en Y , correlatie-coefficient, regressie-coefficient en intercept met hun standaard-afwijkingen) worden significantie-toetsen beschikbaar gesteld en kan men betrouwbaarheids-intervallen laten vastlessen.

De X - en Y -reeksen worden seleverd als eerste, resp. tweede kolom van een matrix, in het voorbeeld de matrix XY . Het argument IMOD bepaalt of het intercept als variabele wordt behandeld. Omdat IMOD hier niet nader wordt sespecificieerd, blijft hij 0, wat volgens de beschrijving inhoudt dat het intercept als variabele in de vereffening meedoet. Als IPRED niet 0 is, wordt om betrouwbaarheids-intervallen gevraagd; de risico-niveaus definieert men in de reeks ALBAP. De X -waarden waarvoor het betrouwbaarheids-interval voor Y moet worden berekend, geeft men op in de eerste kolom van de matrix PRED. De overige kolommen van deze matrix dienen voor 't opbergen van vereffende Y -waarden en betrouwbaarheidsintervallen. DES, ANOVA en STAT bevatten de overige resultaten; zie het handboek. Bij de PRINT-opdrachten in het voorbeeld wordt uit de zojuist genoemde grootheden seput.

In deze toepassing wordt het betrouwbaarheids-interval voor Y berekend bij de X -waarden uit de XY -matrix. Deze matrix echter, wordt volgens de beschrijving van de routine RLONE tijdens de vereffening 'vernietigd' wanneer IMOD = 0, zoals hier het geval is. Om toch over de Y -waarden te kunnen beschikken bij 't uitschrijven van de tabel in nevenstaande output, werd derhalve een reeks genaamd Y met de oorspronkelijke seevens gevuld voordat RLONE werd aangeropen.

V O O R B E E L D 5

PROGRAM VBIMSL5

```

.....
! Niet-lineaire vereffening m.b.v. de IMSL-routine ZXMWD.
! Curve: Y = P1 - P2.P3**X.
! Methode: minimaliseren van de kwadratensom van afwijkenen
! van de y-waarden t.o.v de curve; de kwadratensom
! wordt berekend in de gebruikers-subroutine FCN.
.....

```

```

EXTERNAL FCN
PARAMETER (ND=7,N=3)
DIMENSION A(N), B(N), P(N), WORK(N*(N+1)/2+11*N), IWORK(N)
COMMON X(99), Y(99)

```

```

..... INPUT .....
DATA (X(i),i=1,ND) /1,3,4,5,6,9,10/      ! sesevens X
DATA (Y(i),i=1,ND) /3,5,6,7,8,8, 9/      ! sesevens Y
DATA (A(J),J=1,N) / 0, 0, 0 /            ! ondergrens der par's
DATA (B(J),J=1,N) /77,77, 1 /           ! bovengrens der par's
DATA NSIG / 3 /                           ! betrouwbare cijfers
NSRCH = MIN(2**N+5,100)
.....

```

```
CALL ZXMWD (FCN,N,NSIG,A,B,NSRCH,P,F,WORK,IWORK,IER)
```

```

IDF = ND - N
STAND = SQRT(F/IDF)
PRINT*                                     !""OUTPUT OP TERMINAL""
PRINT*                                     !
PRINT*, 'VEREFFENING: Y = a - b.c**X'     ! probleem-aanduiding
PRINT*, 'STANDAARD AFW. = ', STAND        ! stand.afw. tov. curve
PRINT*, 'VRIJHEIDSGRADEN = ', IDF        ! bijbehorende vrijh.gr.
PRINT*                                     !
PRINT*, 'a = ', P(1)                      ! parameterwaarde
PRINT*, 'b = ', P(2)                      ! parameterwaarde
PRINT*, 'c = ', P(3)                      ! parameterwaarde
.....

```

```

PRINT*, '      X          Y          Y(vereffend)
DO I=1,ND
YCALC = P(1) - P(2) * P(3)**X(I)
PRINT*, X(I), Y(I), YCALC
END DO
PRINT*, 'IER = ', IER
END
..... OUTPUT OP TERMINAL""
!
! X, Y, Y(vereff.)
! -
! IER=0 bij geslaasde
! -----vereffening-----

```

```

C-----
SUBROUTINE FCN (N,P,F)
PARAMETER (ND=7)
DIMENSION P(N)
COMMON X(99), Y(99)
F = 0.
DO I=1,ND
YCALC = P(1) - P(2) * P(3)**X(I)
F = F + (Y(I)-YCALC)**2
END DO
END
..... SUBROUTINE FCN ""
!
! In de subroutine FCN
! definieert men de te
! minimaliseren functie,
! hier een kwadratensom
! van afwijkenen.
.....

```

VEREFFENING: $Y = a - b.c^{**X}$
 STANDAARD AFW. = 0.4518596
 VRIJHEIDSGRADEN = 4

a = 9.717532
 b = 8.514054
 c = 0.8028187

X	Y	Y(vereffend)
1.000000	3.000000	2.882290
3.000000	5.000000	5.312096
4.000000	6.000000	6.180766
5.000000	7.000000	6.878150
6.000000	8.000000	7.438023
9.000000	8.000000	8.538043
10.00000	9.000000	8.770617

IER = 0

De IMSL-routine ZXMWD minimaliseert de functie F zoals die wordt gedefinieerd in de subroutine FCN. Deze subroutine verstrekt de gebruiker zelf. In dit voorbeeld is genoemde functie de som van kwadraten van de verschillende Y-waarden en de waarde volgens een curve (YCALC).

De parameterwaarden die uit de minimalisering voortvloeien kunnen daarom worden beschouwd als goede schattingen van de parameters in de regressiecurve $Y = a - b.c^{**X}$.

Bij deze niet-lineaire vereffening worden de parameter-waarden ingeperkt binnen op te geven onder- en bovengrenzen (A en B). De eis wordt hier gesteld dat de oplossingen der parameters tot in drie significante cijfers nauwkeurig zullen zijn (NSIG is 3). Men heeft bij gebruik van de routine ZXMWD geen besinschattingen te leveren, die worden automatisch voortgebracht (NSRCH stuks). Beschikt men over goede parameterschattingen dan is 't altijd verstandig nauwe grenzen aan te geven omdat men daarmee het proces versnelt.

Door toepassing van ZXMWD wordt een robuust vereffenings-programma verkregen dat niet snel last heeft van rekentechnische problemen. Dit in tegenstelling tot meer verfijnde vereffeningsstechnieken die vaak sneller werken maar daarbij besinschattingen voor de parameters vereisen.

Een vlekkeloos verlopen rekenprocedure leidt tot IER=0, maar het kan voorkomen dat het rekenproces erg langzaam naar de eindoplossing convergeert of dat convergentie naar een plaatselijk optimum optreedt. In dat soort gevallen is IER = 129, 130 of 131 en worden de berekeningen afgebroken. Het seneneren van een groter aantal besinschattingen, dus versnotten van NSRCH, kan dan de remedie zijn. Een andere mogelijkheid is het beter inperken van de parameterwaarden door scherpere besrenzingsen A en B.