'N ALGORITME VIR KLEINSTEKWADRATE-BENADERING

deur

AUGUST SCHAAFSMA

Voorgelê ter vervulling van 'n deel van die vereistes vir die graad

M.Sc. (Toegepaste Wiskunde)

in die Fakulteit Wis- en Natuurkunde,

Universiteit van Pretoria

PRETORIA

Oktober 1977

INHOUDSOPGAWE

Bladsy

Bladsy

1. INLEIDING

1.1 Doel

Die doel van hierdie studie is, om 'n rekenaarprogram te ontwikkel vir die passing van 'n lineêre en/of nie-lineêre funksie, deur 'n aantal eksperimentele datapunte, sodanig dat die som S(a) van die kwadrate van die verskille tussen die eksperimentele en die berekende funksiewaardes in die ooreenstemmende punte 'n minimum is. In wese reduseer die taak dus na die bepaling van die koëffisiënte a_i, i=1,2, --, m in die passingsfunksie F.

Die ongepubliseerde rekenaarprogram PASF [18] is tot dusver gebruik om krommepassings te doen. In dié program word die minimeringsroetine MINIM [4] gebruik wat afgelei is uit die steilste dalingsmetode van Fletcher en Powell [19]. Daar is egter gevind dat die metode relatief stadig na 'n oplossing konvergeer as daar baie eksperimentele data is en/of 'n passingsfunksie met 'n groot aantal koëffisiënte gebruik word. Verder word daar nie in die program voorsiening gemaak vir die berekening van enige statistiese gegewens of die gebruik van 'n grafiektrekker om die eksperimentele data te stip of die passingsfunksie te teken nie.

In die nuwe program PASPLT moet daar dus gepoog word om die nadele en tekortkominge van die ou program uit te skakel. Dus moet daar aan die volgende vereistes voldoen word:

(a) Die metode moet vinnig na 'n oplossing konvergeer al bevat die passingsfunksie 'n groot aantal koëffisiënte a_i en al sou daar ook baie eksperimentele data wees.

(b) Die rekenaargeheue wat gebruik word, moet tot 'n minimum beperk word, dus moet dit maklik wees om die orde van 'n matriks te kan verander.

(c) Die program moet verskillende statistiese parameters kan bereken. Byvoorbeeld korrelasiekoëffisiënte, gemiddeldes, standaardafwykings en regressiekoëffisiënte.

(d) Ook moet die program die fasiliteit besit om uit die gegewe data, die nodige veranderlikes wat in die passing gebruik word, tussen gegewe grense uit te soek.

(e) Die koëffisiënte a_i, $i=1,2,--,n$ van die passingsfunksie, moet tussen gegewe grense beperk kan word, en moet selfs gedurende die minimeringsproses konstant gehou kan word.

(f) Daar moet in die program voorsiening gemaak word vir die stip van eksperimentele data en om die passingsfunksie te kan teken.

(g) In dieselfde lopie moet verskillende funksies gepas kan word.

1.2 Uiteensetting

In hoofstuk 2 word die teorie van die pseudo-inverse van 'n matriks sowel as die Householder-transformasies wat in die passingstegniek gebruik word, bespreek.

In hoofstuk 3 word die Newton-Gauss-Marquardt-passingstegniek bespreek en gewys hoe ons hiervan in ons tegniek gebruik maak. Daar word ook op verskille tussen verskillende bestaande tegnieke gewys.

Die program PASPLT word in hoofstuk 4 beskryf en 'n lys van die verskillende Fortran- en Assemblerroetines verskyn respektiewelik in Bylae A en B. Numeriese voorbeelde ter illustrasie van die verskillende moontlikhede word in Bylae C gedoen.

In hoofstuk 5 word verskillende minimeringsprogramme met PASPLT vergelyk. Dit word gedoen aan die hand van voorbeelde in [20] en [21].

2. PSEUDO-INVERSE EN HOUSHOLDER-TRANSFORMASIES

2.1 Die Norm van 'n Matriks

Definisie

Die norm $||A||$, van 'n (mxn)-matriks A is 'n waarde wat aan die matriks toegeken word, en besit die volgende eienskappe:

(1) $||A|| \geq 0$, $||A|| = 0$ as en slegs as $A=0$

(2) $||cA|| = |c| \; ||A||$ vir c 'n reële getal

(3) $||A+B|| \leq ||A|| + ||B||$

(4) $||AB|| \leq ||A|| \; ||B||$

Eienskap (3) word driehoeksongelykheid genoem en eienskap (4) die Schwarz ongelykheid.

Die norm kan op verskillende maniere gedefinieer word. Die twee bekendste definisies is die spektraalnorm en die Euklidiese norm, nl.

(i) Die Euklidiese norm $||A||_E = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2\right)^{\frac{1}{2}}$

(ii) Die spektraalnorm $||A||_S = \max_i \left[\lambda_i \, (AA^T)\right]^{\frac{1}{2}}$, waar $\lambda_1, ---, \lambda_m$ die eiewaardes van AA^T is, en A^T die getransponeerde van A voorstel.

Kreyszig [13, p.281] het bewys dat dié norms invariant is ten opsigte van ortogonale transformasies.

2.2 Die Pseudo-inverse

Die pseudo-inverse A^+ van 'n (mxn) matriks A speel 'n belangrike rol in die oplos van kleinstekwadraat-probleme. Beskou 'n (mxn)-matriks A as 'n lineêre afbeelding van E_n in E_m.

$$Ax = y \text{ met } x \in E_n,\ y \in E_m. \qquad (2.1)$$

As die afbeelding een-eenduidig en op is, dan bestaan daar 'n unieke inverse afbeelding.

As A egter nie een-eenduidig is nie maar wel op is, bestaan daar meer as een oplossing vir 'n gegewe y, en as A een-eenduidig is en nie op is nie, kan y buite die waardegebied van A wees.

Met 'n kleinstekwadraat-benadering probeer ons 'n unieke x vind waarvan die afbeelding die naaste aan y is. As A nie een-eenduidig of op is nie, kan daar meer as een vektor wees waarvan die afbeeldings ewe ver van y is. Met behulp van die pseudo-inverse word (2.1) opgelos, sodat die unieke oplossing x verkry word.

<u>Definisie 2.2.1</u>

Die pseudo-inverse van 'n (mxn)-matriks A, is 'n (nxm)-matriks A^+, wat $y \in E_m$ op 'n unieke vektor $x \in E_n$ afbeeld sodanig dat

(i) $x \in \{w : \|Aw-y\|_E\}$ 'n minimum is

(ii) x is die element van bogenoemde stelsel met minimum norm.

<u>Definisie 2.2.2</u>

Boullion [17,p.1] definieer die pseudo-inverse van 'n (mxn)-matriks A, as 'n (nxm)-matriks, A^+, wat die Penrose-voorwaardes [12,p.406] bevredig.

(i) $AA^+ A = A$

(ii) $A^+ AA^+ = A^+$

(iii) $(A^+ A)^* = A^+A$

(iv) $(A A^+)^* = A A^+$ waar $(A)^*$ die toegevoegde, getransponeerde van A voorstel.

Penrose [12] bewys dat bogenoemde vergelykings 'n unieke oplossing A^+ vir enige matriks A het.

<u>Stelling 2.2.1</u> Die Penrose-vergelykings het 'n unieke oplossing A^+ vir enige matriks A.

<u>Bewys</u>:

Ons moet bewys dat die vier vergelykings

$$AXA = A \tag{1}$$

$$XAX = X \tag{2}$$

$$(AX)^* = AX \tag{3}$$

$$(XA)^* = XA \tag{4}$$

'n unieke oplossing vir enige A het.

Die vergelykings (2) en (3) kan vervang word deur die vergelyking:

$$XX^*A^* = X \tag{5}$$

want (2) in (3) gesubstitueer gee (5) en (3) in (2) gesubstitueer gee ook (5).

Die vergelyking (1) en (4) kan vervang word deur die vergelyking

$$XAA^* = A^* \tag{6}$$

want (1) in (4) gesubstitueer gee (6) en (4) in (1) gesubstitueer gee (6) Ons moet dus 'n X vind wat (5) en (6) bevredig.

So 'n X bestaan indien 'n B gevind kan word sodanig dat

$$BA^*AA^* = A^* \tag{7}$$

As $X=BA^*$ word (6) bevredig. Deur XA uit (4) in (6) te stel volg dat:

$$A^*X^*A^* = A^*$$

$$BA^*X^*A^* = BA^*$$

Dus bevredig $X = BA^*$ ook (5).

Die uitdrukkings A^*A, $(A^*A)^2$, $(A^*A)^3$, ----- kan nie almal lineêr onafhanklik wees nie. Dus moet

$$\lambda_1 A^*A + \lambda_2 (A^*A)^2 + --- + \lambda_k (A^*A)^k = 0 \qquad (8)$$

waar alle $\lambda_1, ----, \lambda_k$ nie nul is nie.

Laat λ_r die eerste λ wees wat nie nul is nie en stel

$$B = -\lambda_r^{-1} \{\lambda_{r+1} I + \lambda_{r+2} A^*A + -- + \lambda_k (A^*A)^{k-r-1}\}.$$

Dus (8) word $B(A^*A)^{r+1} = (A^*A)^r$,

en deur herhaaldelike toepassing van die feit dat (i) $SQQ^* = RQQ^*$ impliseer dat $SQ = RQ$ (ii) $SQ^*Q = RQ^*Q$ impliseer dat $SQ^* = RQ^*$ volg

$$BA^*AA^* = A^*.$$

Dus bestaan daar 'n B wat (7) bevredig.

Laat nou X en Y twee verskillende oplossings wees, sodanig dat

X (5) en (6) bevredig d.w.s. $X = XX^*A^*$

en $A^* = XAA^*$,

en Y bevredig $Y = A^*Y^*Y$

$A^* = A^*AY$.

Dus is

$$X = XX^*A^* = XX^*A^*AY = XAY = XAA^*Y^*Y = A^*Y^*Y = Y.$$

Dus is X die unieke oplossing vir die vergelykings (1), (2), (3) en (4).

Hierdie unieke oplossing X word die pseudo-inverse van A genoem en geskryf as A^+. Die matriks A hoef nie noodwendig vierkantig te wees nie en kan selfs singulier wees.

Stelling 2.2.2 Definisies 2.2.1 en 2.2.2 is ekwivalent.

Bewys Veronderstel G is die pseudo-inverse van die (mxn)-matriks A volgens definisie 2.2.1.

Beskou die stelsel lineêre vergelykings $Ax=y$, en gestel die rang van A is p.

Laat $\hat{x}$ die vektor van die stelsel $\|Ax-y\|$ wees wat die minimum norm besit en laat $\hat{x} = Gy$ wees.

Vir minimum norm het ons

$$\|r\|^2 = \|Ax-y\|^2 = (Ax-y)^T(Ax-y).$$

Vir 'n minimum is

$$\frac{\partial}{\partial x_j}\|r\|^2 = 2e_j^T A^T A\hat{x} - 2e_j^T A^T y = 0 \qquad j=1,2,---,\ell$$

Dus $\quad A^T A\hat{x} = A^T y$,

d.w.s. $\quad \hat{x} = (A^T A)^{-1}A^T y = Gy$

waar $\quad G = (A^T A)^{-1}A^T$.

G bevredig die Penrose voorwaardes:

(1) $AGA = A(A^TA)^{-1}A^TA = A$

(2) $GAG = (A^TA)^{-1}A^TAG = G$

(3) $(GA)^T = A^TG^T = A^TA(A^TA)^{-1} = (A^TA)^{-1}A^TA = GA$

(4) $(AG)^T = \left[A(A^TA)^{-1}A^T\right]^T = A(A^TA)^{-1}A^T = AG$.

Dus is G die pseudo-inverse van A en dus uniek.

Definisie 2.2.1 impliseer dus definisie 2.2.2.

Veronderstel G is die pseudo-inverse van matriks A volgens definisie 2.2.2, d.w.s. G bevredig die Penrose voorwaardes.

(1) $AGA = A$

(2) $GAG = G$

(3) $(GA)^T = GA$

(4) $(AG)^T = AG$.

Ons het bewys dat die oplossing G van bogenoemde vergelykings uniek is en dit kan maklik aangetoon word dat $G=(A^TA)^{-1}A^T$ die voorwaardes bevredig. Dus is

$$G = (A^TA)^{-1}A^T$$

uniek.

Verder het ons reeds aangetoon dat die norm van die stelsel $y = Ax$ 'n minimum is, as $\hat{x} = (A^TA)^{-1}A^Ty$.

Definisie 2.2.2 impliseer dus definisie 2.2.1.

Albert en Sitler [14] toon dat die pseudo-inverse A^+ ook gedefinieer kan word as $A^+ = \lim_{\varepsilon \to 0} (A^T A + \varepsilon^2 I)^{-1} A^T$.

2.3 Householder-transformasie

Die doel van 'n Householder-transformasie is om met behulp van ortogonale matrikse, 'n simmetriese matriks A te transformeer na 'n simmetriese tridiagonale stelsel, d.i. 'n matriks met nie-nul elemente in die hoof- en die twee aangrensende diagonale terwyl alle ander elemente nul is.

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & 0 \\ a_{12} & a_{22} & a_{23} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & a_{n-1,n} \\ 0 & \cdots & 0 & a_{n-1,n} & a_{nn} \end{bmatrix}$$

A is tridiagonaal en simmetries.

Vir so 'n stelsel is dit redelik maklik om die eiewaardes te bereken.

Deur bogenoemde transformasie word al die elemente van 'n ry en kolom, behalwe die tridiagonaalelemente gelyk aan nul gemaak.

Die verskillende stappe kan dus soos volg voorgestel word. Laat A 'n (4x4) matriks wees

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \xrightarrow{\text{stap } 1} \begin{bmatrix} \times & \times & 0 & 0 \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix} \xrightarrow{\text{stap } 2} \begin{bmatrix} \times & \times & 0 & 0 \\ \times & \times & \times & 0 \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix}$$

As A egter nie simmetries is nie, sal 'n bo- of onderdriehoeksmatriks ontstaan wat soos volg voorgestel kan word.

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \xrightarrow{\text{stap } 1} \begin{bmatrix} \times & \times & 0 & 0 \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \xrightarrow{\text{stap } 2} \begin{bmatrix} \times & \times & 0 & 0 \\ \times & \times & \times & 0 \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}$$

Die vorm van die matriks wat so ontstaan, word die <u>Hessenberg</u>-vorm genoem.

<u>Metode</u>

Laat v 'n vektor wees sodat $v^Tv = 1$, dan is die matriks

$$P = I - 2vv^T \tag{2.3.1}$$

ortogonaal en simmetries, omdat

$$P^T = (I-2vv^T)^T = I - 2(vv^T)^T = I-2vv^T = P$$

en

$$PP^T = (I - 2vv^T)(I - 2vv^T)^T = (I-2vv^T)(I-2vv^T) =$$

$$I - 4vv^T+4vv^Tvv^T = I - 4vv^T+4vv^T = I.$$

Kies nou v sodanig dat die eerste $k-1$-komponente nul is, d.w.s.

$$v_k^T = \left(0, 0, --, v_k^{(k)}, v_k^{(k+1)}, --, v_k^{(n)}\right) \text{ dan is}$$

$$P_k = I - 2v_k v_k^T .$$

Stel $\quad A_k = P_k^T A_{k-1} P_k \qquad k = 2, ---, n-1$

en $\quad A_1 = A .$

Veronderstel die simmetriese matriks $A_{k-1} = (g_{ij})$ het behalwe vir die tridiagonaalelemente, nulle in die eerste k-2 rye en kolomme, dan kan ons A_{k-1} soos volg voorstel.

$$A_{k-1} = \left[\begin{array}{cccccccc}
g_{11} & g_{12} & 0 & \cdots & & & & 0 \\
g_{12} & g_{22} & g_{23} & \ddots & & & & \vdots \\
0 & g_{23} & \ddots & \ddots & \ddots & & & \vdots \\
\vdots & \ddots & \ddots & g_{k-2,k-2} & g_{k-2,k-1} & 0 & \cdots & 0 \\
\vdots & \text{(k-1)de ry} & \ddots & g_{k-2,k-1} & g_{k-1,k-1} & \cdots & \cdots & g_{k-1,n} \\
\vdots & & & 0 & \vdots & \ddots & & \vdots \\
0 & \cdots & \cdots & 0 & g_{k-1,n} & \cdots & \cdots & g_{n,n}
\end{array}\right]$$

$$= \left[\begin{array}{c|c}
A_{k-1}^{(1)} & A_{k-1}^{(2)} \\
\hline
A_{k-1}^{(3)} & A_{k-1}^{(4)}
\end{array}\right]$$

en $\qquad P_k$ is van die vorm

$$P_k = I - 2v_kv_k^T = \begin{array}{r} \\ \\ \\ \\ \\ \text{Kde ry} \\ \\ \\ \end{array} \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & & & \vdots \\ \vdots & & \ddots & 1 & 0 & \cdots & \cdots & 0 \\ \vdots & & & \ddots & & & & \\ \vdots & & & 0 & 1-2\left[v_k^{(k)}\right]^2 & \cdots & \cdots & 2v_k^{(n)}v_k^{(k)} \\ \vdots & & & \vdots & \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 & -2\,v_k^{(n)}v_k^{(k)} & \cdots & \cdots & 1-2\left[v_k^{(n)}\right]^2 \end{bmatrix}$$

$$\left[\begin{array}{c|c} I & 0 \\ \hline 0 & \bar{P}_K \end{array}\right].$$

Die doel is nou om die (n-k+1) elemente van $v_k^{(k)}, \cdots, v_k^{(n)}$ te kies sodat $v^Tv = 1$ en sodat die (n-k) nie-tridiagonaalelemente in die (k-1)-de ry en kolom van A_k nul is.

Definieer nou $S = \sum_{j=k}^{n} g_{k-1,j}^2 = \sum_{j=k}^{n} g_{j,k-1}^2$ (simmetrie) (2.3.2)

en laat $\left[v_k^{(k)}\right]^2 = \frac{1}{2}\left[1 \pm g_{k-1,k} / \sqrt{S}\right]$ (2.3.3)

en $v_k^{(j)} = \pm g_{k-1,j} / (2v_k^{(k)}\sqrt{S})$ $j = k+1, --, n$ wees. (2.3.4)

Die teken in (2.3.3) word gekies sodat die waarde 'n maksimum is. Dieselfde teken word in (2.3.4) gebruik.

Ons moet nou aantoon dat (2.3.2), (2.3.3) en (2.3.4) die vergelyking (2.3.1) bevredig en dat A_k nulle in die gewenste posisies het.

Laat

$$P_k = \begin{bmatrix} I & 0 \\ 0 & \bar{P}_k \end{bmatrix}$$ waar I van orde (k-1) x (k-1) is

en

$$\bar{P}_k = I - 2\bar{v}_k\bar{v}_k^T,$$ van orde (n-k+1) x (n-k+1)

met

$$\bar{v}_k = (v_k^{(k)}, --, v_k^{(n)})^T$$

Dan is $A_k = P_k^T A_{k-1} P_k = P_k A_{k-1} P_k$

$$= \left[\begin{array}{c|c} A_{k-1}^{(1)} & A_{k-1}^{(2)} \bar{P}_k \\ \hline \bar{P}_k A_{k-1}^{(3)} & \bar{P}_k A_{k-1}^{(4)} \bar{P}_k \end{array}\right].$$

Die boonste linkerkant van A_k is dieselfde as die van A_{k-1} en dit is tridiagonaal.

Verder is: $\bar{P}_k A_{k-1}^{(3)} = A_{k-1}^{(3)} - \bar{v}_k \bar{v}_k^T A_{k-1}^{(3)}$

en

$$A_{(k-1)}^{(3)} = \begin{bmatrix} 0 & 0 \cdots\cdots 0 & g_{k,\,k-1} \\ 0 & 0 \cdots\cdots 0 & g_{k+1,\,k-1} \\ \vdots & \vdots \qquad \vdots & \vdots \\ 0 & 0 \cdots\cdots 0 & g_{n,k-1} \end{bmatrix}.$$

Omdat die eerste (k-2) kolomme van $A_{k-1}^{(3)}$ nul is, volg dit dat die eerste (k-2) kolomme van $\bar{P}_k A_{k-1}^{(3)}$ ook nul moet wees en omdat A_k simmetries is, volg dit dat die eerste (k-2) rye van $A_{k-1}^{(2)} \bar{P}_k$ ook nul moet wees.

Dus het A_k nulle op al die plekke waar A_{k-1} nulle het.

Verder is:

$$v^T v = \sum_{i=k}^{n} v_i^2 = \tfrac{1}{2}\left[1 \pm g_{k-1,k}/\sqrt{S}\right] +$$

$$\sum_{j=k+1}^{n} g_{k-1,j}^2 \,/\, 2\sqrt{S}\,(\sqrt{S} \pm g_{k-1,k})$$

$$= \frac{1}{2\sqrt{S}}\left[\sqrt{S} \pm g_{k-1,k} + \frac{1}{\sqrt{S} \pm g_{k-1,k}}\,(S - g_{k-1,k}^2)\right]$$

$$= \frac{1}{2\sqrt{S}}\left[\sqrt{S} \pm g_{k-1,k} + \sqrt{S} \mp g_{k-1,k}\right]$$

$= 1$ en voldoen dus aan (2.3.1)

Ons moet nou nog aantoon dat die (k-1)-de kolom van $\bar{P}_k A_{k-1}^{(3)}$, behalwe die eerste komponente, nul is. Laat u_k hierdie kolom wees, dan is

$$u_k = \bar{a}_k - 2\bar{v}_k \bar{v}_k^T \bar{a}_k \text{ waar } \bar{a}_k = (g_{k,k-1}, --, g_{n,k-1})^T$$

die (k-1)de kolom van $A_{k-1}^{(3)}$ is.

Beskou nou die i-de komponent van u_k

waar i = k+1, ---, n

$$u_k^{(i)} = \bar{a}_k^{(i)} - 2v_k^{(i)} \bar{v}_k^T \bar{a}_k = g_{i,k-1} \mp v_k^T \bar{a}_k \frac{g_{k-1,i}}{v_k^{(k)} S} .$$

Verder is:

$$\bar{v}_k^T \bar{a}_k = v_k^{(k)} g_{k,k-1} \pm \frac{1}{2v_k^{(k)}\sqrt{S}} \sum_{j=k+1}^{n} g_{j,k-1}^2$$

$$= v_k^{(k)} \sqrt{S} \left[\frac{g_{k,k-1}}{\sqrt{S}} \pm \tfrac{1}{2} \frac{1}{(v_k^{(k)})^2 S} (S - g_{k,k-1}^2) \right]$$

$$= v_k^{(k)} \frac{\sqrt{S}}{\sqrt{S}} \left[g_{k,k-1} \pm \frac{(\sqrt{S}+g_{k,k-1})(\sqrt{S}-g_{k,k-1})}{(\sqrt{S} \pm g_{k,k-1})} \right]$$

$$= \pm v_k^{(k)} \sqrt{S}.$$

Dus is $u_k^{(i)} = g_{i,k-1} \mp (\pm g_{k-1,i}) = 0$ omdat

$$g_{i,k-1} = g_{k-1,i} \text{ (simmetrie).}$$

Dit volg dus dat A_k nulle het in die gewenste posisies.

In Franklin [15,p. 252] word 'n rekursieformule afgelei, om die karakteristieke polinoom van die Householder- en Hessenberg-matriks te verkry.

Laat H die Hessenberg-matriks wees met komponente $\{(h_{ij}), i,j = 1, --, n\}$ dan is die rekursieformule vir die karakteristieke polinoom

$$\phi_n(\lambda) = |H_n-\lambda I| = (h_{nn}-\lambda)\ \phi_{n-1}(\lambda) +$$

$$\sum_{r=1}^{n-1} (-1)^{n-r}\ h_{nr}\ \ h_{r,\ r+1}\ h_{r+1,r+2} \cdots h_{n-1,n}\ \phi_{r-1}(\lambda) \text{ vir } n\geq 2$$

$$\text{met } \phi_o(\lambda) = 1 \text{ en } \phi_1(\lambda) = h_{11}-\lambda\ . \qquad (2.3.5)$$

Bostaande formule vereenvoudig vir die Householder-matriks na

$$\phi_n(\lambda) = |H_n-\lambda I| = (h_{nn}-\lambda)\ \phi_{n-1}(\lambda)-h_{n,n-1}h_{n-1,r}\ \phi_{n-1}(\lambda)$$

$$\text{vir } n\geq 2 \text{ en met } \phi_o(\lambda) = 1 \text{ en } \phi_1(\lambda) = h_{11}-\lambda. \qquad (2.3.6)$$

2.3.1 Die Householder-matriks

Stelling 2.3.1.1

Laat v 'n vektor met m nie-nul elemente wees d.w.s. $v^T=(v_1, \cdots, v_m)$, dan bestaan daar 'n ortogonale matriks Q sodat

$$Qv = -\sigma\ ||v||\ e_1 \text{ met } e_1^T = (1,\ 0,\ \cdots,0)$$

en

$$\sigma = \begin{cases} +\ 1 \text{ as } v_1 \geq 0 \\ -\ 1 \text{ as } v_1 < 0\ . \end{cases}$$

Bewys: Definieer $u = v + \sigma\ ||v||\ e_1$

en

$$Q = I_m - \frac{2uu^T}{u^Tu}.$$

Dan is Q simmetries en ortogonaal

en

$$Qv = -\sigma||v||\ e_1$$

want

$$Q^TQv = -\sigma||v||\ Q^Te_1 = -\sigma||v|| \left[\left(I- \frac{2uu^T}{u^Tu}\right)e_1\right]$$

$$= -\sigma\ ||v||\ e_1 + 2\ ||v||\ \sigma\ \frac{uu^T}{u^Tu}\ e_1.$$

Dus $$v = -\sigma \|v\| e_1 + \frac{(v+\sigma\|v\|e_1)(v+\sigma\|v\|e_1)^T e_1}{v_1 + \sigma\|v\|}$$

$$= -\sigma \|v\| e_1 + \frac{(v+\sigma\|v\| e_1)(v_1+\sigma\|v\|)}{v_1 + \sigma\|v\|} .$$

Q word die Householder-transformasiematriks genoem.

2.4 Matriksontbinding

2.4.1 Die QR-ontbinding

Laat A 'n reële (mxn)-matriks wees

met $\ell = \min(m,n)$

$$k = \begin{cases} n, & m>n \\ m-1, & \text{andersins.} \end{cases}$$

A kan ontbind word in 'n produk $A=Q^TR$ waar

(i) Q 'n (mxm) ortogonale simmetriese matriks en

(ii) R 'n (mxn) bo-diagonaalmatriks is.

Q is 'n produk van k elementêre, simmetriese ortogonale Householder-matrikse.

D.w.s. $Q = Q_k Q_{k-1} --- Q_1$

waar
$$Q_i = I-2v_iv_i^T \text{ en } v_i^Tv_i = 1 \qquad (2.4.1)$$

Q_i transformeer die elemente van A_{i-1} in die i-de kolom onder die hoofdiagonaal na nulle, sonder om die nulle in die kolomme 1, 2, ---, i-1 te verander.

Dan is $A_{i-1} = Q_{i-1}\, Q_{i-2} \text{---}\, Q_1\, A$

en $A_i = Q_i\, A_{i-1}$

waar $A_o = A$ (2.4.2)

$K_j(A_i) = Q_i K_j(A_{i-1})$ waar $K_j(A_i)$ die j-de kolom van A_i is

d.w.s. $K_j(A_i) = K_j(A_{i-1}) - 2v_i v_i^T\, K_j(A_{i-1})$ $j=1,\ 2,\ \text{--},n.$ (2.4.3)

Om die eerste (i-1) kolomme onveranderd te laat, is dit voldoende om die eerste (i-1) elemente van v_i gelyk aan nul te stel, sodat

$$v_i^T K_j(A_{i-1}) = 0, \quad j=1,\ 2,\ \text{--},i-1.$$

Dit beteken ook dat die eerste (i-1) rye van A_{i-1} onveranderd bly.

Beskou Q_1

Uit (2.4.2) is $K_1(A_1) = Q_1 K_1(A)$, stel verder $Q_1 K_1(A) = \lambda e_1$

waar

$$e_1^T = (1,0,\ \text{--},0) \quad \text{en } \lambda \text{ 'n konstante is.}$$

d.w.s. $\lambda^2 e_1^T e_1 = \lambda^2 = K_1(A)^T Q_1^T Q_1 K_1(A) = K_1(A)^T K_1(A)$

$$= \sum_{i=1}^{m} a_{i1}^2$$

Stel $c = 2v_1^T K_1(A)$ dus $cv_1 = 2v_1 v_1^T K_1(A)$ dan volg uit (2.4.3) dat

$Q_1 K_1(A) = K_1(A) - 2v_1 v_1^T K_1(A) = K_1(A) - cv_1$

Dus, $cv_1 = K_1(A) - \lambda e_1$

Dui die komponente van v_1 aan met $v_1(i)$, $i=1,2$ --,m en $v_1(1) = (a_{11}-\lambda)/c$

$v_1(i) = a_{i1}/c$ $i=2,3,\ \text{--},\ m$

(2.4.4)

Dan is volgens (2.4.1), $v_1^T v_1 = 1$,

d.w.s. $\sum_{i=1}^{m} v_1^2(i)=1$.

Dus, $v_1^2(1) + \sum_{i=2}^{m} v_1^2(i) = 1$

Dan is volgens (2.4.4)

$$\frac{(a_{11}-\lambda)^2}{c^2} + \frac{1}{c^2} \sum_{i=2}^{m} a_{i1}^2 = 1$$

$$a_{11}^2 + \lambda^2 - 2a_{11}\lambda + \sum_{i=2}^{m} a_{i1}^2 = c^2$$

$$2\lambda^2 - 2a_{11}\,\lambda = c^2$$

$$c = \sqrt{2\lambda\ (\lambda-a_{11})}$$

Deur die teken van λ te kies as die teenoorgestelde van die teken van a_{11}, word voorkom dat c nul word . Sodoende kan v asook Q uniek gedefinieer word.

Die ander transformasies Q_i, i=2,3, --,k word op dieselfde manier verkry, deur gebruik te maak van die submatriks van A_{i-1}, gevorm deur die eerste (i-1) rye en kolomme weg te laat.

Net soos die kolomelemente deur 'n QR-ontbinding nul gemaak kan word, kan die ry-elemente deur 'n RQ-ontbinding nul gemaak word.

2.4.2 Die VDU-ontbinding

Stelling 2.4.2.1

Elke reële (mxn)-matriks A kan as 'n produk $A = VDU^T$ ontbind word waar

(i) V 'n ortogonale (mxm) matriks,

(ii) D 'n diagonale (mxn) matriks en

(iii) U 'n ortogonale (nxn) matriks is.

Bewys:

Laat $A^{(1)}$ die matriks wees wat verkry word deur A links met 'n Householder-matriks te vermenigvuldig. Dan is $a_{11}^{(1)}$ die enigste nie-nulelement in die eerste kolom van $A^{(1)}$ en

$$\left|a_{11}^{(1)}\right| = \left[\sum_{i=1}^{m} (a_{i1})^2\right]^{\frac{1}{2}} \geq |a_{11}| . \tag{1}$$

Laat verder $A^{(2)}$ die matriks wees wat verkry word deur $A^{(1)}$ links met 'n Householder-matriks te vermenigvuldig. Dan is $a_{11}^{(2)}$ die enigste nie-nulelement in die eerste ry van $A^{(2)}$ en

$$\left|a_{11}^{(2)}\right| = \left[\sum_{j=1}^{n} (a_{1j}^{(1)})^2\right]^{\frac{1}{2}} \geq \left|a_{11}^{(1)}\right| . \tag{2}$$

Hierdie proses kan herhaal word, deur alternatiewelik die nie-diagonaalelemente van die eerste kolom en eerste ry na nulle te reduseer uit (1) en (2), volg dat die waarde van $\left|a_{11}^{(K)}\right|$ steeds groter word, maar omdat die norm van 'n ortogonale transformasie konstant bly, is $\left|a_{11}^{(K)}\right|$ begrens, want

$$\left|a_{11}^{(K)}\right| \leq \left[\sum_{i,j} (a_{ij}^{(K)})^2\right]^{\frac{1}{2}} = \left[\sum_{i,j} (a_{ij})^2\right]^{\frac{1}{2}} = \|A\|_E . \tag{3}$$

Dus konvergeer $|a_{11}^{(K)}|$ na 'n bepaalde limiet. Stel die limiet is d_1, waar

$$|d_1| \leq ||A||_E \text{ en } \lim_{K\to\infty} a_{i1}^{(K)} = 0 \text{ en } \lim_{K\to\infty} a_{1j}^{(K)} = 0$$

$$i=2, --, m, j=2, ---, n.$$

Die Householder-transformasies met $v(j)=0$, $j=1,2, ---, i-1$ laat die eerste $(i-1)$ rye en kolomme van die matriks onveranderd. Deur die bogenoemde proses op al die rye en kolomme toe te pas, kan ons 'n matriks D, met d_i as diagonaalelemente ontwikkel.

Dus kan die (mxn)-matriks A as 'n produk ontbind word, d.w.s. $A=VDU^T$, waar V die produk van ortogonale matrikse is, wat die kolom-elemente reduseer en U^T die produk van ortogonale matrikse wat die ry-elemente reduseer. Die diagonaalelemente d_i, van die matriks D kan gekies word sodat $d_i \geq 0$, $i=1,2 ---, \ell$ met $\ell = \min(m,n)$ en

$$d_1 \geq d_2 \geq d_3 --- \geq d_r > d_{r+1} = --- = d_\ell = 0, \quad r \leq \ell \text{ waar } r$$

die rang van A.

Omdat die Euklidiese norm deur ortogonale transformasies onveranderd gelaat word, is

$$||A||_E = ||D||_E = \left[\sum_{i=1}^{\ell} d_i^2 \right]^{\frac{1}{2}}.$$

Vir die minimering van $||A-B||$ is dit volgens Eckart & Young [16] die voordeligste dat vir 'n mxn-matriks $B = VD_TU^T$ van rang s, $s \leq r$, die diagonaalelemente van D_T gegee word deur

$$D_{T_{ii}} = d_i, \; 1 \leq i \leq s$$
$$= 0 \;, \; s < i < \ell \;.$$

Die beste benadering vir die matriks D, is dus die matriks D_T van rang s omdat

$$\|A - B\|_E = \|D - D_T\|_E.$$

2.5 Die Pseudo-Inverse van die VDU matriks

Vir die diagonaalmatriks D met elemente

$$D_{ii} = d_i \quad \forall i = 1, \text{--}$$

$$D_{ij} = 0 \quad \forall i \neq j,$$

volg uit die Penrose-voorwaardes dat

$$D^+_{ii} = d^+_i = 1/d_i, \; d_i \neq 0$$
$$= 0 \qquad d_i = 0$$

en $$D^+_{ij} = 0 \; \forall i \neq j \;.$$

Vir $A = VDU^T$ sal die matriks $A^+ = UD^+V^T$ die Penrosevoorwaardes bevredig.

(i) $AA^+A = VDU^TUD^+V^TVDU^T = VDD^+DU^T = VDU^T = A$

(ii) $A^+AA^+ = UD^+V^TVDU^TUD^+V^T = UD^+DD^+V^T = UD^+V^T = A^+$

(iii) $(AA^+)^T = (VDU^TUD^+V^T)^T = (VDU^T)(UD^+V^T) = AA^+$

(iv) $(A^+A)^T = (UD^+V^TVDU^T) = A^+A$

Aangesien die inverse vir die Penrosevoorwaardes uniek is sal die pseudo-inverse van VDU^T gelyk aan UD^+V^T wees.

Die pseudo-inverse A^+ van A kan dus maklik bereken word as die VDU-ontbinding van A bekend is.

3. FUNKSIEBENADERING DEUR KLEINSTEKWADRATE-TEGNIEKE

In die numeriese wiskunde moet ons dikwels die koëffisiënte van 'n lineêre, 'n nie-lineêre of 'n gesamentlike lineêre en nie-lineêre funksie bepaal uit 'n stel empiries bekende punte. Die funksie moet so bepaal word dat die som van die kwadrate van die verskille tussen die gepaste funksiewaardes en die ooreenstemmende bekende empiriese waardes 'n minimum is.

Ons beskou eerste die lineêre funksie:

3.1 Die Lineêre Passing

3.1.1 Die normaalvergelykings

Met 'n lineêre funksie bedoel ons 'n funksie waarin die koëffisiënte a_i, $i=1,2, --, n$ slegs in die eerste graad voorkom.

Laat f 'n funksie met funksiewaarde f_i in die ry punte $\{x_i\}$, $i=1, --, m$ wees en laat die ooreenstemmende eksperimentele waargenome funksiewaardes y_i wees.

Die verskil tussen bogenoemde waardes word aangedui deur $E_i=f_i-y_i$.

Laat $\{\phi_j\}$, $j=1, -----, n$ 'n funksie wees wat gedefinieer is vir elke x_i. Ons moet dan y_i benader deur 'n lineêre kombinasie van die $\{\phi_j(x)\}$,

d.i. $$f_i = \sum_{j=1}^{n} a_j\phi_j(x), \quad i=1,2, --, m$$

Die a_j's moet so bepaal word dat

$$H(a_1, --, a_n) \equiv \sum_{i=1}^{m} w(x_i) \left[y_i - \sum_{j=1}^{n} a_j \phi_j (x_i) \right]^2$$

$$= \sum_{i=1}^{m} w(x_i) R_i^2 \text{ met } R_i = y_i - \sum_{j=1}^{n} a_j \phi_j (x_i)$$

'n minimum is.

Die funksie w waar $w(x_i) \geq 0$, $i = 1, 2, ---, m$ word die beswaringsfunksie en R_i die residu by x_i genoem.

Vir $w_i = 1$, $i = 1, 2, ---, m$ het ons dus

$$H(a_1, ---, a_n) = \sum_{i=1}^{m} \left[y_i - \sum_{j=1}^{n} a_j \phi_j (x_i) \right]^2 = \sum_{i=1}^{m} R_i^2$$

Vir $\phi_j(x) = x^{j-1}$ sal die gepaste funksie byvoorbeeld 'n polinoom van graad (n-1) wees.

Bostaande formule geskryf in matriksnotasie is:

$$||y - Fa||^2 = ||r||^2 \quad \text{waar } a = \{a_i\},\ i=1, ---,n$$

d.i.

$$||r||^2 = (y-Fa)^T (y-Fa)$$

Vir 'n minimum is

$$\frac{\partial}{\partial a_j} ||r||^2 = 2e_j^T F^T Fa - 2e_j^T F^T y = 0, \ j=1, ---, n$$

D.w.s. $F^T Fa = F^T y$

of

$$a = (F^T F)^{-1} F^T y. \qquad (3.1.1)$$

Hierdie vergelykings staan bekend as die normaalvergelykings.

Om die koëffisiënte a_i, $i=1,2, ---, n$ te verkry moet ons dus die inverse van (F^TF) bereken. Ralston [3] bespreek verskillende metodes om die inverse regstreeks of iteratief te bereken.

Omdat $Fa=0$ as en slegs as $a=0$, is die matriks F_TF positief definiet en nie-singulier, sien Ralson [3,p.135].

Fourie [5] bewys dat as $\phi_j(x)$ benader word deur 'n veelterm die koëffisiëntematriks getransformeer na die interval $[0,1]$, 'n Hilbert-matriks van orde $(m+1)$ is. Om die inversevan dié matriks te bereken, lewer probleme want die matriks is byna singulier omdat die rye van die matriks byna lineêr afhanklik is. In baie gevalle is die metode om die normaalvergelykings op te los, onstabiel veral as m taamlik groot word en dus sal ander metodes gesoek moet word.

3.1.2 Oplossing van die kleinstekwadrate-probleem m.b.v. die pseudo-inverse

Laat in die kleinstekwadrate-probleem die residue gegee word deur r_i, $i=1, --, m$ en laat die bekende eksperimentele data gegee word deur die matriks $X = (\{x_{ij}\}, i=1,2, --, m; j=1,2, --, k)$. Die lineêre funksie kan geskryf word as $F(X,a)$ waar $a=a_i$, $i=1, 2, ---, n$ is. Die a_i's moet sodanig bepaal word dat die residue 'n minimum is. Gestel $F_i(X,a) = a_1x_{i,1}^2 + a_2x_{i,1}x_{i,2} + --+a_nx_{i,n}^2$, $i=1,2, --, m$ en laat:

$$F_{i,1} = x_{i,1}^2$$

$$F_{i,2} = x_{i,1}\, x_{i,2}$$

$$\vdots$$

$$F_{i,n} = x_{i,n}^2, \quad i=1, --, m$$

Laat y_i, i=1, --,m die eksperimentele bekende waardes van die afhanklike veranderlike wees, dan kan die residue geskryf word as:

$$r_i = y_i - \sum_{j=1}^{n} F_{i,j}(a_j), i=1, \text{--}, m$$

In matriksnotasie y + Fa = r.

Die vektor met minimum norm wat $||r||$ minimeer word gegee deur

$$a \doteq F^{+}y$$

Die probleem reduseer dus na die bepaling van die pseudo-inverse van F. Een metode om die pseudo-inverse te bepaal is, deur gebruik te maak van die VDU-ontbinding (die sogenaamde pseudo-inverse metode van paragraaf 2.5). Ander metodes wat gebruik kan word is die gedempte kleinstekwadraat-metode en die dubbele verslappingsmetode.

3.1.3 Die direkte-oplossingsmetode

Uit § 2.4.2 volg dit dat $F=VDU^T$ waar V 'n (mxm) ortogonale matriks, D 'n (mxn) diagonale matriks en U^T 'n (nxn) ortogonale matriks is. Dan is $F^{+}= UD^{+}V^T$ volgens § 2.5.

D.w.s. $a = UD^{+}V^{T}y$.

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ u_{21} & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} d_1^+ & 0 & \cdots & & & 0 \\ 0 & d_2^+ & & & & \vdots \\ \vdots & & \ddots & & & \vdots \\ 0 & \cdots & 0 & d_n^+ & 0 \cdots & 0 \end{bmatrix} \begin{bmatrix} v_{11} & v_{21} & \cdots & v_{m1} \\ v_{12} & v_{22} & \cdots & v_{m2} \\ \vdots & & \ddots & \vdots \\ v_{1m} & \cdots & & v_{mm} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

$$\begin{bmatrix} \frac{u_{11}}{d_1} & \frac{u_{12}}{d_2} & \cdots & \frac{u_{1n}}{d_n} \\ \frac{u_{21}}{d_1} & \frac{u_{22}}{d_2} & & \frac{u_{2n}}{d_n} \\ \vdots & & \ddots & \vdots \\ \frac{u_{n1}}{d_1} & \frac{u_{22}}{d_2} & \cdots & \frac{u_{nn}}{d_n} \end{bmatrix} \begin{bmatrix} v_{11}\ y_1 + v_{21}\ y_2 + \cdots + v_{m1}\ y_m \\ v_{12}\ y_1 + v_{22}\ y_2 + \cdots + v_{m2}\ y_m \\ \\ \\ v_{1m}\ y_1 + v_{2m}\ y_2 + \cdots + v_{mm}\ y_m \end{bmatrix}$$

Stel $K_i(V) = v_i$ en $K_i(U) = u_i$ met $K_i(V)$ en $K_i(U)$ respektiewelik die i-de kolom van V en U. Die algemene oplossing is dan

$$a = \sum_{d_i > o} (y,\ v_i)\ u_i/d_i$$

waar (y,v_i) die binneproduk van y en v_i is.

3.1.4 Die gedempte kleinstekwadrate-oplossing

Hier word die kleinstekwadraat-probleem voorgestel deur:

$$\begin{bmatrix} F \\ B \end{bmatrix} a - \begin{bmatrix} y \\ 0 \end{bmatrix} = \begin{bmatrix} r^{(1)} \\ r^{(2)} \end{bmatrix}$$

dit is: $\left[F^TF + B^TB\right] a = F^Ty$ (uit § 3.1.1) (3.1.2)

Ons beskou die volgende twee gevalle:

(a) $B^TB = \varepsilon^2 I$ die gedempte kleinstekwadraat-metode

(b) $B^TB = \varepsilon^2 I + \varepsilon^2(F^TF + \varepsilon^2 I)^{-1}$ Die dubbele verslappings-metode.

Beskou eerstens $B^TB = \varepsilon^2 I$ dan word (3.1.1)

$\left[F^TF + \varepsilon^2 I\right] a = F^Ty$ deur F te ontbind kry ons

$$\left[(VDU^T)^T VDU^T + \varepsilon^2 I\right] a = (VDU^T)^T y$$

$$\left[UDV^TVDU^T + \varepsilon^2 I\right] a = (UDV^T) y$$

$$\left[UD^2U^T + \varepsilon^2 I\right] a = UDV^T y$$

D.w.s. $a = \sum_{d_i > 0} (y, v_i) \dfrac{d_i}{d_i^2 + \varepsilon^2} u_i$ (sien § 3.1.3). (3.1.3)

Maar $\lim_{\varepsilon \to 0} \dfrac{d_i}{d_i^2 + \varepsilon^2} = \dfrac{1}{d_i} = d_i^+$

Dus $F^+ = \lim_{\varepsilon \to 0} (F^TF + \varepsilon^2 I)^{-1} F^T$

3.1.5 Die dubbele verslappingsmetode

In die geval van die dubbele verslappingsmetode is

$$B^TB = \varepsilon^2 I + \varepsilon^2 (F^TF + \varepsilon^2 I)^{-1}$$

Dus word (3.1.1)

$$\left[(FF^T + \varepsilon^2 I) + \varepsilon^2 (F^TF + \varepsilon^2)^{-1} \right] a = F^T y$$

dus $\quad a = \sum_{i=1}^{n} (y, v_i) \dfrac{d_i}{d_i^2 + \varepsilon^2 + \dfrac{\varepsilon^2}{d_i^2 + \varepsilon^2}} u_i \qquad$ (Sien § 3.1.3)

en F^+ kan dus ook gedefinieer word deur

$$F^+ = \lim_{\varepsilon \to 0} \left[(F^TF + \varepsilon^2 I) + \varepsilon^2 (F^TF + \varepsilon^2 I)^{-1} \right]^{-1} F^T$$

3.2 Die Nie-lineêre Passing

'n Nie-lineêre passing is 'n funksie waarin sommige van die koëffisiënte a_i met funksies vermenigvuldig word wat op hulle beurt van die koëffisiënte afhanklik is.

Beskou byvoorbeeld 'n matriks X met eksperimentele data soos gegee in § 3.1.2 en gestel die passingsfunksie word gedefinieer deur

$$F_i(X,a) = a_1 x_{i,1}^2 + a_2 e^{a_3 x_{i,2} + a_4 x_{i,5}}, i = 1, --, m. \qquad (3.2.1)$$

Die funksie gedefinieer deur

$e^{a_3 x_{i,2} + a_4 x_{i,5}}$ waarmee a_2 vermenigvuldig word, is afhanklik van a_3 en a_4. $F_i(X,a)$ is dus 'n nie-lineêre passing.

Gestel $f_i = F_i(X,a) - y_i$, $i = 1,2, ---, m$, waar y_i die afhanklike veranderlike voorstel. Ons moet dus $\sum_{i=1}^{m} f_i^2$ minimeer.

In Bellman [7,p.65] word die volgende metodes vir die minimering van kleinstekwadrate-probleme bespreek.

(a) Die metode van Newton

(b) Die Gauss-Newton-metode

(c) Die Gauss-Newton-Marquardt-metode

Laasgenoemde metode is 'n verbetering van die eersgenoemde twee metodes.

3.2.1 Die Gauss-Newton-Marquardt-metode

In 'n nie-lineêre kleinstekwadrateprobleem word 'n lineêre stelsel met elke stap opgelos. Dit beteken dat die nie-lineêre probleem omskep word in 'n iteratiewe lineêre kleinstekwadrate-probleem.

Beskou die volgende stelsel nie-lineêre vergelykings wat opgelos moet word:

$$f_i = f_i(a) = F_i(X,a) - y_i, \; i = 1,2, ---, m.$$

Deur die funksie f(a) in 'n Taylor-reeks te ontwikkel vir a^i, die a-vektor in die i-te iterasie, kry ons

$$f(a^i + h^i) = f(a^i) + A^i h^i + 0(h^2)$$

waar h^i die h-vektor en A^i die matriks van parsiële afgeleides in die i-te iterasie is, wat gedefinieer word deur:

$$A^i_{st} = \frac{df_s}{da_t}(a^i), \quad s=1, ---, m, \quad t=1, ---, n.$$

As ons tweede en hoër orde terme in h weglaat, reduseer die stelsel nie-lineêre vergelykings na 'n stelsel van lineêre vergelykings nl.

$$\begin{bmatrix} f^i \\ 0 \end{bmatrix} + \begin{bmatrix} A^i \\ B^i \end{bmatrix} h^i = \begin{bmatrix} r_1{}^i \\ r_2{}^i \end{bmatrix} \qquad (3.2.2)$$

Die h-vektor kan uit (3.2.2) m.b.v. die volgende algoritme bereken word:

(i) Los vir h uit (3.2.2) op

(ii) Bepaal 'n staplengte deur γ so te kies dat

$$\|f(a^i) + \gamma h^i\| < \|f(a^i)\|$$

(iii) Stel $a^{i+1} = a^i + \gamma h^i$ en toets vir konvergensie.

As $B^i = 0$ reduseer die algoritme na dié van Gauss-Newton.

In Lootsma [2,p.178] vind ons die volgende konvergensie-stelling

As B^i so gekies word dat $\delta^i \geq \delta > 0$, dan konvergeer die ry $\{\|f^i\|\}$ en die limietpunte van die ry $\{x^i\}$ is stasionêre punte van $\|f\|$.

<u>Bewys</u>

Om die konvergensie van die algoritme in § 3.2.1 te bewys, maak ons die volgende aannames:

(i) Die dempingsfaktor γ, $0 \leq \gamma \leq 1$, is so gekies dat $\|f(x^i + h^i)\|$ 'n minimum is.

(ii) Die iterasie is begrens in 'n gebied R.

(iii) Die funksie word gedefinieer as

$$f(x+\gamma h) = f(x) + \gamma \nabla f h + \gamma^2 \|h\|^2 w(x,\gamma,h) \tag{3.2.3}$$

waar $\|w\| \leq W$ vir punte in R.

Deur $\nabla f h$ uit vergelyking (3.2.2) in (3.2.3) te vervang, kry ons

$$f(x^i+\gamma h^i) = f^i + \gamma(r_1^{\ i} - f^i) + \gamma^2 \|h^i\|^2 w^i(\gamma).$$

Deur gebruik te maak van die driehoeksongelykheid kry ons

$$\|f(x^i+\gamma h^i)\| \leq (1-\gamma)\|f^i\| + \gamma\|r_1^{\ i}\| + \gamma^2\|h^i\|^2 W = Q^i(\gamma) \tag{3.2.4}$$

D.w.s. $\|f(x^i+\gamma h^i)\| \leq \min\limits_{0 \leq \gamma \leq 1} Q^i(\gamma)$

As $Q^i(\gamma)$ 'n minimum waarde het vir $\gamma<1$ dan moet $\frac{dQ^i}{d\gamma}$ nul wees in die minimum punt

Dus
$$\frac{dQ^i}{d\gamma} = -(\|f^i\| - \|r_1^{\ i}\|) + 2\gamma\|h^i\|^2 W$$
$$= 0.$$

d.w.s.
$$\gamma = \frac{\|f^i\| - \|r_1^{\ i}\|}{2\|h^i\|^2 W}. \tag{3.2.5}$$

Stel (3.2.4) in (3.2.3). Dan is

$$Q^i(\gamma) = \|f^i\| - \frac{\frac{1}{2}(\|f^i\| - \|r_1^{\ i}\|)^2}{2\|h^i\|^2 W} = \|f^i\| - \frac{\gamma}{2}(\|f^i\| - \|r_1^{\ i}\|). \tag{3.2.6}$$

As $\gamma = 1$ die minimumpunt is, dan is

$$\frac{\|f^i\| - \|r_1{}^i\|}{2\|h^i\|^2{}_W} \geq 1$$

en

$$Q^i(1) = \|r_1{}^i\| + \|h^i\|^2{}_W \leq \|f^i\| - \tfrac{1}{2}(\|f^i\| - \|r_1{}^i\|) \tag{3.2.7}$$

Laat $(\gamma_1^i)^* = \min\left(1, \dfrac{\|f^i\| - \|r_1{}^i\|}{2\|h^i\|^2{}_W}\right)$

dan volg uit (3.2.4) en (3.2.6) dat

$$\|f^{i+1}\| \leq \|f^i\| - \frac{(\gamma_1^i)^*}{2}(\|f^i\| - \|r_1{}^i\|).$$

Verder weet ons dat as $A^TA + B^TB = M$ en $\min\limits_{t,\|t\|=1}(t^TMt) = \delta^2$

dan is $\|h\| \leq \frac{1}{\delta}\{\|f\|^2 - \|r\|^2\}^{\frac{1}{2}}$ ([2] p.177 lemma 3).

Dus is

$$(\gamma_1^i)^* \geq \min\left(1, \frac{(\delta_1^i)^2}{2W}\frac{1}{(\|f^i\| + \|r_1{}^i\|)}\right) \geq \min\left(1, \frac{(\delta_1^i)^2}{2W}\frac{1}{2\|f^i\|}\right).$$

Dus, $(\gamma_1^i)^* \geq \min\left(1, \dfrac{(\delta_1^i)^2}{4W\|f^i\|}\right) = \mu > 0$

Die ry $\{\|f^i\|\}$ is dus dalend en begrens na onder. Dus is die ry konvergent.

Verder weet ons ook dat die voorwaardes

(i) $\begin{bmatrix} f^i \\ 0 \end{bmatrix} = r^i$

(ii) $\|f^i\| = \|r^i\|$ en

(iii) r^i 'n stasionêre punt van $\|f(x)\|$ is, ekwivalent is ([2] p.176 lemma 1).

Dus, omdat

$$\|f^i\| - \|r^i\| \leq \|f^i\| - \|r_1^i\| \leq \frac{2}{\mu}(\|f^i\| - \|f^{i+1}\|),$$

is die limietpunte van die ry $\{x^i\}$ stasionêre punte van $\|f\|$.

3.3 Die Saamgestelde Funksie

Die funksie wat in (3.2.1) gedefinieer is deur $F_i(X,a) = a_1x_{i,1}^2 + a_2e^{a_3x_{i,2} + a_4x_{i,5}}$, $i=1,2,$ ---, m bestaan uit 'n lineêre gedeelte $F_i^{(1)}(X,a) = a_1x_{i,1}^2$, en 'n nie-lineêre gedeelte

$$F_i^{(2)}(X,a) = a_2e^{a_3x_{i,2} + a_4x_{i,5}}$$

In die algemeen is die passingsfunksies saamgestelde funksies So 'n funksie kan dus geskryf word as $F_i(x,a) = F_i^{(1)}(X,a) + F_i^{(2)}(X,a)$ waar $F_i^{(1)}(X,a)$ die lineêre gedeelte en $F_i^{(2)}(X,a)$ die nie-lineêre gedeelte is.

Omdat die parsiële afgeleides van die lineêre gedeelte gedurende die iteratiewe proses konstant bly, word hulle slegs eenmaal bereken en gestoor in 'n matriks $A^{(k)}$. Die parsiële afgeleides van die nie-lineêre gedeelte verander by elke iterasie, en hulle word gestoor in die matriks $A^{(v)}$, sodat $A^{(i)}$ voorgestel kan word deur:

$$A^{(i)} = \left[A^{(v)} \; A^{(k)}\right]$$

Ons moet nou die volgende lineêre kleinstekwadrate-probleem oplos:

$$\begin{bmatrix} f_i \\ 0 \end{bmatrix} + \begin{bmatrix} A^{(v)} \\ B^{(v)} \end{bmatrix} h^{(v)} + \begin{bmatrix} A^{(k)} \\ B^{(k)} \end{bmatrix} h^{(k)} = \begin{bmatrix} r_1^{(i)} \\ r_2^{(i)} \end{bmatrix} = r^{(i)}$$

$$\begin{bmatrix} f_i \\ 0 \end{bmatrix} + \begin{bmatrix} A^{(v)} & A^{(k)} \\ B^{(i)} & \end{bmatrix} h^{(i)} = \begin{bmatrix} r_1^{(i)} \\ r_2^{(i)} \end{bmatrix} = r^{(i)}$$

waar $B^{(i)} = \left[B^{(v)}, B^{(k)}\right]$ en $h^{(i)} = \begin{bmatrix} h^{(v)} \\ h^{(k)} \end{bmatrix}$

Hieruit kan die h-vektor bereken en die algoritme is dieselfde as in § 3.2.1.

3.4 Beperkings op die Koëffisiënte

In die praktyk lê die koëffisiënte a_j gewoonlik tussen bepaalde grense sê $g_{1,j} \leq a_j \leq g_{2,j}$. Dit kan byvoorbeeld gebeur dat die nie-lineêre gedeelte van die funksie (3.2.1) nl. die term $e^{(a_3x_{i,1} + a_4x_{i,5})}$ gedurende die iteratiewe proses te groot word indien daar nie 'n beperking op die koëffisiënte a_3 en a_4 is nie.

Ons definieer a_j^* as

(i) $a_j^* = g_{1,j} + (g_{2,j} - g_{1,j}) \sin^2 a_j$ $\forall j$ waarvoor daar beperkings op die koëffisiënte is, d.w.s. $a_j^* \in [g_{1j}; g_{2j}]$

(ii) $a_j^* = a_j$ as daar nie beperkings op die koëffisiënte is nie.

Deur die koëffisiënte a_j van die funksie te vervang deur a_j^* en daarna die funksie $F_i(X,a)$ te minimeer, sal die koëffisiënte aan die beperkings voldoen.

3.5 Gebruik van Skaleringskonstantes

In 'n stelsel lineêre vergelykings Ax=b kan aan die elemente x_j en die b_i fisiese betekenisse geheg word. So kan bv. x 'n verplasingsvektor wees (gemeet in sentimeters) en C 'n kragvektor voorstel (gemeet in dienes). Omdat die fisiese eenhede willekeurig gekies kan word, is dit nodig om die invloed van so 'n keuse na te gaan. As x_j bv. in kilometers i.p.v. sentimeters gemeet word, moet x_j vervang word deur $10^5 x_j$ en elke term $a_{ij}\, x_j$ deur $10^5 a_{ij}\, x_j$. Die hele j-de kolom van A moet dus met 'n faktor 10^5 vermenigvuldig word.

Veronderstel ons vervang die x_j's deur $d_j^{(2)} x_j$ $\forall j = 1, \ldots, n$ en laat

$$D_2 = \begin{bmatrix} d_1^{(2)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_n^{(2)} \end{bmatrix}$$

Dan is

$$x = D_2 x^1$$

Veronderstel verder dat

$$\begin{bmatrix} d_1^{(1)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_n^{(1)} \end{bmatrix}$$

Die tweede nie-singuliere matriks is en dat

$$b = D_1 b^1.$$

Die sisteem $Ax = b$ word dus

$$AD_2 x^1 = D_1 b^1 .$$

Dus $\quad D_1^{-1} A D_2 x^1 = b^1$

d.w.s. $\quad A^1 x^1 = b^1$

waar $\quad A^1 = D_1^{-1} AD_2$ en $b^1 = D_1^{-1} b$.

Hierdie skaleringsfaktore kan 'n belangrike rol speel in die oplos van 'n kleinstekwadraat-probleem. Die keuse van die skaleringsfaktore kan die noukeurigheid van die resultate beïnvloed asook die aantal iterasies wat nodig is om aan die konvergensievereistes te voldoen.

P C Haarhoff bespreek in [4] 'n metode om geskikte skaleringsfaktore te bereken.

Volgens die metode word twee skaleringsfaktore bereken:

(a) Die skaleringsfaktor Fsc vir die objekfunksie word geneem as die som van die funksiewaarde $Fsc = \Sigma F_i^2$.

(b) Die skaleringsfaktore XSC_i vir die skalering van X_i, $i=1, --, n$, moet so gekies word dat die koëffisiënte in alle Taylor uitbreidings na skalering van orde een is (die waardes van sommige koëffisiënte kan selfs nul wees, maar mag net nie groter as een wees nie) en dat die skaleringsfaktore nie te groot of te klein word nie. D.w.s.

$$BOT \leq XSC_i \leq TOP$$

waar TOP en BOT respektiewelik die gegewe bo- en ondergrens is. As die skaleerde waardes van F en X_i deur f en x_i voorgestel word dan is

$$\frac{\partial f}{\partial x_i} \lesssim 1$$

d.w.s.

$$XSC_i \lesssim Fsc \Big/ \left|\frac{\partial F}{\partial X_i}\right| = (XSC_i)_1$$

en

$$XSC_i \lesssim \sqrt{Fsc \Big/ \left|\frac{\partial^2 F}{\partial X_i^2}\right|} = (XSC_i)_2 .$$

Die waarde van XSC_i word dan geneem as

$$XSC_i = \text{maks}\{\min\{(XSC_i)_1, (XSC_i)_2, TOP\}, BOT\}.$$

3.6 Numeriese Berekening van Afgeleides

In die meeste gevalle, waar die minimum van die norm van die kwadrate van 'n nie-lineêre passing gevind moet word, is die analitiese afgeleides van die gegewe funksie onverkrygbaar.

Numeriese metodes moet dan gebruik word om afgeleides te bepaal.

Beskou die Taylor-uitbreiding van 'n funksie f(x) om die punt x, dan is

$$f(x+h) = f(x)+h\, f^1(x) + \frac{h^2}{2} f''(x)+0(h^3) \qquad (3.6.1)$$

en

$$f(x-h) = f(x)-h\, f^1(x) + \frac{h^2}{2} f''(x)+0(h^3). \qquad (3.6.2)$$

Uit (3.6.1) is die eerste orde-benadering $f^1(x) = \frac{f(x+h)-f(x)}{h}$.

As 'n tweede orde-benadering volg uit die verskil van twee Taylor-uitbreidings dat

$$f^1(x) = \frac{f(x+h) - f(x-h)}{2h} .$$

3.7 Die Algoritme vir PASPLT

Om die oplossing van 'n nie-lineêre kleinstekwadrate-probleem te vind, los ons in werklikheid met elke iterasie 'n lineêre probleem op.

Laat $F = f_i = f_i(a)$ en $A = \nabla f(a)$

die Jacobiaan van F wees, dan kan die lineêre probleem wat opgelos moet word, voorgestel word deur:

$$\begin{bmatrix} F \\ 0 \end{bmatrix} + \begin{bmatrix} A \\ B \end{bmatrix} h = r$$

Dus, $$\left[A^T A + B^T B\right] h = -A^T F, \tag{3.7.1}$$

of $$h = -\left[A^T A + B^T B\right]^{-1} A^T F.$$

Laat $a^i = \{a_j^i\}$, $j=1,2,---,\ell$ die koëffisiënte en $S(a^i)$ die som van die kwadrate van die verskille in die i-te iterasie wees.

In die berekening van $(A^T A + B^T B)^{-1}$ word gebruik gemaak van Householder-transformasies en van die pseudo-inverse.

As die Jacobiaan numeries bereken word, word die passingsfunksie soos in par. 3.3 beskryf, geskei in 'n lineêre en 'n nie-lineêre gedeelte. Die matriks $B^T B$ is 'n diagonaalmatriks met diagonaalelemente λV waar V gegee word deur

$$V_k^i = |a_k^i| \Big/ \sum_{j=1}^{\ell} |a_j^i|, \qquad k=1,2,---,\ell$$

en waar die dempingsfaktor λ 'n skalaarveranderlike is wat begrens word deur FNMN en FNMX. Die dempingsterm λV^i speel 'n belangrike rol in die bepaling van die nuwe koëffisiënte a_k^{i+1}.

Die parameter IC word in die begin van elke iterasie gelyk aan nul gestel en met elke berekening van S(a) met een vermeerder. Die dempingsfaktor λ word soos volg deur die program verander.

i As $S(a^i+h) > S(a^i)$ word die waarde van $R=\min\{0{,}2\ \log^2 \frac{\lambda S(a^i)}{S(a^i+h)} + DINC,\ 10\}$ bereken en λ word vervang deur $\lambda = \lambda * R$, waar DINC 'n konstante is.

ii As $S(a^i+h) \leq S(a^i)$ en $IC \neq 1$ bly λ onveranderd in die volgende iterasie en as IC=1 word λ vervang deur $\lambda = \lambda * DECR$, waar $0 < DECR < 1$, 'n konstante is.

Skalering van die veranderlikes kan toegepas word en die koëffisiënte a_j kan gedurende die iteratiewe proses begrens word of waar nodig konstant gehou word.

3.7.1 Algoritme

Die algoritme kan stapsgewys soos volg voorgestel word:

(1) Kies λ_o, DECR, DINC, FNMN en FNMX. Kies beginwaardes vir a^o en bereken $S(a^o)$.

(2) Bereken A^i en ontbind A^i in ortogonale matrikse.

Bereken $V^i = |a^i| \Big/ \sum_{k=1}^{\ell} |a_k^i|$ Stel IC=0.

(3) Voltooi die ontbinding van $\begin{bmatrix} A^i \\ \lambda V^i \end{bmatrix}$

(4) Bereken h uit (3.7.1) en stel IC = IC + 1.

(5) As $S(a^i+h) \leq S(a^i)$ Stel $a^{i+1} = a^i+h$ en i=i+1 gaan na stap 7.

(6) As $S(a^{i+h}) > S(a^i)$ en as $\lambda <$ FNMX, Stel $\lambda = R * \lambda$ met $R=\min\{0{,}2\log^2 \frac{\lambda S(a^i)}{S(a^i+h)} + \text{DINC}, 10\}$ gaan na stap 3. As $\lambda >$ FNMX gaan na stap 9.

(7) As IC=1 en $\lambda >$ FNMN. Stel $\lambda =$ DECR $* \lambda$, anders $\lambda = \lambda$.

(8) Toets vir konvergensie. Vir geen konvergensie, gaan na stap 3.

(9) Einde van program.

In figuur 3.1 word die vloeidiagram van die algoritme gegee.

3.8 Bestaande Algoritmes

Die algoritme wat in PASPLT gebruik word, is net soos 'n paar ander algoritmes gebaseer op die algoritme van Marquardt [23].

Die belangrikste verskil tussen die algoritmes is geleë in die keuse van B^TB.

3.8.1 Die Marquardt-of DLS-algoritme

In die Marquardt algoritme [23] word B^TB as die diagonaal matriks λI geneem.

Die verskillende stappe kan soos volg voorgestel word.

1. Kies λ en $\nu > 1$, waar ν 'n konstante is, gee beginwaardes vir $a=a^o$ en bereken $S(a^o)$ Stel i=1.

2. Stel $\lambda = \lambda / \nu$.

3. Bereken A, A^TA en A^TF.

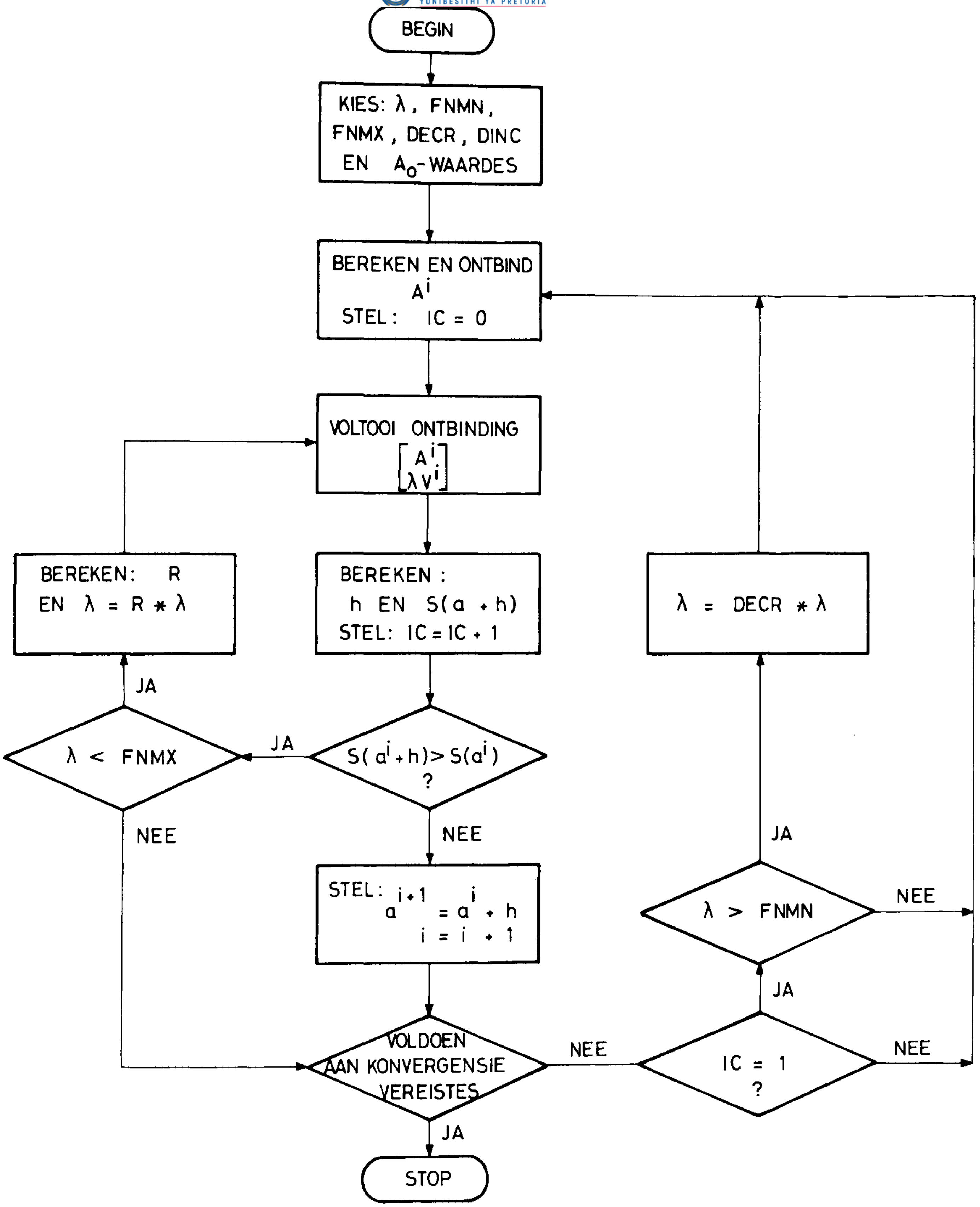


FIG. 3.1 VLOEIDIAGRAM PASPLT - ALGORITME

4. Los h op uit $\left[A^T A + \lambda I\right] h = -A^T F$ en bereken $S(a^i+h)$

5. As $S(a^i+h) \leq S(a^i)$, stel $a^{i+1} = a^i+h$, $i=i+1$ en gaan na stap 7.

6. As $S(a^i+h) > S(a^i)$, stel $\lambda = \nu\, \lambda$ en gaan na stap 2.

7. Toets vir konvergensie, vir geen konvergensie gaan na stap 2, anders eindig die program.

Die algoritme verskil van dié in par. 3.7.1 t.o.v. die keuse van V en die berekening van λ d.w.s. in die keuse van $B^T B$.

Marquardt neem $B^T B$ as λI, terwyl in PASPLT die matriks λVI geneem word. D.w.s. Marquardt neem $B^T B$ as 'n diagonaalmatriks waarvan die elemente almal dieselfde is, terwyl in PASPLT die matriks $B^T B$ 'n diagonaalmatriks is waarin die elemente almal verskil.

Marquardt bepaal 'n nuwe dempingsfaktor vir elke iterasie terwyl PASPLT dit nie doen nie. Verder bepaal Marquardt die nuwe dempingsfaktor deur die vorige dempingsfaktor met dieselfde konstante ν te vermenigvuldig of te deel terwyl in PASPLT hierdie waarde ν nie konstant bly nie.

3.8.2 Die Osborne-algoritme

Die Osborne-algoritme soos beskryf in [8] is 'n variasie van die Marquardt-algoritme. In hierdie algoritme word $B^T B$ ook as die diagonaalmatriks λI geneem en die parameters DECR, DINC en IC word soos in par. 3.7.1 gedefinieer.

Die Osborne-algoritme maak soos die PASPLT algoritme gebruik van Householder-transformasies en van die pseudo-inverse.

Net soos in die geval van die Marquardt-algoritme verskil hierdie algoritme van die PASPLT-algoritme in die keuse van B^TB. Osborne neem B^TB net soos Marquardt as die diagonaalmatriks λI. Die verskil tussen Marquardt en Osborne is daarin geleë dat in die berekening van die nuwe dempingsfaktor Marquardt altyd die ou dempingsfaktor vermenigvuldig of deel met dieselfde konstante waarde ν terwyl Osborne twee verskillende waardes gebruik, een om mee te vermenigvuldig en 'n ander om mee te deel. Verder bereken Osborne net soos PASPLT nie 'n nuwe dempingsfaktor vir elke iterasie nie.

3.8.3 Die MDLS of Gauss-Newton-Marquardt algoritme

Omdat die norms van die kolomme van die Jacobiaan A in nie-lineêre probleme soms baie verskil, kan die elemente van A^TA soms baie groot, baie klein of selfs nul word.

Marquardt het daarom voorgestel dat B^TB as 'n diagonaalmatriks Q met diagonaalelemente $\lambda Q_{ii} = \lambda(A^TA)_{ii}$ geneem word. Dus het ons hier 'n metode waarin die diagonaalelemente van B^TB nie almal dieselfde is nie. Die PASPLT-algoritme verskil dus in die opsig slegs in die keuse van die diagonaalterme. Die nuwe dempingsfaktor word in die geval op dieselfde manier as in die Marquardtmetode bereken.

3.8.4 Die Nash-Marquardt-algoritme

In die geval waar A^TA nulelemente in die diagonaal $(A^TA)_{ii}$ besit, het λ geen effek nie aangesien vir die betrokke elemente $Q_{ii}=0$ is. Om die probleem op te los stel Nash [21] voor dat $(A^TA)_{ii}$ vervang word deur $(A^TA)_{ii}\ (1+\lambda) + \alpha\lambda$ waar α 'n willekeurige konstante is. Die vergelyking (3.7.1) word dus:

$$\{A^TA + (\lambda(A^TA)_{ii} + \lambda\alpha)I\}\ h = -A^TF$$

Om singulariteite in die bepaling van h, tot 'n minimum te beperk, maak hy ook gebruik van matriksontbinding. Verder definieer hy konstantes ν, β en R, waar $\nu>\beta>1$ is. Hierdie konstantes ν en β word gebruik in die bepaling van nuwe λ-waardes terwyl R 'n ondergrens vir λ is.

Vir $S(a^i+h) > S(a^i)$ word $\lambda = \nu\lambda$ gestel en vir $S(a^i+h) \leq S(a^i)$ word $\lambda = \beta\lambda/\nu$ gestel. Die Nash-metode verskil dus van die PASPLT-metode in die bepaling van B^TB. Nash gebruik vir B^TB die diagonaalmatriks $\{\lambda(A^TA)_{ii} + \lambda\alpha\}I$ terwyl in PASPLT vir B^TB $(\lambda V)I$ gebruik word. Nash bepaal net soos Marquardt 'n nuwe dempingsfaktor vir elke iterasie, maar gebruik anders as Marquardt twee verskillende waardes vir ν, een om mee te vermenigvuldig en die ander om mee te deel. Beide algoritmes maak gebruik van matriksontbinding en in beide algoritmes is daar 'n ondergrens vir λ. In die PASPLT-algoritme word daar ook 'n bogrens op λ geplaas.

3.8.5 Die Fletcher-algoritme

Fletcher lê in [22] reëls gegrond op die verhouding R neer vir die vermeerdering en vermindering van λ. Die verhouding is $R = R'/R''$, waar $R' = S(a^i) - S(a^{i+1})$,

$$R'' = \Phi(a^i) - \Phi(a^i+h) = -2h^T(A^TF)^i - h^TA^ih$$

en $\Phi(a)$ die verwagte som van die kwadrate van die verskille voorstel.

As $R < \rho$ word λ verminder en as $R > \sigma$ word λ vermeerder, andersins bly λ konstant. Die konstantes ρ en σ word so gekies dat $0<\rho<\sigma<1$. Die program VA07A, wat in [22] bespreek word, gebruik Fletcher se algoritme, en dit besit verder die moontlikheid om die vektor a te skaleer. Verder is dit ook moontlik om die a-waardes te begrens. Die metode stem behalwe vir die keuse van B^TB en die verandering van λ grootliks met die metode wat in PASPLT gebruik word ooreen.

Fletcher neem B^TB net soos Marquardt, as die diagonaalmatriks λI.

3.8.6 Samevatting

Al die algoritmes berus op die kleinstekwadrate-metode. Die vernaamste verskille en ooreenkomste van voorgaande algoritmes kan in die volgende tabel saamgevat word.

ALGORITME / PARAMETER	PASPLT	MARQUART	OSBORNE	MDLS	NASH	FLETCHER
λ-waardes (i) $S(a^i+h)>S(a^i)$ (ii) $S(a^i+h)\leq S(a^i)$ (iii) Grense	(i) $R=\min\{0{,}2\ \log^2 \frac{\lambda S(a^i)}{S(a^i+h)} + DINC,\ 10\}$ $\lambda=\lambda R$ en $IC=IC+1$ (ii) $IC=1$ $\lambda=\lambda*DECR$ $IC\neq1$ $\lambda=\lambda$ (iii) $FNMN \leq \lambda \leq FNMX$	(i) $\lambda=\lambda*DINC$ (ii) $\lambda=\lambda*DECR$ -	(i) $\lambda=\lambda*DINC$ $IC=IC+1$ (ii) $IC=1$ $\lambda=\lambda*DECR$ $IC\neq1$ $\lambda=\lambda$ (iii) $FNMN \leq \lambda \leq FNMX$	(i) $\lambda=\lambda*DINC$ (ii) $\lambda=\lambda*DECR$	(i) $\lambda=\lambda*DINC$ (ii) $\lambda=\lambda*DECR$ (iii) $\lambda>FNMN$	(i) $R=R'/R''$ $0<\rho<\sigma<1$ Verminder λ as $R<\rho$ Vermeerder λ as $R>\sigma$ anders $\lambda=\lambda$
V-vektor	Vektor met elemente $V_i = \lvert a_i\rvert / \sum_{j=1}^{\ell} \lvert a_j\rvert$	eenheids-vektor	eenheidsvektor	Vektor met elemente $(A^TA)_{ii}$	Vektor met elemente $(A^TA)_{ii}+\alpha$	eenheidsvektor
Matriksont-bindings	JA	NEE	JA	NEE	JA	JA
Skalering	JA	NEE	NEE	NEE	NEE	JA
Beperking op a-waardes	JA	NEE	NEE	NEE	NEE	JA

Tabel 3.8.1 Die vernaamste verskille en ooreenkomste van die verskillende algoritmes

Verder verskil PASPLT van die ander metodes in die opsig dat as die afgeleides numeries bepaal word, word die funksie in 'n lineêre en nie-lineêre gedeelte geskei. Die afgeleides van die lineêre gedeelte word slegs een keer in die minimeringsprosedure bereken, daardeur word baie rekentyd bespaar.

4. BESKRYWING VAN DIE PROGRAM PASPLT

4.1 Inleiding

Ons taak is soos voorheen gestel om die koëffisiënte a_j van 'n gegewe lineêre of nie-lineêre funksie F(x,a), deur 'n aantal eksperimentele datapunte te bereken, sodanig dat die som van die kwadrate van die verskille tussen die eksperimentele waardes en die berekende funksiewaardes in die ooreenstemmende punte 'n minimum is.

Laat

$$\bar{X} = (x_{ij}) = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

die eksperimentele data voorstel, waar i die datastel en j die waarneming aandui.

Verder dui

$$\bar{Y} = (y_i) = [y_1, \cdots, y_m]^T$$

$$= [x_{1s}, x_{2s}, x_{3s}, \cdots, x_{ms}]^T$$ die eksperimentele s-de afhanklike veranderlike

en

$$\bar{Y}^* = (y_i^*) = [y_1^*, \cdots, y_m^*]^T$$ die berekende waardes aan

waar

$$y_i^* = F_i\ (x_{ij},\ a_k), \quad \begin{array}{l} j=1,2,\ \cdots,\ n \\ k=1,\ell \\ i=1,\ m \text{ met } \ell < n \end{array}$$

Die funksie wat geminimeer word, is

$$S = \sum_{i=1}^{m} (y_i - y_i^*)^2 = ||\bar{Y} - \bar{Y}^*||^2$$

Die funksie F wat gepas word kan lineêr en/óf nie-lineêr in a_i, $i=1,2, ---, \ell$ wees. Die koëffisiënt a_j kan begrens word deur

$$g_{a_{1,j}} \leq a_j \leq g_{a_{2,j}} \quad \forall j = 1,2, ---, \ell$$

Dit is moontlik om met die PASPLT-program in 'n enkele lopie 'n aantal verskillende funksies deur dieselfde eksperimentele datapunte te pas.

Die program bied ook die moontlikheid om die eksperimentele punte en/of die gepaste funksie m.b.v. 'n grafiektrekker te teken.

Verder is daar ook voorsiening gemaak om die interkorrelasie tussen die veranderlikes in tabelvorm op te som asook vir die berekening van die kolom gemiddeldes en standaardafwykings.

4.2 Beskrywing van PASPLT

Die programmatuur bestaan uit 'n hoofprogram sowel as Fortran- en Assemblersubroetines. Die verskillende Fortran- en Assemblersubroetines wat gebruik word, word respektiewelik in bylaes A en B gegee.

Die gebruiker verskaf slegs die subroetine PASDIM met ingangspunte PASGEN, PASFUN en PASAF.

Die parameters IR, IK en IA wat in PASDIM gedefinieer word, word deur middel van 'n COMMON-stelling na die subroetine DIMEN oorgedra om die ordes van die matrikse te bepaal.

Verdere veranderlikes wat as onafhanklike of afhanklike veranderlikes in die program gebruik kan word, kan in PASGEN genereer word.

In PASFUN word die funksie gedefinieer. Voordat PASFUN geroep word, word die betrokke stel eksperimentele waardes in 'n vektor x gestoor. Indien meer as een funksie gepas word, word die parameter NPAS deur middel van die COMMON-stelling oorgedra nl.

```
COMMON / FNOM / NPAS
```

As die parameter NUM=0, moet die afgeleides van 'n nie-lineêre funksie $\frac{\partial F}{\partial a_i}$, i=1, ---, ℓ in PASAF gegee word. Die afgeleides word in die vektor F gestoor, terwyl die veranderlikes op dieselfde wyse as in PASFUN in die vektor x gestoor word. Vir NUM=1, word die afgeleides numeries bereken volgens die metode van par. 3.6. PASPLT is in dubbelpresisie vir 'n IBM 370 model 155 geskryf en die CALCOMP 936 grafiektrekker word gebruik om die krommes te trek.

'n Vloeidiagram wat die prosedure illustreer verskyn op bls. 53 figuur 4.1. Indien 'n subroetine, bv. die stipsubroetines, nie beskikbaar is nie, kan dit vervang word deur 'n fopsubroetine, bv.

```
SUBROUTINE PLOTT
RETURN
END
```

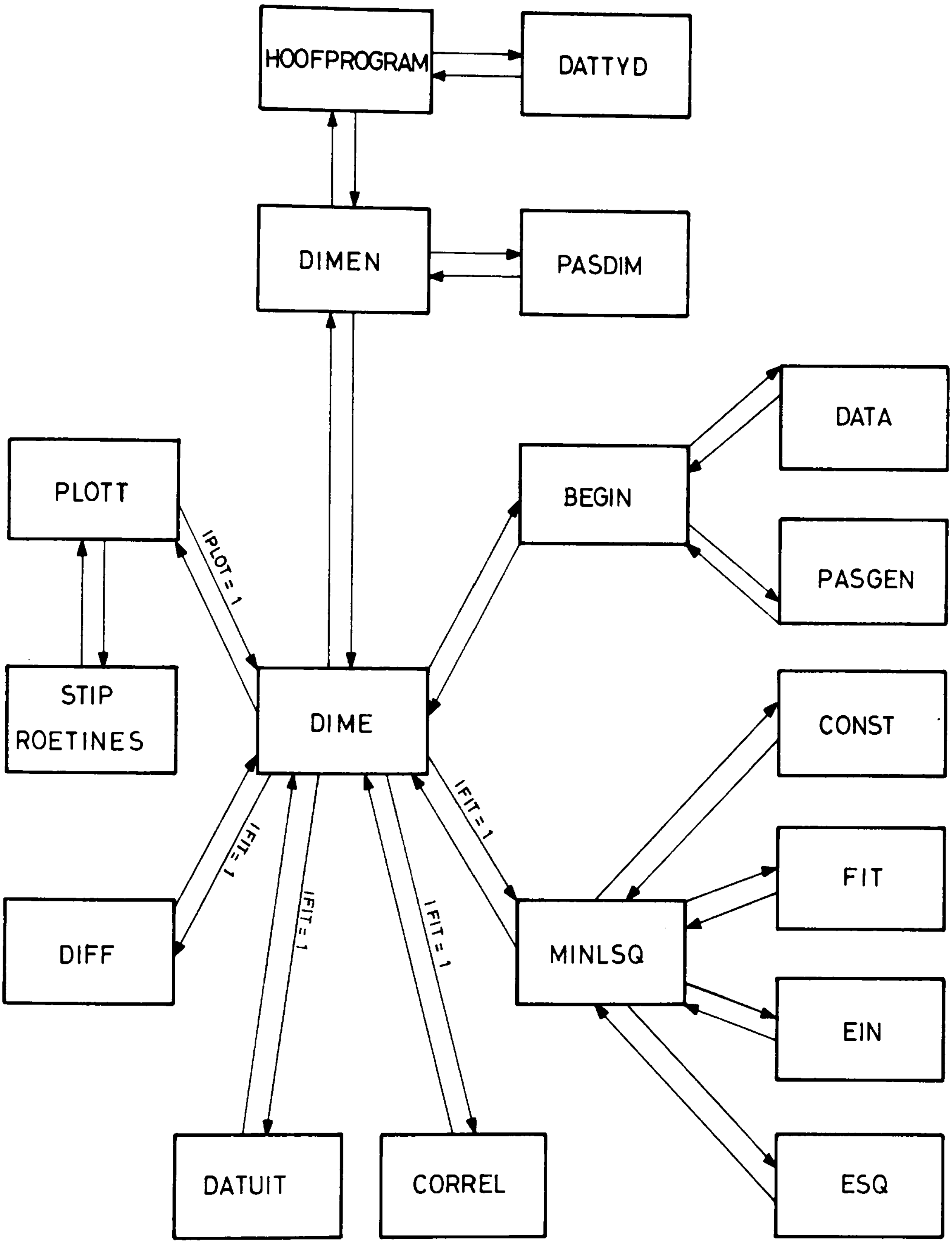


FIG. 4.1 GLOBALE VLOEIDIAGRAM

4.3 Beskrywing van Fortran-subroetines

4.3.1 DATTYD

Die subroetine word in die begin van die hoofprogram geroep en bepaal die datum en tyd, wat uitgedruk word bo-aan die bladsy met resultate.

Die volgende Assemblersubroetines word deur DATTYD geroep:

1. DATE - Om die datum te bepaal
2. HTYD - Om tyd te bereken
3. CPUTIM - Om die verwerkingstyd te bepaal. Om dit te kan doen word die subroetine aan die begin en einde van die program geroep.

4.3.2 DIMEN

Die parameters IR, IK en IA wat in PASDIM gedefinieer is, word in die subroetine DIMEN gebruik om die ordes van matrikse $\bar{X}$, $\overline{XT}$ en COR te bereken, waar $\bar{X}$ die matriks met eksperimentele datapunte, $\overline{XT}$ die matriks met die afgeleides van vektor $\underset{\cdot}{a}$ by elke datapunt en COR die matriks met korrelasiekoëffisiënte voorstel. In DIMEN word die volgende twee subroetines geroep, nl.

1. PASDIM, om die waardes van IR, IK en IA te kry;
2. DMENSI, om stoorplek dinamies toe te ken.

Die subroetine DIME kom as eksterne subroetine in die argumentlys van DIMEN voor.

4.3.3 DIME

In subroetine DIME word eerstens die subroetine BEGIN geroep: In BEGIN word die keuse parameters gedefinieer, die naamlys 'LEES' word ingelees en die invoerparameters word uitgedruk:

Indien die keuseparameter IFIT = 1, roep DIME die volgende subroetines:

(1) Die optimeringsprogram MINLSQ, om die funksie

$$S = \sum_{i=1}^{m} (y_i - y_i^*)^2 \text{ te minimeer.}$$

(2) Die korrelasieprogram CORREL, om die kolomgemiddeldes en variansies te bereken.

(3) DATUIT, om die data ingelees en die korrelasiekoëffisiënte matriks uit te druk.

Indien die keuseparameter IPLOT=1 en die parameter IER=0, word die stipprogram geroep om grafieke te teken. Die program stel IER=0 slegs as IFIT=0 of as IFIT=1 en die konvergensievereistes word bevredig.

4.3.4 BEGIN

In BEGIN word alle keuseparameters gedefinieer. 'n Lys van alle keuseparameters verskyn op bladsy A5 van Bylae A..

Alle datapunte word van kaarte met verwysingnommer 5 of van 'n lêer met verwysingnommer 8 gelees, en die data wat binne die bepaalde voorgeskrewe grense lê word in matriks $\bar{X}$ gestoor. Die subroetine PASGEN word hierna geroep om indien nodig verdere veranderlikes te genereer.

Die totale aantal veranderlikes en datastelle wat gebruik kan word, word deur die program bepaal. Indien die aantal datastelle groter as IR is, word 'n boodskap uitgedruk en slegs die eerste IR stelle word in die passing gebruik. Invoerparameters sal uitgedruk word as IDRUK groter as -1 is, en indien die beginwaardes van a_j $j=1, ---, \ell$ buite die voorgeskrewe grense lê of indien IK se waarde te klein is sal die program 'n boodskap uitdruk en eindig.

Die volgende subroetines word deur BEGIN geroep:

(1) PASGEN om nog veranderlikes voort te bring

(2) DATA, om data te lees en/of op 'n lêer te skryf

(3) ASSE, om die grafiektrekker se buffer leeg te maak.

4.3.5 DATA

Die subroetine word gebruik om data vanaf kaarte in 'n lêer te skryf. Indien NGO=5 word die data vanaf kaarte in 'n permanente of tydelike lêer met verwysingnommer 8 geskryf, sodoende kan die data herhaaldelik uitgesoek en gebruik word. Sodra die data in die lêer geskryf is, word NGO gelyk aan 8 gestel en dit is dus vir die gebruiker nie nodig om NGO te verander nie. Indien IDAT=1 sal die data wat op kaarte of in die lêer verskyn uitgedruk word.

4.3.6 MINLSQ

In subroetine MINLSQ word die koëffisiënte $a_j, j=1, ---,$ van die passing iteratief bereken, sodanig dat die som van die kwadrate van die verskille tussen die berekende funksiewaardes en die ooreenstemmende eksperimentele waardes 'n minimum is. Die metode wat vir hierdie minimering gebruik word, is dié wat in par. 3.8 beskryf is.

Vir 'n beskrywing van die verskillende parameters in die argumentlys sien bl. A15, bylaag A.

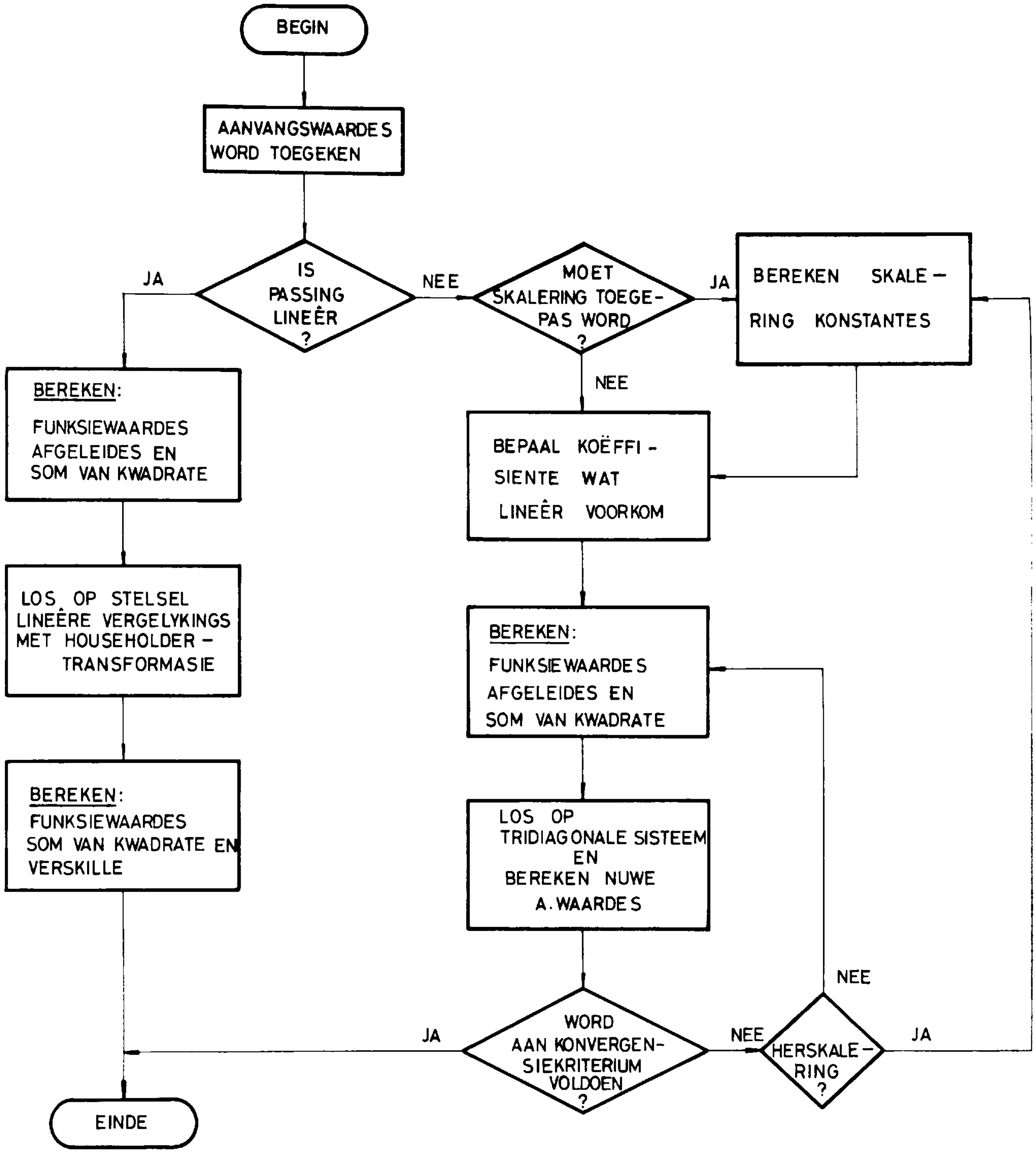
Die volgende subroetines word deur MINLSQ gebruik:

1. FIT, om die funksiewaardes en afgeleides te bereken;
2. CONST, om te bepaal watter koëffisiënte lineêr voorkom;
3. EIN, om die skalare produk van twee vektore te bereken;
4. ESQ, om die som van die kwadrate van die komponente van 'n vektor te bereken.

'n Skematiese voorstelling van die iteratiewe minimeringsprosedure verskyn op bls. 58 figuur 4.2.

4.3.7 FIT

As FIT deur MINLSQ geroep word en IN=0, word die verskille tussen die eksperimentele en berekende waardes bereken en in die vektor F gestoor. As IN=1, word die afgeleides $\frac{\partial F_i}{\partial a_j}$, j=1,2, --, en i = 1,2, ---, m bereken en in die matriks B gestoor, en die verwerkingstyd wat nog beskikbaar is, word bepaal. Indien die beskikbare verwerkingstyd minder of gelyk aan die parameter TYD is, word IN=-1 en IPONS=1 gestel. Die a-waardes word dan op kaarte gepons en die program eindig met IER=5.



FIGUUR 4.2 LOGIESE DIAGRAM VAN MINLSQ

4.3.8 DIFF

In DIFF word die regressiekoëffisiënt, die som van kwadraatafwykings, die som van die absolute waardes en die wortel van die gemiddelde kwadraatafwyking van die persentasiefout, bereken en uitgedruk.

4.3.9 CORREL

In die subroetine CORREL word die gemiddeldes en standaardafwykings bereken asook indien ICOR=1 die korrelasietabel van die eksperimentele data.

4.3.10 DATUIT

In die subroetine DATUIT word, indien IDRUK > 0 is, die gemiddeldes, standaardafwykings en as ICOR=1, die korrelasietabel uitgedruk.

4.3.11 CONST

Die a-waardes wat lineêr voorkom word in CONST bepaal. As a_j lineêr is word 'n parameter IK(J) gelyk aan 1 gestel so nie gelyk aan 0. As IK(J)=1, bly die afgeleides in die j^{de} kolom van B konstant en word net een maal in die subroetine bereken.

4.3.12 PLOTT

As IPLOT=1 en IER=0, word die subroetine PLOTT geroep. As IPUNT=1, word die afhanklike eksperimentele waardes wat tussen gegewe grense lê uitgesoek en teen 'n bepaalde onafhanklike veranderlike geteken. Die IX(NN) veranderlike word teen die IX(NN-1) veranderlike geteken.

As ILYN=1, word die funksie tussen XBEG en XEND, YBEG en YEND geteken. As IDENT > 0 is, word die waarde van X(I, IDENT) by die ooreenstemmende punt geskryf.

4.3.13 ASSE

In die subroetine ASSE word die stipsubroetines beskikbaar gestel en indien IAS=1, word 'n assestelsel geteken, terwyl die opskrif wat in die vektor PASF gestoor is, bo-aan die grafiek geskryf word.

Die waardes van die veranderlikes wat konstant gehou word, word onder die opskrif geskryf.

4.3.14 LYN

Die subroetine LYN, teken die gedeelte van die grafiek wat tussen XBEG en XEND t.o.v. X-as en tussen YBEG en YEND t.o.v. die Y-as lê. Aan die einde van die grafiek word die naam in XNAME(IX (NSOEK)) en die waarde van XP(NSOEK) geskryf.

NDIGD is die aantal intervalle waarin die X-as verdeel word en by elke endpunt van so 'n interval word die funksiewaarde bereken om die grafiek te teken.

4.4 Die Gebruik van PASPLT

4.4.1 Inlees van data

Die eksperimentele data kan òf van kaarte òf van 'n lêer ingelees word. As die data van kaarte gelees word, d.i. NGØ=5, word die data wat op die kaarte verskyn in 'n tydelike of permanente lêer, met lêernommer 8 geskryf, die program stel dan NGØ gelyk aan 8 en as dié data weer gebruik word sal dit outomaties van skyf gelees word.

In die naamlys '&LEES' word die formaatstelling waarop die data ingelees word, in die vektor FORM gestoor bv. FORM = '(8F10.0)'.

Die data wat vir die passing benodig word, word uit die eksperimentele data, soos dit in die lêer geskryf is, uitgesoek en slegs dié data word in die matriks $\bar{X} = (x_{ij})$ gestoor.

Die waarde van die parameter KX dui aan op hoeveel veranderlikes grense is en die kolomindekse van hierdie veranderlikes word in die vektor KVX gestoor. Die grenswaardes word in die matriks (GX(I,J), I=1,2 en J=1,KX) gegee, waar GX(1,J) die onderste en GX(2,J) die boonste grens aandui.

As NGØ=5 en die data dus van kaarte gelees word moet aan die einde van die data 'n kaart geplaas word met 'n ster in kolom 1.

Deur IDAT=1 te stel word 'n lys van al die data wat op die kaarte gepons is, gedruk. As in dieselfde lopie, die grenswaardes verander word vir 'n volgende passing moet die data weer gelees word, en die parameter LEESX moet gelyk aan 1 gestel word, sodat die nodige data weer uitgesoek kan word. Nadat die data uitgesoek is stel die program LEESX=0.

Die program kan maksimaal 50 veranderlikes hanteer, terwyl daar geen beperking op die aantal eksperimentele datastelle is nie. Die aantal veranderlikes wat ingelees en gegenereer word, word deur die program bepaal.

Alle verdere parameters vir die passing word in die naamlys '&LEES' gedefinieer en verskillende naamlyste vir verskillende passings kan na mekaar ingelees word. Sien voorbeelde 3, 4, 5, 6 en 8 in bylaag C.

4.4.2 Die lineêre funksie

Die lineêre funksie $y = a_1 + \sum_{j=2}^{\ell} a_j x_{j-1}$ kan gepas word deur die parameter NN gelyk aan die aantal koëffisiënte te stel en in die vektor IX die kolomindekse van die veranderlikes te stoor wat in die passing gebruik word. Die kolomindekse word so gestoor dat IX(NN) die afhanklike veranderlike is. Bv. as NN=4 en IX=1, 4, 5, 7, 8 sal die volgende funksie gepas word:

$$Y = a_1 + a_2 X_1 + a_3 X_4 + a_4 X_5 + a_5 X_7$$

met X_8 = Y die afhanklike veranderlike. Omdat bogenoemde funksie lineêr in a_j, j=1,2, ---, 5 is, kan hierdie a's regstreeks bepaal word.

Vir die passing moet die parameter IT=0 wees, en omdat dit nie nodig is om die funksie in PASFUN te definieer nie, moet NL=0 gestel word. Sien voorbeeld 5. Alle ander lineêre funksies moet in die subroetine PASDIM onder ingangspunt PASFUN gedefinieer word. Die parameter NL dui die aantal koëffisiënte a_j aan en omdat die funksie lineêr is, moet IT=0 wees. Verder stel NN die aantal (IX(J), J=1,NN) veranderlikes voor met IX(NN) die afhanklike veranderlike.

Vir die lineêre funksie is dit nie nodig om beginwaardes vir die a's te gee nie.

4.4.3 Die nie-lineêre funksie

Vir die passing van 'n nie-lineêre funksie moet redelike aanvangwaardes vir die a_j's in die naamlys verskaf word. Dit is ook moontlik om sommige of selfs alle a-waardes te begrens. Dit word gedoen deur aan die parameter KA, 'n waarde toe te ken gelyk aan die aantal a-waardes wat begrens word en in die vektor KVA aan te dui op watter a-waardes grense gestel is. Die grense word in die matriks (GA(I,J), I=1,2, J=1,KA) gegee, waar GA(1,J) die onderste en GA(2,J) die boonste grens van A(J) voorstel.

Bv.

As ons a_3 en a_5 respektiewelik wil begrens tussen 3,0 en 5,0, 4,0 en 8,0 dan is KA=2, KVA=3,5, en GA=3.0, 5.0, 4.0, en 8.0.

Deur 'n parameter IV(J)=1 te stel is dit moontlik om die ooreenstemmende a_j-waarde gedurende die iteratiewe minimeringsproses konstant te hou. Dit kan in die ontwikkeling van die funksie handig wees. Die parameter IT dui die maksimum aantal iterasies aan, terwyl ITSIK die aantal iterasies voorstel, voordat herskalering van die a-waardes en afgeleides plaasvind.

4.4.4 Die gekombineerde lineêre en nie-lineêre funksie

Gestel die funksie bestaan uit 'n lineêre en 'n nie-lineêre gedeelte. Die funksie kan dan geskryf word as Y=F(X,A) + G(X,A) waar F(X,A) die lineêre gedeelte en G(X,A) die nie-lineêre gedeelte voorstel. Die afgeleides $\frac{\partial F}{\partial a_j}$ van die lineêre gedeelte bly gedurende die iteratiewe minimeringsproses konstant.

Beskou as voorbeeld die volgende funksie, wat ons deur 'n stel datapunte wil pas:

$$Y = a_1 + a_2 X_1^2 + a_3 e^{a_4 X_2}$$

$F = F(X,A) = a_1 + a_2 X_1^2$ stel die lineêre gedeelte

en

$G = G(X,A) = a_3 e^{a_4 X_2}$ stel die nie-lineêre gedeelte voor

dan is: $\frac{dY}{da_1} = 1$ en $\frac{dY}{da_2} = X_1^2$

As NUM=1 word die afgeleides numeries bepaal. Die a's word nou met ΔA vermeerder en die afgeleides word weer bereken.

Indien $\left| \frac{dF}{da_j} - \frac{dF}{d(a_j+\Delta a_j)} \right| \leq 1.D-8$

word aanvaar dat die a_j lineêr voorkom en die afgeleides $\frac{\partial F_i}{\partial a_j}$ is konstant $\forall$ i = 1, 2, ---, m.

As NUM=1 word alle afgeleides soos in par. 3.6 numeries, bepaal. D.w.s.

$$\frac{\partial F}{\partial a_j} = \frac{F(a_j+h) - F(a_j-h)}{2h}$$

waar h deur die program bepaal word.

Die afgeleides wat onafhanklik van a_j's is word in die matriks (B(I,NL+J), I=1, M, J=1,NC) gestoor waar NC die aantal a_j's is wat lineêr in F voorkom. Vir elk van hierdie a_j's word die parameter IK(J) = 1 gestel, en vir elke a_j wat nie-lineêr voorkom word IK(J)=0 gestel. Aangesien daar in die meeste funksies lineêre terme in a_j voorkom, kan baie rekenwerk bespaar word deur die funksie in 'n lineêre en 'n nie-lineêre gedeelte te skei, omdat die afgeleides van die lineêre terme net eenmaal bereken word. Hierdie skeiding is veral voordelig as M en NC groot is.

Die parameter IT en ITSIK het dieselfde betekenis as in 4.4.3.

4.4.5 Subroetine PASDIM

Die subroetine PASDIM besit 4 ingangspunte nl. PASDIM, PASGEN, PASFUN en PASAF.

PASDIM het geen argumentlys nie. Die parameters IR, IK en IA wat in PASDIM gedefinieer is, word deur middel van 'n COMMON-stelling met die naam DIM na die subroetine DIMEN oorgedra. Die waardes wat aan hierdie parameters toegeken word, moet aan die volgende vereistes voldoen:

IR - Die parameter moet groter of gelyk wees aan die maksimum aantal datastelle wat in die passings gebruik word.

IK - Die parameter moet groter of gelyk wees aan die aantal veranderlikes in 'n datastel, maar kleiner of gelyk aan 50.

IA - Die parameter moet groter of gelyk wees aan die som van twee keer die aantal koëffisiënte wat lineêr en eenkeer die aantal koëffisiënte wat nie-lineêr in die passings voorkom, maar kleiner of gelyk aan 50.

Slegs bogenoemde parameters is in PASDIM gedefinieer om daardeur meer rekenaargeheue vir die program beskikbaar te stel. Dit is dus nie nodig om enige dimensiestellings te verander of te definieer nie.

PASGEN word gebruik om veranderlikes te verander of om nog veranderlikes voort te bring. Die argument X en M word in die argumentlys oorgedra, waar X 'n vektor met fopdimensies is. Die subroetine PASGEN word vir elke datastel geroep, maar voordat PASGEN geroep word, word die M veranderlikes van die betrokke datastel in die vektor X gestoor. Gestel ons wil die volgende funksie pas:

nl. $Y = a_1 + a_2X_1 + a_3X_1^2 + a_4X_1^3$

D.w.s. M=2, en ons kan X_1^2 en X_1^3 in PASGEN bereken deur:

```
X(M+1) = X(1) * X(1)
X(M+2) = X(1) * X(1) * X(1)
```

Die funksie is in dieselfde vorm as die wat in par. 4.4.2 bespreek is, en dit is dus slegs nodig om die parameters NN en IX te definieer. In bogenoemde geval is NN=4, en IX=1, 3, 4, 2. Die funkise kan ook soos gegee in PASFUN gedefinieer word.

Die funksie word in PASFUN gedefinieer. Sy argumentlys bevat die volgende 3 veranderlikes nl. X, A en FPAS, met X en A vektore en FPAS 'n skalaarwaarde. Voordat PASFUN geroep word, word die betrokke stel veranderlikes in die vektor X, gestoor, terwyl die koëffisiënte a_j in die vektor A en die berekende funksiewaarde in FPAS gestoor word.

Die parameter NPAS wat in die naamlys gedefinieer word stel ons in staat om verskillende funksies in dieselfde lopie te pas. Hierdie parameter word deur middel van 'n COMMON-stelling genaamd FNOM na PASDIM oorgedra. Verder kan die betrokke datastelnommer I en die aantal datastelle NX deur 'n COMMON-stelling genaamd XIJ na PASDIM oorgedra word.

Die gebruik van NPAS word in die volgende voorbeeld geïllustreer:

```
      SUBROUTINE PASDIM
      IMPLICIT REAL * 8 (A-H, O-Z)
      DIMENSION A(1), X(1)
      COMMON/DIM/IR, IK, IA
      COMMON/FNOM/NPAS  /XIJ / I, NX
      IR = 10
      IK = 10
      IA = 5
      RETURN
      ENTRY PASAF (X,A,F)
      ENTRY PASGEN (X,M)
      X(M+1) = X(1) * X(1)
      RETURN
      ENTRY PASFUN (X, A, FPAS)
      GO TO (1, 2, 3), NPAS
1     FPAS = A(1) + A(2) * X(1)
      RETURN
2     FPAS = A(1) + A(2) * X(3)
      RETURN
3     FPAS = A(1) + DEXP(X(1) * A(2)).
      RETURN
      END
```

4.4.6 Die minimeringsproses

In die lineêre geval word die normaalvergelykings opgelos m.b.v. Householder-transformasies. Geen iterasies is nodig nie dus is IT=0.

Vir die nie-lineêre geval word 'n iteratiewe metode, naamlik die van Gauss-Newton-Marquardt [2] gebruik. Die parameter IT dui die maksimum aantal iterasies aan. Indien minder as IT iterasies nodig is vir konvergensie, word die parameter IER nul gestel en na die roepprogram teruggekeer. Indien die konvergensie nie in IT iterasies bereik word nie is IER=1. Vir die betekenis van ander waardes van IER sien bylaag A p.A-1.

Daar is twee soorte konvergensies nl. absolute en relatiewe konvergensie.

Vir hierdie twee gevalle konvergeer die oplossings as die volgende voorwaardes bevredig word:

(a) Vir absolute konvergensie as

$$\left| a_i^j - a_i^{j-1} \right| \leq \text{EPS} \ \forall i = 1, ---, \ell$$

(b) Vir relatiewe konvergensie as:

$$\left| \frac{a_i^j - a_i^{j-1}}{a_i^j} \right| \leq \text{EPS} \ \forall \ i=1, ---, \ell$$

waar a_i^j die waarde van a_i se j-de iterasie en EPS die konvergensie-norm is.

M.b.v. die parameter ICON bepaal ons die soort konvergensie wat ons wil gebruik. Vir 'n absolute konvergensietoets moet ICON=0 en vir 'n relatiewe konvergensietoets moet ICON=1 wees.

As NSK=1 word skaleringsfaktore bereken, om die objekfunksie en koëffisiënte te skaal. Indien geen skalering nodig is nie word NSK=0 gestel. Na elke ITSIK-iterasies word nuwe skaleringsfaktore bereken.

4.4.7 Parameters wat die uitdruk beheer

Verskillende parameters beheer die drukstuk vir die invoer en die resultate. Hierdie parameters word in die naamlys '&LEES' gedefinieer en kan na willekeur verander word. Die funksie en die betekenis van die verskillende waardes wat aan die parameters toegeken word verskyn in bylaag A op p.A4 en A5.

4.4.8 Die grafiekprogram

Indien 'n grafiek verlang word, moet die parameter IPLOT=1 wees. 'n Grafiek sal slegs geteken word as IER=0 is. Die parameter IER sal die waarde 0 aanneem indien geen passing gedoen word nie, d.w.s. IFIT=0, òf as die passing die konvergensievoorwaardes bevredig. As IPLOT=1, maar die aantal iterasies IT nie genoeg om konvergensie te bereik nie, sal IER=1 wees, en dus geen grafiek geteken word nie.

Dit is moontlik om die eksperimentele waarde vir die afhanklike veranderlike teen enige ander veranderlike te stip om 'n prentjie te kry van hoe die punte versprei lê. Die punte wat abnormaal ver buite die gebied lê as gevolg van eksperimentele of ponsfoute, kan reggestel word. Dit is soms ook moontlik om die grense so te kies dat die ongewenste punte buite die grense val en dus nie in die passing gebruik sal word nie. Sien Bylae C voorbeeld 1 op bladsy C1, waar die vierde punt ver afwyk van al die ander punte.

Die funksie wat in PASFUN gedefinieer is, kan ook geteken word, deur een van die onafhanklike veranderlikes oor 'n gebied te varieer, terwyl die ander veranderlikes oor die gebied konstant gehou word. Die parameter NSOEK wat in die naamlys gedefinieer moet word, dui die aantal veranderlikes aan wat konsant gehou moet word, terwyl die vektor XP hierdie konstante waardes bevat. Die veranderlikes wat oor die gebied XBEG tot XEND varieer, is dié wat in IX(NN-1) gestoor is.

Om die funksie wat in PASFUN gedefinieer is te teken moet IPLOT=1 en ILYN=1 wees. As IPUNT=1 is, sal die afhanklike teen die onafhanklike eksperimentele datapunte gestip word, d.w.s. IX(NN) word teen IX(NN-1) gestip sodanig dat die NSOEK veranderlikes tussen die grense (XP(I) ± TOLX(I), I=1, NSOEK), lê. Sien Bylae C voorbeeld 3, 4 en 5.

Dit is ook moontlik om die grafieke van die funksies wat in PASFUN gedefinieer is, vir verskillende stelle koëffisiënte te teken. Die nut hiervan is daarin geleë dat verskillende funksies geteken kan word om sodoende vooraf redelike aanvangwaardes vir die koëffisiënte te kry. Sien Bylae C voorbeeld 7. Dit kan 'n besparing in verwerkingstyd teweeg bring omdat daar moontlik minder iterasies vir konvergensie nodig sal wees.

4.4.9 Numeriese Voorbeelde

Ter illustrasie van die gebruik van PASPLT word die volgende voorbeelde in Bylae C gedoen:

1. Die stip van eksperimentele data deur die grafiektrekker. p.C1.

2. Die voortbringing van veranderlikes en die passing van 'n funksie. p.C8.

3. Passingsgrafieke. p.C13.

4. Passingsgrafieke met begrensde veranderlikes. p.C20.

5. Die toepassing van verskillende funksies in dieselfde lopie. p.C27.

6. 'n Toepassing waar slegs grafieke geteken word. p.C34.

7. 'n Nie-lineêre passing. p.C38.

8. Bepaling van die rekenaargeheue wat gebruik word deur 'n lineêre passing. p.C44.

5. RESULTATE

5.1 Voorbeelde

Om die doeltreffendheid van PASPLT te vergelyk met programme gebaseer op die algoritmes wat in par. 3.8 beskryf is, gebruik ons die voorbeelde en data van Meyer en Roth [20]. Dieselfde voorbeelde is ook deur Nash [21] in sy vergelyking van die verskillende algoritmes gebruik. Ons sou graag ons vergelyking wou baseer op die verwerkingstye maar omdat die gegewens vir die metodes nie beskikbaar is nie, is ons verplig om die vergelyking te maak aan hand van die gegewens wat uit bogenoemde artikels geneem is.

Voorbeeld 1

Funksie: $F = \dfrac{a_1\ a_2\ x_1}{1 + a_1\ x_1 + a_2\ x_2}$

DATA		
x_1	x_2	y
1,0	1,0	0,126
2,0	1,0	0,219
1,0	2,0	0,076
2,0	2,0	0,126
0,1	0,0	0,186

Tabel 5.1.1 Data vir voorbeeld 1

Beginwaardes: a^o = (10,39; 48,83; 0,74); $S(a^o)$ = 0,0365
Eindwaardes: a = (3,13; 15,16; 0,78) ; $S(a)$ = 0,4E-4

Die resultate is dieselfde vir al die verskillende algoritmes. In die geval was vir λ_o = 1.D-5 ses iterasies (minimum aantal interasies) en 17 funksieberekeninge deur PASPLT gedoen.

```
VOORBEELD 1

                                                ** INVOER **

IA    =   3      IPR   =   1      IFIT  =   1      IDENT =  -1      ITSIK =   9      MM    =   3
IR    =  65      ICON  =   1      ILYN  =   1      IDRUK =   0      KA    =   0      NL    =   3
IK    =   5      ICOR  =   0      IPEN  =   1      IPLOT =   0      KX    =   0      NN    =   3
IT    =  20      IDAT  =   0      IXAS  =   2      IPONS =   0      LEESX =   1      NGO   =   5
IAS   =   1      IDES  =   1      IYAS  =   3      IPUNT =   0      M     =   3      NSK   =   0

IX    =   1    2    3
IV    =   0    0    0
EPS   =  1.00D-08   TYD   =  1.00E 01   YDUIM =  1.10E 01   FNU   =  1.00D-05
FORM  =(8F10.0)
A(I)  =        10.390000            48.830000            0.74000000

                                                ** RESULTATE **

MINIMERING SOM VAN VIERKANTE      N =  3      M =   5
ITERASIE    0                SKW   = 3.6552445D-02      FNU  = 1.0000000D-05      FSC =  1.000000
                 A    =      10.390000         48.830000          0.74000000

ITERASIE    1    ITF   =  2  SKW = 7.8747318D-02      FNU  = 7.1888675D-05      FNN        2.154
                 V     =     0.17328219        0.81437625         1.23415610D-02
                 A     =    -0.81947796       -3.3876726          0.75656954

ITERASIE    2    ITF   = 11  SKW = 4.3227711D-02      FNU  = 0.2276534          FNN        0.5489
                 V     =     0.16509351        0.68248663         0.15241986
                 A     =     1.0693162         3.3149324          0.33219778

ITERASIE    3    ITF   = 12  SKW = 1.7911507D-02      FNU  = 0.2276534          FNN        0.4144
                 V     =     0.22672073        0.70284535         7.04339165D-02
                 A     =     5.9314598         16.566841          0.31670559

ITERASIE    4    ITF   = 15  SKW = 6.3687026D-03      FNU  = 0.3568880          FNN        0.3556
                 V     =     0.25998064        0.72613790         1.38814598D-02
                 A     =     3.0760349         14.650440          0.59867976

ITERASIE    5    ITF   = 16  SKW = 4.4989115D-05      FNU  = 0.3568880          FNN        7.0641D-03
                 V     =     0.16785860        0.79947156         3.26698336D-02
                 A     =     3.1362283         15.300905          0.78177615

ITERASIE    6    ITF   = 17  SKW = 4.3552735D-05      FNU  = 0.1784440          FNN        0.9681
                 V     =     0.16318451        0.79613805         4.06774449D-02
                 A     =     3.1311994         15.158771          0.78012198
```

Voorbeelde 2 en 3

Funksie: $f_1 = 10(a_2-a_1^2)$ $f_2 = (1-a_1)$ met $S(a) = f_1^2 + f_2^2$

Beginwaardes: Voorbeeld 2 $a^o=(-1,2;\ 1,0)$ $S(a^o) = 24,2$

Voorbeeld 3 $a^o=(-0,86;\ 1,14)$ $S(a^o) = 19,5$

Eindwaardes: Voorbeeld 2 en 3 $a=(1;\ 1)$, $S(a) = 0$

Dit word die Rosenbrock- paraboliese valleiprobleem genoem. Vir PASPLT was slegs 3 of 4 iterasies nodig om dieselfde resultate te verkry waarvoor die ander algoritmes, minstens 15 iterasies benodig.

```
VOORBEELD 2                                          ** RESULTATE **

MINIMERING SOM VAN VIERKANTE         N =   2        M =     2
ITERASIE    0                    SKW    =  24.20000              FNU = 1.0000000D-06      FSC =   1.000000
                  A       =     -1.2000000            1.0000000

ITERASIE    1     ITE     =   2   SKW =   2342.457             FNU = 1.0000000D-05      FNN           96.80
                  V       =      0.54545455           0.45454545
                  A       =      0.99997581          -3.8399419

ITERASIE    2     ITE     =   3   SKW = 3.7219746D-11          FNU = 1.0000000D-05      FNN         1.5889D-14
                  V       =      0.20661008           0.79338992
                  A       =      0.99999391           0.99998778

ITERASIE    3     ITE     =   4   SKW = 1.3794043D-19          FNU = 5.0000000D-06      FNN         3.7061D-09
                  V       =      0.50000153           0.49999847
                  A       =       1.0000000            1.0000000


VOORBEELD 3                                          ** RESULTATE **

MINIMERING SOM VAN VIERKANTE         N =   2        M =     2
ITERASIE    0                    SKW    =  19.49162              FNU = 1.0000000D-06      FSC =   1.000000
                  A       =     -0.86000000           1.1400000

ITERASIE    1     ITE     =   2   SKW =   1196.870             FNU = 1.0000000D-05      FNN           61.40
                  V       =      0.43000000           0.57000000
                  A       =      0.99999496          -2.4595913

ITERASIE    2     ITE     =   3   SKW = 1.2261505D-11          FNU = 1.0000000D-05      FNN         1.0245D-14
                  V       =      0.28905045           0.71094955
                  A       =      0.99999650           0.99999299

ITERASIE    3     ITE     =   4   SKW = 1.4989489D-20          FNU = 5.0000000D-06      FNN         1.2225D-09
                  V       =      0.50000088           0.49999912
                  A       =       1.0000000            1.0000000
```

Voorbeeld 4 en 5

Funksie: $$F = a_3\left(e^{-a_1x_1} + e^{-a_2x_2}\right)$$

DATA			
		Voorbeeld 4	Voorbeeld 5
x_1	x_2	y_1	y_2
0,0	0,0	40,2	40,0
0,6	0,4	11,0349	10,0
0,6	1,0	4,48869	5,0
1,4	1,4	2,46137	2,5
2,6	1,4	2,45137	2,5
3,2	1,6	1,82343	2,0
0,8	2,0	1,00094	1,0
1,6	2,2	0,741352	0,7
2,6	2,2	0,741352	0,8
4,0	2,2	0,741352	0,7
1,2	2,6	0,406863	0,4
2,0	2,6	0,406862	0,4
4,6	2,8	0,301411	0,3
3,2	3,0	0,223291	0,22
1,6	3,2	0,165418	0,2
4,2	3,4	0,122545	0,1
2,0	3,8	0,067254	0,05
3,2	3,8	0,067254	0,07
2,8	4,2	0,036910	0,03
4,2	4,2	0,036910	0,03
5,4	4,4	0,027343	0,03
5,6	4,8	0,015006	0,02
3,2	5,0	0,011117	0,01

Tabel 5.1.2 Data vir voorbeelde 4 en 5

```
     VOORBEELD 5                                ** RESULTATE **

MINIMERING SOM VAN VIERKANTE        N =  3       M =   23
ITERASIE    0                  SKW    =  226.8520         FNU = 1.00000000D-02      FSC =  1.000000
                 A       =     12.000000          1.0000000           25.000000

ITERASIE    1    ITE     =  1  SKW =   8.038339         FNU = 1.00000000D-02     FNN         3.5434D-02
                 V       =    0.31578947      2.63157895D-02      0.65789474
                 A       =     25.102007       1.2604175          19.837471

ITERASIE    2    ITE     =  2  SKW =   1.423041         FNU = 5.00000000D-03     FNN         0.1770
                 V       =    0.54333472      2.72818266D-02      0.42938345
                 A       =     25.121997       1.4619351          19.892340

ITERASIE    3    ITE     =  3  SKW =   1.252565         FNU = 2.50000000D-03     FNN         0.8802
                 V       =    0.54053383      3.14555162D-02      0.42801065
                 A       =     25.171994       1.5046640          19.917958

ITERASIE    4    ITE     =  4  SKW =   1.251896         FNU = 1.25000000D-03     FNN         0.9995
                 V       =    0.54023397      3.22926582D-02      0.42747338
                 A       =     25.358027       1.5074939          19.920221

ITERASIE    5    ITE     =  5  SKW =   1.251894         FNU = 6.25000000D-04     FNN          1.000
                 V       =    0.54200331      3.22122246D-02      0.42577547
                 A       =     26.027448       1.5076092          19.920343

ITERASIE    6    ITE     =  6  SKW =   1.251893         FNU = 3.12500000D-04     FNN          1.000
                 V       =    0.54846125      3.17689698D-02      0.41976978
                 A       =     27.871000       1.5076134          19.920348

ITERASIE    7    ITE     =  7  SKW =   1.251892         FNU = 1.56250000D-04     FNN          1.000
                 V       =    0.56534659      3.05810373D-02      0.40407237
                 A       =     30.504324       1.5076135          19.920349
```

Beginwaardes: Voorbeeld 4 $a^o = (12,0;\ 1,0;\ 25,0)$, $S(a^o) = 216,0$

Voorbeeld 5 $a^o = (12,0;\ 1,0;\ 25,0)$, $S(a^o) = 226,9$

Eindwaardes: Voorbeeld 4 $a = (14,3;\ 1,5;\ 20,1)$, $S(a) = 10^{-12}$

Voorbeeld 5 $a = (30,5;\ 1,5;\ 19,9)$, $S(a) = 1,25$

In voorbeeld 4 het Meyer en Roth gevind dat $S(a^o) = 187$ is, terwyl $S(a^o)=216,03$ is as PASPLT gebruik word. Die verskil is waarskynlik te wyte aan die feit dat PASPLT in dubbelnoukeurigheid werk. Die a_1-waarde het in sommige gevalle baie groot geword. In PASPLT kan die probleem voorkom word deur a_1 byvoorbeeld tydens die eerste iterasie konstant te hou.

In voorbeeld 5 het a_1 vir sommige λ_o groter as 40 geword, hoewel die som van die kwadrate van die verskille onveranderd gebly het. Omdat slegs drie x_1-waardes kleiner as 1 is, het die waarde van a_1 in $e^{-a_1x_1}$ 'n baie geringe invloed op die funksiewaarde. Die minimum aantal iterasies wat PASPLT vir voorbeelde 4 en 5 gebruik is resp. 8 en 5. Dit word slegs deur die Nash-metode oortref.

Voorbeeld 6 en 7

Funksie: $F = a_1 + a_2\ e^{a_3x}$

DATA		
x	y-voorbeeld 6	y-voorbeeld 7
1	16,7242	16,7
5	16,8262	16,8
10	16,9657	16,9
15	17,1198	17,1
20	17,2902	17,2
25	17,4785	17,4
30	17,6865	17,6
35	17,9165	17,9
40	18,1706	18,1
50	18,7619	18,7

Tabel 5.1.3 Data vir voorbeeld 6 en 7

```
MINIMERING SOM VAN VIERKANTE       N =   3       M =    16
ITERASIE    0                      SKW    = 1.1289259D 12        FNU
                A        =    2.00000000D-02       4000.0000

ITERASIE    1   ITE      =  1  SKW = 1.2806113D 11       FNU =
                V        =    4.70586023D-06       0.94117204
                A        =   -0.29658439           3365.2507

ITERASIE    2   ITE      =  2  SKW = 1.4204664D 10       FNU =
                V        =    8.26044978D-05       0.93728749
                A        =   -22.163269            2776.9625

ITERASIE    3   ITE      =  3  SKW = 1.5411610D 09       FNU =
                V        =    7.39183241D-03       0.92616486
                A        =   -305.88293            2277.5693

ITERASIE    4   ITE      =  4  SKW = 1.6994044D 08       FNU =
                V        =    0.11082331           0.82517772
                A        =   -145.65755            1830.0948

ITERASIE    5   ITE      =  5  SKW = 1.8326329D 07       FNU =
                V        =    6.83763421D-02       0.85910543
                A        =   -63.568298            1430.3004

ITERASIE    6   ITE      =  6  SKW =   1889303.          FNU =
                V        =    3.90895716D-02       0.87952379
                A        =   -29.071362            1081.1312

ITERASIE    7   ITE      =  7  SKW =   178214.2          FNU =
                V        =    2.38155460D-02       0.88567334
                A        =   -14.524521            788.76030

ITERASIE    8   ITE      =  8  SKW =   13490.07          FNU =
                V        =    1.62689107D-02       0.88349014
                A        =   -8.5491814            565.42699

ITERASIE    9   ITE      =  9  SKW =   536.4321          FNU =
                V        =    1.32573115D-02       0.87681397
                A        =   -6.3776932            428.77229

ITERASIE   10   ITE      = 10  SKW =   3.350622          FNU =
                V        =    1.29441842D-02       0.87023745
                A        =   -5.8490645            379.93979
ITERASIE   11   ITE      = 11  SKW = 0.4066671           FNU =
                V        =    1.33659978D-02       0.86821993
                A        =   -5.8071667            374.69565
```

Beginwaardes: A^o = (20; 2; 0,5), $S(a^o)$ = 1,E22

Eindwaardes : Voorbeeld 6 a = (15,5; 1,2; 0,02), S(a)=1, E-12

Voorbeeld 7 a = (15,67; 0,999; 0,022), S(a) = 0,006

Voorbeeld 8

Funksie: $F = a_1 + e^{a_2/(x+a_3)}$

DATA	
x	y
50	34,780
55	28,610
60	23,650
65	19,630
70	16,370
75	13,720
80	11,540
85	9,744
90	8,261
95	7,030
100	6,005
105	5,147
110	4,427
115	3,820
120	3,307
125	2,872

Tabel 5.1.4 Data vir voorbeeld 8

Beginwaardes: a^o = (0,02; 4000; 250), $S(a^o)$ = 1,E12

Eindwaardes: a = (-5,8; 374,7; 51), S(a) = 0,4066

In hierdie voorbeeld vind Meyer en Roth S(a) = 88 vir a=(0,0056; 6181,4; 345,2), terwyl Nash vind dat S(a) = 20137 vir a=(-735,732; 2786,66; 215,92). Die resultate van PASPLT oortref bogenoemde ver aangesien S(a) = 0,4066 vir a = (-5,8; 374,7; 51)

```
  VOORBEELD 8

= 1.0000000D-06      FSC =   1.000000
        250.00000

1.0000000D-06        FNN          0.1134
       5.88232526D-02
        224.86732

5.0000000D-07        FNN          0.1109
       6.26299037D-02
        199.22000

2.5000000D-07        FNN          0.1085
       6.64433031D-02
        176.64327

1.2500000D-07        FNN          0.1103
       6.39989710D-02
        154.48073

6.2500000D-08        FNN          0.1078
       7.25182279D-02
        132.35270

3.1250000D-08        FNN          0.1031
       8.13866402D-02
        110.48587

1.5625000D-08        FNN          9.4328D-02
       9.05111128D-02
        89.492882

7.8125000D-09        FNN          7.5696D-02
       0.10024095
        70.889227

3.9062500D-09        FNN          3.9765D-02
       0.10992872
        57.557256

1.9531250D-09        FNN          6.2461D-03
       0.11681837
        51.818916
9.7656250D-10        FNN          0.1214
       0.11841407
        51.017065
```

5.2 Vergelyking van Resultate

In tabel 5.2.1 word die resultate van [20] en [21] saam met ons resultate vir PASPLT getabuleer. In die tabel word die aantal kere wat die Jacobiaan en die aantal kere wat die som van die kwadrate van die verskille bereken word aangedui. In al die voorbeelde is die afgeleides analities bereken.

Voorbeeld	NASH		FLETCHER		DLS	MDLS	PLASPLT					
	$\beta = 4$ α=1.E-4	$\beta = 4$ $\alpha = 1$	Skaal elke Iterasie	Skaal eenmaal	λ_o=1.E-3	λ_o=1.E-3	λ_o=1.D-1	λ_o=1.D-2	λ_o=1.D-3	λ_o=1.D-4	λ_o=1.D-5	λ_o=1.D-6
1	15,22	26,39	9,15	10,16	19,29	4,4	8,12	23,44	8,15	7,15	6,17	6,18
2	25,29	32,32	23,30	15,18	38,61	17,32	23,30	17,23	6,7	4,5	3,4	2,3
3	22,25	21,25	21,26	15,19	29,46	16,29	27,36	19,24	20,29	4,5	3,4	3,4
4	6,9	13,21	18,19	64,65	-	10,25	12,12	8,8	20,23	17,21	13,17	-
5	7,11	15,22	14,18	52,55	-	14,46	9,9	6,6	5,5	5,5	5,5	5,5
6	66,89	69,97	58,82	36,39	-	24,40	45,63	47,67	53,76	51,77	51,75	52,78
7	121,170	67,95	60,87	17,40	-	22,35	42,59	44,63	39,56	41,63	51,76	51,79
8	13,23 *	15,25 *	8,14 *	10,21 *	-	7,12 *	22,27	28,37	13,13	12,12	12,12	12,12

* - Verkeerde optimum

Tabel 5.2.1 Minimum aantal iterasies en aantal funksieberekeninge deur verskillende metodes vir voorgaande voorbeelde

-81-

Uit die tabel 5.1 blyk dit dat PASPLT in baie gevalle beter resultate (d.w.s. minder iterasies en minder funksie-berekeninge as die ander algoritmes lewer. Dit blyk ook dat λ_o = 1.D-5 'n goeie beginwaarde vir al die voorbeelde van λ is.

5.3 Samevatting

Uit voorgaande is dit duidelik dat PASPLT baie goed vergelyk met programme wat op ander algoritmes gegrond is.

Funksies met meer as 1000 eksperimentele datastelle en 25 koëffisiënte is suksesvol gepas, terwyl konvergensie baie vinniger verkry is, as met die program PASF wat ons tot dusver gebruik het.

Die program PASPLT voldoen dus aan al die vereistes wat in paragraaf 1.1 gestel is.

8. OPSOMMING

TITEL : 'n Algoritme vir kleinstekwadrate-benadering

LEIER : Prof. N Sauer

DEPARTEMENT : Toegepaste Wiskunde

GRAAD : M.Sc. (Toegepaste Wiskunde)

Die doel van hierdie verhandeling is om 'n algoritme vir kleinstekwadrate-benaderings te vind en om met behulp van die algoritme 'n rekenaarprogram te ontwikkel vir die minimering van die kleinstekwadrate-benadering en die passings van krommes deur gegewe eksperimentele data.

Eerstens word die norm en die pseudo-inverse van matrikse sowel as Householder-transformasies bespreek, aangesien ons daarvan gebruik maak in die oplossing van ons probleem.

In die lineêre kleinstekwadrate-probleem word die koëffisiënte matriks A ontbind in 'n produk van 'n ortogonale en 'n bo-driehoeksmatriks. Hierdie produk word dan gebruik om die pseudo-inverse van A te verkry.

Die algoritme wat ontwikkel is vir die nie-lineêre kleinstekwadrate-probleem is gebaseer op die Gauss-Newton-Marquardt-metode, waar met elke stap 'n lineêre kleinstekwadrate-probleem opgelos word. Resultate wat met behulp van die algoritme verkry is, word ook vergelyk met die resultate van ander algoritmes.

9. SYNOPSIS

TITLE : An algorithm for least squares approximation

LEADER : Prof N Sauer

DEPARTMENT : Applied Mathematics

DEGREE : M.Sc. (Applied Mathematics)

The purpose of this thesis is to obtain an algorithm for a least squares approximation and to develop with the aid of this algorithm a computer programme which minimises a sum of the squares and to fit a curve through experimental data.

Firstly the norm and the pseudo-inverse of matrices as well as the Householder-transformations are discussed since these concepts are utilised to solve the problem.

In the linear least squares problem the coefficient matrix A is written as a product of an orthogonal and toptriangularised matrix. This product is then used to find the pseudo-inverse of A.

The algorithm developed for non-linear least squares problems is based on the Gauss-Newton-Marquardt method, where in each step a linear least squares problem is solved.

Results obtained with the aid of this algorithm are then also compared with the results obtained from other algorithms.

10. BYLAE A

In hierdie bylaag verskyn die gedokumenteerde program op bladsy A1, sowel as die verskillende FORTRAN-subroetines wat in die program gebruik word.

Die betekenis van die verskillende parameters wat in die naamlys voorkom word op p.A2 gegee en op p.A5 verskyn 'n lys van die keuseparameters met hulle onderskeie betekenisse.

Die volgende lyste van FORTRAN-subroetines verskyn in bylaag A: DATTYD, DIMEN, DIME, PASDIM, DATA, BEGIN, MINLSQ, DIFF, FIT, CORREL, DATUIT, CONST, PLOTT, ASSE, LYN.

```
C   DOEL
C   ----
C
C   DIE PROGRAM BEREKEN DIE KOEFFISIENTE VAN 'N LINEERE OF NIE-LINEERE
C   MEERVOUDIGE REGRESSIEVERGELYKING,IN DIE VORM  Y=F(X,A), WAAR X DIE
C   VEKTOR MET DIE ONAFHANKLIKE VERANDERLIKES EN A DIE KOEFFISIENTE
C   VEKTOR VOORSTEL.
C
C   METODE
C   ------
C
C   DIE METODE WAT GEBRUIK WORD IS DIE GAUSS-NEWTON-MARQUARDT METODE, MET
C   TOEPASSING VAN HOUSEHOLDER TRANSFORMASIES, DIE REGRESSIELYN WORD
C   GESKEI IN 'N LINEERE EN NIE-LINEERE GEDEELTE.
C
C   GEBRUIK
C   -------
C
C   UIT GEGEWE DATA X(I,J) WORD DIE ONAFHANKLIKE SOWEL AS DIE AFHANKLIKE
C   VERANDERLIKES, TUSSEN GEGEWE GRENSE GX UITGESOEK, WAAR
C   GX(1,J) DIE ONDERSTE GRENS EN GX(2,J) DIE BOONSTE GRENS VOORSTEL.
C   AAN DIE EINDE VAN DIE DATA X(I,J) MOET 'N KAART GEPLAAS WORD MET 'N
C   '*' IN KOLOM 1,DIE RES VAN DIE KAART KAN GEBRUIK WORD VIR KOMMENTAAR.
C   DIE FUNKSIE Y=F(X,A) WORD IN DIE SUBROETINE PASDIM MET INGANGSPUNT
C   'ENTRY PASFUN(X,A,FPAS)' DEUR DIE GEBRUIKER GEDEFINIEER.
C   PASDIM WORD DEUR DIE GEBRUIKER VOORSIEN.
C   DIE KOEFFISIENTE VAN DIE PASSING WORD IN DIE VEKTOR (A(I),I=1,NL)
C   GESTOOR, EN DEUR IV(J) = 1 TE STEL, KAN DIE OOREENSTEMMENDE A(J)
C   GEDURENDE DIE MINIMERINGSPROSES KONSTANT GEHOU WORD, DIT IS OOK
C   MOONTLIK OM DIE A'S TUSSEN GEGEWE GRENSE TE BEGRENS.
C   PROGRAMPARAMETERS WORD DEUR MIDDEL VAN 'N 'NAMELIST' INGELEES, BAIE
C   VAN HIERDIE PARAMETERS IS OPSIONEEL EN DUS NIE NODIG OM IN TE LEES
C   NIE.
C   DIE PROGRAM BESIT DIE FASILITEIT OM DIE GEPASTE KROMME TE TEKEN, DEUR
C   VERSKILLENDE VERANDERLIKES KONSTANT TE HOU.
C   DIE DIMENSIES VAN DIE PROGRAM WORD IN DIE SUBROETINE PASDIM GEDEFINI-
C   EER EN DIT IS MOONTLIK OM NOG VERANDERLIKES IN DIE SUBROETINE PASDIM
C   ONDER INGANGSPUNT PASGEN TE GENEREER.
C   'N KORRELASIETABEL KAN UITGEDRUK WORD, WAT DIE KORRELASIE TUSSEN DIE
C   VERSKILLENDE VERANDERLIKES WEERGEE.
C   INDIEN DIE FUNKSIE LINEER IN SY A-WAARDES IS, KAN DIE OPLOSSING
C   REGSTREEKS VERKRY WORD DEUR IT = 0 TE STEL.
C
C   DIE PROGRAM IS IN DUBBELNOUKEURIGHEID VIR 'N IBM 370/155  EN
C   'N CALCOMP 936 MODEL STIPPER, GESKRYF.
C
C   BELANGRIKE UITSETPARAMETERS.
C
C   IER      = 'N UITSETPARAMETER MET DIE VOLGENDE BETEKENIS.
C              IER   = 0   GEEN FOUT, NORMALE KONVERGENSIE.
C              IER   = 1   GEEN KONVERGENSIE NA IT ITERASIES.
C              IER   = 2   AANTAL DATAPUNTE IS MINDER AS KOEFFISIENTE.
C              IER   = 3   GEEN KONVERGENSIE KAN VERKRY WORD.
C              IER   = 4   RANG VAN MATRIKS MET AFGELEIDES IS KLEINER AS
C                          AANTAL KOEFFISIENTE.
C              IER   = 5   TE MIN TYD TOEGELAAT.
C
C   IT       = AANTAL ITERASIES GEDOEN
```

```
C
C   BESKRYWING VAN PARAMETERS WAT DEUR MIDDEL VAN 'N NAAMLYS INGELEES
C   KAN WORD.
C
C   PASSINGPARAMETERS.
C   ------------------
C
C   FORM       = 'N 9 DIMENSIONELE *8-VEKTOR WAT DIE FORMAATSTELLING BEVAT
C                WAARMEE DIE X'E INGELEES WORD.
C
C   NGO        = DATASTELVERWYSINGSNOMMER WAARVAN GELEES MOET WORD.
C                NGO = 5 INDIEN VAN KAARTE GELEES WORD.
C                NGO = 8 INDIEN VAN SKYF GELEES WORD.
C
C   M          = AANTAL VERANDERLIKES WAT INGELEES WORD.
C
C   NN         = AANTAL VERANDERLIKES WAT IN DIE FUNKSIE VOORKOM,
C                (AFHANKLIKE + ONAFHANKLIKE).
C
C   IX         = VEKTOR MET DIMENSIE 50, EN DUI DIE ONAFHANKLIKE VERANDERLI-
C                KES AAN WAT GENEEM MOET WORD, TERWYL IX(NN) DIE AFHANKLI-
C                KE VERANDERLIKE VOORSTEL.
C                INDIEN GRAFIEK GETREK WORD SAL IX(NN-1) SE VERANDERLIKE VIR
C                DIE X-AS GEBRUIK WORD.
C
C   KX         = AANTAL X'E WAAROP GRENSE IS.
C
C   KVX        = VEKTOR MET DIMENSIE 50, EN DUI AAN WATTER X-VERANDERLI-
C                KES BEGRENS IS.
C
C   GX         = 'N 2*50 MATRIKS WAT DIE GRENSE OP DIE GESPESIFISEERDE
C                KVX(J),J=1,KX BEVAT.
C                GX(1,J) = ONDERSTE GRENS MET J=1,KX
C                GX(2,J) = BOONSTE GRENS MET J=1,KX.
C
C   LEESX      = INLEES-PARAMETER MET DIE VOLGENDE BETEKENIS.
C                LEESX = 0 GEEN X-WAARDES WORD INGELEES.
C                LEESX = 1 X-WAARDES WORD WEER VAN SKYF GELEES.
C                LEESX = 2 NUWE X-WAARDES WORD VAN KAARTE GELEES.
C
C   A          = 'N 50 DIMENSIONELE VEKTOR WAT DIE AANVANGSWAARDES VAN A(I)
C                BEVAT, EN WORD GELYK AAN NUL GESTEL INDIEN NIE INGELEES
C                NIE. BY TERUGKEER BEVAT DIE A-VEKTOR DIE KOEFFISIENTE VAN
C                DIE PASSING.
C
C   KA         = DIE AANTAL A'S WAAROP GRENSE IS.
C
C   KVA        = 'N 50 DIMENSIONELE VEKTOR WAT AANDUI WATTER A'S BEGRENS IS.
C
C   GA         = 'N 2*50 MATRIKS WAT DIE GRENSE OP DIE GESPESIFISEERDE
C                KVA(J),J=1,KA BEVAT.
C                GA(1,J) = ONDERSTE GRENS MET J=1,KA
C                GA(2,J) = BOONSTE GRENS MET J=1,KA.
```

```
C
C  IV        = 'N 50 DIMENSIONELE VEKTOR WAT AANDUI WATTER A-WAARDES GEDU-
C              RENDE MINIMERING KONSTANT GEHOU MOET WORD. IV(J) MOET GELYK
C              AAN 1 GESTEL WORD INDIEN A(J) KONSTANT GEHOU MOET WORD.
C
C  NL        = AANTAL A'S WAT IN FUNKSIE GEDEFINIEER IN PASDIM GEBRUIK
C              WORD.
C              INDIEN NL=0, DAN SAL DIE REGRESSIELYN VAN IX(NN) OP DIE
C              REGRESSIEVLAK VAN IX(NN-1)....IX(1) BEREKEN WORD, MET NN
C              KOEFFISIENTE, EN DIT IS DUS NIE NODIG OM DIE FUNKSIE
C              IN PASFUN TE DEFINIEER NIE.
C
C  IFIT      = PARAMETER MET DIE VOLGENDE BETEKENIS.
C              IFIT = 0 GEEN PASSING WORD GEDOEN NIE.
C              IFIT = 1 FUNKSIE WORD GEPAS.
C
C  IPONS     = PARAMETER MET DIE VOLGENDE BETEKENIS.
C              IPONS = 0 A-WAARDES WORD NIE GEPONS AAN EINDE VAN PROGRAM
C                        NIE.
C              IPONS = 1 A-WAARDES WORD GEPONS AAN EINDE VAN PROGRAM.
C
C  ICOR      = PARAMETER MET DIE VOLGENDE BETEKENIS.
C              ICOR  = 1 KORRELASIEKOEFFISIENTE WORD BEREKEN EN UITGEDRUK.
C              ICOR  = 0 KORRELASIEKOEFFISIENTE WORD NIE BEREKEN NIE.
C
C  IT        = DIE TOTALE AANTAL ITERASIES WAT GEDOEN MOET WORD.
C              INDIEN IT=0,  WORD FUNKSIE BESKOU AS LINEER IN SY A-WAARDES
C              EN DIE NORMAALVERGELYKINGS WORD REGSTREEKS OPGELOS.
C
C  ITSIK     = DIE AANTAL ITERASIES TUSSEN OPEENVOLGENDE HERSKALERINGS.
C
C  EPS       = KONVERGENSIENORM.
C
C  NSK       = PARAMETER MET DIE VOLGENDE BETEKENIS.
C              NSK = 1 SKALERING VAN A-WAARDES WORD GEDOEN.
C              NSK = 0 GEEN SKALERING WORD GEDOEN NIE.
C
C  ICON      = PARAMETER MET DIE VOLGENDE BETEKENIS.
C              ICON = 0 'N ABSOLUTE KONVERGENSIETOETS WORD GEBRUIK.
C              ICON = 1 'N RELATIEWE KONVERGENSIETOETS WORD GEBRUIK.
C
C  TYD       = SODRA REKENAARTYD NOG BESKIKBAAR KLEINER AS TYD IS, SAL DIE
C              PROGRAMRESULTATE UITDRUK EN EINDIG.
C              TYD IN SEKONDES.
C
C  NPAS      = VERWYSINGSPARAMETER NA DIE FUNKSIE WAT GEPAS MOET WORD.
C              INDIEN VERSKILLENDE FUNKSIES MET DIESELFDE PROGRAM GEPAS
C              WORD.
C
C  NUM       = PARAMETER WAT BEPAAL OF AFGELEIDES NUMERIES OF ANALITIES
C              BEREKEN MOET WORD.
C              NUM = 0 AFGELEIDES MOET IN PASAF GEGEE WORD.
C              NUM = 1 AFGELEIDES WORD NUMERIES BEREKEN DEUR GEBRUIK TE
C                      MAAK VAN 'N EERSTE ORDE VERSKILVERGELYKING.
C              NUM = 2 AFGELEIDES WORD NUMERIES BEREKEN DEUR GEBRUIK TE
C                      MAAK VAN 'N TWEEDE ORDE VERSKILVERGELYKING.
C
C  FNU       = DEMPINGSFAKTOR.
```

```
C  UITDRUKPARAMETERS.
C  -----------------
C
C  IDRUK     = 1 ALLE RESULTATE WORD UITGEDRUK : DATA X(I,J), KORR.MATRIKS
C                UITGESOEKTE PUNTE VIR DIE STIPPER, VERSKILLE, ENS.
C            = 0 ALLEEN INVOERDATA, EN BELANGRIKE RESULTATE WORD UITGEDRUK
C            =-1 ALLEEN INVOERDATA WORD UITGEDRUK.
C            <-1 GEEN UITDRUK.
C  IPR       =   BY ELKE IPR-DE ITERASIE SAL RESULTATE UITGEDRUK WORD.
C  IDAT      = 1 DATA SOOS OP KAARTE GEPONS WORD UITGEDRUK.
C            = 0 DATA SOOS OP KAARTE GEPONS WORD NIE UITGEDRUK NIE.
C  OPSK      =   'N 50 DIMENSIONELE *8-VEKTOR WAT DIE OPSKRIFTE VAN DIE
C                INGELEESDE SOWEL AS DIE GEGENEREERDE X'E BEVAT.
C  PASF      =   'N 10 DIMENSIONELE *8-VEKTOR WAT 'N BOODSKAP BEVAT OM
C                UITGEDRUK TE WORD, EN SAL AS OPSKRIF BO GRAFIEK GEBRUIK
C                WORD.
C
C  STIPPARAMETERS.
C  --------------
C
C  IPLOT     = 0  GEEN GRAFIEK SAL GETEKEN WORD NIE.
C            = 1  GRAFIEK SAL GETEKEN WORD.
C  IAS       = 1  ASSESTELSEL SAL GETEKEN WORD.
C            = 0  GEEN ASSESTELSEL SAL GETEKEN WORD NIE.
C  ILYN      = 1  DIE PASSINGSKROMME WORD GETEKEN.
C            = 0  GEEN LYN WORD GETEKEN NIE.
C  IPEN      =    BEPAAL DIE KLEUR VAN DIE PEN WAT GEBRUIK MOET WORD EN
C                 KAN DIE WAARDES 1(BLOU), 2(ROOI) EN 3(GROEN) AANNEEM.
C  IPUNT     = 1  DIE UITGESOEKTE PUNTE WORD GESTIP.
C            = 0  GEEN PUNTE WORD GESTIP NIE.
C  IDENT     > 0  DIE WAARDE VAN DIE X(I,IDENT) WORD BY DIE GESTIPTE PUNTE
C                 GESKRYF.
C            < 1  GEEN WAARDE WORD BY GESTIPTE PUNTE GESKRYF NIE.
C  IDES      =    DIE AANTAL DESIMALE NA DIE PUNT VAN DIE UITGESOEKTE
C                 X-WAARDES WAT GESKRYF WORD BY DIE GESTIPTE PUNT.
C  YDUIM     =    BESKIKBARE BREEDTE VAN GRAFIEKPAPIER.
C  XLEN      =    LENGTE X-AS IN DUIME.
C  YLEN      =    LENGTE Y-AS IN DUIME.
C  XBEG      =    BEGINWAARDE X-AS.
C  XEND      =    EINDWAARDE X-AS.
C  YBEG      =    BEGINWAARDE Y-AS.
C  YEND      =    EINDWAARDE  Y-AS.
C  NSOEK     =    DIE AANTAL X-VERANDERLIKES WAT KONSTANT GEHOU MOET WORD
C                 IN DIE FUNKSIE, OM GRAFIEKE TE TEKEN.
C  XP        =    'N 50 DIMENSIONELE *4-VEKTOR, WAT DIE KONSTANTE WAARDES
C                 BEVAT WAT GEBRUIK MOET WORD IN DIE PASSING OM GRAFIEKE
C                 TE TEKEN.
C  TOLX      =    'N 50 DIMENSIONELE *4-VEKTOR, WAT DIE TOLERANSIES BEVAT
C                 VIR DIE X-WAARDES, ALLE NSOEK X-WAARDES IN DIE GEBIED
C                 XP-TOLX EN XP+TOLX SAL GESTIP WORD.
C  NDIGD     =    DIE LENGTE VAN X-AS WORD VERDEEL IN NDIGD DELE, EN BY
C                 ELKE PUNT WORD FUNKSIEWAARDE BEREKEN OM LYN TE TEKEN.
C  IDXP      =    VEKTOR WAT DIE AANTAL DESIMALE NA DIE DESIMALE PUNT
C                 WAT GESTIP MOET WORD, AANDUI.
C  FAKT      =    GRAFIEKVERGROTINGSFAKTOR. (FAKT=2.54 MATE IN DUIME)
C  IDG       =    VEKTOR WAT DIE AANTAL DESIMALE VAN DIE GRENSE WAT
C                 GESTIP MOET WORD AANDUI.
```

```
C  PARAMETERS WAT OPSIONEEL IS EN WAARDES TOEGEKEN INDIEN NIE IN NAAMLYS
C  GESPESIFISEER.
C
C    ************************************************
C    *   PARAMETER    *     **     * WAARDE TOEGEKEN *
C    ************************************************
C    *   A(50)        *     **     *    0.           *
C    *   EPS          *     **     *    1.D-6        *
C    *   FAKT         *     **     *    2.54         *
C    *   FNU          *     **     *    1.D-4        *
C    *   GA(2,50)     *     **     * -1.E20,1.E20    *
C    *   GX(2,50)     *     **     * -1.E20,1.E20    *
C    *   IAS          *     **     *    0            *
C    *   ICON         *     **     *    1            *
C    *   ICOR         *     **     *    1            *
C    *   IDAT         *     **     *    0            *
C    *   IDENT        *     **     *   -1            *
C    *   IDES         *     **     *    1            *
C    *   IDG(50)      *     **     *    2            *
C    *   IDXP(50)     *     **     *    2            *
C    *   IDRUK        *     **     *    0            *
C    *   IFIT         *     **     *    1            *
C    *   ILYN         *     **     *    1            *
C    *   IPEN         *     **     *    1            *
C    *   IPLOT        *     **     *    0            *
C    *   IPONS        *     **     *    0            *
C    *   IPR          *     **     *    1            *
C    *   IPUNT        *     **     *    0            *
C    *   IT           *     **     *    0            *
C    *   ITSIK        *     **     *    0            *
C    *   IV(50)       *     **     *    0            *
C    *   IX(50)       *     **     *    1,2,3,4,...  *
C    *   FORM(9)      *     **     * BLANK           *
C    *   KA           *     **     *    0            *
C    *   KX           *     **     *    0            *
C    *   LEESX        *     **     *    1            *
C    *   M            *     **     *    0            *
C    *   NDIGD        *     **     *    100          *
C    *   NGO          *     **     *    5            $
C    *   NL           *     **     *    0            *
C    *   NN           *     **     *    2            *
C    *   NPAS         *     **     *    1            *
C    *   NSK          *     **     *    1            *
C    *   NSOEK        *     **     *    0            *
C    *   NUM          *     **     *    2            *
C    *   OPSK(50)     *     **     * BLANK           *
C    *   PASF(10)     *     **     * BLANK           *
C    *   TOLX(50)     *     **     *    0.05         *
C    *   TYD          *     **     *    10.          *
C    *   XLEN         *     **     *    10.          *
C    *   XBEG         *     **     *    0.           *
C    *   XEND         *     **     *    10.          *
C    *   YDUIM        *     **     *    11.          *
C    *   YLEN         *     **     *    8.           *
C    *   YBEG         *     **     *    0.           *
C    *   YEND         *     **     *    8.           *
C    ************************************************
```

```
C***   HOOFPROGRAM
C
C   DATUM EN TYD WORD BEREKEN
C
      CALL DATTYD
C
C   DIMENSIES WORD BEREKEN
C
      CALL DIMEN
      CALL EXIT
      END

C
      SUBROUTINE DATTYD
      DOUBLE PRECISION DAT
      DATA N/0/
C
C   DATUM WORD VERKRY UIT SUBROETINE DATE
C
      CALL DATE (DAT)
C
C   TYD WORD VERKRY UIT SUBROETINE HTYD
C
      CALL HTYD(I)
C
C   PROSESSEERTYD WORD VERKRY UIT SUBROETINE CPUTIM
C
      CALL CPUTIM(T)
      I1=I/360000
      I2=(I-I1*360000)/6000
      I3=(I-I1*360000-I2*6000)/100
C
C   INDIEN N=0 WORD DATUM EN TYD UITGEDRUK
C
      IF(N.EQ.0) KZ=I
      I=I-KZ
1     FORMAT(1X,'DATUM',1X,A8,100X,'TYD',1X,I3,'-',I2,'-',I2,/)
      IF(N.EQ.0) WRITE(6,1)DAT,I1,I2,I3
      T1=FLOAT(I)/100.
C
C   INDIEN N=1 WORD TOTALE TYD EN PROSESSEERTYD UITGEDRUK
C
      IF(N.NE.0)WRITE(6,2)T1,T
2     FORMAT(//,T38,'    TOTALE TYD IN REKENAAR =',F8.2,' SEK.',/,
     1T38,' WERKLIKE TYD IN REKENAAR =',F8.2,' SEK.',//)
      N=N+1
      RETURN
       ENTRY CPUTYD(J)
      CALL CPUTIM(T)
      GO TO (3,4),J
3     T1=T
      RETURN
4     T2=T-T1
      WRITE(6,10)T2
10    FORMAT(/,1X,'MINIMERINGSTYD IS',F8.2,' SEK.')
      RETURN
      END
```

```
C
      SUBROUTINE DIMEN
C
C   SUBROETINE BEPAAL DIE DIMENSIES VAN DIE PROGRAM
C
      IMPLICIT REAL*8(A-H,O-Z)
      EXTERNAL DIME
C
C   N1,N2 EN N3 WORD IN PASDIM GEDEFINIEER
C
      COMMON/DIM/N1,N2,N3
      COMMON/DIM2/N4
      DIMENSION COR(1,1),X(1,1),X1(1,1)
C
C   N1   =  MAKSIMUM AANTAL RYE VAN MATRIKS X
C   N2   =  MAKSIMUM AANTAL KOLOMME IN MATRIKS X
C   N3   =  MAKSIMUM AANTAL KOEFFISIENTE WAT IN PASSING GEBRUIK WORD
C
      CALL PASDIM
      N4=MAX0(N2,N3)
C   ELKE PARAMETER WORD MET TWEE VERMEERDER EN RUIMTE WORD GEBRUIK VIR
C   WERKAREA.
      N1=N1+2
      N2=N2+2
      N4=N4+2
      N3=N3+2
      NA=N1*N2*2
      NC=N1*N3*2
      NB=N4*N4*2
C
C   SUBROETINE DMENSI KEN DIMENSIES TOE AAN DIE VERSKILLENDE MATRIKSE
C
      CALL DMENSI(DIME,X,NA,X1,NC,COR,NB)
      RETURN
      END



      SUBROUTINE DIME(X,X1,COR)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 XBEG,XEND,YBEG,YEND,XLEN,YLEN,XP,TOLX,GA,GX,TYD
      COMMON/BLOS/ EPS,    PASF(10),   TYD
      COMMON/DIM/N1,N2,N3
      COMMON/DIM2/N4
      COMMON/BLOK/GX(2,52),OPSK(52),IX(52),KVX(52),KVA(52),GA(2,52)
      COMMON/INTR/IP(50), IV(50),  N, KA, KX, MM, NL, IT, ITSIK, IPR,
     1 ICON, NSK, ICOR, IER,  NN,   NLL,   IPONS
      COMMON/IPLT/NSOEK, IXAS,  IYAS, IPUNT, ILYN, IAS, NDIGD,   LONG,
     1          IDENT, IDES,  LSIM,  NV,  IPLOT, IFIT, IPEN
      COMMON/APLT/ XBEG, XEND,  YBEG,  YEND, XLEN,YLEN,XP(50),TOLX(50),
     1        IDXP(50),IDG(50)
      COMMON/BLAB/A(50),B(50)/DGIN/ M
      DIMENSION COR(N4,1),X(N1,1),X1(N1,1)
```

```
C
C   SUBROETINE BEGIN LEES BEGINPARAMETERS EN DATAPUNTE IN, EN VERSKILLEND
C   PARAMETERS WORD UITGEDRUK
C
10      CALL BEGIN(X,N1,N2,N3)
        NLL=NL
        IF(NL.NE.0)GO TO 20
        NLL=NN
        IT=0
20      DO 30 I=1,NLL
30       B(I)=A(I)
C
C   VIR IFIT = 0  GEEN PASSING WORD GEDOEN
C
        IF(IFIT.EQ.0)GO TO 65
C   INDIEN IT=0, BETEKEN FUNKSIE IS LINEER.
        IF(IT.EQ.0)GO TO 50
C
C   VIR KA = 0  GEEN GRENSE OP A'S
C
        IF(LONG.GE.0)CALL CPUTYD(1)
        IF(KA.EQ.0) GO TO 60
        DO 40 K=1,KA
        O=GA(1,K)
        D=GA(2,K)
40       B(KVA(K))=DARSIN(DSQRT(( B(KVA(K))-O)/(D-O)))
        GO TO 60
50      KA=0
C
C   OPTIMISASIE-PROGRAM WORD GEROEP
C
60      CALL MINLSQ(NLL,N,A, B,IV,IT,ICON,IPR,NSK,X(1,N2-1),X,X1,COR,
       1IP,EPS,ITSIK,IER,N1,N4,N3)
        IF(LONG.GE.0)CALL CPUTYD(2)
C
C   CORREL BEREKEN KORRELASIEKOEFFISIENTE, GEMIDDELDES EN STANDAARDAFWYKI
C
65      IF(N.LE.1.OR.M.EQ.0)GO TO 80
        CALL CORREL(N,MM,X,COR,B,ICOR,N1,N2,N4)
C
C   DATUIT DRUK DATAPUNTE INGELEES UIT
C
        CALL DATUIT(X,COR,A,N,MM,LONG,ICOR,N1,N2,N4)
C
C   DIFF BEREKEN VERSKILLE TUSSEN DIE AFHANKLIKE EN GEPASTE WAARDE
C
        IF(IFIT.EQ.0)GO TO 80
70      CALL DIFF(A,B,X,OPSK,IX,N,LONG,N1,N2)
C
C   INDIEN IPLOT = 1 EN IER = 0 WORD PLOTPROGRAM GEROEP
C
80      IF(IPLOT.EQ.1.AND.IER.EQ.0) CALL PLOTT(A,X,B,N1)
        IF(M.EQ.0)WRITE(6,85)IER
85      FORMAT(1X,'IER =',I3)
        GO TO 10
90      RETURN
        END
```

```
      SUBROUTINE PASDIM
      IMPLICIT REAL*8(A-H,O-Z)
C
C   IN SUBROETINE PASDIM WORD WAARDES AAN DIE PARAMETERS IR,IK,IA GEGEE.
C
      DIMENSION X(1),A(1)
C
C   BESKRYWING PARAMETERS
C
C   IR    = MAKSIMUM AANTAL DATAPUNTE WAT IN DIE PASSINGS GEBRUIK WORD,
C           EN MOET GROTER AS NUL WEES.
C   IK    = TOTALE AANTAL VERANDERLIKES, INGELEES PLUS GEGENEREER,
C           MET  0 < IK < 50 .
C   IA    = AANTAL A-WAARDES WAT IN FUNKSIE PLUS A-WAARDES WAT LINEER
C           VOORKOM, MET  0 < IA < 50 .
C
      COMMON/DIM/IR,IK,IA
      IR=10
      IK=10
      IA=10
      RETURN
C
C   IN PASGEN KAN NOG VERANDERLIKES GEGENEREER OF VERANDER WORD
C
      ENTRY PASGEN(X,M)
C
C   GENEREER OF VERANDER NOG AFHANKLIKES
C
C     X(M+1)=X(4)*X(5)
      RETURN
C
C   IN PASAF WORD DIE AFGELEIDES BEREKEN AS NUM=0
C
      ENTRY PASAF(X,A,F)
      DIMENSION F(1)
      RETURN
C
C   IN PASFUN WORD DIE FUNKSIE GEDEFINIEER
C
      ENTRY PASFUN(X,A,FPAS)
C
C   BEREKEN DIE WAARDE VAN DIE PASSINGSFUNKSIE BY DIE I-DE DATAPUNT
C
      FPAS=A(1)+A(2)*X(2)+A(3)*X(2)*X(2)+DEXP(X(2)*A(4))+A(5)*X(1)
      RETURN
      END
```

```
      SUBROUTINE DATA(NGO,IDAT)
C
C  SUBROETINE OM DATA IN TE LEES EN ONDER DATASTELVERWYSINGSNOMMER 8
C  OP SKYF TE SKRYF.
C
      DIMENSION A(80)
      DATA STER/1H*/
C
C  IDAT = 0 DATA WORD VAN KAARTE GELEES EN OP SKYF GESKRYF
C  IDAT = 1 DATA WORD VAN SKYF GELEES EN UITGEDRUK
C
      IF(IDAT.EQ.1)GO TO 30
      NDISC=8
      I=0
5     READ(5,10,END=20)A
      IF(A(1).EQ.STER)GO TO 20
      I=I+1
C
C  DATA WORD OP SKYF GESKRYF.
C
      WRITE(NDISC,10)A
10    FORMAT(80A1)
      GO TO 5
20    CONTINUE
      NGO=NDISC
      IF(I.EQ.0)RETURN
      REWIND NDISC
      RETURN
30    I=0
      WRITE(6,50)
50    FORMAT(//,1X,'DATALYS SOOS OP KAARTE GEPONS',/)
55    READ(NDISC,10,END=100)A
      I=I+1
      WRITE(6,101)I,A
101   FORMAT(1X,I10,8X,80A1)
      GO TO 55
100   CONTINUE
      REWIND NDISC
      RETURN
      END
```

```
      SUBROUTINE BEGIN(X,N1,N2,N3)
C
C  SUBROETINE BEGIN LEES BEGINWAARDES IN, OPSIONELE PARAMETERS WORD GE-
C  DEFINIEER, EN UITGEDRUK
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 XBEG,XEND,YBEG,YEND,XLEN,YLEN,XP,TOLX,GA,GX,TYD,YDUIM,FAKT
      NAMELIST/LEES/FORM,M,NN,IDRUK,IX,NGO,IDAT,GX,KVX,KX,LEESX,OPSK,GA,
     1EPS,PASF,A,NSK,IT,IPR,IV,ICON,ITSIK,NL,KA,KVA,IPLOT,YDUIM,FNU,N,
     2 XBEG,XEND,YBEG,YEND,TOLX,IDENT,IDES,NDIGD,XLEN,YLEN,IPUNT,FAKT,
     3ILYN,NSOEK,XP,IAS,IDXP,IDG,IFIT,IPONS,TYD,NPAS,ICOR,IPEN,NUM,MM
      DIMENSION X(N1,1),FORM(10)
      COMMON/IPLT/NSOEK, IXAS,  IYAS, IPUNT, ILYN, IAS, NDIGD,  IDRUK,
     1             IDENT, IDES,  LSIM,  NV,  IPLOT, IFIT, IPEN
      COMMON/APLT/ XBEG, XEND,  YBEG,  YEND, XLEN,YLEN,XP(50),TOLX(50),
     1          IDXP(50),IDG(50)
      COMMON/INTR/IP(50), IV(50),  N, KA, KX, MM, NL,NIT, ITSIK, IPR,
     1 ICON, NSK, ICOR, IER,  NN,  NLL,  IPONS
      COMMON/BLOK/GX(2,52),OPSK(52),IX(52),KVX(52),KVA(52),GA(2,52)
      COMMON/BLAB/A(50),B(50)/DGIN/ M/FNUK/FNU/BGAS/YDUIM,FAKT
      COMMON/BLOS/ EPS, PASF(10),TYD/FNOM/NPAS/AFGE/NUM/XIJ/IJ,NX
      DATA BLANK/1H /,NTEL/0/
      IF(NTEL.GT.0)GO TO 20
C
C  WAARDES AAN OPSIONELE PARAMETERS WORD TOEGEKEN
C
1     NPAS=1
      M=0
      MM=2
      IPONS=0
      TYD=10.
      DO  2 I=1,50
      OPSK(I)=BLANK
      IV(I)=0
      IX(I)=I
      IDXP(I)=2
      IDG(I)=2
      GX(1,I)=-1.E20
      GA(1,I)=-1.E20
      GX(2,I)=1.E20
      GA(2,I)=1.E20
      A(I)=0.D0
2     TOLX(I)=.05
      IDENT=-1
      NSOEK=0
      NDIGD=100
      XLEN=10.
      YLEN=8.
      IPUNT=0
      ILYN=1
      IAS=1
      NN=2
      IPLOT=0
      ICOR=1
      IT=0
      IDES=1
      NUM=2
      ITSIK=0
```

```
      NSK=1
      IPR=1
      ICON=1
      IPEN=1
      IDAT=0
      KX=0
      KA=0
      IER=0
      IXAS=1
      IYAS=2
      NL=0
      IFIT=1
      IDRUK=0
      LEESX=1
      NGO=5
      NGO1=NGO
      EPS=1.D-6
      YDUIM=11.
      FNU=1.D-4
      FAKT=2.54
      DO 5 J=1,N2
      DO 5 I=1,N1
      X(I,J)=BLANK
      IF(J.LE.2)X(I,J)=0.D0
5     CONTINUE
      DO 10 I=1,10
      FORM(I)=BLANK
      PASF(I)=BLANK
10    CONTINUE
20    NTEL=NTEL+1
C
C  INDIEN NGO = 5 WORD DATA VAN KAARTE GELEES EN OP SKYF GESKRYF
C
      IF(IER.EQ.5)GO TO 390
      IF(NGO1.EQ.5)CALL DATA(NGO1,0)
C
C  NAMELIST WORD INGELEES
C
30    READ(5,LEES,END=390)
      IF(LEESX.EQ.2)CALL DATA(NGO1,0)
      IF(NGO.NE.5)NGO1=NGO
      IF(ITSIK.EQ.0)ITSIK=IT
      IF(IDRUK.LT.0)ICOR=0
      IF(NN.GT.0)IYAS=IX(NN)
      IF(NN.GT.1)IXAS=IX(NN-1)
      IX(52)=IX(NN)
      NIT=IT
40    IF(LEESX.EQ.0)GO TO 160
      DO 45 J=1,50
45    B(J)=BLANK
      I=1
C
C  DATA WORD VAN SKYF GELEES EN GESTOOR IN MATRYS X(I,J)
C
50    READ(NGO1,FORM,END=150)(X(I,J),J=1,M)
      IF(KX.EQ.0)GO TO 70
      DO 60 K3=1,KX
      IF(X(I,KVX(K3)).LT.GX(1,K3).OR.X(I,KVX(K3)).GT.GX(2,K3))
```

```
     1GO TO 50
60    CONTINUE
70    I=I+1
      DO 80 J=1,M
80    B(J)  =   X(I-1,J)
C
C  PASGEN GENEREER NOG VERANDERLIKES
C
      IJ=I-1
      CALL PASGEN(B,M)
      IF(I.NE.2)GO TO 110
C
C  OM TOTALE AANTAL VERANDERLIKES MM TE BEREKEN
      DO 90 J=1,50
      IF(B(J).EQ.BLANK)GO TO 100
      MM=J
90    CONTINUE
100   IF(MM.GT.N2-2)GO TO 405
110   DO 120 J=1,MM
120   X(I-1,J)=     B(J)
      IF(I.GE.N1-1)GO TO 130
      GO TO 50
130   NN1=N1-2
      WRITE(6,140)NN1
140   FORMAT(//,' SLEGS EERSTE  ',I3, ' GENEEM',//)
150   N=I-1
      REWIND NGO1
C
C  PARAMETERS WORD UITGEDRUK
C
160   IF(IDRUK.LT.-1)GO TO 370
      IF(NTEL.GT.1)WRITE(6,170)
170   FORMAT('1')
      WRITE(6,180)
180   FORMAT(50X,'** INVOER **',/)
      NN1=N1-2
      NN2=N2-2
      NN3=N3-2
      WRITE(6,190)NN3,IPR,IFIT,IDENT,ITSIK,MM,NPAS,NN1,ICON,ILYN,IDRUK,
     1KA,NL,NDIGD,NN2,ICOR,IPEN,IPLOT,KX,NN,NSOEK,IT,IDAT,IXAS,IPONS,
     2LEESX,NGO,NUM,IAS,IDES,IYAS,IPUNT,M,NSK,N
190   FORMAT(' IA     =',I5,6X,'IPR    =',I5,6X,'IFIT   =',I5,6X,'IDENT =',
     1  I5,6X,'ITSIK =',I5,6X,'MM     =',I5,6X,'NPAS   =',I5,
     2   /,1X,'IR     =',I5,6X,'ICON   =',I5,6X,'ILYN   =',I5,6X,'IDRUK =',
     3  I5,6X,'KA     =',I5,6X,'NL     =',I5,6X,'NDIGD =',I5,
     4   /,1X,'IK     =',I5,6X,'ICOR   =',I5,6X,'IPEN   =',I5,6X,'IPLOT =',
     5  I5,6X,'KX     =',I5,6X,'NN     =',I5,6X,'NSOEK =',I5,
     6   /,1X,'IT     =',I5,6X,'IDAT   =',I5,6X,'IXAS   =',I5,6X,'IPONS =',
     7  I5,6X,'LEESX =',I5,6X,'NGO    =',I5,6X,'NUM    =',I5,
     8   /,1X,'IAS    =',I5,6X,'IDES   =',I5,6X,'IYAS   =',I5,6X,'IPUNT =',
     9  I5,6X,'M      =',I5,6X,'NSK    =',I5,6X,'N      =',I5,/)
      WRITE(6,200)(IX(I),I=1,NN)
200   FORMAT(1X,'IX     =',20I5,(/,8X,20I5))
      IF(NL.NE.0)WRITE(6,210)(IV(I),I=1,NL)
210   FORMAT(1X,'IV     =',20I5,(/,8X,20I5))
      IF(IPLOT.NE.1)GO TO 215
      WRITE(6,211)(IDXP(I),I=1,NN)
      WRITE(6,212)(IDG(I),I=1,NN)
```

```
211   FORMAT(1X,'IDXP  =',20I5,(/,8X,20I5))
212   FORMAT(1X,'IDG   =',20I5,(/,8X,20I5))
215   IF(KX.EQ.0)GO TO 240
      WRITE(6,230)(KVX(I),I=1,KX)
      WRITE(6,220)(GX(1,J),OPSK(KVX(J)),GX(2,J),J=1,KX)
220   FORMAT(1X,'GX    =',(1X,3(G10.4,' < ',A8,' < ',G10.4,' ! ')))
230   FORMAT(1X,'KVX(I)=',20I5)
240   IF(KA.EQ.0)GO TO 270
      WRITE(6,250)(KVA(I),I=1,KA)
250   FORMAT(1X,'KVA(I)=',20I5)
      WRITE(6,260)(GA(1,J),KVA(J),GA(2,J),J=1,KA)
260   FORMAT(1X,'GA    =',(1X,3(G10.4,' < ','  A(',I2,') ',' < ',G10.4,
     1' ! ')))
270   WRITE(6,280)EPS,TYD,YDUIM,FNU
280   FORMAT(' EPS   =',1PD10.2,'  TYD   =',1PD10.2,'  YDUIM =',1PD10.2,
     1'  FNU   =',1PD10.2)
      IF(IPLOT.EQ.0) GO TO 320
      WRITE(6,290)XBEG,XEND,YBEG,YEND,XLEN,YLEN
290   FORMAT(1X,'XBEG  =',F6.2,6X,'XEND  =',F6.2,6X,'YBEG  =',F6.2,6X,
     1'YEND  =',F6.2,6X,'XLEN  =',F6.2,6X,'YLEN  =',F6.2,/)
      NMIN2=NN-2
      WRITE(6,310)(XP(I),I=1,NMIN2)
      IF(NSOEK.NE.0)WRITE(6,300)(TOLX(I),I=1,NSOEK)
300   FORMAT(1X,'TOLX  =',1P8D10.2)
310   FORMAT(1X,'XP(I) =',1P8D10.2)
320   WRITE(6,360)FORM
      IF(KA.EQ.0)   GO TO 340
      DO   330   J=1,KA
      IF(GA(2,J).LT.GA(1, J))   GO TO 401
      AK1=A(KVA(J))
      IF(AK1.GT.GA(2,J).OR.AK1.LT.GA(1,J)) GO TO 403
330   CONTINUE
340   IF(NL.NE.0)WRITE(6,350)(A(I),I=1,NL)
350   FORMAT(1X,'A(I)  =',1P6G20.8/(8X,6G20.8))
360   FORMAT(1X,'FORM  =',10A8,//)
370   LEESX=0
      IF(IDAT.EQ.1)CALL DATA(NG01,1)
      IF(IDRUK.GT.-1)WRITE(6,380)
380   FORMAT(//,50X,'** RESULTATE **',/)
      GO TO 400
390   CALL DATTYD
      IFIT=-1
      IF(IPLOT.EQ.1.AND.IER.EQ.0)CALL ASSE
      CALL EXIT
400   RETURN
401   WRITE(6,402) GA(1,J),GA(2,J),J
402   FORMAT(1X,'*** WAARSKUWING GRENSE ***'/1X,'GA(1,J) =',1PD16.4,
     110X,'GA(2,J) =',1PD16.8,10X,'J=',I4)
      CALL EXIT
403   WRITE(6,404)KVA(J),KVA(J),AK1,J,GA(1,J),J,GA(2,J)
404   FORMAT(///,1X,'*** WAARSKUWING!  WAARDE VAN A(',I2,') LE BUITE VOO
     1RGESKREWE GRENSE',/,1X,'A(',I2,')=',F16.5,2X,'GA(1,',I2,')=',F10.4
     2,2X,'GA(2,',I2,')=',F10.4)
      CALL EXIT
405    NN2=N2-2
      WRITE(6,406)MM,NN2
406   FORMAT(1X,'AANTAL KOLOMME IN MATRYS TE MIN GEDEFINIEER',2I5)
      END
```

```
C
C   SUBROETINE MINLSQ
C
C   DOEL
C
C      OM 'N LINEERE OF NIE-LINEERE FUNKSIE DEUR GEGEWE DATAPUNTE X(I,J)
C      TE PAS, DEUR GEBRUIK TE MAAK VAN DIE METODE VAN KLEINSTEKWADRATE.
C
C   GEBRUIK
C
C      CALL MINLSQ(NL,M,A,XX,IV,IL,ICON,IPR,NSK,F,X,B,C,IP,EPS,
C     1,ITSIK,IER,NB,NC,ND)
C
C   BESKRYWING PARAMETERS
C
C      NL     AANTAL KOEFFISIENTE WAT IN PASSING GEBRUIK WORD
C      M      AANTAL DATAPUNTE WAT IN PASSING GEBRUIK WORD
C      A      VEKTOR MET DIMENSIE VAN MINSTENS NL, EN BEVAT BEGINWAARDES
C             VAN DIE ARGUMENTE WAT IN PASSING GEBRUIK WORD, BY TERUG-
C             KEER BEVAT A DIE EINDWAARDES WAT DIE FUNKSIE MINIMEER
C      XX     VEKTOR MET DIMENSIE VAN MINSTENS NL, EN BEVAT DIE GESKAALDE
C             A-WAARDES
C      IV     VEKTOR MET DIMENSIE VAN MINSTENS NL, AS IV(I)=1, DIE VERAN-
C             DERLIKE A(I) WORD GEDURENDE ITERASIE PROSES NIE VERANDER
C             NIE, ANDERS IS IV(I)=0
C      IL     PARAMETER WAT DIE MAKSIMUM AANTAL ITERASIES WAT UITGEVOER
C             WORD, AANDUI. AS IL=0, WORD FUNKSIE BESKOU AS LINEER IN SY
C             A-WAARDES
C      ICON   AS ICON=0 WORD 'N ABSOLUTE KONVERGENSIE TOETS GEBRUIK EN
C             VIR ICON=1 WORD 'N RELATIEWE KONVERGENSIE TOETS GEBRUIK
C      IPR    UITDRUKPARAMETER. AS IPR=0 WORD GEEN RESULTATE UITGEDRUK
C             NIE. AS IPR.NE.0 WORD ELKE IPR-DE ITERASIE RESULTATE UITGE-
C             DRUK.
C      NSK    AS NSK=0 WORD GEEN SKALERING VAN A-WAARDES GEDOEN NIE, EN
C             AS NSK=1 WORD SKALERING GEDOEN
C      F      VEKTOR MET DIMENSIE M EN BEVAT BY TERUGKEER DIE WAARDES VAN
C             DIE GEPASTE FUNKSIE
C      X      MATRIKS MET DIMENSIE X(I,J),I=1, M EN J=1, MM WAAR MM DIE
C             AANTAL VERANDERLIKES VOORSTEL
C      B      MATRIKS MET DIMENSIE B(M+2,NL+2), EN BEVAT DIE AFGELEIDES
C             BY ELKE DATAPUNT MET BETREKKING TOT A.  B(I,J)-WAARDES WORD
C             IN SUBROETINE FIT BEREKEN
C      C      MATRIKS MET DIMENSIE C(NC,NC),EN WORD GEBRUIK VIR WERKAREA.
C      EPS    KONVERGENSIEKRITERIUM WAT A-WAARDES MOET BEVREDIG.
C      ITSIK  AANTAL ITERASIES WAT UITGEVOER MOET WORD VOORDAT HERSKALING
C             PLAASVIND
C      IER    UITSETFOUTPARAMETER MET DIE VOLGENDE BETEKENIS
C             IER = 0 GEEN FOUT, NORMALE KONVERGENSIE
C             IER = 1 GEEN KONVERGENSIE NA IL ITERASIES
C             IER = 2  M IS KLEINER AS NL
C             IER = 3  GEEN KONVERGENSIE KAN VERKRY WORD
C             IER = 4  RANG VAN MATRYS B IS KLEINER AS NL
C             IER = 5  TE MIN TYD GEGEE.
C      NB     EERSTE INDEKS VAN MATRIKS X EN B, NB MOET MINSTENS M+2 WEES
C      NC     DIMENSIE VAN MATRIKS C, NC MOET MINSTENS MAKS(MM,NL)+2 WEES
C      ND     TWEEDE INDEKS VAN MATRIKS B, ND MOET MINSTENS NL+2 WEES.
C
```

```
C
C   SUBROETINES GEROEP DEUR MINLSQ
C
C      FIT      -   OM FUNKSIEWAARDES EN AFGELEIDES TE BEREKEN
C      CONST    -   OM LINEERE A'S UIT TE SOEK
C
C      METODE
C      DIE GAUSS-NEWTON-MARQUARDT METODE WORD GEBRUIK EN HOUSEHOLDER
C      TRANSFORMASIES WORD TOEGEPAS
C
      SUBROUTINE MINLSQ(NL,M,A,XX,IV,IL,ICON,IPR,NSK ,F,X,B,C,IP,
     1  EPS,ITSIK,IER,NB,NC,ND)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XX(1),IV(1),F(1),B(NB,1),C(NC,1),IP(1)
      DIMENSION FSS(3),X(NB,1),A(1),HH(50)
      COMMON/FNUK/ FN   /AFGE/ NUM
      N=NL
      IF(M.LT.N) GOTO 400
      DO 1 I=1,N
      B(I,ND)=1.D-6
    1 B(I,ND-1)=1.D0
      FSC=1.D0
      DO 2 I=1,3
    2 FSS(I)=1.D3*DFLOAT(I)
      ARC=1.
      IF(ICON.EQ.1) ARC=1.D-6
      IT=1
      IT2=1
      IER=0
      FNU=FN
      IDR=0
      FNN=1.D0
      FNMN=1.D-10
      FNMX=1.D32
      DECR=.5D0
      DINC=1.5
      NTEST=0
C
C   SKALINGSFAKTORE WORD BEREKEN
C
      DEL=1.D-12
      BOT=1.D-5
      TOP=1.D5
      IC1=0
    3 ICC=0
      IF(IL.EQ.0.OR.(NSK+NUM).EQ.0)GO TO 9
      IF(NSK.EQ.0) GO TO 7
C
C   EERSTE AFGELEIDES WORD BEREKEN
C
      DEL1=DSQRT(DEL)
      DEL3=DEL**(1.D0/3.D0)
      FSC=1.D0
      DO 4 I=1,N
      XX(I)=XX(I)*B(I,ND-1)
      B(I,ND)=1.D-6
    4 B(I,ND-1)=1.D0
      CALL FIT(0,A,X,XX,B(1,ND-1),F,FSC,B,NB)
```

```
      DO 16 I=1,N
      HH(I)=DABS(XX(I))+1.D-10
      SOM=SOM+HH(I)
   16 CONTINUE
C
C  KYK OF A-WAARDES KONSTANT BLY
C
      NN=N
      IS=0
      DO 40 I=1,N
      IF(IV(I).EQ.0) GOTO 20
      NN=NN-1
      IS=IS+1
      GOTO 40
   20 IF(IS.EQ.0) GOTO 40
      IJ=I-IS
      HH(IJ)=HH(I)
      DO 30 J=1,M
   30 B(J,IJ)=B(J,I)
   40 CONTINUE
      IC=0
C
C  ORTOGONALE TRANSFORMASIES VAN B EN F
C
      DO 50 I=1,NN
      C(NC-1,I)=FSQ(M,B(1,I))
      C(NC,I)=C(NC-1,I)
   50 IP(I)=I
      DO 140 K=1,NN
      HH(K)=HH(K)/SOM
      KP1=K+1
      MK=M-K+1
C
C  KOLOMME EN/OF RYE WORD OMGERUIL INDIEN NOODSAAKLIK
C
      IF(K.EQ.NN) GOTO 100
   75 R=C(NC-1,K)
      L=K
      DO 80 I=KP1,NN
      IF(R.GE.C(NC-1,I)) GOTO 80
      R=C(NC-1,I)
      L=I
   80 CONTINUE
      IF(K.EQ.L) GOTO 100
      C(NC-1,L)=C(NC-1,K)
      C(NC-1,K)=R
      DO 90 I=1,M
      R=B(I,K)
      B(I,K)=B(I,L)
   90 B(I,L)=R
      J=IP(K)
      IP(K)=IP(L)
      IP(L)=J
  100 R=B(K,K)
      SIG=DSQRT(FSQ(MK,B(K,K)))
      IF(SIG.NE.0.D0) GOTO 106
      IER=4
      IPK=IP(K)
```

```
      FF=ESQ(M,F)
      DO 5 I=1,N
      XX(I)=XX(I)+DEL3
      CALL FIT(0,A,X,XX,B(1,ND-1),F,FSC,B,NB)
      FP=ESQ(M,F)
      XX(I)=XX(I)-2.D0*DEL3
      CALL FIT(0,A,X,XX,B(1,ND-1),F,FSC,B,NB)
      FM=ESQ(M,F)
      XX(I)=XX(I)+DEL3
      IF(DABS(FP-FM).LT.1.D-15)FP=FP+1.D-15*FP
      C(NC,I)=(FP+FM-2.D0*FF)/DEL3/DEL3
      XSC1=FF/DABS((FP-FM)/DEL3/2.D0)
      XSC2=DSQRT(FF/DABS((FP+FM-2.D0*FF)/DEL3/DEL3))
      XSC    =DMIN1(XSC1,XSC2,TOP)
      XSC    =DMAX1(XSC,BOT)
      B(I,ND-2)=XSC
      B(I,ND)=B(I,ND-2)*DEL1
    5 CONTINUE
      FSC=FF
      DO 6  I=1,N
      B(I,ND-1)=B(I,ND-2)
    6 XX(I)=XX(I)/B(I,ND-1)
C
C   LINEERE A-WAARDES WORD BEPAAL.
C
    7  CALL CONST(A,X,B,C(1,1),C(1,2),C(1,3),B(1,ND),B(1,ND-1),FSC,M,NL)
      IF(FSC.LE.1.D1.OR.IT.GT.5)IT2=ITSIK
C
C   FUNKSIEWAARDES WORD BEREKEN.
C
    9 IN=0
      CALL FIT(IN,A,X,XX,B(1,ND-1),F,FSC,B,NB)
      SSQ=ESQ(M,F)
      SKW=SSQ*FSC*FSC
      SK=SKW
      FSS(3)=SKW
      IF(IL.EQ.0.OR.IPR.EQ.0) GO TO 15
      IT3=IT-1
      WRITE(6,10)N,M,IT3,SKW,FNU,FSC
   10 FORMAT('0MINIMERING VAN SOM VAN VIERKANTE',5X,'N =',I3,5X,'M =',I5
     1,/,' ITERASIE ',I3,16X,'SKW    =',1PG14.7,5X,'FNU =',1PG14.7,
     15X,'FSC =',1PG14.7)
      WRITE(6,13)(A (I),I=1,N)
      IF(NSK.EQ.0)GO TO 15
      WRITE(6,12)(XX(I),I=1,N)
      WRITE(6,14)(B(I,ND-1),I=1,N)
      WRITE(6,11)(B(I,ND),I=1,N)
   11 FORMAT(16X,' DELDIF=', 1P6G18.8,/,(24X,1P6G18.8))
   12 FORMAT(16X,' X      =', 1P6G18.8,/,(24X,1P6G18.8))
   13 FORMAT(16X,' A      =', 1P6G18.8,/,(24X,1P6G18.8))
   14 FORMAT(16X,' XSC    =', 1P6G18.8,/,(24X,1P6G18.8))
C
C   AFGELEIDES VAN A WORD BEREKEN EN IN MATRIKS B GESTOOR.
C
   15 IN=1
      CALL FIT(IN,A,X,XX,B(1,ND-1),B(1,ND),FSC,B,NB)
      IF(IN.LT.0) GOTO 408
      SOM=0.
```

```
      DO 105 II=1,IPK
  105 IF(IV(II).GT.0)IPK=IPK+1
      IV(IPK)=2
      GO TO 15
  106 BETA=1.D0/(SIG*(SIG+DABS(R)))
      IF(R.GE.0.D0) SIG=-SIG
      C(NC-1,K)=SIG
      B(K,K)=R-SIG
      IF(K.EQ.NN) GOTO 130
      DO 120 J=KP1,NN
      R=BETA*EIN(MK,B(K,K),B(K,J))
      DO 110 I=K,M
  110 B(I,J)=B(I,J)-B(I,K)*R
  120 C(NC-1,J)=C(NC-1,J)-B(K,J)**2
  130 R=BETA*EIN(MK,B(K,K),F(K))
      DO 140 I=K,M
  140 F(I)=F(I)-R*B(I,K)
C
  145 IF(IL.NE.0) GOTO 170
      DO 160 I=1,NN
      C(I,NC-1)=F(I)
      IF(I.EQ.NN) GOTO 160
      I1=I+1
      DO 150 J=I1,NN
  150 C(J,I)=B(I,J)
  160 C(I,I)=C(NC-1,I)
      GOTO 260
  170 DO 250 K=1,NN
      FNK=FNU*HH(K)
      KM1=K-1
      KP1=K+1
      R=C(NC-1,K)
      RR=C(NC,K)
      SIG=R*R+(FNK*RR)**2
      IF(K.EQ.1) GOTO 180
      SIG=SIG+ESQ(KM1,C(1,K))
  180 SIG=DSQRT(SIG)
      BETA=1.D0/(SIG*(SIG+DABS(R)))
      IF(R.GE.0.D0) SIG=-SIG
      C(K,K)=SIG
      T=R-SIG
      IF(K.EQ.NN) GOTO 220
      DO 210 J=KP1,NN
      R=T*B(K,J)
      IF(K.EQ.1) GOTO 190
      R=R+EIN(KM1,C(1,K),C(1,J))
  190 R=BETA*R
      C(J,K)=B(K,J)-R*T
      IF(K.EQ.1) GOTO 210
      DO 200 I=1,KM1
  200 C(I,J)=C(I,J)-R*C(I,K)
  210 C(K,J)=-R*FNK*RR
  220 R=T*F(K)
      IF(K.EQ.1) GOTO 230
      R=R+EIN(KM1,B(1,1),C(1,K))
  230 R=BETA*R
      C(K,NC-1)=F(K)-R*T
      B(K,1)=-R*FNK*RR
```

```
      IF(K.EQ.1) GOTO 250
      DO 240 I=1,KM1
  240 B(I,1)=B(I,1)-R*C(I,K)
  250 CONTINUE
C
C  LOS OP DRIEDIAGONALE SISTEEM D.M.V. TERUGVERVANGING.
C
  260 C(NN,NC-1)=-C(NN,NC-1)/C(NN,NN)
      IF(NN.EQ.1) GOTO 290
      NNM1=NN-1
      NNP1=NN+1
      DO 280 J=1,NNM1
      K=NN-J
      R=C(K,NC-1)
      DO 270 I=1,J
  270 R=R+C(NNP1-I,NC-1)*C(NNP1-I,K)
  280 C(K,NC-1)=-R/C(K,K)
C
C  BEREKEN NUWE A-WAARDES
C
  290 DO 300 I=1,NN
  300 C(IP(I),1)=C(I,NC-1)
      DO 305 I=1,N
  305 C(I,NC-1)=XX(I)
      NP1=N+1
      ISS=IS
      DO 320 I=1,N
      K=NP1-I
      IF(IV(K).EQ.0) GOTO 310
      XX(K)=C(K,NC-1)
      ISS=ISS-1
      GOTO 320
  310 XX(K)=C(K,NC-1)+C(K-ISS,1)
  320 CONTINUE
      IN=0
      CALL FIT(IN,A,X,XX,B(1,ND-1),B,FSC,B,NB)
      IC1=IC1+1
      SQ=ESQ(M,B)
      SK = SQ*FSC*FSC
      IF(IL.EQ.0) GOTO 340
      IC=IC+1
      IF(IC.GT.99)GO TO 340
      FNN=DABS(SK/SKW)
      IF(IT.EQ.1.AND.IC.EQ.2)SKW=SK
      IF(SK.LT.SKW*1.0000001D0) GOTO 340
      IF(IDR.EQ.1 .AND. IC.EQ.1) WRITE(6,325)
  325 FORMAT(16X,'ITER',2X,'SOM VAN VIERK.',9X,'FNU',9X,'FNN')
      FN1=FNU/FNN
      IF(FN1.LT.1.D-20)FN1=1.D-20
      FNN=0.2D0*DLOG10(FN1)*DLOG10(FN1)+DINC
      FNN=DMIN1(FNN,10.D0)
      IF(IDR.EQ.1) WRITE(6,335)IC,IC1,SK,FNU,FNN
      FNU=FNU*FNN
      IF(FNU.GT.FNMX) GOTO 405
  335 FORMAT(14X,2I3,3X,1PG14.7,4(5X,1PG9.3) )
      DO 336 I=1,N
  336 XX(I)=C(I,NC-1)
      GOTO 170
```

```
  340 IF(IL.EQ.0) GO TO 382
      DO 341 I=2,3
  341 FSS(I-1)=FSS(I)
      FSS(3)=SK
      IF(IPR.EQ.0)GO TO 342
      IF(IT/IPR*IPR-IT.NE.0) GO TO 342
      WRITE(6,343)IT,IC1,SK ,FNU ,FNN
      WRITE(6,344)(HH(I),I=1,N)
      IF(NSK.NE.0)WRITE(6,12)(XX(I),I=1,N)
      WRITE(6,13)(A  (I),I=1,N)
  342 IF(DABS(FSS(3)-FSS(1)).LT.EPS)NTEST=NTEST+1
  343 FORMAT(/,1X,'ITERASIE ',I3,4X,'ITF    =',I3,2X,'SKW =',1PG14.7,5X,
     1'FNU =',1PG14.7,5X,'FNN',1PG17.4)
  344 FORMAT(16X,' V      =', 1P6G18.8,/,(24X,1P6G18.8))
      IF(NTEST.GT.1)GO TO 382
      SKW=SK
      IF(IC.GT.1) GOTO 345
      IF(FNU.GT.FNMN)FNU=DECR*FNU
C
C  KONVERGENSIETOETS
C
  345 DO 350 I=1,N
      XSC=B(I,ND-1)
      IF(DABS(XSC*(XX(I)-C(I,NC-1))).GT.EPS*(DABS(XSC*XX(I))+ARC))
     1GO TO 360
  350 CONTINUE
      GOTO 382
  360 IF(IT.GE.IL) GOTO 390
      IT=IT+1
      DO 380 I=1,M
  380 F(I)=B(I,1)
      IF(NSK.EQ.0)GO TO 15
      IT1=IT-1
      IF((IT1/IT2  )*IT2  -IT1.EQ.0)GO TO 3
      GOTO 15
  382 DO 384 I=1,M
  384 F(I)=B(I,1)
      GOTO 420
C
  390 IER=1
      GOTO 420
  400 IER=2
      GOTO 420
  405 IER=3
      GOTO 415
  408 IER=5
      GO TO 420
  415 DO 416 I=1,N
  416 XX(I)=C(I,NC-1)
      CALL FIT(0,A,X,XX,B(1,ND-1),F,FSC,B,NB)
  420 IL=IT
      RETURN
      END
```

```
      SUBROUTINE DIFF(A,B,X,OPSK,IX,N,LONG,N1,N2)
C
C  IN SUBROETINE DIFF WORD DIE FUNKSIEWAARDES VAN DIE GEPASTE FUNKSIE
C  BEREKEN, DIE VERSKILLE TUSSEN WERKLIKE EN GEPASTE FUNKSIEWAARDES,
C  DIE REGRESIEKOEFFISIENT EN SOM VAN KWADRAATAFWYKINGS.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(N1,1),OPSK(1),IX(1),A(1),B(1)
      COMMON/INTR/IP(50), IV(50), IN, KA, KX, MM, NL, IT, ITSIK, IPR,
     1 ICON, NSK, ICOR, IER,  NN,   NLL,   IPONS
      COMMON/BLOS/ EPS,   PASF(10),  TYD  /XIJ/ I,   NX
      DATA X1/' BEREKEN'/,X2/' RESIDU '/
      SY=X(N1-1,IX(NN))
      NN2=NN+2
      OPSK(N2-1)=X1
      OPSK(N2)=X2
      IX(NN+1)=N2-1
      IX(NN+2)=N2
      NN1=NN-1
      S6=0.D0
      S4=0.D0
      S5=0.D0
C
C  BEREKENING VAN FUNKSIEWAARDES EN VERSKILLE
C
      DO 50 I=1,N
      IF(NL.EQ.0)GO TO 20
      DO 10 J=1,MM
10     B(J)=X(I,J)
      CALL PASFUN(B,A,FPAS)
      X(I,N2-1)=FPAS
      GO TO 40
20    X(I,N2-1)=A(1)
      DO 30 J=1,NN1
30    X(I,N2-1)=X(I,N2-1)+     (X(I,IX(J)))*A(J+1)
40    X(I,N2)   =     (X(I,IX(NN)))-X(I,N2-1)
      S5=S5+DABS(X(I,N2))
      S4=S4+X(I,N2-1)
      S6=S6+X(I,N2)*X(I,N2)
50    CONTINUE
      IF(LONG.LT.0)GO TO 110
      I4=12
      I1=NN2/I4
      IF(I1*I4.NE.NN2)I1=I1+1
      DO 100 I=1,I1
      I2=I4*(I-1)+1
      I3=I2+I4-1
      IF(I3.GT.NN2)I3=NN2
      WRITE(6,60)(OPSK(IX(J)),J=I2,I3)
60    FORMAT(2X,'NR.',1X,12A10)
      WRITE(6,70)
70    FORMAT( )
C
      DO 90 J=1,N
      WRITE(6,80)J,(X(J,IX(I5)),I5=I2,I3)
```

```
80     FORMAT(1X,I3,12F10.4)
90     CONTINUE
       WRITE(6,170)
100    CONTINUE
C
C  BEREKENING VAN REGRESSIEKOEFFISIENT, SOM VAN KWADRAATAFWYKINGS, SOM
C  ABSOLUTE FOUT, EN WORTEL VAN GEMIDDELDE KWADRAATAFWYKING VAN DIE
C  PERSENTASIE FOUT.
C
110    S1=S4/DFLOAT(N)
       S2=0.
       S3=0.
       S4=0.D0
       F1=0.D0
       DO 130 I=1,N
       S2=S2+(X(I,N2-1)-S1)**2
       S3=S3+(X(I,IX(NN))-SY)*(X(I,N2-1)-S1)
       S4=S4+(X(I,IX(NN))-SY)**2
       IF(X(I,IX(NN)).EQ.0.D0)GO TO 120
       RES1=X(I,N2)/X(I,IX(NN))
120    F1=F1+RES1*RES1
130    CONTINUE
       RMS1=DSQRT(F1/N)*1.D2
       RMS=DSQRT(S6/N)
       COR=S3/DSQRT(S2*S4)
C
C  SKRYF RESULTATE UIT
C
       IF(IER.EQ.5)IT=IT-1
       WRITE(6,140)IER,IT,COR,S6,S5,RMS,RMS1
140    FORMAT('0','RESULTATE',//,1X,'IER=',I2,2X,'IT=',I2,2X,'REGRESSIEKO
      1EFFISIENT',40X,'=',1P1D16.8,/,
      216X,'SOM VAN KWADRAATAFWYKINGS',35X,'=',1P1D16.8,/,
      316X,'SOM ABSOLUTE FOUT',43X,'=',1PD16.8,/,
      316X,'WORTEL GEMIDDELDE KWADRAATAFWYKING',26X,'=',1PD16.8,/,
      416X,'WORTEL GEMIDDELDE KWADRAATAFW. VAN DIE PERSENTASIEFOUT',6X,'=
      5',1P1D16.8,2X,'PERSENT')
       WRITE(6,150)(A(I),I=1,NLL)
150    FORMAT(//,1X,'A(I)=',1P5D20.10,/,(6X,1P5D20.10))
       IF(IER.NE.4) GO TO 155
       WRITE(6,151)
151    FORMAT(//,1X,'DIE RANG VAN DIE MATRIKS MET AFGELEIDES IS MINDER
      1AS NL, KYK NA DIE VOLGENDE A-WAARDES.')
       DO 152 I=1,NLL
152    IF(IV(I).EQ.2) WRITE(6,153)I
153    FORMAT(9X,'A(',I2,')')
155    WRITE(6,160) N
       WRITE(6,160) N
160    FORMAT(1H0,'PASSING GEBASEER OP',I4,' PUNTE'/)
       WRITE(6,170)
170    FORMAT(//)
       WRITE(6,180)PASF
180    FORMAT(1X,10A8)
       IF(IPONS.EQ.0)RETURN
       WRITE(7,190)(A(I),I=1,NLL)
190    FORMAT(6X,'A=',/,(3(6X,1PD17.10,',')))
       RETURN
       END
```

```
      SUBROUTINE FIT(IN,A,XV,XX,XSC,F,FSC,B,NB)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4   GX,GA,T1,TYD
      COMMON/DIM/N1,N2,N3
      COMMON/BLOK/GX(2,52),OPSK(52),IX(52),KVX(52),KVA(52),GA(2,52)
      COMMON/INTR/IP(50), IV(50),  N, KA, KX, MM, NL, IT, ITSTK, IPR,
     1 ICON, NSK, ICOR, IER,  NN,   NLL,   IPONS
      COMMON/BLOS/ EPS,   PASF(10),  TYD  /XIJ/ I,   NX
      COMMON/ICONS/ IK(50), NL1/AFGE/ NUM
      DIMENSION XSC(1),F(1),B(NB,1),XV(NB,1),XX(1),A(1),X(50)
C
C     XSC     VEKTOR MET DIMENSIE NL BEVATTENDE DIE SKALERINGSFAKTORE
C     DELDIF  VEKTOR MET DIMENSIE NL EN BEVAT DIE INKREMENTE OM NUMERIESE
C     AFGELEIDES TE BEREKEN
C
C
C  TERUGSKALERING VIND PLAAS.
C
      NX=N
      DO 10 I=1,NLL
10    A(I)=XX(I)*XSC(I)
      IF(KA.EQ.0.OR.IT.EQ.0.OR.NL.EQ.0) GO TO 30
      DO 20 K=1,KA
      O=GA(1,K)
      C=GA(2,K)
20    A(KVA(K))=O+(C-O)*DSIN(A(KVA(K)))**2
30    CONTINUE
      K=1
C
C  INDIEN IN = 1 MOET AFGELEIDES BEREKEN WORD, ANDERS FUNKSIEWAARDES
C
      IF(IN.EQ.1) GO TO 90
      IF(NL.EQ.0)GO TO 60
      DO 50 I=1,N
      DO 40 J=1,MM
40    X(J)=XV(I,J)
      Y=XV(I,IX(52))
      CALL PASFUN(X,A,FPAS)
      F(I)=(FPAS-Y)/FSC
50    CONTINUE
      RETURN
60    DO 80 I=1,N
      F(I)=A(1)
      DO 70 J=2,NLL
70    F(I)=F(I)+A(J)*XV(I,IX(J-1))
      F(I)=F(I)-XV(I,IX(52))
80    CONTINUE
      RETURN
90    CONTINUE
C
C  TI = VERWERKINGSTYD NOG BESKIKBAAR. INDIEN TI.LE.TYD KEER PROGRAM
C       TERUG NA HOOFPROGRAM.
C
      CALL CPUREM(T1)
      IF(T1.LE.TYD)GO TO 170
      IF(IT.EQ.0.AND.NL.EQ.0)GO TO 150
      IF(NUM.EQ.0) GO TO 180
```

```
      SIGN=1.D0
      DO 91 I1=1,NLL
      DO 94 J=1,NUM
91    B(NB-J+1,I1)=XX(I1)*XSC(I1)+F(I1)*SIGN
      IF(KA.EQ.0.OR.IT.EQ.0.OR.NL.EQ.0) GO TO 93
      DO 92 K=1,KA
      O=GA(1,K)
      C=GA(2,K)
92    B(NB-J+1,KVA(K))=O+(C-O)*DSIN(B(NB-J+1,KVA(K)))**2
93    CONTINUE
      SIGN=-SIGN
94    CONTINUE
C
C  AFGELEIDES M.B.T. A WORD BY ELKE PUNT BEREKEN EN IN B GESTOOR.
C
      DO 140 I=1,N
      L=0
      DO 100 M=1,MM
100   X(M)=XV(I,M)
      IF(NUM.EQ.1)CALL PASFUN(X,A,F2)
      DO 130  J=1,NL
      IF(IT.EQ.0)GO TO 110
      IF(IK(J).EQ.0)GO TO 120
110    TEMP=A(J)
      A(J)=B(NB,J)
      CALL PASFUN(X,A,F1)
      IF(NUM.EQ.1)GO TO 115
      A(J)=B(NB-1,J)
      CALL PASFUN(X,A,F2)
115   A(J)=TEMP
      B(I,J)=(F1-F2)/F(J)/DFLOAT(NUM)
      B(I,J)=B(I,J)/FSC*XSC(J)
      GO TO 130
120   L=L+1
      B(I,J)=B(I,NL+L)
130   CONTINUE
140   CONTINUE
      RETURN
150   CONTINUE
      DO 160 I=1,N
      B(I,1)=1.D0
      DO 160 J=2,NLL
160   B(I,J)=XV(I,IX(J-1))
      RETURN
170   IN=-1
      IPONS=1
      RETURN
180   CONTINUE
      DO 220 I=1,NX
      DO 210 J=1,MM
210   X(J)=XV(I,J)
      CALL PASAF(X,A,F)
      DO 205 J=1,NLL
205   B(I,J)=F(J)/FSC*XSC(J)
220   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CORREL(N,MM,X,COR,B,ICOR,N1,N2,N3)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(N1,1),COR(N3,1),B(1)
      DO 100 J=1,MM
      X(N1,J)=0.D0
100   X(N1-1,J)=0.D0
      DO 101 I=1,MM
      DO 101 J=1,MM
101   COR(I,J)=0.D0
C
C  BEREKEN GEMIDDELDES
C
      DO 108 J=1,MM
      DO 107 I=1,N
107   X(N1-1,J)=X(N1-1,J)+     X(I,J)
108   X(N1-1,J)=X(N1-1,J)/DFLOAT(N)
      IF(ICOR.EQ.0)RETURN
C
C  GEMIDDELDES WORD VAN DATA AFGETREK
C
      DO 109 J=1,MM
      DO 109 I=1,N
109   X(I,J)=X(I,J)-X(N1-1,J)
C
C  BEREKEN STANDAARDAFWYKINGS
C
      DO 111 J=1,MM
      DO 110 I=1,N
110   X(N1,J)=X(N1,J)+X(I,J)*X(I,J)
      IF(X(N1,J).EQ.0.D0)X(N1,J)=1.D-30
111   CONTINUE
C
C  BEREKEN KORRELASIEKOEFFISIENTE
C
      DO 115 JJ=1,MM
      DO 114 J=JJ,MM
      DO 113 I=1,N
113   COR (JJ,J)=COR(JJ,J)+X(I,J)*X(I,JJ)
      IF(J.EQ.JJ)B(J)=COR(JJ,J)
      DEEL=DSQRT(X(N1,J)*X(N1,JJ))
      COR(JJ,J)=COR(JJ,J)/DEEL
114   COR(J,JJ)=COR(JJ,J)
115   CONTINUE
      DO 116 I=1,MM
116   X(N1,I)=DSQRT(X(N1,I)/(N-1))
      DO 118 J=1,MM
      DO 118 I=1,N
118   X(I,J)=X(I,J)+X(N1-1,J)
      RETURN
      END
```

```
      SUBROUTINE DATUIT(X,COR,A,N,MM,LONG,ICOR,N1,N2,N3)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4    GX,GA
      COMMON/BLOK/GX(2,52),OPSK(52),IX(52),KVX(52),KVA(52),GA(2,52)
      DIMENSION X(N1,1),COR(N3,1),A(50)
      WRITE(6,11)
C
C   VIR LONG.LE.0 WORD X-WAARDES NIE UITGEDRUK NIE.
C
      IF(LONG.LE.0) GO TO 30
11    FORMAT(//)
      I4=10
      I1=MM/I4
      IF(I1*I4.NE.MM)I1=I1+1
      DO 25 II=1,I1
      I2=I4*(II-1)+1
      I3=I2+I4-1
      IF(I1.EQ.II)I3=MM
      WRITE(6,19)(OPSK(J),J=I2,I3)
17    FORMAT(/,1X,'STD AFW.',2X,10F12.4)
18    FORMAT(/,1X,'GEMM.',5X,10F12.4)
19    FORMAT(8X,'NR.',1X,10A12)
      WRITE(6,33)
      DO 20 I=1,N
      WRITE(6,21)I,(X(I,J),J=I2,I3)
20    CONTINUE
21    FORMAT(6X,I5,10F12.4)
      WRITE(6,18)(X(N1-1,J),J=I2,I3)
      WRITE(6,17)(X(N1,J),J=I2,I3)
      WRITE(6,11)
25    CONTINUE
30    CONTINUE
C
C   VIR LONG.LT.0 OF ICOR = 0 WORD KORRELASIETABEL NIE UITGEDRUK NIE.
C
      IF(LONG.LT.0.OR.ICOR.EQ.0)RETURN
      WRITE(6,31)
31    FORMAT(' MATRIKS MET KORRELASIEKOEFFISIENTE ',/)
      I4=20
      I1=MM/I4
      IF(I1*I4.NE.MM)I1=I1+1
      DO 50 II=1,I1
      I2=I4*(II-1)+1
      I3=I4+I2-1
      IF(I3.GT.MM)I3=MM
      WRITE(6,32)(I,I=I2,I3)
      WRITE(6,33)
32    FORMAT('      I',4X,20I6)
33    FORMAT( )
      DO 35 I=1,MM
      WRITE(6,34)I,(COR(I,J),J=I2,I3)
34    FORMAT(I5,5X,20F6.2)
35    CONTINUE
      WRITE(6,11)
50    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CONST(A,XV,B,D1,D2,X,F,XSC,FSC,M,NL)
C
C   SUBROETINE CONST SOEK A'S UIT, WAT IN DIE PASSINGFUNKSIE LINEER
C   VOORKOM. INDIEN A(J) LINEER VOORKOM WORD IK(J) GELYK AAN 1 GESTEL
C   SO NIE IS A(J) GELYK AAN 0.
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/DIM/N1,N2,N3
      COMMON/INTR/IP(50), IV(50),  N, KA, KX, MM,NNL, IT, ITSIK, IPR,
     1 ICON, NSK, ICOR, IER,  NN,   NLL,   IPONS
      COMMON/ICONS/ IK(50), NL1  /XIJ/ I,  NX
      COMMON/BLOS/ EPS,   PASF(10),  TYD
      DIMENSION XV(N1,1),B(N1,1),X(1),A(1),F(1),D1(1),D2(1),XSC(1)
      NL1=N3-NL-2
      DO 5 I=1,50
5     IK(I)=1
      IF(NL1.EQ.0)RETURN
      I=2
      DO 10 J=1,MM
10    X(J)=XV(I,J)
C
C   AFGELEIDES WORD BEREKEN.
C
      L=0
      DO 20 J=1,NL
      A(J)=A(J)+F(J)
      CALL PASFUN(X,A,F1)
      A(J)=A(J)-F(J)*2.D0
      CALL PASFUN(X,A,F2)
      A(J)=A(J)+F(J)
      D1(J)=(F1-F2)/F(J)/2.D0
20    CONTINUE
      DO 30 J=1,NL
30    A(J)=A(J)+F(J)*1.D0
      DO 40 J=1,NL
      A(J)=A(J)+F(J)
      CALL PASFUN(X,A,D2(J))
      A(J)=A(J)-F(J)*2.D0
      CALL PASFUN(X,A,F2)
      A(J)=A(J)+F(J)
40    D2(J)=(D2(J)-F2)/F(J)/2.D0
      DO 50 J=1,NL
50    A(J)=A(J)-F(J)*1.D2
      DO 70 J=1,NL
      IF(L.GE.NL1)GO TO 59
      IF(DABS(D2(J)-D1(J)).LT.EPS)GO TO 60
59    IK(J)=1
      GO TO 70
60    L=L+1
      IK(J)=0
      DO 65 I=1,M
C
C   INDIEN A LINEER WORD B-WAARDES BEREKEN.
C
      DO 61 II=1,MM
61    X(II)=XV(I,II)
```

```
C
      CALL PASFUN(X,A,F2)
      A(J)=A(J)+F(J)
      CALL PASFUN(X,A,F1)
      A(J)=A(J)-F(J)
      B(I,NL+L)=(F1-F2)/F(J)
65    B(I,NL+L)=B(I,NL+L)/FSC*XSC(J)
70    CONTINUE
      RETURN
      END




      SUBROUTINE PLOTT(A,X,XF,N1)
      DIMENSION A(1),XF(1)
      COMMON/INTR/IP(50), IV(50),  N, KA, KX, MM, NL, IT, ITSIK, IPR,
     1 ICON, NSK, ICOR, IER,  NN,   NLL,   IPONS
      COMMON/BLOK/GX(2,52),XNAME(52),IX(52),KVX(52),KVA(52),GA(2,52)
      COMMON/IPLT/NSOEK, IXAS,  IYAS, IPUNT, ILYN, IAS, NDIGD, IPRINT,
     1            IDENT, IDES,  LSIM,  NV,  IPLOT, IFIT, IPEN
      COMMON/APLT/ XBEG, XEND,  YBEG,  YEND, XLEN,YLEN,XP(50),TOLX(50),
     1          IDXP(50),IDG(50)
      REAL*8 X,XF,XNAME,A
      DIMENSION X(N1,1)
      DATA NOP/0/
      IF(IAS.EQ.1) LSIM=0
C
C   SUBROETINE ASSE TREK ASSESTELSEL
C
      IF(LSIM.EQ.0.OR.NOP.EQ.0)CALL ASSE
      NOP=1
      NV=0
      IF(NSOEK.LT.1) GO TO 35
      IF(NSOEK.LT.1.AND.KX.LT.1.AND.IPUNT.NE.1.AND.ILYN.NE.1)
     1GO TO 10
      IF(IPRINT.LT.0)GO TO 10
      WRITE(6,1)
      WRITE(6,2)(XNAME(IX(I)),XP(I),I=1,NSOEK)
      WRITE(6,8)(XNAME(IX(I)),TOLX(I),I=1,NSOEK)
1     FORMAT('0','PUNTE WORD GEKARAKTERISEER DEUR')
2     FORMAT((1X,6(2X,A8,'=',G10.4)))
3     FORMAT((1X,3(5X,A8,'TUSSEN ',G10.4,'EN ',G10.4)))
4     FORMAT(1X/(1X,10(A8,5X)))
5     FORMAT(1X,'GEEN PUNTE HIER NIE')
6     FORMAT(1X,10G13.5)
8     FORMAT((1X,'BREUKTOLERANSIES ',6('--',A8,':',F8.6)))
C
C   INDIEN IPUNT = 1 WORD PUNTE GETEKEN
C
      IF(IPUNT.NE.1.AND.ILYN.NE.1) GO TO 20
      WRITE(6,3)XNAME(IXAS),XBEG,XEND,XNAME(IYAS),YBEG,YEND
```

```
C
10    CONTINUE
      IF(LSIM.EQ.0) CALL SYMBOL(0.,YLEN+.55,.10,XNAME(IX(NSOEK)),0.,8)
      IF(LSIM.EQ.0) XNUMB=1.0
      CALL SYMBOL(XNUMB,YLEN+.62,.10,LSIM,0.0,-1)
      CALL NUMBER(XNUMB+0.15,YLEN+.55,.10,XP(NSOEK),0.,IDXP(NSOEK))
      CALL WHERE(XW,YW)
      XNUMB=XW+.2
20    CONTINUE
35    IF(IPUNT.LT.1)GO TO 75
      DO 70 I=1,N
      IF(NSOEK.LT.1)GO TO 60
      DO 40 J=1,NSOEK
      B=(XP(J)+TOLX(J))
      O=(XP(J)-TOLX(J))
      IF(X(I,IX(J)).LT.O.OR.X(I,IX(J)).GT.B)GO TO 70
40    CONTINUE
50    IF(IPUNT.LT.1.AND.ILYN.LT.1) GO TO 60
      IF(X(I,IXAS).LT.XBEG.OR.X(I,IXAS).GT.XEND)GO TO 70
      IF(X(I,IYAS).LT.YBEG.OR.X(I,IYAS).GT.YEND) GO TO 70
60    CONTINUE
      NV=NV+1
C
C  INDIEN IPRINT = 1 WORD UITGESOEKTE DATAPUNTE UITGEDRUK
C
      IF(IPRINT.LE.0)GO TO 65
      IF(NV.EQ.1)WRITE(6,4)(XNAME(K),K=1,MM)
      WRITE(6,6)(X(I,K),K=1,MM)
65    XX1   =(X(I,IXAS)-XBEG)/(XEND-XBEG)*XLEN
      YY1  =(X(I,IYAS)-YBEG)/(YEND-YBEG)*YLEN
      CALL SYMBOL(XX1,YY1,0.07,LSIM,0.0,-1)
C
C  INDIEN IDENT > 0 WORD 'N SIMBOOL BY ELKE PUNT GETEKEN
C
      IF(IDENT.GE.1)CALL NUMBER(XX1-.15,YY1+.05,.07,X(I,IDENT),0.,IDES)
70    CONTINUE
      IF(NV.LT.1) WRITE(6,5)
C
C  INDIEN ILYN = 1 WORD LYN GETREK
C
75    IF(ILYN.EQ.1) CALL LYN(A,X,XF,N1)
      LSIM=LSIM+1
      IPEN=IPEN+1
      IF(IPEN.GT.3)IPEN=1
      CALL NEWPEN(IPEN)
      RETURN
      END
```

```
      SUBROUTINE ASSE
C
C  SUBROETINE OM ASSESTELSEL TE TEKEN
C
      REAL*8   XNAME,EPS,PASF,TYD
      COMMON/BLOK/GX(2,52),XNAME(52),IX(52),KVX(52),KVA(52),GA(2,52)
      COMMON/IPLT/NSOEK, IXAS,  IYAS, IPUNT, ILYN, IAS, NDIGD, IPRINT,
     1              IDENT, IDES,  LSIM,  NV,  IPLOT, IFIT, IPEN
      COMMON/INTR/IP(50), IV(50),  N, KA, KX, MM, NL, IT, ITSIK, IPR,
     1 ICON, NSK, ICOR, IER,  NN,   NLL,   IPONS
      COMMON/APLT/ XBEG, XEND,  YBEG,  YEND, XLEN,YLEN,XP(50),TOLX(50),
     1          IDXP(50),IDG(50)
      COMMON/BLOS/ EPS,   PASF(10),  TYD
      COMMON/BGAS/  YDUIM, FAKT
      DATA IJ/0/
      IF(IFIT.EQ.-1)GO TO 90
      IF(IJ.NE.0) GO TO 40
      CALL IHCPLT(48,'PROGRAM GAAN PLOT - PEN WORD DEUR PLOTTER GESTEL')
      CALL FACTOR(FAKT)
      CALL PLOT(2.,-11.,-3)
      CALL PLOT(0.0,1.0,-3)
      XL=0.
      YL=0.
      IJ=1
C
C  INDIEN IAS = 0 WORD GEEN ASSESTELSEL GETEKEN NIE
C
      IF(IAS.EQ.0)RETURN
      GO TO 50
40    IF(YLEN*2.+YL+3.0.GT.YDUIM) GO TO 45
      YL=YL+YLEN+3.0
      CALL PLOT(0.,YLEN+3,-3)
      GO TO 50
45    YL=-YL
      CALL PLOT(XLEN+3.5,YL,-3)
      YL=0.
50    DX=(XEND-XBEG)/XLEN
      DY=(YEND-YBEG)/YLEN
      XA=0.0
      YA=0.0
C
C  X - Y ASSE WORD GETEKEN
C
      CALL NEWPEN(1)
      CALL AXIS(XA,YA,XNAME(IYAS),8,YLEN,90.,YBEG,DY,10.)
      CALL AXIS(XA,YA,XNAME(IXAS),-8,XLEN,0.0,XBEG,DX,10.)
      CALL SYMBOL(0.,YLEN+1.25,.10,PASF,0.0,80)
      IF(NSOEK.LE.1) GO TO 70
      NSOEK1=NSOEK-1
      XA=.0
      DO 60 I=1,NSOEK1
      CALL SYMBOL(XA,YLEN+.85,.10,XNAME(IX(I)),0.0,8)
      CALL WHERE(XW,YW)
      CALL SYMBOL(XW+.1,YLEN+.85,.10,'=',0.0,1)
      CALL NUMBER(XW+.35,YLEN+.85,.10,XP(I),0.0,IDXP(I))
      CALL WHERE (XW,YW)
60    XA=XW+.2
70    IF(KX.LT.1) RETURN
```

```
      XA=0.
      DO 80 I=1,KX
      CALL SYMBOL(XA,YLEN+1.05,.10,XNAME(KVX(I)),0.0,8)
      CALL WHERE(XA,YA)
      CALL SYMBOL(XA+.2,YLEN+1.05,.10,' TUSSEN',0.0,7)
      CALL NUMBER(XA+1.1,YLEN+1.05,.10,GX(1,I),0.0,IDG(I))
      CALL WHERE(XW,YW)
      XA=XW+.15
      CALL SYMBOL(XA,YLEN+1.05,.10,'EN',0.0,2)
      CALL NUMBER(XA+.30,YLEN+1.05,.10,GX(2,I),0.0,IDG(I))
      CALL WHERE(XW,YW)
80    XA=XW+.2
      RETURN
C
C  INDIEN IFIT = -1 WORD BUFFER LEEGGEMAAK
C
90    CALL PLOT(XLEN+4.,YLEN,999)
      CALL EXIT
      END




C
C
      SUBROUTINE LYN(A,X,XF,N1)
C
C  SUBROETINE LYN TEKEN DIE GEPASTE FUNKSIE.
C
      REAL*8 X,XNAME,A,YX,XF
      DIMENSION X(N1,1),A(1)
      COMMON/BLOK/GX(2,52),XNAME(52),IX(52),KVX(52),KVA(52),GA(2,52)
      COMMON/IPLT/NSOEK, IXAS,  IYAS, IPUNT, ILYN, IAS, NDIGD, IPRINT,
     1             IDENT, IDES,  LSIM,  NV,  IPLOT, IFIT, IPEN
      COMMON/INTR/IP(50), IV(50), KN, KA, KX, MM, NL, IT, ITSTK, IPR,
     1 ICON, NSK, ICOR, IER,  NN,   NLL,   IPONS
      COMMON/APLT/ XBEG, XEND,  YBEG,  YEND, XLEN,YLEN,XP(50),TOLX(50),
     1           IDXP(50),IDG(50)
      DIMENSION X1(102),Y1(102),XF(1)
      IF(NSOEK.LT.1) GO TO 30
      DO 10 J=1,MM
10    XF(J)=X(1,J)
      NN2=NN-2
      IF(NN2.LT.1)GO TO 30
      DO 20 J=1,NN2
20    XF(IX(J))=DBLE(XP(J))
30    NPTE=0
      IKYK=0
      ITEL=0
      N=NDIGD
      IF(N.GT.100)N=100
      DO 50 I=1,N
```

```
C
      XX=(I-1)*XLEN/(N-1)
      XXX=XX/XLEN*(XEND-XBEG)+XBEG
      IF(NN.GE.1)GO TO 35
      XF(1)=DBLE(XXX)
      GO TO 36
35    XF(IX(NN-1))=DBLE(XXX)
      IF(NL.EQ.0) GO TO 37
36    CALL PASFUN(XF,A,YX)
      GO TO 39
37    YX=A(1)
      DO 38 J=2,NN
38    YX=YX+A(J)*XF(IX(J-1))
39    YY=SNGL(YX)
      IF(YY.GE.YBEG.AND.YY.LE.YEND) GO TO 40
      IF(IKYK.EQ.1) GO TO 60
      GO TO 50
40    IKYK=1
      NPTE=NPTE+1
      X1(NPTE)=XX
      Y1(NPTE)=(YY-YBEG)/(YEND-YBEG)*YLEN
50    CONTINUE
60    CONTINUE
      IF(NPTE.LT.2) RETURN
      X1(NPTE+1)=0.
      X1(NPTE+2)=1.
      Y1(NPTE+1)=0.
      Y1(NPTE+2)=1.
      CALL LINE(X1,Y1,NPTE,1,0,-1)
C
C   SKRYF DIE OPSKRIF
C
      IF(NSOEK.LT.1) RETURN
      THETA1=ATAN2((Y1(NPTE)-Y1(NPTE-1)),(X1(NPTE)-X1(NPTE-1)))
      THETA=THETA1/3.14159*180.0
      XX=X1(NPTE)
      YY=Y1(NPTE)
      CALL SYMBOL(XX,YY,.07,XNAME(IX(NSOEK)),THETA,8)
      CALL WHERE(XW,YW)
      S=SQRT((XX-XW)**2+(YY-YW)**2)
      XX=XX+(S+.05)*COS(THETA1)
      YY=YY+(S+.05)*SIN(THETA1)
      CALL SYMBOL(XX,YY,.07,'=',THETA,1)
      XX=XX+.07*COS(THETA1)
      YY=YY+.07*SIN(THETA1)
C
C   TEKEN DRYFPUNTGETALLE
C
      CALL NUMBER(XX,YY,0.07,XP(NSOEK),THETA,IDXP(NSOEK))
      RETURN
      END
```

BYLAE B

In hierdie bylae verskyn lyste van die volgende Assembler-roetines [1].

(1) EIN

'n Dubbele noukeurigheidroetine om die skalaarproduk van twee vektore X en Y van lengte N te bereken [1].

(2) ESQ

'n Dubbele noukeurigheidroetine om die som van die kwadrate van die komponente van 'n vektor X met lengte N te bereken [1].

(3) CPUTIM

Die roetine word gebruik om die werklike verwerkingstyd te bepaal [1].

(4) CPUREM

Die roetine bepaal die tyd wat nog beskikbaar is vir die uitvoering van die taak [1].

(5) DMENSI

Subroetine DMENSI word gebruik om stoorplek dinamies toe te ken terwyl die program uitgevoer word en nie as die program vertaal word nie [1].

```
EIN      CSECT                     PROGRAM EIN
         B      8(15)
         DC     X'03'              PURPOSE
         DC     CL3'EIN'             THIS PROGRAM COMPUTES THE INNER PRODUCT
         STM    14,12,12(13)         (SUM X(I)*Y(I)) OF TWO VECTORS X AND Y
         BALR   12,0                 OF DIMENSION N. X,Y AND THE RESULT ARE
         USING  *,12                 DOUBLE PRECISION, BUT THE COMPUTATION
         LM     3,5,0(1)             IS DONE IN EXTENDED PRECISION AND THE
         L      6,0(3)               RESULT IS ROUNDED BEFORE RETURN.
         LD     0,ZERO
         LD     2,ZERO             USAGE (AS A FORTRAN FUNCTION SUBPROGRAM)
LOOP     LD     4,0(4)               EIN(N,X,Y)
         MXD    4,0(5)
         AXR    0,4
         A      4,EIGHT
         A      5,EIGHT
         BCT    6,LOOP
         LRDR   0,0
         LM     2,12,28(13)
         MVI    12(13),X'FF'
         BR     14
         DS     0D
ZERO     DC     D'0'
EIGHT    DC     F'8'
         END    EIN
```

```
ESQ      CSECT                     PROGRAM ESQ
         B      8(15)
         DC     X'03'              PURPOSE
         DC     CL3'ESQ'             THIS PROGRAM COMPUTES THE SUM OF
         STM    14,12,12(13)         SQUARES (SUM X(I)**2) OF THE COMPONENTS
         BALR   12,0                 OF A VECTOR X OF DIMENSION N  X AND THE
         USING  *,12                 RESULT ARE DOUBLE PRECISION, BUT THE
         LM     3,4,0(1)             COMPUTATION IS DONE IN EXTENDED
         L      6,0(3)               PRECISION AND THE RESULT IS ROUNDED
         LD     0,ZERO               BEFORE RETURN.
         LD     2,ZERO
LOOP     LD     4,0(4)             USAGE (AS A FORTRAN FUNCTION SUBPROGRAM)
         MXD    4,0(4)               ESQ(N,X)
         AXR    0,4
         A      4,EIGHT
         BCT    6,LOOP
         LRDR   0,0
         LM     2,12,28(13)
         MVI    12(13),X'FF'
         BR     14
         DS     0D
ZERO     DC     D'0'
EIGHT    DC     F'8'
         END    ESQ
```

```
AEB      TITLE 'CPUTIM - ROUTINE TO DETERMINE ELAPSED CPU TIME'
**********************************************************************
*                                                                    *
*       *****   ******  *     * ******* ******* *     *              *
*      *     *  *     * *     *    *       *    **   **              *
*      *        *     * *     *    *       *    * * * *              *
*      *        ******  *     *    *       *    *  *  *              *
*      *        *       *     *    *       *    *     *              *
*      *     *  *       *     *    *       *    *     *              *
*       *****   *        *****     *    ******* *     *              *
*                                                                    *
**********************************************************************
*                                                                    *
* USAGE....... CALL CPUTIM (TCPU)      (FORTRAN LINKAGE CONVENTIONS) *
*                                                                    *
* FUNCTIONS... CPUTIM RETURNS THE CPU TIME USED SINCE THE FIRST 'CALL *
*              CPUTIM' IN THE PROGRAM.                               *
*                                                                    *
* PARAMETERS.. ON ENTRY R1 IS ASSUMED TO POINT TO AN ALIGNED FULLWORD *
*              CONTAINING THE ADDRESS OF TCPU. TCPU IS ALSO ASSUMED  *
*              TO BE A FULLWORD ALIGNED ON A FULLWORD BOUNDARY.      *
*              ON EXIT TCPU CONTAINS THE ELAPSED CPU TIME IN SECONDS *
*              EXPRESSED AS A SHORT FLOATING POINT NUMBER.           *
*                                                                    *
* ATTRIBUTES.. NONE.                                                 *
*                                                                    *
* REGISTER.... R0 - NOT USED                                         *
* USAGE        R1 - ADDRESS OF TREM                                  *
*              R2 - ADDRESS OF TIMER QUEUE ELEMENT                   *
*              R3 - ADDRESS OF SIX HOUR PSEUDO CLOCK                 *
*              R4 TO R7 - WORK REGISTERS                             *
*              R8 TO RB - NOT USED                                   *
*              RC - BASE REGISTER                                    *
*              RD - ADDRESS OF PREVIOUS SAVE AREA                    *
*              RE - RETURN ADDRESS                                   *
*              RF - RETURN CODE                                      *
*              F0 - WORK REGISTER                                    *
*                                                                    *
* NOTES....... SEE THE NOTES ON CPUREM.                              *
*                                                                    *
* AUTHOR...... G. LADNER.                                            *
*                                                                    *
* DATE........ 28/09/73.                                             *
*                                                                    *
**********************************************************************
*--------------------------------------------------------------------*
**********************************************************************
CPUTIM   CSECT
**********************************************************************
*        REGISTER EQUATES                                            *
**********************************************************************
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
```

```
R6       EQU   6
R7       EQU   7
RC       EQU   12
RD       EQU   13
RE       EQU   14
RF       EQU   15
F0       EQU   0
***********************************************************************
*        PROLOGUE                                                     *
***********************************************************************
         B     12(RF)                  BRANCH AROUND ID
         DC    AL1(6)                  ID LENGTH
         DC    CL6'CPUTIM'             IDENTIFIER
         DC    X'00'                   FOR HALFWORD ALIGNMENT
         STM   RE,RC,12(RD)            SAVE REGISTERS
         BALR  RC,0                    ADDRESS OF NEXT INSTRUCTION IN RC
         USING *,RC                    RC IS BASE REGISTER
         L     R1,0(R1)                R1 CONTAINS ADDRESS OF TCPU
***********************************************************************
*        DETERMINE T=(TOX-(SHPC-TIMER))                               *
***********************************************************************
         L     R2,16                   R2 CONTAINS ADDRESS OF CVT
         L     R3,88(R2)               R3 CONTAINS ADDRESS OF SHPC
         L     R2,0(R2)                R2 CONTAINS ADDRESS OF TCB WORDS
         L     R2,4(R2)                R2 CONTAINS ADDRESS OF CURRENT TCB
         L     R2,124(R2)              R2 CONTAINS ADDRESS OF JOB STEP TCB
         L     R2,132(R2)              R2 CONTAINS ADDRESS OF INITIATOR TCB
         L     R2,120(R2)              R2 CONTAINS ADDRESS OF TIMER ELEMENT
CPULPC   L     R5,0(R3)                R5 CONTAINS VALUE IN SHPC
         L     R6,12(R2)               R6 CONTAINS TOX
         L     R4,80                   R4 CONTAINS VALUE IN TIMER
         L     R7,0(R3)                R7 CONTAINS VALUE IN SHPC
         CR    R5,R7                   HAS AN INTERRUPT MODIFIED SHPC
         BNE   CPULPC                  YES, TRY AGAIN
         SRL   R4,1                    CONVERT TIMER UNITS TO SHPC UNITS
         SR    R5,R4                   R5 CONTAINS (SHPC-TIMER)
         SR    R6,R5                   R6 CONTAINS T=(TOX-(SHPC-TIMER))
***********************************************************************
*        CONVERT T TO FLOATING POINT                                  *
***********************************************************************
         SR    R7,R7                   ZERO R7
         SRDL  R6,8                    R6,R7 CONTAIN T FRACTION
         STM   R6,R7,CPUDFL            STORE T FRACTION
         MVI   CPUDFL,X'48'            CPUDFL CONTAINS TREM/26.04166E-6
***********************************************************************
*        CONVERT THE CONTENTS OF CPUDFL TO SECONDS                    *
***********************************************************************
         LD    F0,CPUFAC               F0 CONTAINS 26.04166E-6
         MD    F0,CPUDFL               F0 CONTAINS TREM
***********************************************************************
*        FIRST TIME TEST                                              *
***********************************************************************
         CLI   CPUFLG,X'FF'            TEST FOR FIRST TIME THROUGH
         BE    CPUNXT                  NOT FIRST
         MVI   CPUFLG,X'FF'            SET FLAG FOR NEXT TIME
         STE   F0,CPUSTR               STORE REMAINING CPU TIME
```

```
         LE    F0,CPUSF0              ZERO FOR FIRST TIME
CPUSTE   STE   F0,0(R1)               STORE CPU TIME AT TCPU
************************************************************************
*        EPILOGUE                                                       *
************************************************************************
         LM    RE,RC,12(RD)           RESTORE REGISTERS
         SR    RF,RF                  RC=0
         BR    RE                     RETURN
************************************************************************
*        DETERMINE TCPU                                                 *
************************************************************************
CPUNXT   SE    F0,CPUSTR              NEGATIVE VALUE OF TCPU
         LPER  F0,F0                  POSITIVE VALUE OF TCPU
         B     CPUSTE                 GO STORE TCPU
************************************************************************
*        CONSTANTS, STORAGE AREAS                                       *
************************************************************************
CPUFAC   DC    D'26.04166E-6'         UNIT OF TIME FOR SHPC (SECS.)
CPUDFL   DS    D                      INTERMEDIATE STORAGE FOR TREM
CPUSF0   DC    E'0.0'                 SHORT FLOATING POINT ZERO
CPUSTR   DS    E                      STORAGE FOR TREM
CPUFLG   DC    X'00'                  FIRST TIME FLAG
         END   CPUTIM
```

```
AEB       TITLE 'CPUREM - ROUTINE TO DETERMINE REMAINING CPU TIME'
*******************************************************************************
*                                                                             *
*          *****    ******    *     *   ******    *******   *     *           *
*         *     *   *     *   *     *   *     *   *         **   **           *
*         *         *     *   *     *   *     *   *         * * * *           *
*         *         ******    *     *   ******    *****     *  *  *           *
*         *         *         *     *   *   *     *         *     *           *
*         *     *   *         *     *   *    *    *         *     *           *
*          *****    *          *****    *     *   *******   *     *           *
*                                                                             *
*******************************************************************************
*                                                                             *
* USAGE....... CALL CPUREM (TREM)       (FORTRAN LINKAGE CONVENTIONS)         *
*                                                                             *
* FUNCTIONS... CPUREM MAY BE INVOKED BY A JOB STEP TASK OR SUBTASK.           *
*              THE VALUE OF TREM AFTER THE CALL GIVES THE AMOUNT OF           *
*              CPU TIME STILL AVAILABLE TO THE JOB I.E. THE VALUE             *
*              SPECIFIED IN THE 'TIME' PARAMETER ON THE JOB CARD              *
*              MINUS THE CPU TIME OF THE CURRENT STEP AND ALL PREVIOUS*
*              STEPS.                                                         *
*                                                                             *
* PARAMETERS.. ON ENTRY R1 IS ASSUMED TO POINT TO AN ALIGNED FULLWORD         *
*              CONTAINING THE ADDRESS OF TREM. TREM IS ALSO ASSUMED           *
*              TO BE A FULLWORD ALIGNED ON A FULLWORD BOUNDARY.               *
*              ON EXIT TREM CONTAINS THE REMAINING CPU TIME IN                *
*              SECONDS EXPRESSED AS A SHORT FLOATING POINT NUMBER.            *
*                                                                             *
* ATTRIBUTES.. SERIALLY REUSABLE.                                             *
*                                                                             *
* REGISTER.... R0 - NOT USED                                                  *
* USAGE        R1 - ADDRESS OF TREM                                           *
*              R2 - ADDRESS OF TIMER QUEUE ELEMENT                            *
*              R3 - ADDRESS OF SIX HOUR PSEUDO CLOCK                          *
*              R4 TO R7 - WORK REGISTERS                                      *
*              R8 TO RB - NOT USED                                            *
*              RC - BASE REGISTER                                             *
*              RD - ADDRESS OF PREVIOUS SAVE AREA                             *
*              RE - RETURN ADDRESS                                            *
*              RF - RETURN CODE                                               *
*              F0 - WORK REGISTER                                             *
*                                                                             *
* NOTES....... 1. CPUREM MAY BE INVOKED BY LANGUAGES OTHER THAN               *
*                 FORTRAN. PARAMETERS ARE PASSED AS DESCRIBED ABOVE           *
*                 (SEE 'PARAMETERS') AND NORMAL OS LINKAGE CONVENTIONS*
*                 ARE FOLLOWED.                                               *
*              2. THE INTERVAL TIMER (SYSTEM/370 MODEL 155) IS                *
*                 DECREMENTED EVERY 1/300TH OF A SECOND BY INTERNAL           *
*                 TIMING CIRCUITS. THIS IS A HARDWARE LIMITATION              *
*                 WHICH COULD RESULT IN ERRORS OF UP TO 2/300THS OF           *
*                 A SECOND IN TIMING AN INTERVAL.                             *
*              3. INTERRUPTS ARE HANDLED AT THE EXPENSE (IN TERMS OF          *
*                 CPU TIME) OF THE INTERRUPTED PROGRAM. THIS CAN              *
*                 RESULT IN DIFFERENT CPU TIMES FOR IDENTICAL                 *
*                 INSTRUCTION SEQUENCES.                                      *
*                                                                             *
```

```
* AUTHOR...... G. LADNER.                                             *
*                                                                     *
* DATE......... 15/05/73.                                             *
*                                                                     *
***********************************************************************
*---------------------------------------------------------------------*
***********************************************************************
CPUREM    CSECT
***********************************************************************
*         REGISTER EQUATES                                            *
***********************************************************************
R1        EQU   1
R2        EQU   2
R3        EQU   3
R4        EQU   4
R5        EQU   5
R6        EQU   6
R7        EQU   7
RC        EQU   12
RD        EQU   13
RE        EQU   14
RF        EQU   15
F0        EQU   0
***********************************************************************
*         PROLOGUE                                                    *
***********************************************************************
          B     12(RF)              BRANCH AROUND ID
          DC    AL1(6)              ID LENGTH
          DC    CL6'CPUREM'         IDENTIFIER
          DC    X'00'               FOR HALFWORD ALIGNMENT
          STM   RE,RC,12(RD)        SAVE REGISTERS
          BALR  RC,0                ADDRESS OF NEXT INSTRUCTION IN RC
          USING *,RC                RC IS BASE REGISTER
          L     R1,0(R1)            R1 CONTAINS ADDRESS OF TREM
***********************************************************************
*         DETERMINE T=(TOX-(SHPC-TIMER))                              *
***********************************************************************
          L     R2,16               R2 CONTAINS ADDRESS OF CVT
          L     R3,88(R2)           R3 CONTAINS ADDRESS OF SHPC
          L     R2,0(R2)            R2 CONTAINS ADDRESS OF TCB WORDS
          L     R2,4(R2)            R2 CONTAINS ADDRESS OF CURRENT TCB
          L     R2,124(R2)          R2 CONTAINS ADDRESS OF JOB STEP TCB
          L     R2,132(R2)          R2 CONTAINS ADDRESS OF INITIATOR TCB
          L     R2,120(R2)          R2 CONTAINS ADDRESS OF TIMER ELEMENT
CPULPC    L     R5,0(R3)            R5 CONTAINS VALUE IN SHPC
          L     R6,12(R2)           R6 CONTAINS TOX
          L     R4,80               R4 CONTAINS VALUE IN TIMER
          L     R7,0(R3)            R7 CONTAINS VALUE IN SHPC
          CR    R5,R7               HAS AN INTERRUPT MODIFIED SHPC
          BNE   CPULPC              YES, TRY AGAIN
          SRL   R4,1                CONVERT TIMER UNITS TO SHPC UNITS
          SR    R5,R4               R5 CONTAINS (SHPC-TIMER)
          SR    R6,R5               R6 CONTAINS T=(TOX-(SHPC-TIMER))
***********************************************************************
*         CONVERT T TO FLOATING POINT                                 *
***********************************************************************
```

```
         SR     R7,R7                    ZERO R7
         SRDL   R6,8                     R6,R7 CONTAIN T FRACTION
         STM    R6,R7,CPUDFL             STORE T FRACTION
         MVI    CPUDFL,X'48'             CPUDFL CONTAINS TREM/26.04166E-6
***********************************************************************
*        CONVERT THE CONTENTS OF CPUDFL TO SECONDS                    *
7-667777 $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
         LD     F0,CPUFAC                F0 CONTAINS 26.04166E-6
         MD     F0,CPUDFL                F0 CONTAINS TREM
         STE    F0,0(R1)                 STORE F0 AT TREM
***********************************************************************
*        EPILOGUE                                                     *
***********************************************************************
         LM     RE,RC,12(RD)             RESTORE REGISTERS
         SR     RF,RF                    RC=0
         BR     RE                       RETURN
***********************************************************************
*        CONSTANTS, STORAGE AREA                                      *
***********************************************************************
CPUFAC   DC     D'26.04166E-6'           UNIT OF TIME FOR SHPC (SECS.)
CPUDFL   DS     D                        INTERMEDIATE STORAGE FOR TREM
         END    CPUREM
```

```
DMENSI   CSECT
*
* STEL AREA AANGEVRA BESKIKBAAR AAN FORTRAN.PROGRAMME
*
*   1STE ADRES AANGEGEE IS DIE VAN 'N SUBROETINE WAARAAN
*    DMENSI BEHEER OORGEE NADAT AREA VERKRY IS
*    DAARNA VOLG ARRAY-ADRES EN -GROOTTE IN PARE
*
*
*
*
*   REGISTERS GEBRUIK
*        3   ADRES VAN ADRESLYS IN MAIN
*        4   ADRES VAN ADRESLYS IN DMENSI
*        5   MAKS AANTAL ARRAYS
*        6   AANTAL ARRAYS
*        7   AREA AANGEVRA IN BYTES
*        8   ALGEMEEN
*        9   AREA-ADRESSE VIR FREEMAIN
*        10
*        11
*        12  BASIES
*
         ISEQ  77,80
A010     STM   14,12,12(13)
         BALR  12,0
         USING *,12
         LR    4,13
         ST    13,DSAVE+4
         LA    13,DSAVE
         ST    13,8(0,4)
         LR    3,1                        ADRES VAN ADRESLYS
         L     8,0(0,3)         ADRES VAN SUBROETINE SOOS AANGEGEE
         L     8,0(0,8)
         LA    8,0(8,0)         00 IN HOE ORDE BYTE
         ST    8,ASUBR
         LA    3,4(0,3)
         LA    4,ADLYS
         LA    9,FREE
         NI    A250+1,X'0F'
         NC    A300+4(4),=X'FF000000'     ABEND-KODE
         XR    6,6
A060     LA    5,100            VOORSIENING VIR 100 ARRAYS
A070     LA    6,1(0,6)         TEL AANTAL ARRAYS
         TM    0(3),X'80'       ARRAY ADRES
         BZ    A080
         A     6,=F'2000'       GEEN GROOTTE AANGEGEE - FOUT 4
         B     A280
A080     TM    4(3),X'80'       GROOTTE-ADRES
         BZ    A090
         OI    A250+1,X'F0'     EINDE VAN LYS
         NI    4(3),X'7F'
A090     L     7,4(0,3)
         L     7,0(0,7)         AREA IN WOORDE
         SLL   7,2              AREA IN BYTES
         ST    7,A200+4         PLAAS AREA IN GETMAIN
```

```
         ST      7,4(9)
A200     GETMAIN  EC,LV=2000,A=GETMADR
         C       15,=F'0'           AREA TOEGEKEN
         BE      A210
         A       6,=F'1000'        )TEL 1000 BY AS GEEN
         B       A280              )AREA BESKIKBAAR
A210    L       8,GETMADR
         LA      8,0(8,0)
         ST      8,0(4)            WYSIG ARRAY-ADRES IN LYS
         ST      8,0(9)
A250    BC      0,A350
         LA      3,8(0,3)
         LA      4,4(0,4)
         LA      9,8(0,9)
         BCT     5,A070
         TM      A250+1,X'F0'       PRESIES MAKS AANTAL
         BO      A350
         A       6,=F'2000'         TE VEEL ARRAYS AANGEGEE
A280    O        6,A300+4
         ST      6,A300+4
A300    ABEND 33,DUMP
A350    MVI     8(9),X'FF'
         LA      1,ADLYS
         OI      0(4),X'80'       END VAN LYS
         L       15,ASUBR
         BALR    14,15
         LA      5,40
         LA      9,FREE
A360    CLI     0(9),X'FF'       EINDE VAN LYS
         BE      A370
         L       0,4(9)
         L       1,0(9)
         LA      9,8(0,9)
         FREEMAIN  R,LV=(0),A=(1)
         BCT     5,A360
A370    L      13,DSAVE+4        RETURN
         LM      14,12,12(13)
         MVI     12(13),X'FF'
         BR      14
         LTORG
        DS          0F
DSAVE   DS          18F
ADLYS   DS         100F
GETMADR DS          F
ASUBR   DS          F
FREE    DS         200F
         DS           CL1
         END
```

BYLAE C

NUMERIESE VOORBEELDE

1. Die Stip van Eksperimentele Data Deur die Grafiektrekker

In die matriks van eksperimentele datapunte E(I,J) wat op bladsy C3 verskyn, stel die ry-indeks I die I-de eksperiment en die kolomindeks J die J-de waarneming voor. Ons noem die waarnemings X1, X2, ---. Gestel X3 is die afhanklike veranderlike. Verskillende eksperimente is gedoen waar die waarde van X1 bykans konstant gehou is. Ons moet dus 'n uitdrukking vir X3 in terme van die onafhanklike veranderlikes vind. Deur die waardes van X3 teen die verskillende veranderlikes te stip, kan ons 'n idee vorm van hoe die punte versprei lê. Op bladsy C7 verskyn 'n grafiek van X3 as 'n funksie van X2. Dit is duidelik dat een punt (nl. die punt gemerk 4) ver van die ander punte lê. Uit die data op bladsy C5 blyk dit dat die waarnemings in eksperimente 3 en 4 dieselfde is behalwe vir die waarneming X2. Dit kan waarskynlik die gevolg wees van 'n foutiewe waarneming of 'n ponsfout.

'n Lys van die subroetine en data ingelees verskyn op bl. C2 en C3, terwyl die resultate op bl. C4 uitgedruk is. Die eksperimentele data is in 'n permanente lêer geskryf.

```
  PASDIM VIR VOORBEELD 1
  ------ --- --------- -
C
      SUBROUTINE PASDIM
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(1),A(1)
      COMMON/DIM/IR,IK,IA
C
C   BESKRYWING PARAMETERS
C
C   IR     = MAKSIMUM AANTAL DATAPUNTE WAT IN DIE PASSINGS GEBRUIK WORD,
C              EN MOET GROTER AS NUL WEES.
C   IK     = TOTALE AANTAL VERANDERLIKES, INGELEES PLUS GEGENEREER,
C              MET   0 < IK < 50 .
C   IA     = AANTAL A-WAARDES WAT IN FUNKSIE PLUS A-WAARDES WAT LINEER
C              VOORKOM, MET   0 < IA < 50 .
C
      IR=40
      IK=11
      IA=9
      RETURN
      ENTRY PASAF(X,A,F)
      ENTRY PASGEN(X,M)
      DATA NOM/1/
C
C   IN PASGEN KAN NOG VERANDERLIKES GEGENEREER OF VERANDER WORD
C   M+1 IS EERSTE BESKIKBARE KOLOM.
C
      X(M+1)=NOM
      NOM=NOM+1
      RETURN
      ENTRY PASFUN(X,A,FPAS)
C
C   IN PASFUN WORD DIE FUNKSIE GEDEFINIEER
C
C    NL=0, DUS NIE NODIG OM FUNKSIE TE DEFINEER.
C
      RETURN
      END
```

DATA VIR VOORBEELD 1

100.2	0.6	3.05	50.3	0.054	0.21	0.8	0.01
100.0	1.	2.5	40.2	0.055	0.03	1.	0.03
400.0	1.	6.5	40.2	0.055	0.03	1.	0.03
400.0	4.	6.5	40.2	0.055	0.03	1.	0.03
99.8	1.45	2.1	10.0	0.059	0.25	0.8	0.05
100.0	2.2	1.2	4.02	0.065	0.01	1.0	0.01
99.9	1.1	2.45	20.1	0.058	0.14	0.9	0.04
100.0	2.7	0.75	6.	0.042	0.05	1.0	0.05
100.1	3.1	0.25	8.30	0.032	0.16	0.9	0.06
200.5	0.6	4.4	3.6	0.042	0.53	0.5	0.03
200.4	1.	3.7	8.3	0.052	0.42	0.6	0.02
200.3	1.25	3.32	7.6	0.031	0.31	0.7	0.01
200.1	1.51	3.05	3.2	0.025	0.12	0.9	0.02
199.6	1.83	2.75	6.3	0.031	0.61	0.4	0.01
199.8	2.65	1.78	7.4	0.025	0.83	0.2	0.03
200.5	3.1	1.3	8.5	0.031	0.56	0.5	0.06
200.4	3.6	0.9	3.2	0.052	0.47	0.6	0.07
300.3	0.8	5.7	6.1	0.061	0.34	0.7	0.04
300.2	1.	5.3	7.5	0.071	0.23	0.8	0.03
300.1	1.42	4.7	8.4	0.085	0.12	0.9	0.02
100.1	1.75	1.75	5.01	0.063	0.16	0.9	0.06
300.0	1.81	4.05	9.3	0.093	0.01	1.	0.01
300.5	2.25	3.5	10.5	0.01	0.52	0.5	0.02
300.3	2.7	3.0	10.6	0.02	0.36	0.7	0.06
100.0	1.	2.9	40.2	0.055	0.03	1.	0.03
100.1	1.1	2.15	20.1	0.058	0.14	0.9	0.04
100.0	1.9	1.6	4.02	0.065	0.01	1.0	0.01
199.6	2.05	2.25	6.3	0.031	0.61	0.4	0.01
300.1	3.25	2.35	74.	0.025	0.18	0.9	0.08
300.4	3.85	2.02	8.3	0.083	0.47	0.6	0.07
299.5	4.35	1.45	9.5	0.073	0.56	0.5	0.06
400.3	2.	5.55	6.3	0.082	0.35	0.7	0.05
400.2	2.5	4.9	7.5	0.032	0.24	0.8	0.04
400.6	3.1	4.25	8.6	0.053	0.62	0.4	0.02
399.8	3.7	3.6	7.6	0.052	0.88	0.2	0.08
399.9	4.25	2.83	4.3	0.061	0.99	0.1	0.09
400.2	4.62	2.53	5.8	0.075	0.81	0.8	0.01

```
*    EINDE DATAPUNTE
 &LEES
 PASE='VOORBEELD 1',
 IDRUK=1,
 OPSK='X1','X2','X3','X4','X5','X6','X7','X8','X9','X10','X11','X12','X13',
 M=8,
 FORM='(8F10.0)',
 NN=3,IX=1,2,3,
 IDAT=1,IDENT=9,
 IPLOT=1,IPUNT=1,
 XLEN=6.,YLEN=7.,
  XBEG=0.,XEND=6.,YBEG=0.,YEND=7.,
 ILYN=0,NSOEK=0,IFIT=0,
 &END
```

DATUM 77-09-22 TYD 17-22-45

** INVOER **

IA	=	9	IPR	=	1	IFIT	=	0	IDENT	=	9	ITSTK	=	0	MM	=	9	NPAS	= 1
IR	=	40	ICON	=	1	ILYN	=	0	IDRUK	=	1	KA	=	0	NL	=	0	NDIGO	= 100
IK	=	11	ICOR	=	1	IPEN	=	1	IPLOT	=	1	KX	=	0	NN	=	3	NSOEK	= 0
IT	=	0	IDAT	=	1	IXAS	=	2	IPONS	=	0	LEESX	=	1	NGO	=	5	NUM	= 2
IAS	=	1	IDES	=	1	IYAS	=	3	IPUNT	=	1	M	=	8	NSK	=	1	N	= 37

IX = 1 2 3
IDXP = 2 2 2
IDG = 2 2 2
EPS = 1.00D-06 TYD = 1.00E 01 YDUIM = 1.10E 01 FNU = 1.000D-04
XBEG = 0.0 XEND = 6.00 YBEG = 0.0 YEND = 7.00 XLEN = 6.00 YLEN = 7.00

XP(I) = 0.0
FORM =(8F10.0)

DATALYS SOOS OP KAARTE GEPONS

1	100.2	0.6	3.05	50.3	0.054	0.21	0.8	0.01
2	100.0	1.	2.5	40.2	0.055	0.03	1.	0.03
3	400.0	1.	6.5	40.2	0.055	0.03	1.	0.03
4	400.0	4.	6.5	40.2	0.055	0.03	1.	0.03
5	99.8	1.45	2.1	10.0	0.059	0.25	0.8	0.05
6	100.0	2.2	1.2	4.02	0.065	0.01	1.0	0.01
7	99.9	1.1	2.45	20.1	0.058	0.14	0.9	0.04
8	100.0	2.7	0.75	6.	0.042	0.05	1.0	0.05
9	100.1	3.1	0.25	8.30	0.032	0.16	0.9	0.06
10	200.5	0.6	4.4	3.6	0.042	0.53	0.5	0.03
11	200.4	1.	3.7	8.3	0.052	0.42	0.6	0.02
12	200.3	1.25	3.32	7.6	0.031	0.31	0.7	0.01
13	200.1	1.51	3.05	3.2	0.025	0.12	0.9	0.02
14	199.6	1.83	2.75	6.3	0.031	0.61	0.4	0.01
15	199.8	2.65	1.78	7.4	0.025	0.83	0.2	0.03
16	200.5	3.1	1.3	8.5	0.031	0.56	0.5	0.06
17	200.4	3.6	0.9	3.2	0.052	0.47	0.6	0.07
18	300.3	0.8	5.7	6.1	0.061	0.34	0.7	0.04
19	300.2	1.	5.3	7.5	0.071	0.23	0.8	0.03
20	300.1	1.42	4.7	8.4	0.085	0.12	0.9	0.02
21	100.1	1.75	1.75	5.01	0.063	0.16	0.9	0.06
22	300.0	1.81	4.05	9.3	0.093	0.01	1.	0.01
23	300.5	2.25	3.5	10.5	0.01	0.52	0.5	0.02
24	300.3	2.7	3.0	10.6	0.02	0.36	0.7	0.06
25	100.0	1.	2.9	40.2	0.055	0.03	1.	0.03
26	100.1	1.1	2.15	20.1	0.058	0.14	0.9	0.04
27	100.0	1.9	1.6	4.02	0.065	0.01	1.0	0.01
28	199.6	2.05	2.25	6.3	0.031	0.61	0.4	0.01
29	300.1	3.25	2.35	74.	0.025	0.18	0.9	0.08
30	300.4	3.85	2.02	8.3	0.083	0.47	0.6	0.07
31	299.5	4.35	1.45	9.5	0.073	0.56	0.5	0.06
32	400.3	2.	5.55	6.3	0.082	0.35	0.7	0.05
33	400.2	2.5	4.9	7.5	0.032	0.24	0.8	0.04
34	400.6	3.1	4.25	8.6	0.053	0.62	0.4	0.02
35	399.8	3.7	3.6	7.6	0.052	0.88	0.2	0.08
36	399.9	4.25	2.83	4.3	0.061	0.99	0.1	0.09
37	400.2	4.62	2.53	5.8	0.075	0.81	0.8	0.01

** RESULTATE **

C 4

NR.	X1	X2	X3	X4	X5	X6	X7	X8	X9
1	100.2000	0.6000	3.0500	50.3000	0.0540	0.2100	0.8000	0.0100	1.0000
2	100.0000	1.0000	2.5000	40.2000	0.0550	0.0300	1.0000	0.0300	2.0000
3	400.0000	1.0000	6.5000	40.2000	0.0550	0.0300	1.0000	0.0300	3.0000
4	400.0000	4.0000	6.5000	40.2000	0.0550	0.0300	1.0000	0.0300	4.0000
5	99.8000	1.4500	2.1000	10.0000	0.0590	0.2500	0.8000	0.0500	5.0000
6	100.0000	2.2000	1.2000	4.0200	0.0650	0.0100	1.0000	0.0100	6.0000
7	99.9000	1.1000	2.4500	20.1000	0.0580	0.1400	0.9000	0.0400	7.0000
8	100.0000	2.7000	0.7500	6.0000	0.0420	0.0500	1.0000	0.0500	8.0000
9	100.1000	3.1000	0.2500	8.3000	0.0320	0.1600	0.9000	0.0600	9.0000
10	200.5000	0.6000	4.4000	3.6000	0.0420	0.5300	0.5000	0.0300	10.0000
11	200.4000	1.0000	3.7000	8.3000	0.0520	0.4200	0.6000	0.0200	11.0000
12	200.3000	1.2500	3.3200	7.6000	0.0310	0.3100	0.7000	0.0100	12.0000
13	200.1000	1.5100	3.0500	3.2000	0.0250	0.1200	0.9000	0.0200	13.0000
14	199.6000	1.8300	2.7500	6.3000	0.0310	0.6100	0.4000	0.0100	14.0000
15	199.8000	2.6500	1.7800	7.4000	0.0250	0.8300	0.2000	0.0300	15.0000
16	200.5000	3.1000	1.3000	8.5000	0.0310	0.5600	0.5000	0.0600	16.0000
17	200.4000	3.6000	0.9000	3.2000	0.0520	0.4700	0.6000	0.0700	17.0000
18	300.3000	0.8000	5.7000	6.1000	0.0610	0.3400	0.7000	0.0400	18.0000
19	300.2000	1.0000	5.3000	7.5000	0.0710	0.2300	0.8000	0.0300	19.0000
20	300.1000	1.4200	4.7000	8.4000	0.0850	0.1200	0.9000	0.0200	20.0000
21	100.1000	1.7500	1.7500	5.0100	0.0630	0.1600	0.9000	0.0600	21.0000
22	300.0000	1.8100	4.0500	9.3000	0.0930	0.0100	1.0000	0.0100	22.0000
23	300.5000	2.2500	3.5000	10.5000	0.0100	0.5200	0.5000	0.0200	23.0000
24	300.3000	2.7000	3.0000	10.6000	0.0200	0.3600	0.7000	0.0600	24.0000
25	100.0000	1.0000	2.9000	40.2000	0.0550	0.0300	1.0000	0.0300	25.0000
26	100.1000	1.1000	2.1500	20.1000	0.0580	0.1400	0.9000	0.0400	26.0000
27	100.0000	1.9000	1.6000	4.0200	0.0650	0.0100	1.0000	0.0100	27.0000
28	199.6000	2.0500	2.2500	6.3000	0.0310	0.6100	0.4000	0.0100	28.0000
29	300.1000	3.2500	2.3500	74.0000	0.0250	0.1800	0.9000	0.0800	29.0000
30	300.4000	3.8500	2.0200	8.3000	0.0830	0.4700	0.6000	0.0700	30.0000
31	299.5000	4.3500	1.4500	9.5000	0.0730	0.5600	0.5000	0.0600	31.0000
32	400.3000	2.0000	5.5500	6.3000	0.0820	0.3500	0.7000	0.0500	32.0000
33	400.2000	2.5000	4.9000	7.5000	0.0320	0.2400	0.8000	0.0400	33.0000
34	400.6000	3.1000	4.2500	8.6000	0.0530	0.6200	0.4000	0.0200	34.0000
35	399.8000	3.7000	3.6000	7.6000	0.0520	0.8800	0.2000	0.0800	35.0000
36	399.9000	4.2500	2.8300	4.3000	0.0610	0.9900	0.1000	0.0900	36.0000
37	400.2000	4.6200	2.5300	5.8000	0.0750	0.8100	0.8000	0.0100	37.0000
GEMM.	237.9405	2.2186	3.0508	14.2527	0.0517	0.3349	0.7189	0.0376	19.0000
STD AFW.	113.9422	1.1674	1.5877	16.0753	0.0201	0.2746	0.2559	0.0233	10.8244

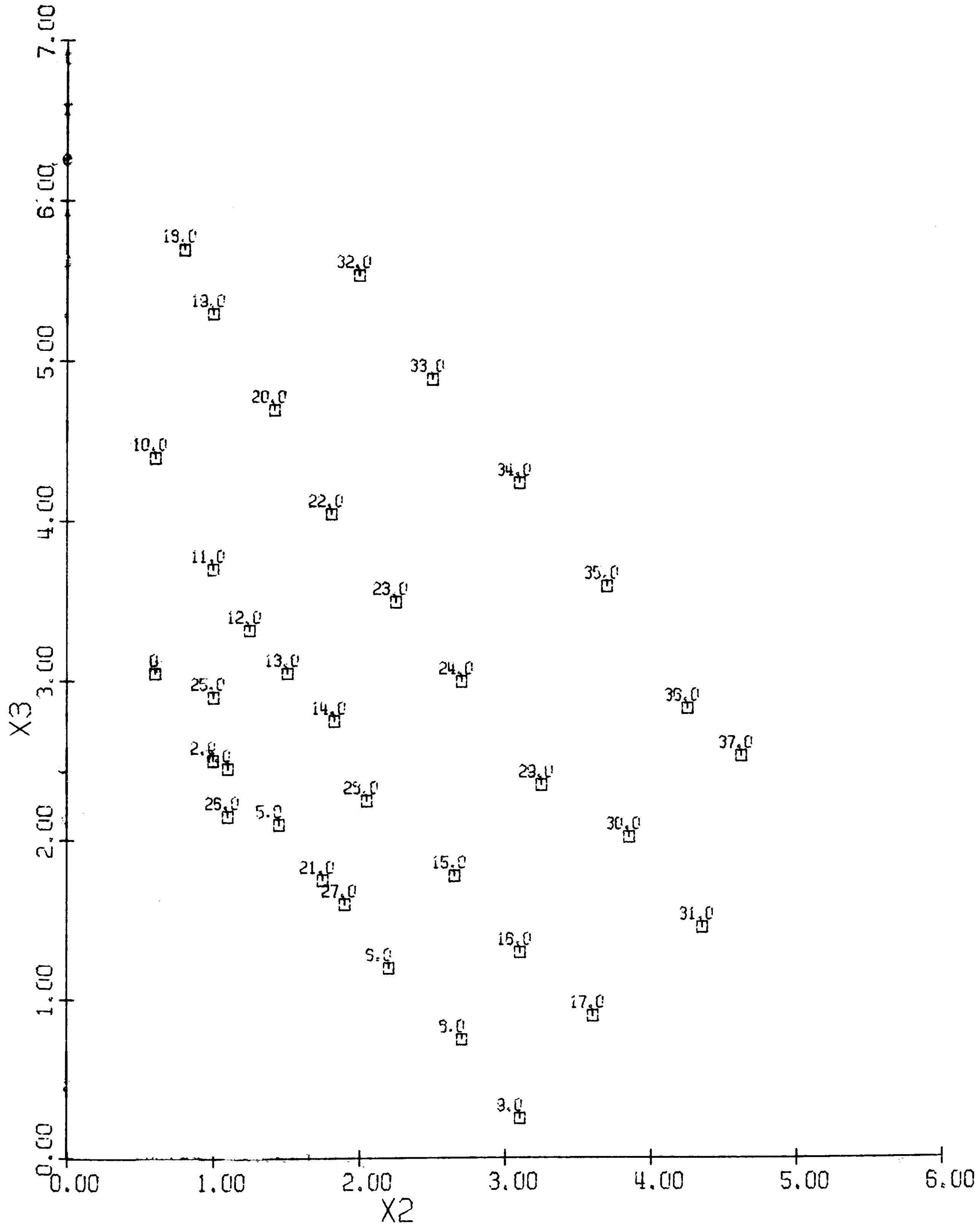
MATRIKS MET KORRELASIEKOEFFISIENTE

I	1	2	3	4	5	6	7	8	9
1	1.00	0.48	0.65	-0.03	0.15	0.43	-0.35	0.18	0.54
2	0.48	1.00	-0.29	-0.13	0.01	0.51	-0.37	0.50	0.52
3	0.65	-0.29	1.00	0.16	0.19	-0.09	0.06	-0.24	0.01
4	-0.03	-0.13	0.16	1.00	-0.10	-0.38	0.38	0.06	-0.24
5	0.15	0.01	0.19	-0.10	1.00	-0.17	0.26	-0.02	0.17
6	0.43	0.51	-0.09	-0.38	-0.17	1.00	-0.94	0.27	0.51
7	-0.35	-0.37	0.06	0.38	0.26	-0.94	1.00	-0.28	-0.41
8	0.18	0.50	-0.24	0.06	-0.02	0.27	-0.28	1.00	0.30
9	0.54	0.52	0.01	-0.24	0.17	0.51	-0.41	0.30	1.00

X1	X2	X3	X4	X5	X6	X7	X8	X9
100.20	0.60000	3.0500	50.300	0.54000D-01	0.21000	0.80000	0.10000D-01	1.0000
100.00	1.0000	2.5000	40.200	0.55000D-01	0.30000D-01	1.0000	0.30000D-01	2.0000
400.00	1.0000	6.5000	40.200	0.55000D-01	0.30000D-01	1.0000	0.30000D-01	3.0000
400.00	4.0000	6.5000	40.200	0.55000D-01	0.30000D-01	1.0000	0.30000D-01	4.0000
99.800	1.4500	2.1000	10.000	0.59000D-01	0.25000	0.80000	0.50000D-01	5.0000
100.00	2.2000	1.2000	4.0200	0.65000D-01	0.10000D-01	1.0000	0.10000D-01	6.0000
99.900	1.1000	2.4500	20.100	0.58000D-01	0.14000	0.90000	0.40000D-01	7.0000
100.00	2.7000	0.75000	6.0000	0.42000D-01	0.50000D-01	1.0000	0.50000D-01	8.0000
100.10	3.1000	0.25000	8.3000	0.32000D-01	0.16000	0.90000	0.60000D-01	9.0000
200.50	0.60000	4.4000	3.6000	0.42000D-01	0.53000	0.50000	0.30000D-01	10.000
200.40	1.0000	3.7000	8.3000	0.52000D-01	0.42000	0.60000	0.20000D-01	11.000
200.30	1.2500	3.3200	7.6000	0.31000D-01	0.31000	0.70000	0.10000D-01	12.000
200.10	1.5100	3.0500	3.2000	0.25000D-01	0.12000	0.90000	0.20000D-01	13.000
199.60	1.8300	2.7500	6.3000	0.31000D-01	0.61000	0.40000	0.10000D-01	14.000
199.80	2.6500	1.7800	7.4000	0.25000D-01	0.83000	0.20000	0.30000D-01	15.000
200.50	3.1000	1.3000	8.5000	0.31000D-01	0.56000	0.50000	0.60000D-01	16.000
200.40	3.6000	0.90000	3.2000	0.52000D-01	0.47000	0.60000	0.70000D-01	17.000
300.30	0.80000	5.7000	6.1000	0.61000D-01	0.34000	0.70000	0.40000D-01	18.000
300.20	1.0000	5.3000	7.5000	0.71000D-01	0.23000	0.80000	0.30000D-01	19.000
300.10	1.4200	4.7000	8.4000	0.85000D-01	0.12000	0.90000	0.20000D-01	20.000
100.10	1.7500	1.7500	5.0100	0.63000D-01	0.16000	0.90000	0.60000D-01	21.000
300.00	1.8100	4.0500	9.3000	0.93000D-01	0.10000D-01	1.0000	0.10000D-01	22.000
300.50	2.2500	3.5000	10.500	0.10000D-01	0.52000	0.50000	0.20000D-01	23.000
300.30	2.7000	3.0000	10.600	0.20000D-01	0.36000	0.70000	0.60000D-01	24.000
100.00	1.0000	2.9000	40.200	0.55000D-01	0.30000D-01	1.0000	0.30000D-01	25.000
100.10	1.1000	2.1500	20.100	0.58000D-01	0.14000	0.90000	0.40000D-01	26.000
100.00	1.9000	1.6000	4.0200	0.65000D-01	0.10000D-01	1.0000	0.10000D-01	27.000
199.60	2.0500	2.2500	6.3000	0.31000D-01	0.61000	0.40000	0.10000D-01	28.000
300.10	3.2500	2.3500	74.000	0.25000D-01	0.18000	0.90000	0.80000D-01	29.000
300.40	3.8500	2.0200	8.3000	0.83000D-01	0.47000	0.60000	0.70000D-01	30.000
299.50	4.3500	1.4500	9.5000	0.73000D-01	0.56000	0.50000	0.60000D-01	31.000
400.30	2.0000	5.5500	6.3000	0.82000D-01	0.35000	0.70000	0.50000D-01	32.000
400.20	2.5000	4.9000	7.5000	0.32000D-01	0.24000	0.80000	0.40000D-01	33.000
400.60	3.1000	4.2500	8.6000	0.53000D-01	0.62000	0.40000	0.20000D-01	34.000
399.80	3.7000	3.6000	7.6000	0.52000D-01	0.88000	0.20000	0.80000D-01	35.000
399.90	4.2500	2.8300	4.3000	0.61000D-01	0.99000	0.10000D 00	0.90000D-01	36.000
400.20	4.6200	2.5300	5.8000	0.75000D-01	0.81000	0.80000	0.10000D-01	37.000

TOTALE TYD IN REKENAAR = 30.34 SEK.
WERKLIKE TYD IN REKENAAR = 2.86 SEK.

VOORBEELD 1



2. Die Voortbrenging van Veranderlikes en die Passing van 'n Funksie

In hierdie voorbeeld word (i) 'n korrelasietabel uitgedruk (ii) verdere veranderlikes voortgebring en (iii) 'n funksie met lineêre en nie-lineêre terme in a_j gepas. Die eksperimentele data wat gebruik word is dié wat in voorbeeld 1 gebruik is en wat in 'n permanente lêer geskryf is. Die aantal veranderlikes in 'n datastel is 8, dus moet in die naamlys M=8 wees. Die eerste beskikbare kolom om verdere veranderlikes te stoor is in X(9) of X(M+1), hierdie veranderlikes word geskep of verander in PASGEN. Sien bladsy C9.

Uit die korrelasietabel, in voorbeeld 1 bladsy C5, blyk dat X3 met X1 en X2 korreleer, ons sal dus X1 en X2 in die passing gebruik.

Gestel ons wil die volgende funksie pas

$$Y = a_1 + a_2X_1 + a_3X_2 + a_4X_1X_2 + e^{a_5X_2}$$

Die funksie is lineêr in a_i, i=1,2,3,4 dus moet IA in PASDIM minstens 9 wees. Aangesien 3 veranderlikes gebruik word is NN=3 en die vektor IX=1, 2, 3 waar IX(NN)=3 as afhanklike veranderlike gebruik sal word.

Die funksie wat dus geminimeer word is

$$S = \sum_{i=1}^{m} (X_{i3} - y_i)^2,$$

waar m die aantal eksperimentele datastelle voorstel.

Omdat IPONS=1 is, sal die a-waardes op kaarte gepons word.

```
  PASDIM VIR VOORBEELD 2
  ------ --- --------- -
C
      SUBROUTINE PASDIM
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(1),A(1)
      COMMON/DIM/IR,IK,IA
C
C   BESKRYWING PARAMETERS
C
C   IR    = MAKSIMUM AANTAL DATAPUNTE WAT IN DIE PASSINGS GEBRUIK WORD,
C           EN MOET GROTER AS NUL WEES.
C   IK    = TOTALE AANTAL VERANDERLIKES, INGELEES PLUS GEGENEREER,
C           MET  0 < IK < 50 .
C   IA    = AANTAL A-WAARDES WAT IN FUNKSIE PLUS A-WAARDES WAT LINEER
C           VOORKOM, MET  0 < IA < 50 .
C
      IR=40
      IK=11
      IA=9
      RETURN
      ENTRY PASAF(X,A,F)
      ENTRY PASGEN(X,M)
C
C   IN PASGEN KAN NOG VERANDERLIKES GEGENEREER OF VERANDER WORD
C   M+1 IS EERSTE BESKIKBARE KOLOM.
C
      X(M+1)=X(1)-X(6)-X(7)
      X(M+2)=X(1)+X(7)-1.
      RETURN
      ENTRY PASFUN(X,A,FPAS)
C
C   IN PASFUN WORD DIE FUNKSIE GEDEFINIEER
C   BEREKEN DIE WAARDE VAN DIE PASSINGSFUNKSIE BY DIE I-DE DATAPUNTE
C
      FPAS=A(1)+A(2)*X(1)+A(3)*X(2)+A(4)*X(1)*X(2)+DEXP(A(5)*X(2))
      RETURN
      END
```

DATA VIR VOORBEELD 2
---- --- --------- -

100.2	0.6	3.05	50.3	0.054	0.21	0.8	0.01
100.0	1.	2.5	40.2	0.055	0.03	1.	0.03
400.0	1.	6.5	40.2	0.055	0.03	1.	0.03
99.8	1.45	2.1	10.0	0.059	0.25	0.8	0.05
100.0	2.2	1.2	4.02	0.065	0.01	1.0	0.01
100.0	2.7	0.75	6.	0.042	0.05	1.0	0.05
100.1	3.1	0.25	8.30	0.032	0.16	0.9	0.06
200.5	0.6	4.4	3.6	0.042	0.53	0.5	0.03
200.4	1.	3.7	8.3	0.052	0.42	0.6	0.02
200.3	1.25	3.32	7.6	0.031	0.31	0.7	0.01
200.1	1.51	3.05	3.2	0.025	0.12	0.9	0.02
199.6	1.83	2.75	6.3	0.031	0.61	0.4	0.01
199.8	2.65	1.78	7.4	0.025	0.83	0.2	0.03
200.5	3.1	1.3	8.5	0.031	0.56	0.5	0.06
200.4	3.6	0.9	3.2	0.052	0.47	0.6	0.07
99.9	1.1	2.45	20.1	0.058	0.14	0.9	0.04
300.3	0.8	5.7	6.1	0.061	0.34	0.7	0.04
300.2	1.	5.3	7.5	0.071	0.23	0.8	0.03
300.1	1.42	4.7	8.4	0.085	0.12	0.9	0.02
300.0	1.81	4.05	9.3	0.093	0.01	1.	0.01
100.1	1.75	1.75	5.01	0.063	0.16	0.9	0.06
300.5	2.25	3.5	10.5	0.01	0.52	0.5	0.02
300.3	2.7	3.0	10.6	0.02	0.36	0.7	0.06
100.0	1.	2.9	40.2	0.055	0.03	1.	0.03
100.1	1.1	2.15	20.1	0.058	0.14	0.9	0.04
100.0	1.9	1.6	4.02	0.065	0.01	1.0	0.01
199.6	2.05	2.25	6.3	0.031	0.61	0.4	0.01
300.1	3.25	2.35	74.	0.025	0.18	0.9	0.08
300.4	3.85	2.02	8.3	0.083	0.47	0.6	0.07
299.5	4.35	1.45	9.5	0.073	0.56	0.5	0.06
400.3	2.	5.55	6.3	0.082	0.35	0.7	0.05
400.2	2.5	4.9	7.5	0.032	0.24	0.8	0.04
400.6	3.1	4.25	8.6	0.053	0.62	0.4	0.02
399.8	3.7	3.6	7.6	0.052	0.88	0.2	0.08
399.9	4.25	2.83	4.3	0.061	0.99	0.1	0.09
400.2	4.62	2.53	5.8	0.075	0.81	0.8	0.01

```
*    EINDE DATAPUNTE
 &LEES PASF='VOORBEELD 2',M=8,
 OPSK='X1','X2','X3','X4','X5','X6','X7','X8','X9','X10','X11','X12','X13',
 FORM='(8F10.0)',IPONS=1,TYD=5.,IPR=8,IT=40,ITSIK=10,
 NL=5,NN=3,IX=1,2,3,   &END
```

```
DATUM 77-09-23                                                                                    TYD    9-14-21

                                              ** INVOER **

IA   =    9    IPR  =   8    IFIT =   1    IDENT =  -1    ITSIK =  10    MM  =  10    NPAS   =   1
IR   =   40    ICON =   1    ILYN =   1    IDRUK =   0    KA    =   0    NL  =   5    NDIGD  = 100
IK   =   11    ICOR =   1    IPEN =   1    IPLOT =   0    KX    =   0    NN  =   3    NSOEK  =   0
IT   =   40    IDAT =   0    IXAS =   2    IPONS =   1    LEESX =   1    NGO =   5    NUM    =   2
IAS  =    1    IDES =   1    IYAS =   3    IPUNT =   0    M     =   8    NSK =   1    N      =  36

IX   =    1    2    3
IV   =    0    0    0    0    0
EPS  =  1.00D-06   TYD   =  5.00E 00   YOUIM =  1.10E 01   FNU   =  1.00D-04
FORM = (8F10.0)

A(I) =       0.0             0.0             0.0             0.0             0.0

                                              ** RESULTATE **

MINIMERING VAN SOM VAN VIERKANTE     N =  5     M =   36
ITERASIE   0                 SKW   =  216.1160       FNU = 1.0000000D-04     FSC =  216.1160
               A       =   0.0              0.0              0.0              0.0              0.0
               X       =   0.0              0.0              0.0              0.0              0.0
               XSC     =   1.5353510        5.38778699D-03   0.70862507       2.34716046D-03   0.84992954
               DELDIF=     1.53535095D-06   5.38778699D-09   7.08625068D-07   2.34716046D-09   8.49929538D-07

MINIMERING VAN SOM VAN VIERKANTE     N =  5     M =   36
ITERASIE   1                 SKW   =  103.4946       FNU = 5.0000000D-05     FSC =  103.4946
               A       =   1.4389629        1.27994904D-02  -0.20185405       2.57172124D-04  -1.0042553
               X       =   1.3658897        3.2758450       -0.57281126       0.21171595      -0.65767364
               XSC     =   1.0534987        3.90723322D-03   0.35239190       1.21470356D-03   1.5269813
               DELDIF=     1.05349865D-06   3.90723322D-09   3.52391905D-07   1.21470356D-09   1.52698129D-06

MINIMERING VAN SOM VAN VIERKANTE     N =  5     M =   36
ITERASIE   2                 SKW   = 0.9731602       FNU = 2.5000000D-05     FSC = 0.9731602
               A       =   1.7914274        1.33542966D-02  -0.98509819      -4.67614769D-05  -0.90956309
               X       =   15.408959        29.671881       -20.716401       -0.29689080      -2.4324791
               XSC     =   0.11625882       4.50065720D-04   4.75516083D-02   1.57503961D-04   0.37392432
               DELDIF=     1.16258820D-07   4.50065720D-10   4.75516083D-08   1.57503961D-10   3.73924321D-07

ITERASIE   8   ITE     =  8   SKW = 0.9729673      FNU = 7.8125000D-07     FNN        1.000
               V       =   0.22596715       0.43089731       0.30212723       4.18307476D-03   3.68252411D-02
               X       =   15.551776        29.666346       -20.798186       -0.28790325      -2.5316396
               A       =   1.8080311        1.33518056D-02  -0.98898719      -4.53459019D-05  -0.94664163

MINIMERINGSTYD IS    2.63 SEK.
```

MATRIKS MET KORRELASIEKOEFFISIENTE

I	1	2	3	4	5	6	7	8	9	10
1	1.00	0.44	0.62	-0.11	0.15	0.50	-0.41	0.20	1.00	1.00
2	0.44	1.00	-0.43	-0.21	0.01	0.59	-0.44	0.53	0.44	0.44
3	0.62	-0.43	1.00	0.07	0.20	-0.03	-0.01	-0.24	0.62	0.62
4	-0.11	-0.21	0.07	1.00	-0.11	-0.35	0.35	0.08	-0.11	-0.11
5	0.15	0.01	0.20	-0.11	1.00	-0.17	0.26	-0.02	0.15	0.15
6	0.50	0.59	-0.03	-0.35	-0.17	1.00	-0.93	0.27	0.50	0.49
7	-0.41	-0.44	-0.01	0.35	0.26	-0.93	1.00	-0.27	-0.41	-0.41
8	0.20	0.53	-0.24	0.08	-0.02	0.27	-0.27	1.00	0.20	0.20

C 11

9	1.00	0.44	0.62	-0.11	0.15	0.50	-0.41	0.20	1.00	1.00
10	1.00	0.44	0.62	-0.11	0.15	0.49	-0.41	0.20	1.00	1.00

NR.	X1	X2	X3	BEREKEN	RESIDU
1	100.2000	0.6000	3.0500	3.1164	-0.0664
2	100.0000	1.0000	2.5000	2.5377	-0.0377
3	400.0000	1.0000	6.5000	6.5297	-0.0297
4	99.8000	1.4500	2.1000	1.9534	0.1466
5	100.0000	2.2000	1.2000	1.0821	0.1179
6	100.0000	2.7000	0.7500	0.5383	0.2117
7	100.1000	3.1000	0.2500	0.1178	0.1322
8	200.5000	0.6000	4.4000	4.4529	-0.0529
9	200.4000	1.0000	3.7000	3.8737	-0.1737
10	200.3000	1.2500	3.3200	3.5411	-0.2211
11	200.1000	1.5100	3.0500	3.2121	-0.1621
12	199.6000	1.8300	2.7500	2.8235	-0.0735
13	199.8000	2.6500	1.7800	1.9123	-0.1323
14	200.5000	3.1000	1.3000	1.4442	-0.1442
15	200.4000	3.6000	0.9000	0.9238	-0.0238
16	99.9000	1.1000	2.4500	2.4020	0.0480
17	300.3000	0.8000	5.7000	5.4844	0.2156
18	300.2000	1.0000	5.3000	5.2017	0.0983
19	300.1000	1.4200	4.7000	4.6520	0.0480
20	300.0000	1.8100	4.0500	4.1791	-0.1291
21	100.1000	1.7500	1.7500	1.5967	0.1533
22	300.5000	2.2500	3.5000	3.6832	-0.1832
23	300.3000	2.7000	3.0000	3.1882	-0.1882
24	100.0000	1.0000	2.9000	2.5377	0.3623
25	100.1000	1.1000	2.1500	2.4047	-0.2547
26	100.0000	1.9000	1.6000	1.4210	0.1790
27	199.6000	2.0500	2.2500	2.5707	-0.3207
28	300.1000	3.2500	2.3500	2.6026	-0.2526
29	300.4000	3.8500	2.0200	1.9850	0.0350
30	299.5000	4.3500	1.4500	1.4620	-0.0120
31	400.3000	2.0000	5.5500	5.2891	0.2609
32	400.2000	2.5000	4.9000	4.7274	0.1726
33	400.6000	3.1000	4.2500	4.0877	0.1623
34	399.8000	3.7000	3.6000	3.4499	0.1501
35	399.9000	4.2500	2.8300	2.8850	-0.0550
36	400.2000	4.6200	2.5300	2.5111	0.0189

RESULTATE

```
IER= 0  IT= 8  REGRESSIEKOEFFISIENT                                      =  9.937852690-01
               SOM VAN KWADRAATAFWYKINGS                                 =  9.729673450-01
               SOM ABSOLUTE FOUT                                         =  5.025655340 00
               WORTEL GEMIDDELDE KWADRAATAFWYKING                        =  1.643985120-01
               WORTEL GEMIDDELDE KWADRAATAFW. VAN DIE PERSENTASIEFOUT    =  1.181588370 01   PERSENT

A(I)=   1.8080311423D 00   1.3351805597D-02   -9.8898718753D-01   -4.5345901950D-05   -9.4664162877D-01

PASSING GEBASEER OP  36 PUNTE
```

C 12

3. Grafieke van Voorbeeld 2

In voorbeeld 2 het ons 'n funksie gepas en uit die resultate blyk dit dat die passing suksesvol was, want IER=0.

Ons wil nou grafieke deur die eksperimentele punte trek, maar met X1 in die omgewing van respektiewelik 100, 200, 300 en 400. Die afhanklike veranderlike IX(NN) word langs die Y-as gestip en die onafhanklike IX(NN-1) word langs die X-as gestip. Die derde veranderlike word konstant gehou, dus NSOEK, die aantal veranderlikes wat konstant gehou moet word, is 1. Vir die eerste grafiek is XP=100 en TOLX=0,5. Dus alle punte met XP=100 ± 0,5 sal gestip word. As ILYN=1 sal die gepaste kromme ook deur die punte geteken word. Vir die tweede grafiek word al die punte met XP tussen 200 ± 0,5 geneem word en omdat IAS=0, sal die grafiek op dieselfde assestelsel geteken word. Net so vir die ander grafieke met XP=300 en 400. Die waarde van XP(NSOEK) word by die eindpunte van die grafiek geskryf.

PASDIM VIR VOORBEELD 3
------ --- --------- -

```
C
      SUBROUTINE PASDIM
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(1),A(1)
      COMMON/DIM/IR,IK,IA
C
C  BESKRYWING PARAMETERS
C
C  IR    = MAKSIMUM AANTAL DATAPUNTE WAT IN DIE PASSINGS GEBRUIK WORD,
C          EN MOET GROTER AS NUL WEES.
C  IK    = TOTALE AANTAL VERANDERLIKES, INGELEES PLUS GEGENEREER,
C          MET  0 < IK < 50 .
C  IA    = AANTAL A-WAARDES WAT IN FUNKSIE PLUS A-WAARDES WAT LINEER
C          VOORKOM, MET  0 < IA < 50 .
C
      IR=40
      IK=11
      IA=9
      RETURN
      ENTRY PASGEN(X,M)
      ENTRY PASAF(X,A,F)
      RETURN
      ENTRY PASFUN(X,A,FPAS)
C
C  IN PASFUN WORD DIE FUNKSIE GEDEFINIEER
C  BEREKEN DIE WAARDE VAN DIE PASSINGSFUNKSIE BY DIE I-DE DATAPUNTE
C
      FPAS=A(1)+A(2)*X(1)+A(3)*X(2)+A(4)*X(1)*X(2)+DEXP(A(5)*X(2))
      RETURN
      END
```

DATA VIR VOORBEELD 3
---- --- --------- -

```
*   DATA STAAN OP SKYF.
 &LEES   PASF='VOORBEELD 3',
 OPSK='X1','X2','X3','X4','X5','X6','X7','X8','X9','X10','X11','X12','X13',
  XBEG=0.,XEND=6.,YBEG=0.,YEND=7.,XLEN=6.,YLEN=7.,NN=3,IX=1,2,3,NGO=8,
 FORM='(8F10.0)',M=8,NL=5,IPLOT=1,NSOEK=1,IFIT=0,
 A=  1.8080478262D 00,  1.3351806177D-02, -9.8899046807D-01,
 -4.5346180232D-5,-9.4667716271D-01,
 IDXP(1)=0,IPUNT=1,ILYN=1,XP=100.,TOLX=.5, &END
 &LEES   XP=200., IAS=0, IFIT=0,&END
 &LEES   XP=300.,IAS=0, &END
 &LEES   XP=400.,   &END
```

DATUM 77-09-22 TYD 10-52-26

** INVOER **

```
IA    =   9     IPR   =  1    IFIT  =  0    IDENT =  -1   ITSIK =  0    MM    =  8    NPAS   =   1
IR    =  40     ICON  =  1    ILYN  =  1    IDRUK =   0   KA    =  0    NL    =  5    NDIGD  = 100
IK    =  11     ICOR  =  1    IPEN  =  1    IPLOT =   1   KX    =  0    NN    =  3    NSOEK  =   1
IT    =   0     IDAT  =  0    IXAS  =  2    IPONS =   0   LEESX =  1    NGO   =  8    NUM    =   2
IAS   =   1     IDES  =  1    IYAS  =  3    IPUNT =   1   M     =  8    NSK   =  1    N      =  36

IX    =   1     2     3
IV    =   0     0     0     0     0
IDXP  =   0     2     2
IDG   =   2     2     2
EPS   =  1.00D-06   TYD   =  1.00E 01   YDUIM =  1.10E 01   FNU   =  1.00D-04
XBEG  =  0.0        XEND  =  6.00       YBEG  =  0.0        YEND  =  7.00      XLEN  =  6.00     YLEN  =  7.00

XP(I) =  1.00E 02
TOLX  =  5.00E-01
FORM  = (8F10.0)

A(I)  =     1.8080478     1.335180620D-02     -0.98899047     -4.534618020D-05     -0.94667716
```

** RESULTATE **

MATRIKS MET KORRELASIEKOEFFISIENTE

I	1	2	3	4	5	6	7	8
1	1.00	0.44	0.62	-0.11	0.15	0.50	-0.41	0.20
2	0.44	1.00	-0.43	-0.21	0.01	0.59	-0.44	0.53
3	0.62	-0.43	1.00	0.07	0.20	-0.03	-0.01	-0.24
4	-0.11	-0.21	0.07	1.00	-0.11	-0.35	0.35	0.08
5	0.15	0.01	0.20	-0.11	1.00	-0.17	0.26	-0.02
6	0.50	0.59	-0.03	-0.35	-0.17	1.00	-0.93	0.27
7	-0.41	-0.44	-0.01	0.35	0.26	-0.93	1.00	-0.27
8	0.20	0.53	-0.24	0.08	-0.02	0.27	-0.27	1.00

```
PUNTE WORD GEKARAKTERISEER DEUR
  X1       = 100.0
BREUKTOLERANSIES --X1      ±0.500000--
     X2       TUSSEN 0.0        EN  6.000        X3       TUSSEN 0.0        EN  7.000
```

C 15

** INVOER **

IA	=	9	IPR	=	1	IFIT	=	0	IDENT	=	-1	ITSIK	=	0	MM	=	8	NPAS	= 1
IR	=	40	ICON	=	1	TLYN	=	1	IDRUK	=	0	KA	=	0	NL	=	5	NDIGD	= 100
IK	=	11	ICOR	=	1	IPEN	=	2	IPLOT	=	1	KX	=	0	NN	=	3	NSOEK	= 1
IT	=	0	IDAT	=	0	IXAS	=	2	IPONS	=	0	LEESX	=	0	NGO	=	8	NUM	= 2
IAS	=	0	IDES	=	1	IYAS	=	3	IPUNT	=	1	M	=	8	NSK	=	1	N	= 36

IX = 1 2 3
IV = 0 0 0 0 0
IDXP = 0 2 2
IDG = 2 2 2
EPS = 1.00D-06 TYD = 1.00E 01 YDUIM = 1.10E 01 FNU = 1.00D-04
XBEG = 0.0 XEND = 6.00 YBEG = 0.0 YEND = 7.00 XLEN = 6.00 YLEN = 7.00

XP(I) = 2.00E 02
TOLX = 5.00E-01
FORM =(8F10.0)

A(I) = 1.8080478 1.335180620D-02 -0.98899047 -4.534618020D-05 -0.94667716

** RESULTATE **

MATRIKS MET KORRELASIEKOEFFISIENTE

I	1	2	3	4	5	6	7	8
1	1.00	0.44	0.62	-0.11	0.15	0.50	-0.41	0.20
2	0.44	1.00	-0.43	-0.21	0.01	0.59	-0.44	0.53
3	0.62	-0.43	1.00	0.07	0.20	-0.03	-0.01	-0.24
4	-0.11	-0.21	0.07	1.00	-0.11	-0.35	0.35	0.08
5	0.15	0.01	0.20	-0.11	1.00	-0.17	0.26	-0.02
6	0.50	0.59	-0.03	-0.35	-0.17	1.00	-0.93	0.27
7	-0.41	-0.44	-0.01	0.35	0.26	-0.93	1.00	-0.27
8	0.20	0.53	-0.24	0.08	-0.02	0.27	-0.27	1.00

PUNTE WORD GEKARAKTERISEER DEUR
X1 = 200.0
BREUKTOLERANSIES --X1 ±0.500000--
X2 TUSSEN 0.0 EN 6.000 X3 TUSSEN 0.0 EN 7.000

C 16

** INVOER **

IA	=	9	IPR	=	1	IFIT	=	0	IDENT	=	-1	ITSIK	=	0	MM	=	8	NPAS	=	1
IR	=	40	ICON	=	1	ILYN	=	1	IDRUK	=	0	KA	=	0	NL	=	5	NDIGO	=	100
IK	=	11	ICOR	=	1	IPEN	=	3	IPLOT	=	1	KX	=	0	NN	=	3	NSOEK	=	1
IT	=	0	IDAT	=	0	IXAS	=	2	IPONS	=	0	LEESX	=	0	NGO	=	8	NUM	=	2
IAS	=	0	IDES	=	1	IYAS	=	3	IPUNT	=	1	M	=	8	NSK	=	1	N	=	36

IX = 1 2 3
IV = 0 0 0 0 0
IDXP = 0 2 2
IDG = 2 2 2
EPS = 1.00D-06 TYD = 1.00E 01 YDUIM = 1.10E 01 FNU = 1.00D-04
XBEG = 0.0 XEND = 6.00 YBEG = 0.0 YEND = 7.00 XLEN = 6.00 YLEN = 7.00

XP(I) = 3.00E 02
TOLX = 5.00E-01
FORM =(8F10.0)

A(I) = 1.8080478 1.335180620-02 -0.98899047 -4.534618020-05 -0.94667716

** RESULTATE **

MATRIKS MET KORRELASIEKOEFFISIENTE

I	1	2	3	4	5	6	7	8
1	1.00	0.44	0.62	-0.11	0.15	0.50	-0.41	0.20
2	0.44	1.00	-0.43	-0.21	0.01	0.59	-0.44	0.53
3	0.62	-0.43	1.00	0.07	0.20	-0.03	-0.01	-0.24
4	-0.11	-0.21	0.07	1.00	-0.11	-0.35	0.35	0.08
5	0.15	0.01	0.20	-0.11	1.00	-0.17	0.26	-0.02
6	0.50	0.59	-0.03	-0.35	-0.17	1.00	-0.93	0.27
7	-0.41	-0.44	-0.01	0.35	0.26	-0.93	1.00	-0.27
8	0.20	0.53	-0.24	0.08	-0.02	0.27	-0.27	1.00

PUNTE WORD GEKARAKTERISEER DEUR
X1 = 300.0
BREUKTOLERANSIES --X1 ±0.500000--
X2 TUSSEN 0.0 EN 6.000 X3 TUSSEN 0.0 EN 7.000

C 17

** INVOER **

```
IA    =    9      IPR   =   1     IFIT  =   0     IDENT =  -1     ITSIK =   0     MM    =   8     NPAS  =   1
IR    =   40      ICON  =   1     ILYN  =   1     IDRUK =   0     KA    =   0     NL    =   5     NDIGO = 100
IK    =   11      ICOR  =   1     IPEN  =   1     IPLOT =   1     KX    =   0     NN    =   3     NSOEK =   1
IT    =    0      IDAT  =   0     IXAS  =   2     IPONS =   0     LEESX =   0     NGO   =   8     NUM   =   2
IAS   =    0      IDES  =   1     IYAS  =   3     IPUNT =   1     M     =   8     NSK   =   1     N     =  36

IX    =    1    2    3
IV    =    0    0    0    0    0
IDXP  =    0    2    2
IDG   =    2    2    2
EPS   =  1.00D-06   TYD   =  1.00E 01   YDUIM =  1.10E 01   FNU   =  1.00D-04
XBEG  =  0.0        XEND  =  6.00       YBEG  =  0.0        YEND  =  7.00      XLEN  =  6.00     YLEN  =  7.00

XP(I) =  4.00E 02
TOLX  =  5.00E-01
FORM  = (8F10.0)


A(I)  =      1.8080478        1.33518062D-02     -0.98899047       -4.53461802D-05     -0.94667716
```

** RESULTATE **

MATRIKS MET KORRELASIEKOEFFISIENTE

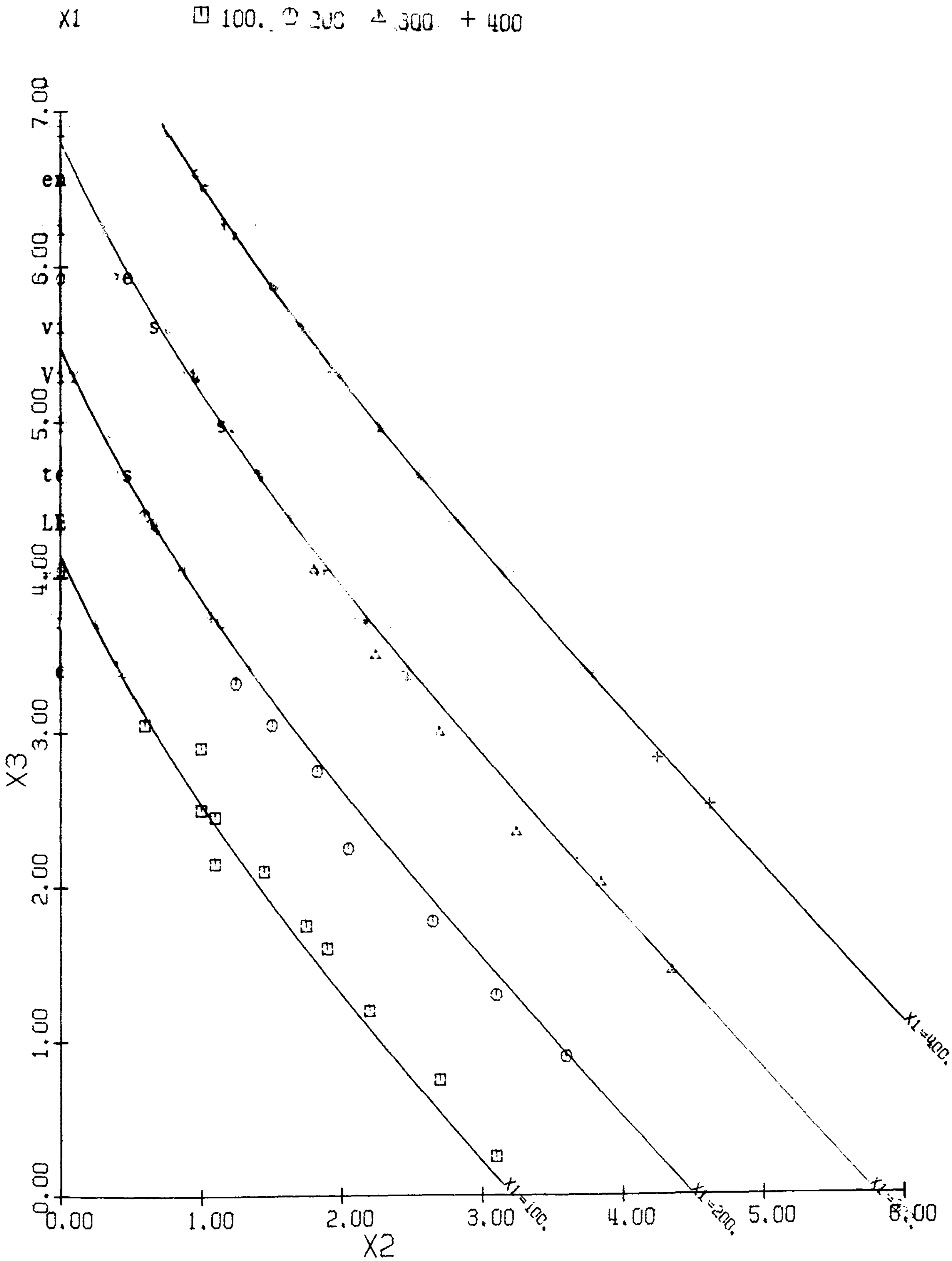
I	1	2	3	4	5	6	7	8
1	1.00	0.44	0.62	-0.11	0.15	0.50	-0.41	0.20
2	0.44	1.00	-0.43	-0.21	0.01	0.59	-0.44	0.53
3	0.62	-0.43	1.00	0.07	0.20	-0.03	-0.01	-0.24
4	-0.11	-0.21	0.07	1.00	-0.11	-0.35	0.35	0.08
5	0.15	0.01	0.20	-0.11	1.00	-0.17	0.26	-0.02
6	0.50	0.59	-0.03	-0.35	-0.17	1.00	-0.93	0.27
7	-0.41	-0.44	-0.01	0.35	0.26	-0.93	1.00	-0.27
8	0.20	0.53	-0.24	0.08	-0.02	0.27	-0.27	1.00

```
PUNTE WORD GEKARAKTERISEER DEUR
  X1       = 400.0
BREUKTOLERANSIES --X1      ±0.500000--
     X2       TUSSEN 0.0        EN   6.000          X3      TUSSEN 0.0       EN   7.000

                        TOTALE TYD IN REKENAAR =    94.40 SEK.
                      WERKLIKE TYD IN REKENAAR =     2.59 SEK.
```

C 18

Voorbeeld 3 met Begrensde Veranderlikes

VOORBEELD 3



4. Voorbeeld 3 met Begrensde Veranderlikes

Ons gebruik weer die data van voorbeeld 2 met die verskil dat ons nou eers die funksie deur die punte in die omgewing van 100 wil pas, daarna respektiewelik in die omgewing van 200, 300 en 400. Die eksperimentele punte word weer gestip en die grafieke geteken. Vir elke passing moet IFIT=1 wees, daar sal dus vier passings gedoen en vier grafieke geteken word. Die data van elke grafiek moet uitgesoek word, vir die eerste grafiek bv. tussen 90 en 110. Dit word gedoen deur grense vir die X1-waardes te stel. Daar is dus vir een stel X-waardes grense, d.w.s. KX=1, KVX=1 en GX=99, 101. Vir die punte wat gestip moet word is NSOEK=1 en XP=100, met TOLX=0,5. Alle punte tussen XP$\pm$0,5 sal dus geteken word. Omdat die program telkens as data van skyf gelees word, LEESX gelyk aan 0 stel, moet LEESX=1 gestel word vir elke kromme wat geteken moet word. As IAS =0 is, word die tweede grafiek op dieselfde assestelsel geteken. Vir die tweede grafiek is XP=200, KX=1 en GX=199, 201. Soortgelyke veranderinge moet aangebring word vir grafieke 3 en 4.

PASDIM VIR VOORBEELD 4
------ --- --------- -

```
C
      SUBROUTINE PASDIM
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(1),A(1)
      COMMON/DIM/IR,IK,IA
C
C  BESKRYWING PARAMETERS
C
C  IR   = MAKSIMUM AANTAL DATAPUNTE WAT IN DIE PASSINGS GEBRUIK WORD,
C         EN MOET GROTER AS NUL WEES.
C  IK   = TOTALE AANTAL VERANDERLIKES, INGELEES PLUS GEGENEREER,
C         MET  0 < IK < 50 .
C  IA   = AANTAL A-WAARDES WAT IN FUNKSIE PLUS A-WAARDES WAT LINEER
C         VOORKOM, MET  0 < IA < 50 .
C
      IR=40
      IK=11
      IA=9
      RETURN
      ENTRY PASAF(X,A,F)
      ENTRY PASGEN(X,M)
C     X(M+1)=X(4)*X(5)
      X(M+1)=X(1)-X(6)-X(7)
      X(M+2)=X(1)+X(7)-1.
      RETURN
      ENTRY PASFUN(X,A,FPAS)
      FPAS=A(1)+A(2)*X(1)+A(3)*X(2)+A(4)*X(1)*X(2)+DEXP(A(5)*X(2))
      RETURN
      END
```

DATA VIR VOORBEELD 4
---- --- --------- -

```
*   DATA STAAN OP SKYF.
 &LEES PASF='VOORBEELD 4',
 OPSK='X1','X2','X3','X4','X5','X6','X7','X8','X9','X10','X11','X12','X13',
 FORM='(8F10.0)',M=8,IPLOT=1,ICOR=0,IPR=0,IT=40,ITSIK=10,NSOEK=1,NL=5,
  XBEG=0.,XEND=6.,YBEG=0.,YEND=7.,XLEN=6.,YLEN=7.,NN=3,IX=1,2,3,
 XP=100.,TOLX=1.0,NGO=8,IPUNT=1,ILYN=1,IDXP=-1,
 KX=1,KVX=1,GX=99.,101.,   &END
 &LEES   XP=200., IAS=0, LEESX=1,IDRUK=0,ICOR=0,GX=199.,201.,    &END
 &LEES   XP=300.,LEESX=1,IDRUK=-1,GX=299.,301.,   &END
 &LEES   XP=400.,LEESX=1,GX=399.,401.,KA=1,KVA=4,GA=-6.,1.,FNU=1.D-2, &END
```

```
DATUM 77-09-22                                                                                              TYD  11- 1-56

                                                  ** INVOER **

IA     =    9      IPR   =   0      IFIT  =   1      IDENT  =  -1      ITSIK  =  10      MM    =  10      NPAS   =   1
IP     =   40      ICON  =   1      ILYN  =   1      IDRUK  =   0      KA     =   0      NL    =   5      NDIGD  = 100
IK     =   11      ICOR  =   0      IPEN  =   1      IPLOT  =   1      KX     =   1      NN    =   3      NSOEK  =   1
IT     =   40      IDAT  =   0      IXAS  =   2      IPONS  =   0      LEESX  =   1      NGO   =   8      NUM    =   2
IAS    =    1      IDES  =   1      IYAS  =   3      IPUNT  =   1      M      =   8      NSK   =   1      N      =  11

IX     =    1    2    3
IV     =    0    0    0    0    0
IDXP   =   -1    2    2
IDG    =    2    2    2
KVX(I) =    1
GX     =   99.00        < X1        <  101.0     ;
EPS    =  1.00D-06   TYD    =  1.00E 01   YDUIM  =  1.10E 01   FNU   =  1.00D-04
XBEG   =  0.0        XEND   =  6.00       YBEG   =  0.0        YEND  =  7.00       XLEN  =  6.00      YLEN  =  7.00

XP(I)  =  1.00F 02
TOLX   =  1.00F 00
FORM   =(8F10.0)

A(I)   =       0.0            0.0            0.0            0.0            0.0
```

** RESULTATE **

MINIMERINGSTYD IS   2.32 SEK.

NR.	X1	X2	X3	BEREKEN	RESIDU
1	100.2000	0.6000	3.0500	3.0124	0.0376
2	100.0000	1.0000	2.5000	2.5845	-0.0845
3	99.8000	1.4500	2.1000	2.1446	-0.0446
4	100.0000	2.2000	1.2000	1.2293	-0.0293
5	100.0000	2.7000	0.7500	0.7223	0.0277
6	100.1000	3.1000	0.2500	0.2737	-0.0237
7	99.9000	1.1000	2.4500	2.5035	-0.0535
8	100.1000	1.7500	1.7500	1.6672	0.0828
9	100.0000	1.0000	2.9000	2.5845	0.3155
10	100.1000	1.1000	2.1500	2.4268	-0.2768
11	100.0000	1.9000	1.6000	1.5511	0.0489

```
RESULTATE

IER= 0   IT=16   REGRESSIEKOEFFISIENT                                    =  9.86914314D-01
                 SOM VAN KWADRAATAFWYKINGS                               =  2.01016522D-01
                 SOM ABSOLUTE FOUT                                       =  1.02487039D 00
                 WORTEL GEMIDDELDE KWADRAATAFWYKING                      =  1.35182208D-01
                 WORTEL GEMIDDELDE KWADRAATAFW. VAN DIE PERSENTASIEFOUT  =  6.38105045D 00   PERSENT

A(I)=    2.2292719493D 01   -1.9462460136D-01   1.5564686417D 01   -1.7132559205D-01   2.7923024513D-01
```

C 22

** INVOER **

IA	=	9	IPR	=	0	IFIT	=	1	IDENT	=	-1	ITSIK	=	10	MM	=	10	NPAS	= 1
IR	=	40	ICON	=	1	ILYN	=	1	IDRUK	=	0	KA	=	0	NL	=	5	NDIGD	= 100
IK	=	11	ICOR	=	0	IPEN	=	2	IPLOT	=	1	KX	=	1	NN	=	3	NSOEK	= 1
IT	=	40	IDAT	=	0	IXAS	=	2	IPONS	=	0	LEESX	=	1	NGO	=	8	NUM	= 2
IAS	=	0	IDES	=	1	IYAS	=	3	IPUNT	=	1	M	=	8	NSK	=	1	N	= 9

```
IX     =   1    2    3
IV     =   0    0    0    0    0
IDXP   =  -1    2    2
IDG    =   2    2    2
KVX(I)=   1
GX     =  199.0      < X1       <  201.0     I
EPS    =  1.000-06   TYD   =  1.00E 01   YDUIM =  1.10E 01   FNU   =  1.00D-04
XBEG   =  0.0        XEND  =  6.00       YBEG  =  0.0        YEND  =  7.00      XLEN  =  6.00      YLEN  =  7.00

XP(I)  =  2.00E 02
TOLX   =  1.00E 00
FORM   = (8F10.0)

A(I)   =      22.292719      -0.19462460      15.564686      -0.17132559      0.27923025
```

** RESULTATE **

MINIMERINGSTYD IS 0.95 SEK.

NR.	X1	X2	X3	BEREKEN	RESIDU
1	200.5000	0.6000	4.4000	4.3458	0.0542
2	200.4000	1.0000	3.7000	3.7420	-0.0420
3	200.3000	1.2500	3.3200	3.3834	-0.0634
4	200.1000	1.5100	3.0500	3.0299	0.0201
5	199.6000	1.8300	2.7500	2.6368	0.1132
6	199.8000	2.6500	1.7800	1.7406	0.0394
7	200.5000	3.1000	1.3000	1.2623	0.0377
8	200.4000	3.6000	0.9000	0.9283	-0.0283
9	199.6000	2.0500	2.2500	2.3808	-0.1308

RESULTATE

```
IER= 0   IT= 8   REGRESSIEKOEFFISIENT                                  =  9.979714500-01
                 SOM VAN KWADRAATAFWYKINGS                             =  4.283220440-02
                 SOM ABSOLUTE FOUT                                     =  5.291286630-01
                 WORTEL GEMIDDELDE KWADRAATAFWYKING                    =  6.898647560-02
                 WORTEL GEMIDDELDE KWADRAATAFW. VAN DIE PERSENTASIEFOUT =  2.996464590 00   PERSENT

A(I)=   -1.86380631630 01    1.14390097540-01    1.39033392400 01    -7.93347310890-02    3.72710991870-01
```

PASSING GEBASEER OP 9 PUNTE

C 23

```
                                                   ** INVOER **

IA     =    9       IPR    =    0      IFIT   =    1      IDENT  =   -1      ITSIK  =   10      MM     =   10      NPAS   =    1
IR     =   40       ICON   =    1      ILYN   =    1      IDRUK  =   -1      KA     =    0      NL     =    5      NDIGD  =  100
IK     =   11       ICOR   =    0      IPEN   =    3      IPLOT  =    1      KX     =    1      NN     =    3      NSOEK  =    1
IT     =   40       IDAT   =    0      IXAS   =    2      IPONS  =    0      LEESX  =    1      NGO    =    8      NUM    =    2
IAS    =    0       IDES   =    1      IYAS   =    3      IPUNT  =    1      M      =    8      NSK    =    1      N      =    9

IX     =    1    2    3
IV     =    0    0    0    0    0
IDXP   =   -1    2    2
IDG    =    2    2    2
KVX(I)=     1
GX     =  299.0      < X1       <  301.0      !
EPS    =  1.00D-06   TYD    =  1.00E 01   YDUIM =  1.10E 01   FNU    =  1.00D-04
XBEG   =  0.0        XEND   =  6.00       YBEG  =  0.0        YEND   =  7.00       XLEN   =  6.00      YLEN   =  7.00

XP(I)  =  3.00E 02
TOLX   =  1.00E 00
FORM   =(8F10.0)


A(I)   =      -18.638063         0.11439010          13.903339         -7.93347311D-02        0.37271099



RESULTATE

IER= 0   IT= 4   REGRESSIEKOEFFISIENT                                         =  9.99807406D-01
                 SOM VAN KWADRAATAFWYKINGS                                    =  6.83713861D-03
                 SOM ABSOLUTE FOUT                                            =  2.05024381D-01
                 WORTEL GEMIDDELDE KWADRAATAFWYKING                           =  2.75623306D-02
                 WORTEL GEMIDDELDE KWADRAATAFW. VAN DIE PERSENTASIEFOUT       =  6.94928708D-01  PERSENT

A(I)=    5.3954455092D 01   -1.5948763563D-01   -4.3478250896D 01    1.3750848417D-01    3.8096921589D-01

PASSING GEBASEER OP   9 PUNTE



VOORBEELD 4
```

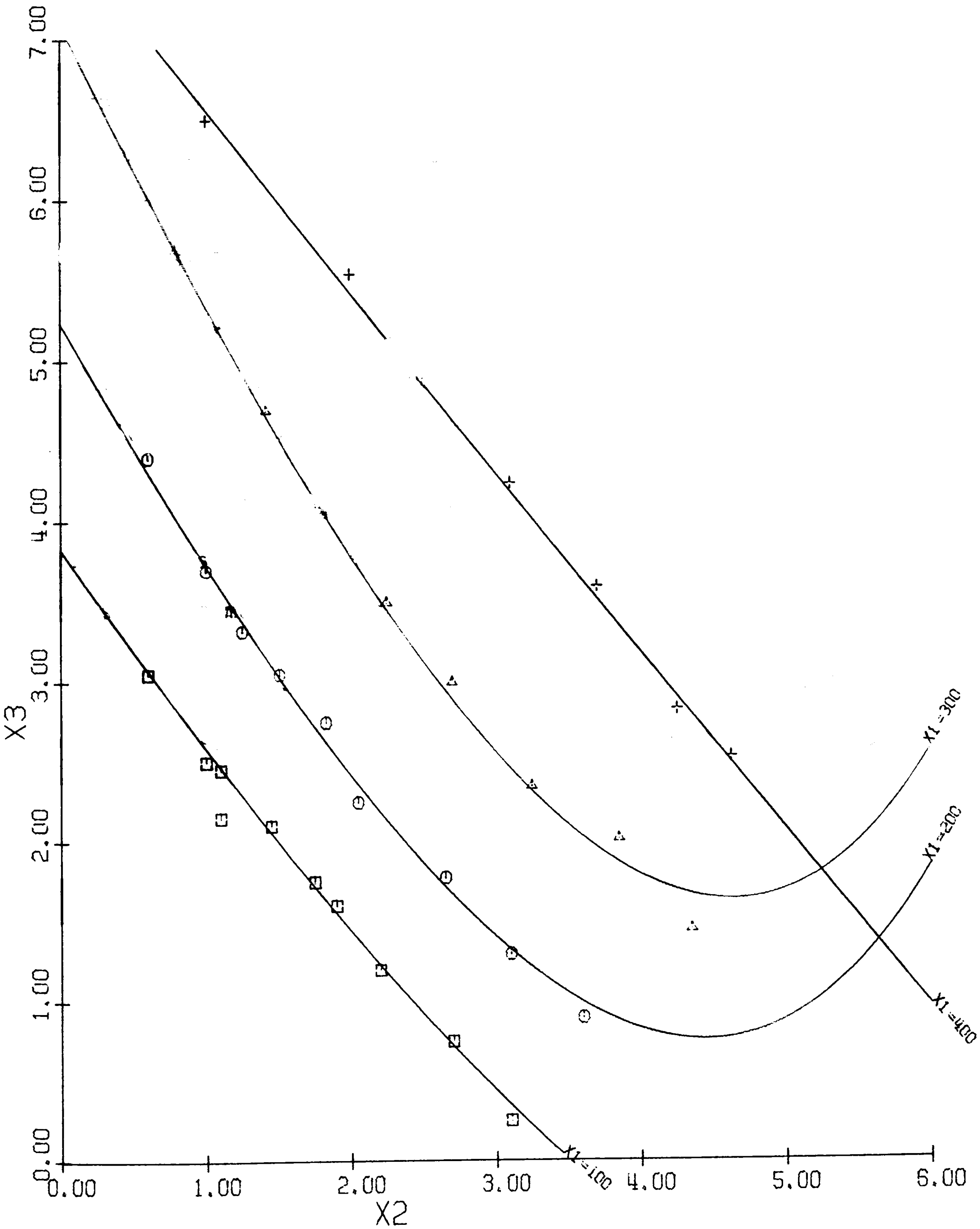
C 24

```
                                        ** INVOER **

IA     =    9      IPR    =    0      IFIT   =    1      IDENT  =   -1      ITSIK  =   10      MM     =   10      NPAS   =    1
IR     =   40      ICON   =    1      ILYN   =    1      IDRUK  =   -1      KA     =    1      NL     =    5      NDIGD  =  100
IK     =   11      ICOR   =    0      IPEN   =    1      IPLOT  =    1      KX     =    1      NN     =    3      NSOEK  =    1
IT     =   40      IDAT   =    0      IXAS   =    2      IPONS  =    0      LEESX  =    1      NGO    =    8      NUM    =    2
IAS    =    0      IDES   =    1      IYAS   =    3      IPUNT  =    1      M      =    8      NSK    =    1      N      =    7

IX     =    1    2    3
IV     =    0    0    0    0    0
IDXP   =   -1    2    2
IDG    =    2    2    2
KVX(I)=     1
GX     =   399.0      < X1        <   401.0      :
KVA(I)=     4
GA     =  -6.000      <    A( 4)  <   1.000      :
EPS    =   1.00D-06   TYD    =  1.00E 01   YDUIM =  1.10E 01   FNU    =  1.00D-02
XBEG   =   0.0        XEND   =  6.00       YBEG  =  0.0        YEND   =  7.00        XLEN   =  6.00       YLEN   =  7.00

XP(I)  =   4.00E 02
TOLX   =   1.00E 00
FORM   =(8F10.0)


A(I)   =        53.954455        -0.15948764        -43.478251        0.13750848        0.38096922



RESULTATE

IER= 0   IT=38   REGRESSIEKOEFFISIENT                                        =  9.991378720-01
                 SOM VAN KWADRAATAFWYKINGS                                   =  2.162736420-02
                 SOM ABSOLUTE FOUT                                           =  3.164010110-01
                 WORTEL GEMIDDELDE KWADRAATAFWYKING                          =  5.558438140-02
                 WORTEL GEMIDDELDE KWADRAATAFW. VAN DIE PERSENTASIEFOUT      =  1.584587290 00   PERSENT

A(I)=    -1.02242496260 02     2.72277078910-01     2.29800242310 01    -6.02372957430-02    -1.16109162880-03

PASSING GEBASEER OP    7 PUNTE


VOORBEELD 4

                         TOTALE TYD IN REKENAAR =    83.27 SEK.
                        WERKLIKE TYD IN REKENAAR =     9.91 SEK.
```

VOORBEELD 4

X1 TUSSEN 99.00 EN 101.00

X1 □ 100 ⊙ 200 △ 300 + 400



5. 'n Toepassing van Verskillende Funksies in Dieselfde Lopie

In hierdie voorbeeld pas ons verskillende funksies deur die stelle punte in 'n omgewing waar X1 respektiewelik 100, 200, 300 en 400 is.

Ons maak gebruik van die parameter NPAS wat oorgedra word deur die Common-stelling: COMMON/FNOM/NPAS. Vir die eerste kromme vir waardes van X1 tussen die grense 100 $\pm$ 1 word NPAS in die naamlys 1 gestel en die funksie in PASFUN na stellingnommer 1 word gepas.

Vir die tweede grafiek d.w.s. vir X1 tussen die grense 200 $\pm$ 1, word 'n lineêre funksie gepas en dit is dus nie nodig om iterasies te doen nie, dus IT=0 en NPAS=2.

Vir die derde grafiek met X1=300 $\pm$ 1, word 299 $\leq$ X1 $\leq$ 301 NPAS=3 gestel. In die geval word grense waarbinne a_4 moet lê gestel dus moet KA=1, KVA=4 en GA = -8, 1, d.w.s. $-8 \leq A_4 \leq 1$ wees.

In die vierde passing word NL=0 en IT=0 gestel terwyl NN=3 en IX=1, 2, 3. Dit bring mee dat die volgende funksie gepas word

$$Y = a_1 + a_2X_1 + a_3X_2$$

Met IX(NN) = IX(3) as afhanklike veranderlike.

PASDIM VIR VOORBEELD 5
------ --- --------- -

```
C
      SUBROUTINE PASDIM
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(1),A(1)
      COMMON/DIM/IR,IK,IA
      COMMON/FNUM/ NPAS
C
C  BESKRYWING PARAMETERS
C
C  IR   = MAKSIMUM AANTAL DATAPUNTE WAT IN DIE PASSINGS GEBRUIK WORD,
C         EN MOET GROTER AS NUL WEES.
C  IK   = TOTALE AANTAL VERANDERLIKES, INGELEES PLUS GEGENEREER,
C         MET   0 < IK < 50 .
C  IA   = AANTAL A-WAARDES WAT IN FUNKSIE PLUS A-WAARDES WAT LINEER
C         VOORKOM, MET  0 < IA < 50 .
C
      IR=40
      IK=11
      IA=9
      RETURN
      ENTRY PASGEN(X,M)
C
C  NOG VERANDERLIKES WORD GEGENEREER.
C  M+1 IS EERSTE BESKIKBARE KOLOM.
C
      X(M+1)=X(1)-X(6)-X(7)
      X(M+2)=X(1)+X(7)-1.
      RETURN
      ENTRY PASAF(X,A,F)
      ENTRY PASFUN(X,A,FPAS)
C
C  VERSKILLENDE FUNKSIES WORD GEPAS.
C  NPAS MOET OORGEDRA WORD MET 'N COMMON STELLING 'COMMON/FNOM/ NPAS'
C
      GO TO (1,2,3),NPAS
1     CONTINUE
C
C  FUNKSIE GEPAS VIR NPAS = 1
C
      FPAS=A(1)+A(2)*X(1)+A(3)*X(2)+A(4)*X(2)*X(2)+DEXP(A(5)*X(2))
      RETURN
2     CONTINUE
C
C  FUNKSIE GEPAS VIR NPAS = 2
C
      FPAS=A(1)+A(2)*X(2)+A(3)*X(2)*X(2)+A(4)*X(2)*X(2)*X(2)+A(5)*X(1)
      RETURN
3     CONTINUE
C
C  FUNKSIE GEPAS VIR NPAS = 3
C
      FPAS=A(1)+A(2)*X(2)+A(3)*X(2)*X(2)+A(4)*X(2)*X(2)*X(2)+A(5)/X(1)
      RETURN
      END
```

DATA VIR VOORBEELD 5
---- --- --------- -

```
*   DATA STAAN OP SKYF.
&LEFS
PASF='VOORBFELD 5',
OPSK='X1','X2','X3','X4','X5','X6','X7','X8','X9','X10','X11','X12','X13',
FORM='(8F10.0)',M=8,IPLOT=1,IPR=0,IT=40,ITSIK=10,NSOEK=1,
NL=5,IDRUK=0,ICOR=0,TOLX=1.0,
NN=3,IX=1,2,3,IDXP=-1,IPUNT=1,ILYN=1,XLEN=6.,YLEN=7.,XP=100.,
 XBFG=0.,XEND=6.,YBFG=0.,YEND=7.,
KX=1,KVX=1,GX= 99.,101.,  NGO=8, &END
&LEFS  XP=200., IAS=0, LEFSX=1,IT=0,NPAS=2,IDRUK=0,ICOR=0,
KX=1,KVX=1,GX=199.,201.,  &END
&LEFS  XP=300.,LFESX=1,NPAS=3,IDRUK=-1,
KA=1,KVA=4,GA=-8.,1.,
KX=1,KVX=1,GX=299.,301.,  &END
&LEFS  XP=400.,LEFSX=1,NL=0,
KX=1,KVX=1,GX=399.,401.,  &END
```

```
DATUM 77-09-22                                                                          TYD  11-21-45

                                        ** INVOER **

IA    =   9      IPR   =   0    IFIT  =   1    IDENT  =  -1    ITSIK =  10    MM   =  10    NPAS  =   1
IR    =  40      ICON  =   1    ILYN  =   1    IDRUK  =   0    KA    =   0    NL   =   5    NDIGD = 100
IK    =  11      ICOR  =   0    IPEN  =   1    IPLOT  =   1    KX    =   1    NN   =   3    NSOEK =   1
IT    =  40      IDAT  =   0    IXAS  =   2    IPONS  =   0    LEESX =   1    NGO  =   8    NUM   =   2
IAS   =   1      IDES  =   1    IYAS  =   3    IPUNT  =   1    M     =   8    NSK  =   1    N     =  11

IX      =   1    2    3
IV      =   0    0    0    0    0
IDXP    =  -1    2    2
IDG     =   2    2    2
KVX(I)=     1
GX      =  99.00        < X1      <  101.0     1
EPS     =  1.00D-06   TYD   =  1.00E 01   YDUIM =  1.10E 01   FNU   =  1.00D-04
XBEG    =  0.0        XEND  =  6.00       YBEG  =  0.0        YEND  =  7.00     XLEN  =  6.00     YLEN  =  7.00

XP(I)   =  1.00E 02
TOLX    =  1.00E 00
FORM    =(8F10.0)

A(I)    =      0.0           0.0           0.0           0.0           0.0

                                        ** RESULTATE **

MINIMERINGSTYD IS    2.42 SEK.

 NR.    X1          X2        X3         BEREKEN   RESIDU

  1  100.2000     0.6000    3.0500     3.0350     0.0150
  2  100.0000     1.0000    2.5000     2.5757    -0.0757
  3   99.8000     1.4500    2.1000     2.1504    -0.0504
  4  100.0000     2.2000    1.2000     1.2452    -0.0452
  5  100.0000     2.7000    0.7500     0.7270     0.0230
  6  100.1000     3.1000    0.2500     0.2615    -0.0115
  7   99.9000     1.1000    2.4500     2.5038    -0.0538
  8  100.1000     1.7500    1.7500     1.6668     0.0832
  9  100.0000     1.0000    2.9000     2.5757     0.3243
 10  100.1000     1.1000    2.1500     2.3993    -0.2493
 11  100.0000     1.9000    1.6000     1.5597     0.0403

RESULTATE

IFR= 0   IT=20   REGRESSIEKOEFFISIENT                                  =  9.876390490-01
                 SOM VAN KWADRAATAFWYKINGS                             =  1.899546520-01
                 SOM ABSOLUTE FOUT                                     =  9.716825950-01
                 WORTEL GEMIDDELDE KWADRAATAFWYKING                    =  1.314100630-01
                 WORTEL GEMIDDELDE KWADRAATAFW. VAN DIE PERSENTASIEFOUT =  5.66519025D 00  PERSENT

A(I)=    5.5735515581D 01   -5.2280969901D-01   -9.9695453713D-01   -5.3866767436D-03   -2.0916493641D 00
```

C 30

** INVOER **

```
IA     =    9      IPR    =   0      IFIT   =   1      IDENT  =  -1      ITSIK  =  10      MM     =  10      NPAS   =    2
IR     =   40      ICON   =   1      ILYN   =   1      IDRUK  =   0      KA     =   0      NL     =   5      NDIGD  =  100
IK     =   11      ICOR   =   0      IPEN   =   2      IPLOT  =   1      KX     =   1      NN     =   3      NSOEK  =    1
IT     =    0      IDAT   =   0      IXAS   =   2      IPONS  =   0      LEESX  =   1      NGO    =   8      NUM    =    2
IAS    =    0      IDES   =   1      IYAS   =   3      IPUNT  =   1      M      =   8      NSK    =   1      N      =    9

IX     =    1    2    3
IV     =    0    0    0    0    0
IDXP   =   -1    2    2
IDG    =    2    2    2
KVX(I) =    1
GX     =  199.0      < X1       <  201.0     I
EPS    =  1.00D-06   TYD    =  1.00E 01   YDUIM  =  1.10E 01   FNU    =  1.00D-04
XBEG   =  0.0        XEND   =  6.00       YBEG   =  0.0        YEND   =  7.00      XLEN   =  6.00      YLEN   =  7.00

XP(I)  =  2.00E 02
TOLX   =  1.00E 00
FORM   =(8F10.0)


A(I)   =      55.735516      -0.52280970      -0.99695454      -5.38667674D-03      -2.0916494
```

** RESULTATE **

MINIMERINGSTYD IS 3.58 SEK.

NR.	X1	X2	X3	BEREKEN	RESIDU
1	200.5000	0.6000	4.4000	4.3909	0.0091
2	200.4000	1.0000	3.7000	3.7214	-0.0214
3	200.3000	1.2500	3.3200	3.3499	-0.0299
4	200.1000	1.5100	3.0500	3.0044	0.0456
5	199.6000	1.8300	2.7500	2.6431	0.1069
6	199.8000	2.6500	1.7800	1.7546	0.0254
7	200.5000	3.1000	1.3000	1.2937	0.0063
8	200.4000	3.6000	0.9000	0.9045	-0.0045
9	199.6000	2.0500	2.2500	2.3874	-0.1374

RESULTATE

```
IER= 0  IT= 1  REGRESSIEKOEFFISIENT                                       =  9.98365975D-01
               SOM VAN KWADRAATAFWYKINGS                                  =  3.45086074D-02
               SOM ABSOLUTE FOUT                                          =  3.86516535D-01
               WORTEL GEMIDDELDE KWADRAATAFWYKING                         =  6.19216417D-02
               WORTEL GEMIDDELDE KWADRAATAFW. VAN DIE PERSENTASIEFOUT     =  2.54565300D 00  PERSENT

A(I)=   2.3679686983D 01   -2.1949019127D 00   3.4478223987D-01   -2.7022798333D-02   -9.0225068695D-02

PASSING GEBASEER OP   9 PUNTE
```

C 31

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

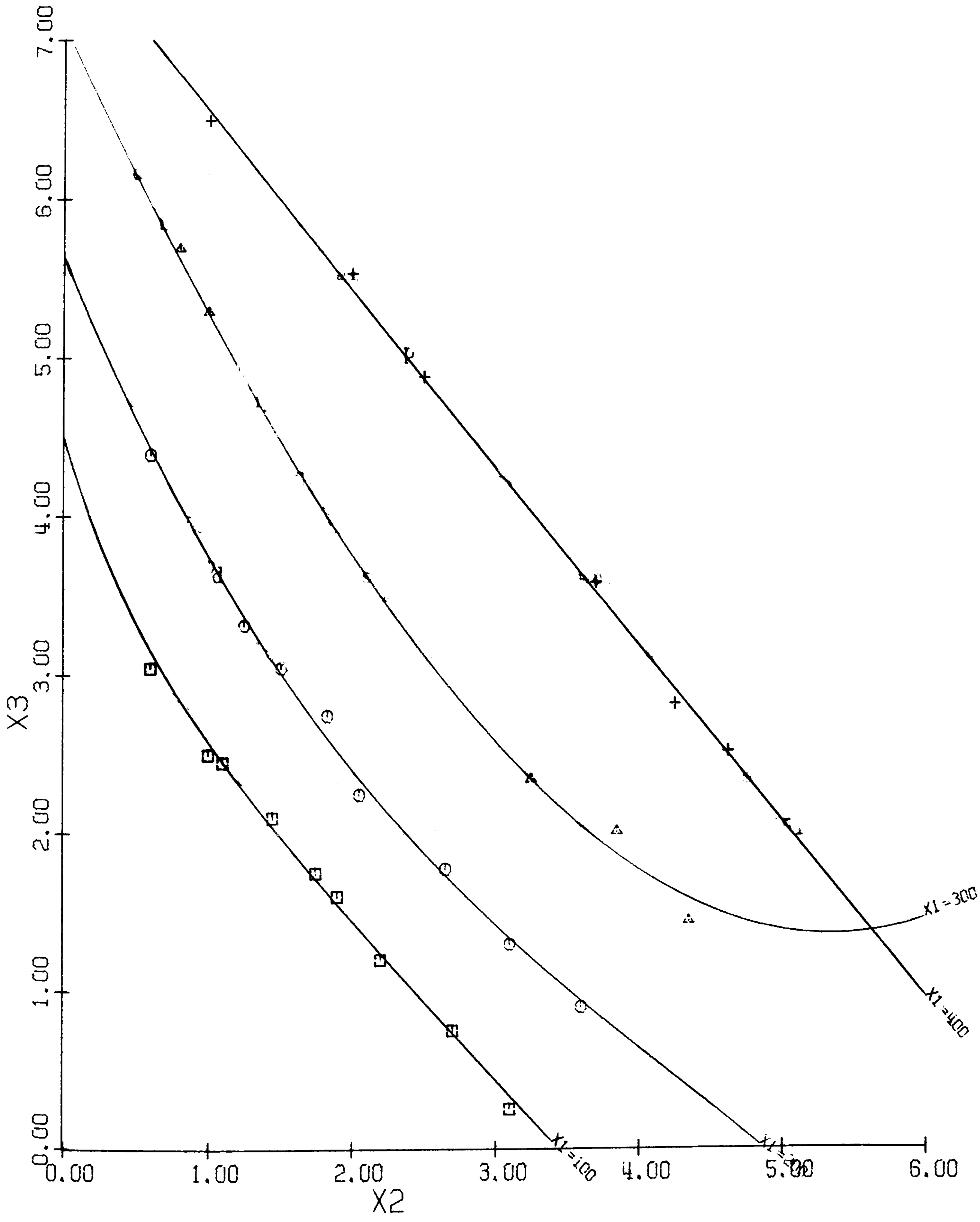
** INVOER **

```
IA     =    9      IPR    =    0      IFIT   =    1      IDENT  =   -1      ITSIK  =   10      MM     =   10      NPAS   =    3
IR     =   40      ICON   =    1      ILYN   =    1      IDRUK  =   -1      KA     =    1      NL     =    5      NDIGD  =  100
IK     =   11      ICOR   =    0      IPEN   =    3      IPLOT  =    1      KX     =    1      NN     =    3      NSOEK  =    1
IT     =    0      IDAT   =    0      IXAS   =    2      IPONS  =    0      LEESX  =    1      NGO    =    8      NUM    =    2
IAS    =    0      IDES   =    1      IYAS   =    3      IPUNT  =    1      M      =    8      NSK    =    1      N      =    9

IX     =    1     2     3
IV     =    0     0     0     0     0
IDXP   =   -1     2     2
IDG    =    2     2     2
KVX(I)=     1
GX     =  299.0      < X1       <  301.0     !
KVA(I)=     4
GA     = -8.000      <   A( 4)  <  1.000     !
EPS    =  1.00D-06   TYD    =  1.00E 01   YDUIM =  1.10E 01   FNU    =  1.00D-04
XBEG   =  0.0        XEND   =  6.00       YBEG  =  0.0        YEND   =  7.00        XLEN   =  6.00      YLEN   =  7.00

XP(I)  =  3.00E 02
TOLX   =  1.00E 00
FORM   =(8F10.0)


A(I)   =       23.679687        -2.1949019         0.34478224        -2.7022798 3D-02     -9.0225068 7D-02



RESULTATE

IER= 0   IT= 1   REGRESSIEKOEFFISIENT                                              =  9.99625541D-01
                 SOM VAN KWADRAATAFWYKINGS                                         =  1.32921631D-02
                 SOM ABSOLUTE FOUT                                                 =  2.91555610D-01
                 WORTEL GEMIDDELDE KWADRAATAFWYKING                                =  3.84305478D-02
                 WORTEL GEMIDDELDE KWADRAATAFW. VAN DIE PERSENTASIEFOUT            =  1.28326061D 00   PERSENT

A(I)=     8.08340808800 01    -1.92526172880 00    1.19018682730-01    7.68199184150-03    -2.21247598410 04


PASSING GEBASEER OP    9 PUNTE



VOORBEELD 5
```

C 32

VOORBEELD 5
X1 TUSSEN 99.00 EN 101.00

X1 □ 100 ⦶ 200 △ 300 + 400



6. 'n Toepassing Waar Slegs Grafieke Geteken Word

In hierdie voorbeeld word die program gebruik om slegs grafieke te teken. Daar is geen eksperimentele data, slegs die kaart met 'n ster in kolom 1 en die naamlyskaarte moet as data ingelees word.

Die funksie word in PASFUN, en die stipparameters word in die naamlys gedefinieer. Deur telkens die A-waardes te verander, kan verskillende grafieke op dieselfde assestelsel geteken word. Vir 'n nuwe assestelsel moet ons IAS=1 stel.

In Bylaag A p.A4 en A5, word die verskillende stipparameters verduidelik.

Die grafieke word geteken in die gebied begrens deur XBEG en XEND langs die X-as en deur YBEG en YEND langs die Y-as. Punte buite die gebied sal nie geteken word nie.

Die opskrifte bo-aan die grafiek word gestoor in die vektor PASF.

Omdat die funksie in PASFUN gedefinieer word, moet daar op gelet word dat NL > 0 is, verder moet IPLOT=1, IFIT=0, en LEESX=0 wees.

PASDIM VIR VOORBEELD 6
------ --- --------- -

```
C
      SUBROUTINE PASDIM
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(1),A(1)
      COMMON/DIM/IR,IK,IA
C
C  IN SUBROETINE PASDIM WORD DIE PARAMETERS IR,IK,IA GEGEE.
C
      IR=1
      IK=1
      IA=1
      RETURN
      ENTRY PASAF(X,A,F)
      ENTRY PASGEN(X,M)
C
C  GEEN DATA WORD GEGENEREER NIE.
C
      RETURN
      ENTRY PASFUN(X,A,FPAS)
C
C  FUNKSIE WAT GETEKEN WORD, WORD HIER GEDEFINEER.
C
      FPAS=A(1)*X(1)*X(1)+A(2)*X(1)+A(3)
      RETURN
      END
```

DATA VIR VOORBEELD 6
---- --- --------- -

```
   GEEN DATAPUNTE
 &LEES
 PASF='VOORBEELD 6    Y=AX**2+BX+C',
 XLEN=6.,YLEN=7.,IPLOT=1,IFIT=0,LEESX=0,NL=3,
  XBEG=0.,XEND=6.,YBEG=0.,YEND=7.,A=0.66666,-4.,6.,    &END
 &END
 &LEES IPEN=1,IAS=0,IDRUK=-2,A=0.33333,-2.,6.,  &END
 &LEES IPEN=1,A=0.55555,-3.33333,  &END
 &LEES IPEN=1,A=0.44444,-2.66666, &END
 &LEES IPEN=1,A=0.222222,-1.3333, &END
 &LEES IPEN=1,A=0.11111,-0.66666, &END
 &LEES IPEN=1,A=-0.11111,0.66666, &END
 &LEES IPEN=1,A=0.,0., &END
```

DATUM 77-09-22

TYD 13-20-36

** INVOER **

IA	=	1	IPR	=	1	IFIT	=	0	IDENT	=	-1	ITSIK	=	0	MM	=	2	NPAS	=	1
IR	=	1	ICON	=	1	ILYN	=	1	IDRUK	=	0	KA	=	0	NL	=	3	NDIGD	=	100
IK	=	1	ICOR	=	1	IPEN	=	1	IPLOT	=	1	KX	=	0	NN	=	2	NSOEK	=	0
IT	=	0	IDAT	=	0	IXAS	=	1	IPONS	=	0	LEESX	=	0	NGO	=	5	NUM	=	2
IAS	=	1	IDES	=	1	IYAS	=	2	IPUNT	=	0	M	=	0	NSK	=	1	N	=	0

IX = 1 2
IV = 0 0 0
IDXP = 2 2
IDG = 2 2
EPS = 1.00D-06 TYD = 1.00E 01 YDUIM = 1.10E 01 FNU = 1.00D-04
XBEG = 0.0 XEND = 6.00 YBEG = 0.0 YEND = 7.00 XLEN = 6.00 YLEN = 7.00

XP(I) = 0.0
FORM =

A(I) = 0.66666000 -4.0000000 6.0000000

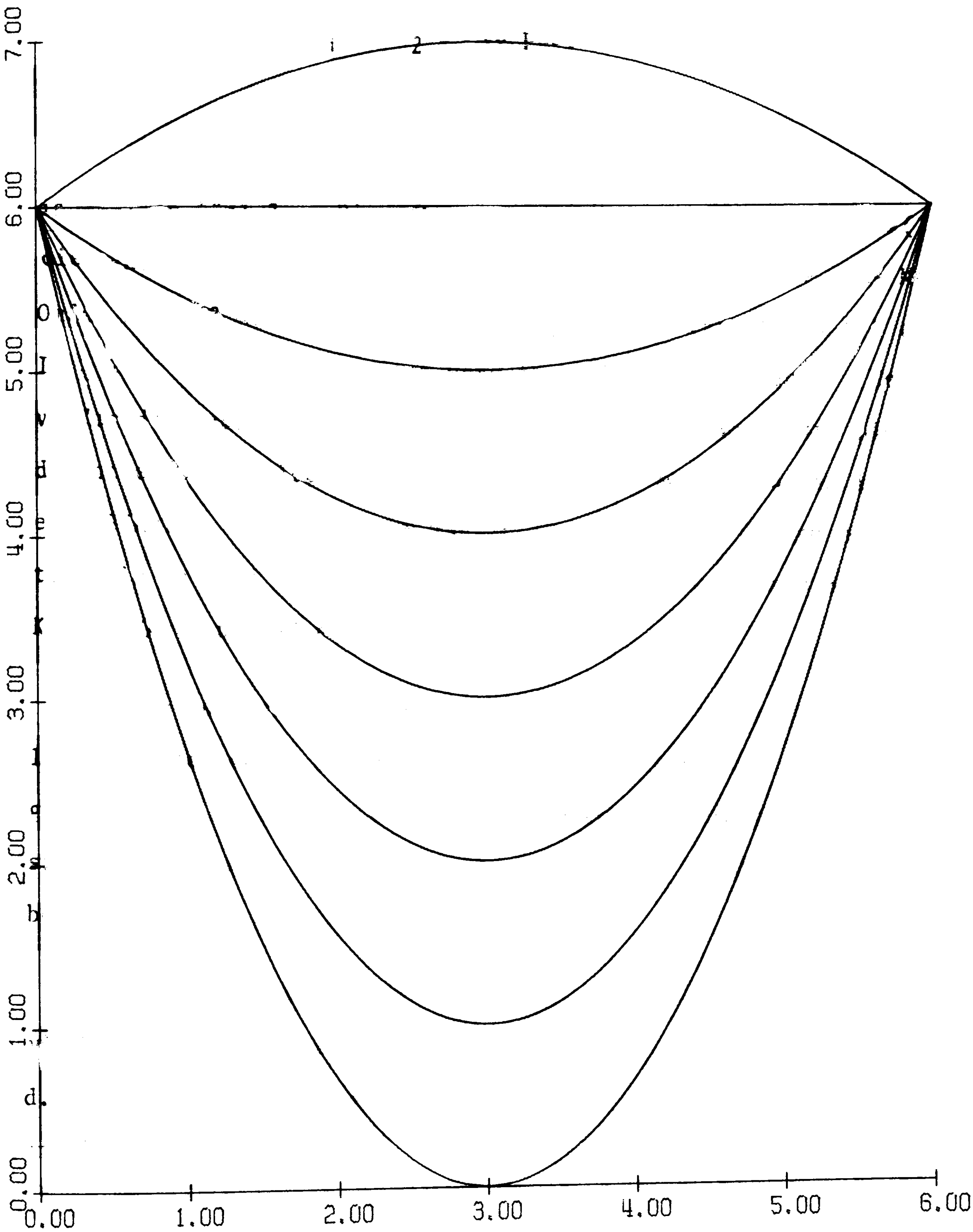
** RESULTATE **

TOTALE TYD IN REKENAAR = 33.82 SEK.
WERKLIKE TYD IN REKENAAR = 0.99 SEK.

C 36

7.

VOORBEELD 6   Y=AX**2+BX+C



7. 'n Nie-lineêre PASSING

Om die volgende funksie

$$y = a_1 \frac{1-e^{-a_2 X}}{1+e^{-a_2 X}}$$

wat nie-lineêr in a_1 en a_2 is te pas, moet ons die parameters soos volg kies.

Omdat daar twee verskillende a's is, moet NL=2 wees. Verder is NN=2 en IX=1,2. Die afhanklike veranderlike is IX(NN)=IX(2) en die aantal veranderlikes wat ingelees moet word is twee, dus M=2. Omdat daar 10 datastelle is moet IR $\geq$ 10 en verder is IK $\geq$ 2 en IA $\geq$ 2. Hierdie parameters word in PASDIM gedefinieer. Gestel verder dat daar nie meer as 40 iterasies gedoen moet word nie en dat na elke 10de iterasie herskalering gedoen moet word dan is IT=40 en ITSIK=10. Die maksimum waarde van X1 is 1. Om nou $e^{-a_2 X}$ nie te klein te laat word nie kan ons bv. a_2 begrens deur 0 en 6, dus KA=1, KVA=2 en GA=0,6.

In voorbeeld 7A word die datapunte gestip en vir verskillende stelle gekose a's die ooreenstemmende funksies geteken. Die grafieke van voorbeeld 7A toon dat die maksimum waarde van die funksie na a_1 streef. Die maksimumwaarde van X_2 sal dus 'n redelike beginwaarde vir a_1 wees.

In voorbeelde 7B is 'n passing met 1, 5 en 3 as beginwaardes vir die a's gekies. Indien die konvergensievereiste bevredig word, d.w.s. as IER=0 en as IPLOT=1, ILYN=1 en IPUNT=1, sal die funksie en die punte geteken word.

In voorbeeld 7C word NUM=0 gestel en die afgeleides

$\frac{\partial F}{\partial a_i}$, i=1, ---, ℓ word in PASAF bereken.

```
PASDIM VIR VOORBEELD 7
------ --- --------- -

C
      SUBROUTINE PASDIM
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(1),A(1),F(1)
      COMMON/DIM/IR,IK,IA
      IR=20
      IK=11
      IA=9
      IA=3
      RETURN
      ENTRY PASGEN(X,M)
      ENTRY PASFUN(X,A,FPAS)
      FPAS=A(1)*(1.D0-DEXP(-X(1)*A(2)))/(1.D0+DEXP(-X(1)*A(2)))
      RETURN
      ENTRY PASAF(X,A,F)
      A12=-A(2)*X(1)
      F(1)=(1.D0-DEXP(A12))/(1.D0+DEXP(A12))
      F(2)=2.D0*X(1)*A(1)*DEXP(A12)/(1.D0+DEXP(A12))**2
      RETURN
      END
0.1         0.3948
0.2         0.6599
0.3         1.2741
0.4         1.3281
0.5         1.7232
0.6         1.4673
0.7         1.6707
0.8         1.9433
0.9         1.7936
1.          1.9281


DATA VIR VOORBEELD 7
---- --- --------- -


*   EINDE DATAPUNTE
 &LEES PASF='VOORBEELD 7-A   Y=A(1)*(1-EXP(-X**A(2)))/(1+EXP(-X**A(2)))',
 FORM='(2F10.0)',M=2,IT=60,ITSIK=10,NL=2,IPR=0,NSOEK=1,XP=1.,A=1.5,3.0,
 KA=1,KVA=2,GA=0.0,6., OPSK='X1','X2',YBEG=0.,YEND=5.,YLEN=5.0,IFIT=0,
 XBEG=0.,XEND=2.5,XLEN=5.0,IDXP=-1,TOLX=5.,IPLOT=1,IPUNT=1, &END
 &LEES  IAS=0,IFIT=0,IPUNT=0,A=2.5,3.5,XP=2.,ICOR=0,IDRUK=-1,  &END
 &LEES  IAS=0,IFIT=0,IPUNT=0,A=1.,2.,XP=3.,  &END
 &LEES  IAS=0,IFIT=0,IPUNT=0,A=3.,5.,XP=4.,GA(2,1)=8.,   &END
 &LEES  IAS=0,IFIT=0,IPUNT=0,A=4.,8.,XP=5.,GA(2,1)=8.,  &END
 &LEES PASF='VOORBEELD 7-C',IPLOT=0,XP=4.,A=1.5,3.,NUM=0,KA=0,IFIT=1, &END
 &LEES PASF='VOORBEELD 7-B                                                  ',
 IAS=1,A=1.5,3.0,IFIT=1,XP=1.,IPUNT=1,IPEN=1,IDRUK=0,NUM=1,IPLOT=1, &END
```

TYD 12- 5-39

** INVOER **

0	IDENT	=	-1	ITSIK	=	10	MM	=	2	NPAS	=	1
1	IDRUK	=	0	KA	=	1	NL	=	2	NDIGD	=	100
1	IPLOT	=	1	KX	=	0	NN	=	2	NSOEK	=	1
1	IPONS	=	0	LEESX	=	1	NGO	=	5	NUM	=	2
2	IPUNT	=	1	M	=	2	NSK	=	1	N	=	10

1.10E 01 FNU = 1.000D-04
0.0 YEND = 5.00 XLEN = 5.00 YLEN = 5.00

** RESULTATE **

C 40

X2 TUSSEN 0.0 EN 5.000

```
DATUM 77-09-22

IA       =     3       IPR     =     0       IFIT   =
IP       =    20       ICON    =     1       ILYN   =
IK       =    11       ICOR    =     1       IPEN   =
IT       =    60       IDAT    =     0       IXAS   =
IAS      =     1       IDES    =     1       IYAS   =

IX       =     1      2
IV       =     0      0
IDXP     =    -1      2
IDG      =     2      2
KVA(I)=        2
GA       = 0.0            <    A( 2)   <   6.000        I
EPS      =  1.00D-06   TYD     =  1.00E 01   YDUIM =
XBEG     =  0.0        XEND    =  2.50       YBEG  =

XP(I)    =  1.00E 00
TOLX     =  5.00E 00
FORM     =(2F10.0)

A(I)     =        1.5000000           3.0000000




MATRIKS MET KORRELASIEKOEFFISIENTE

    I          1       2

    1        1.00    0.90
    2        0.90    1.00



PUNTE WORD GEKARAKTERISEER DEUR
   X1          = 1.000
BREUKTOLERANSIES --X1          I5.000000--
      X1         TUSSEN 0.0          EN   2.500
```

** INVOER **

```
IA    =   3      IPR   =  0     IFIT  =  1     IDENT =  -1    ITSIK =  10    MM    =  2     NPAS  =   1
IR    =  20      ICON  =  1     ILYN  =  1     IDRUK =   0    KA    =   0    NL    =  2     NDIGD = 100
IK    =  11      ICOR  =  0     IPEN  =  1     IPLOT =   1    KX    =   0    NN    =  2     NSOEK =   1
IT    =  60      IDAT  =  0     IXAS  =  1     IPONS =   0    LEESX =   0    NGO   =  5     NUM   =   1
IAS   =   1      IDES  =  1     IYAS  =  2     IPUNT =   1    M     =   2    NSK   =  1     N     =  10

IX    =   1   2
IV    =   0   0
IDXP  =  -1   2
IDG   =   2   2
EPS   =  1.00D-06   TYD   =  1.00E 01   YDUIM =  1.10E 01   FNU   =  1.00D-04
XBEG  =  0.0        XEND  =  2.50       YBEG  =  0.0        YEND  =  5.00      XLEN  =  5.00      YLEN  =  5.00

XP(I) =  1.00E 00
TOLX  =  5.00E 00
FORM  = (2F10.0)


A(I)  =        1.5000000          3.0000000
```

** RESULTATE **

MINIMERINGSTYD IS   0.31 SEK.

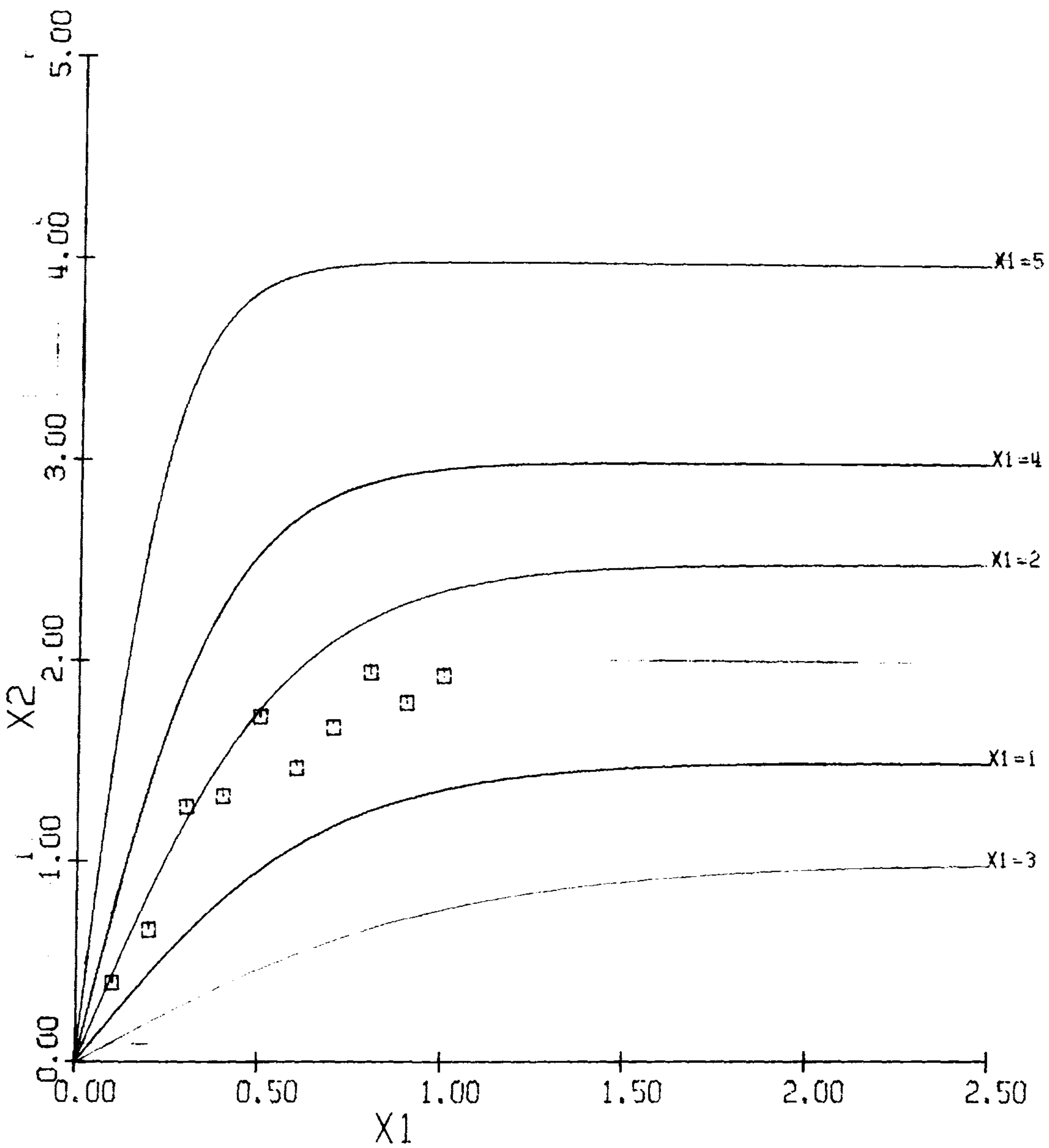
NR.	X1	X2	BEREKEN	RESIDU
1	0.1000	0.3948	0.4186	-0.0238
2	0.2000	0.6599	0.7994	-0.1395
3	0.3000	1.2741	1.1171	0.1570
4	0.4000	1.3281	1.3636	-0.0355
5	0.5000	1.7232	1.5443	0.1789
6	0.6000	1.4673	1.6713	-0.2040
7	0.7000	1.6707	1.7579	-0.0872
8	0.8000	1.9433	1.8159	0.1274
9	0.9000	1.7936	1.8541	-0.0605
10	1.0000	1.9281	1.8791	0.0490

RESULTATE

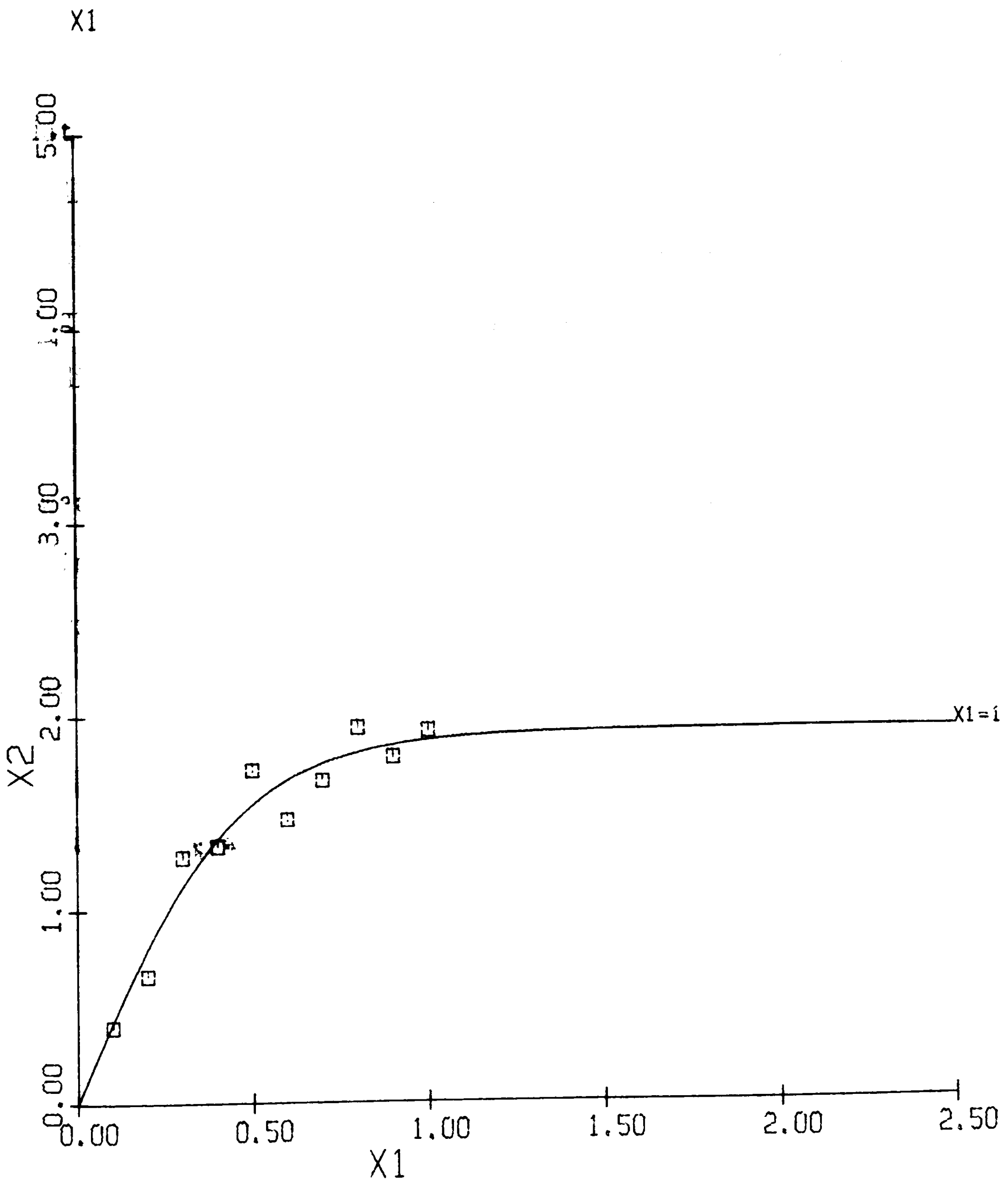
```
IER= 0   IT= 7   REGRESSIEKOEFFISIENT                                          =  9.697793170-01
                 SOM VAN KWADRAATAFWYKINGS                                     =  1.494651040-01
                 SOM ABSOLUTE FOUT                                             =  1.062921590 00
                 WORTEL GEMIDDELDE KWADRAATAFWYKING                            =  1.222559220-01
                 WORTEL GEMIDDELDE KWADRAATAFW. VAN DIE PERSENTASIEFOUT        =  1.015635560 01   PERSENT

A(I)=     1.92482095510 00     4.42025518770 00

PASSING GEBASEER OP  10 PUNTE
```

VOORBEELD 7-A Y=A(1)*(1-EXP(-X**A(2)))/(1+EXP(-X**A(2)))

X1 □ 1 ○ 2 △ 3 + 4 × 5



VOORBEELD 7-B



8. Bepaling Grootte van Program vir 'n Lineêre Passing

Ons moet nog bepaal watter waarde aan K (rekenaargeheue) op die taakkaart toegeken moet word vir verskillende waardes van IR, IK en IA wat in PASDIM gedefinieer word. Ons het verskillende eksperimentele lopies gedoen en die hoeveelheid geheue K wat gebruik is genoteer. Sien eksperimentele data op bladsy C46. Die eerste kolom stel IR, die tweede IK, die derde die IA en die vierde die hoeveelheid K gebruik, voor. Die volgende funksie is deur die data gepas.

$$K = a_1 + a_2 \quad (IR+2)(IK+IA+4) + \{maks\ (IK,\ IA) + 2\}^2$$

Omdat die funksie lineêr is, word IT=0 gestel. Die aantal veranderlikes wat in die passing gebruik word is vier, dus NN=4 en IX=2,1,3,4 met IX(NN) = IX(4) = 4, die afhanklike veranderlike. Die grafieke van K teen IA is geteken, dus IX(NN-1) = IX(3) = 3 vir die X-as. In totaal is daar 4 veranderlikes, dus 2 moet konstant gehou word, dit beteken dat NSOEK=2 is. Ons hou nou die IK-waarde vir elke assestelsel konstant en teken die verskillende grafieke met IR respektiewelik 10, 50, 100, 200, 300, 400 en 500. Die volgorde van die veranderlikes wat varieer is dus 2, 1, 3 met 3 die X-as-veranderlike, dus XP = 10, 10, vir die eerste grafiek.

Die waarde wat op die taakkaart aan K toegeken moet word kan van die grafieke afgelees word.

PASDIM VIR VOORBEELD 8
------ --- --------- -

```
C
      SUBROUTINE PASDIM
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(1),A(1)
      COMMON/DIM/IR,IK,IA
C
C  BESKRYWING PARAMETERS
C
C  IR    = MAKSIMUM AANTAL DATAPUNTE WAT IN DIE PASSINGS GEBRUIK WORD,
C          EN MOET GROTER AS NUL WEES.
C  IK    = TOTALE AANTAL VERANDERLIKES, INGELEES PLUS GEGENEREER,
C          MET  0 < IK < 50 .
C  IA    = AANTAL A-WAARDES WAT IN FUNKSIE PLUS A-WAARDES WAT LINEER
C          VOORKOM, MET  0 < IA < 50 .
C
      IR=100
      IK=10
      IA=2
      RETURN
      ENTRY PASAF(X,A,F)
      ENTRY PASGEN(X,M)
      X1=X(1)+2.
      X2=X(2)+2.
      X3=X(3)+2.
      X(5)=DMAX1(X2,X3)
      X(5)=X(5)*X(5)
      X(6)=X(5)/X1
      X6=X(6)
      X(7)=X1*(X2+X3+X6)
      RETURN
      ENTRY PASFUN(X,A,FPAS)
C
C  BEREKEN DIE WAARDE VAN DIE PASSINGSFUNKSIE BY DIE I-DE DATAPUNTE
C
      X1=X(1)+2.
      X2=X(2)+2.
      X3=X(3)+2.
      X5   =DMAX1(X2,X3)
      X5=X5*X5
      X6=X5/X1
      X7   =X1*(X2+X3+X6)
      FPAS=A(1)+A(2)*X7
      RETURN
      END
```

DATA VIR VOORBEELD 8
---- --- --------- -

```
100.        10.         20.         124.
100.        20.         10.         124.
100.        20.         20.         132.
100.        30.         40.         166.
100.        30.         50.         182.
100.        40.         20.         158.
100.        40.         50.         190.
100.        50.         50.         198.
```

```
300.      10.       10.       152.
300.      10.       40.       236.
300.      20.       30.       228.
300.      20.       50.       290.
300.      30.       10.       206.
300.      30.       40.       282.
300.      30.       50.       314.
300.      40.       50.       338.
500.      50.       20.       406.
300.      50.       50.       362.
500.      10.       50.       366.
500.      20.       30.       314.
500.      20.       50.       406.
500.      30.       20.       314.
500.      30.       50.       444.
500.      40.       50.       484.
500.      50.       50.       522.
*   EINDE DATAPUNTE
 &LEFS PASF='VOORBEELD 8',IDRUK=1,NL=2,M=4,FORM='(8F10.0)',
 OPSK='IR','IK','IA','K',NN=4,IX=2,1,3,4,IPUNT=0,XLEN=6.,YLEN=7.,
 IT=0,IDAT=0,IPLOT=1,ILYN=1,NSOEK=2,XBEG=0.,XEND=60.,YBEG=0.,
 YEND=700.,IFIT=1,XP=10.,100.,IDXP=-1,-1,IDENT=-1,&END
 &LEFS XP=10.,200.,IAS=0,IDRUK=-2,IFIT=0,IDAT=0,   &END
 &LEFS XP=10.,300., IAS=0,   &END
 &LEFS XP=10.,400., IAS=0,   &END
 &LEFS XP=10.,500., IAS=0,   &END
 &LEFS XP=10.,600., IAS=0,   &END
 &LEFS XP=10.,50., IAS=0, IPUNT=0, &END
 &LEFS XP=10.,10., IAS=0,  &END
 &LEFS XP=20.,50., IAS=1,IPEN=1, &END
 &LEFS XP=20.,100.,IAS=0,  &END
 &LEFS XP=20.,200.,IAS=0,  &END
 &LEFS XP=20.,300.,IAS=0,  &END
 &LEFS XP=20.,400.,IAS=0,  &END
 &LEFS XP=20.,500.,IAS=0,  &END
 &LEFS XP=20.,600.,IAS=0,  &END
 &LEFS XP=30.,50., IAS=1,IPEN=1, &END
 &LEFS XP=30.,100.,IAS=0,  &END
 &LEFS XP=30.,200.,IAS=0,  &END
 &LEFS XP=30.,300.,IAS=0,  &END
 &LEFS XP=30.,400.,IAS=0,  &END
 &LEFS XP=30.,500.,IAS=0,  &END
 &LEFS XP=30.,600.,IAS=0,  &END
 &LEFS XP=40.,50., IAS=1,IPEN=1, &END
 &LEFS XP=40.,100.,IAS=0,  &END
 &LEFS XP=40.,200.,IAS=0,  &END
 &LEFS XP=40.,300.,IAS=0,  &END
 &LEFS XP=40.,400.,IAS=0,  &END
 &LEFS XP=40.,500.,IAS=0,  &END
 &LEFS XP=40.,600.,IAS=0,  &END
 &LEFS XP=50.,50., IAS=1,IPEN=1, &END
 &LEFS XP=50.,100.,IAS=0,  &END
 &LEFS XP=50.,200.,IAS=0,  &END
 &LEFS XP=50.,300.,IAS=0,  &END
 &LEFS XP=50.,400.,IAS=0,  &END
 &LEFS XP=50.,500.,IAS=0,  &END
 &LEFS XP=50.,600.,IAS=0,  &END
```

DATUM 77-09-23 TYD 9-12-20

** INVOER **

IA	=	2	IPR	=	1	IFIT	=	1	IDENT	=	-1	ITSIK	=	0	MM	=	7	NPAS	=	1
IR	=	100	ICON	=	1	ILYN	=	1	IDRUK	=	1	KA	=	0	NL	=	2	NDIGD	=	100
IK	=	10	ICOR	=	1	IPEN	=	1	IPLOT	=	1	KX	=	0	NN	=	4	NSOEK	=	2
IT	=	0	IDAT	=	0	IXAS	=	3	IPONS	=	0	LEESX	=	1	NGO	=	5	NUM	=	2
IAS	=	1	IDES	=	1	IYAS	=	4	IPUNT	=	0	M	=	4	NSK	=	1	N	=	25

IX = 2 1 3 4
IV = 0 0
IDXP = -1 -1 2 2
IDG = 2 2 2 2
EPS = 1.00D-06 TYD = 1.00E 01 YDUIM = 1.10E 01 FNU = 1.00D-04
XBEG = 0.0 XEND = 60.00 YBEG = 0.0 YEND =700.00 XLEN = 6.00 YLEN = 7.00

XP(I) = 1.00E 01 1.00E 02
TOLX = 5.00E-02 5.00E-02
FORM =(8F10.0)

A(I) = 0.0 0.0

** RESULTATE **

MINIMERINGSTYD IS 0.90 SEK.

NR.	IR	IK	IA	K			
1	100.0000	10.0000	20.0000	124.0000	484.0000	4.7451	3952.0000
2	100.0000	20.0000	10.0000	124.0000	484.0000	4.7451	3952.0000
3	100.0000	20.0000	20.0000	132.0000	484.0000	4.7451	4972.0000
4	100.0000	30.0000	40.0000	166.0000	1764.0000	17.2941	9312.0000
5	100.0000	30.0000	50.0000	182.0000	2704.0000	26.5098	11272.0000
6	100.0000	40.0000	20.0000	158.0000	1764.0000	17.2941	8292.0000
7	100.0000	40.0000	50.0000	190.0000	2704.0000	26.5098	12292.0000
8	100.0000	50.0000	50.0000	198.0000	2704.0000	26.5098	13312.0000
9	300.0000	10.0000	10.0000	152.0000	144.0000	0.4768	7392.0000
10	300.0000	10.0000	40.0000	236.0000	1764.0000	5.8411	18072.0000
11	300.0000	20.0000	30.0000	228.0000	1024.0000	3.3907	17332.0000
12	300.0000	20.0000	50.0000	290.0000	2704.0000	8.9536	25052.0000
13	300.0000	30.0000	10.0000	206.0000	1024.0000	3.3907	14312.0000
14	300.0000	30.0000	40.0000	282.0000	1764.0000	5.8411	24112.0000
15	300.0000	30.0000	50.0000	314.0000	2704.0000	8.9536	28072.0000
16	300.0000	40.0000	50.0000	338.0000	2704.0000	8.9536	31092.0000
17	500.0000	50.0000	20.0000	406.0000	2704.0000	5.3865	39852.0000
18	300.0000	50.0000	50.0000	362.0000	2704.0000	8.9536	34112.0000
19	500.0000	10.0000	50.0000	366.0000	2704.0000	5.3865	34832.0000
20	500.0000	20.0000	30.0000	314.0000	1024.0000	2.0398	28132.0000
21	500.0000	20.0000	50.0000	406.0000	2704.0000	5.3865	39852.0000
22	500.0000	30.0000	20.0000	314.0000	1024.0000	2.0398	28132.0000
23	500.0000	30.0000	50.0000	444.0000	2704.0000	5.3865	44872.0000
24	500.0000	40.0000	50.0000	484.0000	2704.0000	5.3865	49892.0000
25	500.0000	50.0000	50.0000	522.0000	2704.0000	5.3865	54912.0000
GEMM.	300.0000	29.2000	36.4000	277.5200	1916.0000	8.7803	23495.2000
STD AFW.	163.2993	13.2035	15.5134	117.1029	927.3439	7.7493	14958.9780

MATRIKS MET KORRELASIEKOEFFISIENTE

I	1	2	3	4	5	6	7
1	1.00	0.04	0.20	0.86	0.29	-0.61	0.86
2	0.04	1.00	0.31	0.39	0.57	0.43	0.39
3	0.20	0.31	1.00	0.54	0.86	0.42	0.54
4	0.86	0.39	0.54	1.00	0.64	-0.31	1.00
5	0.29	0.57	0.86	0.64	1.00	0.45	0.64
6	-0.61	0.43	0.42	-0.31	0.45	1.00	-0.31
7	0.86	0.39	0.54	1.00	0.64	-0.31	1.00

NR.	IK	IR	IA	K	BEREKEN	RESIDU
1	10.0000	100.0000	20.0000	124.0000	124.5332	-0.5332
2	20.0000	100.0000	10.0000	124.0000	124.5332	-0.5332
3	20.0000	100.0000	20.0000	132.0000	132.5179	-0.5179
4	30.0000	100.0000	40.0000	166.0000	166.4920	-0.4920
5	30.0000	100.0000	50.0000	182.0000	181.8351	0.1649
6	40.0000	100.0000	20.0000	158.0000	158.5073	-0.5073
7	40.0000	100.0000	50.0000	190.0000	189.8198	0.1802
8	50.0000	100.0000	50.0000	198.0000	197.8045	0.1955
9	10.0000	300.0000	10.0000	152.0000	151.4620	0.5380
10	10.0000	300.0000	40.0000	236.0000	235.0665	0.9335
11	20.0000	300.0000	30.0000	228.0000	229.2736	-1.2736
12	20.0000	300.0000	50.0000	290.0000	289.7068	0.2932
13	30.0000	300.0000	10.0000	206.0000	205.6327	0.3673
14	30.0000	300.0000	40.0000	282.0000	282.3484	-0.3484
15	30.0000	300.0000	50.0000	314.0000	313.3478	0.6522
16	40.0000	300.0000	50.0000	338.0000	336.9888	1.0112
17	50.0000	500.0000	20.0000	406.0000	405.5633	0.4367
18	50.0000	300.0000	50.0000	362.0000	360.6298	1.3702
19	10.0000	500.0000	50.0000	366.0000	366.2660	-0.2660
20	20.0000	500.0000	30.0000	314.0000	313.8175	0.1825
21	20.0000	500.0000	50.0000	406.0000	405.5633	0.4367
22	30.0000	500.0000	20.0000	314.0000	313.8175	0.1825
23	30.0000	500.0000	50.0000	444.0000	444.8605	-0.8605
24	40.0000	500.0000	50.0000	484.0000	484.1577	-0.1577
25	50.0000	500.0000	50.0000	522.0000	523.4550	-1.4550

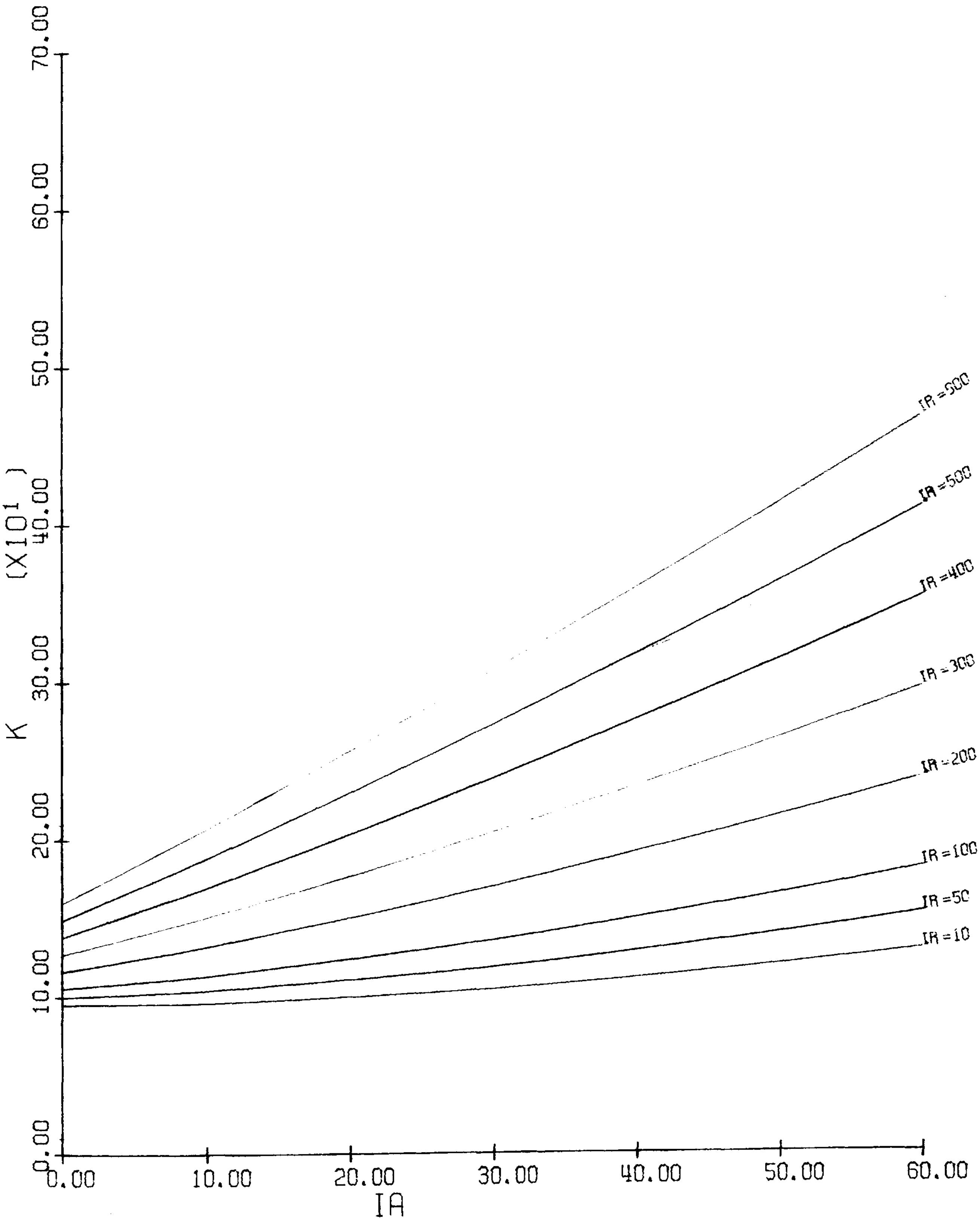
RESULTATE

```
IER= 0   IT= 1   REGRESSIEKOEFFISIENT                                  =  9.99982854D-01
                 SOM VAN KWADRAATAFWYKINGS                             =  1.12860280D 01
                 SOM ABSOLUTE FOUT                                     =  1.38894818D 01
                 WORTEL GEMIDDELDE KWADRAATAFWYKING                    =  6.71893683D-01
                 WORTEL GEMIDDELDE KWADRAATAFW. VAN DIE PERSENTASIEFOUT =  2.69820712D-01  PERSENT

A(I)=     9.3596385873D 01     7.8281357097D-03

PASSING GEBASEER OP  25 PUNTE
```

VOORBEELD 8

IK = 10

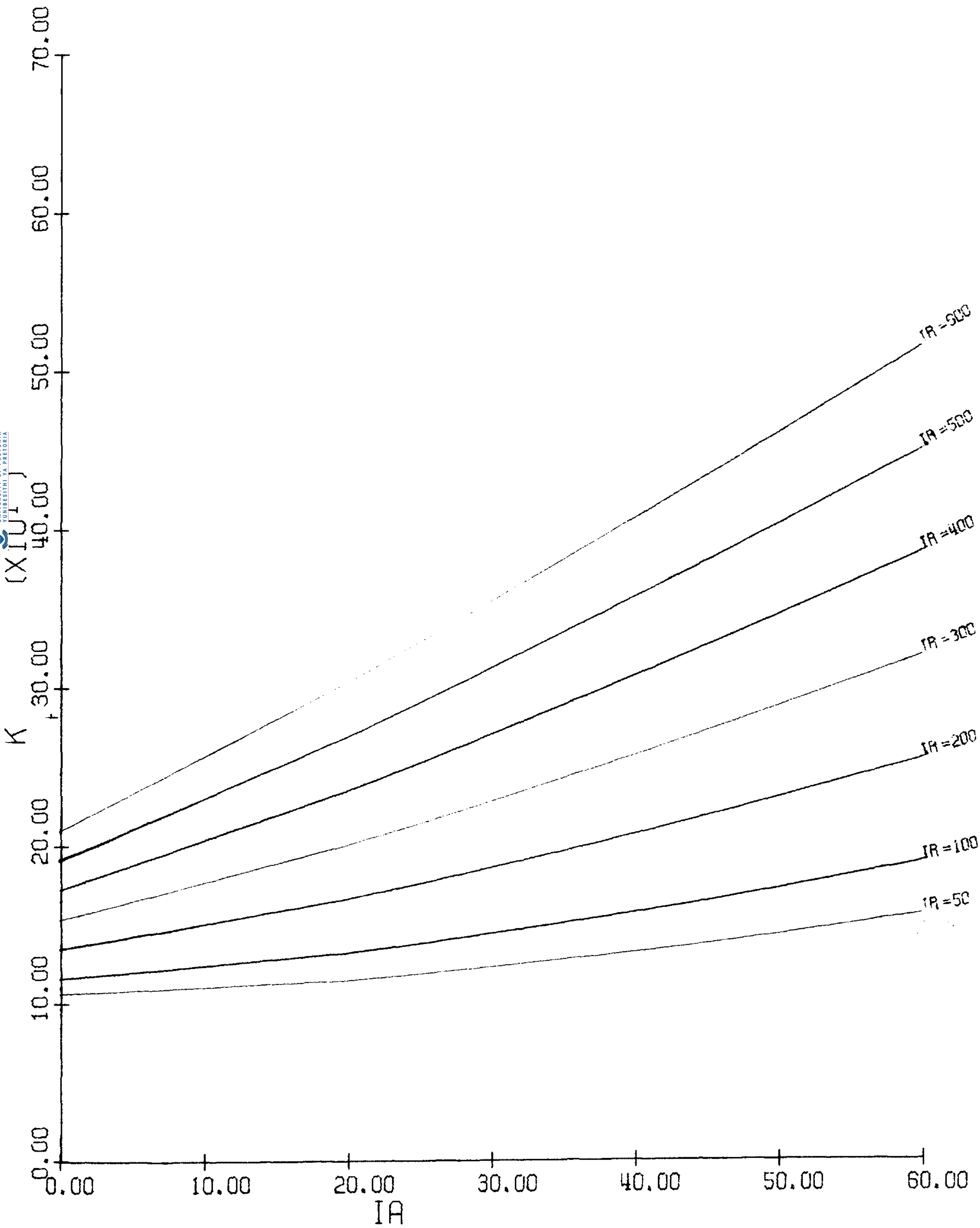
IR □ 100 ○ 200 △ 300 + 400 × 500 ◇ 600 ↑ 50 ⊠ 10



VOORBEELD 8

IK = 20

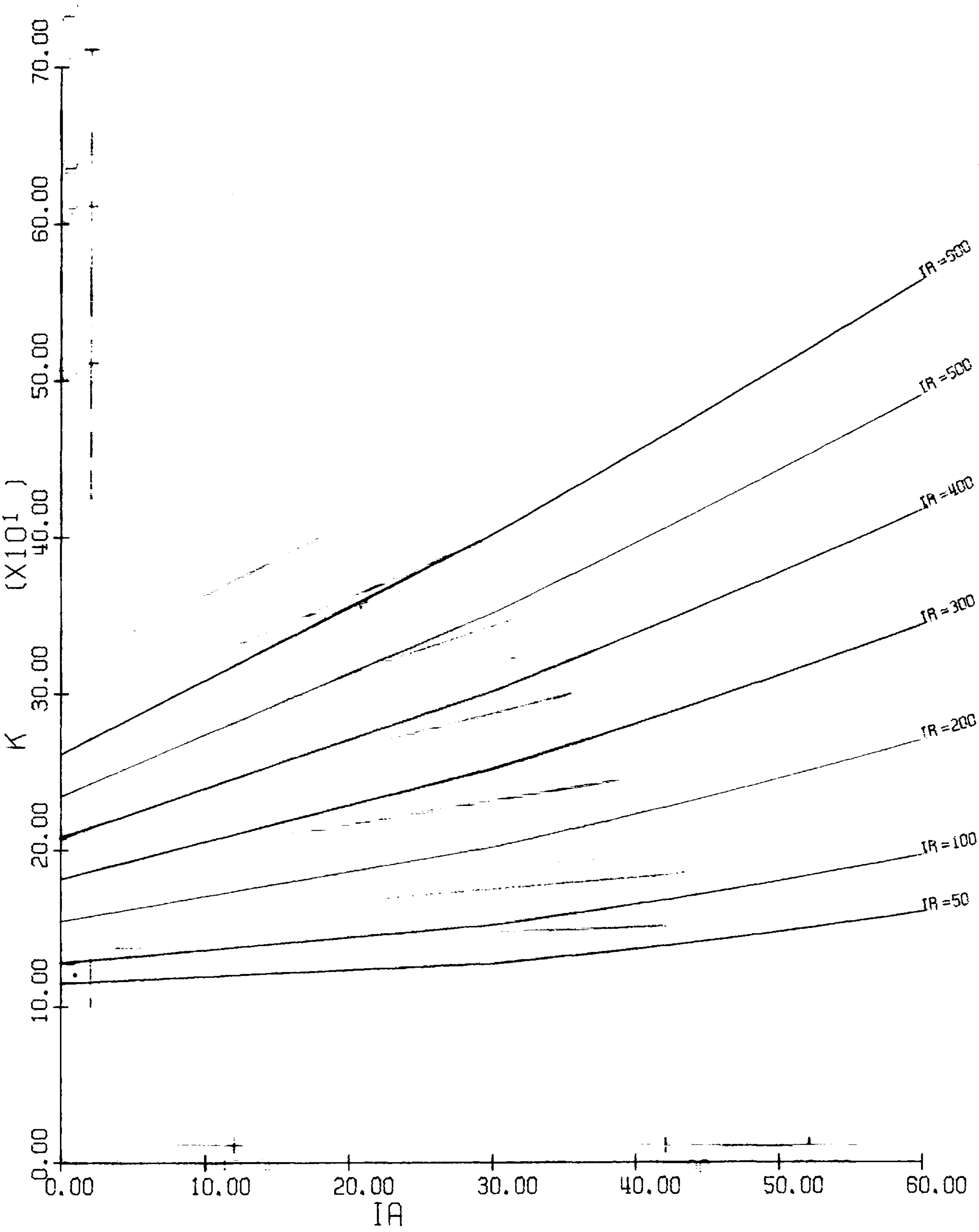
IR 50 100 200 300 400 500 600



VOORBEELD 8

IK ^{IK} = 30

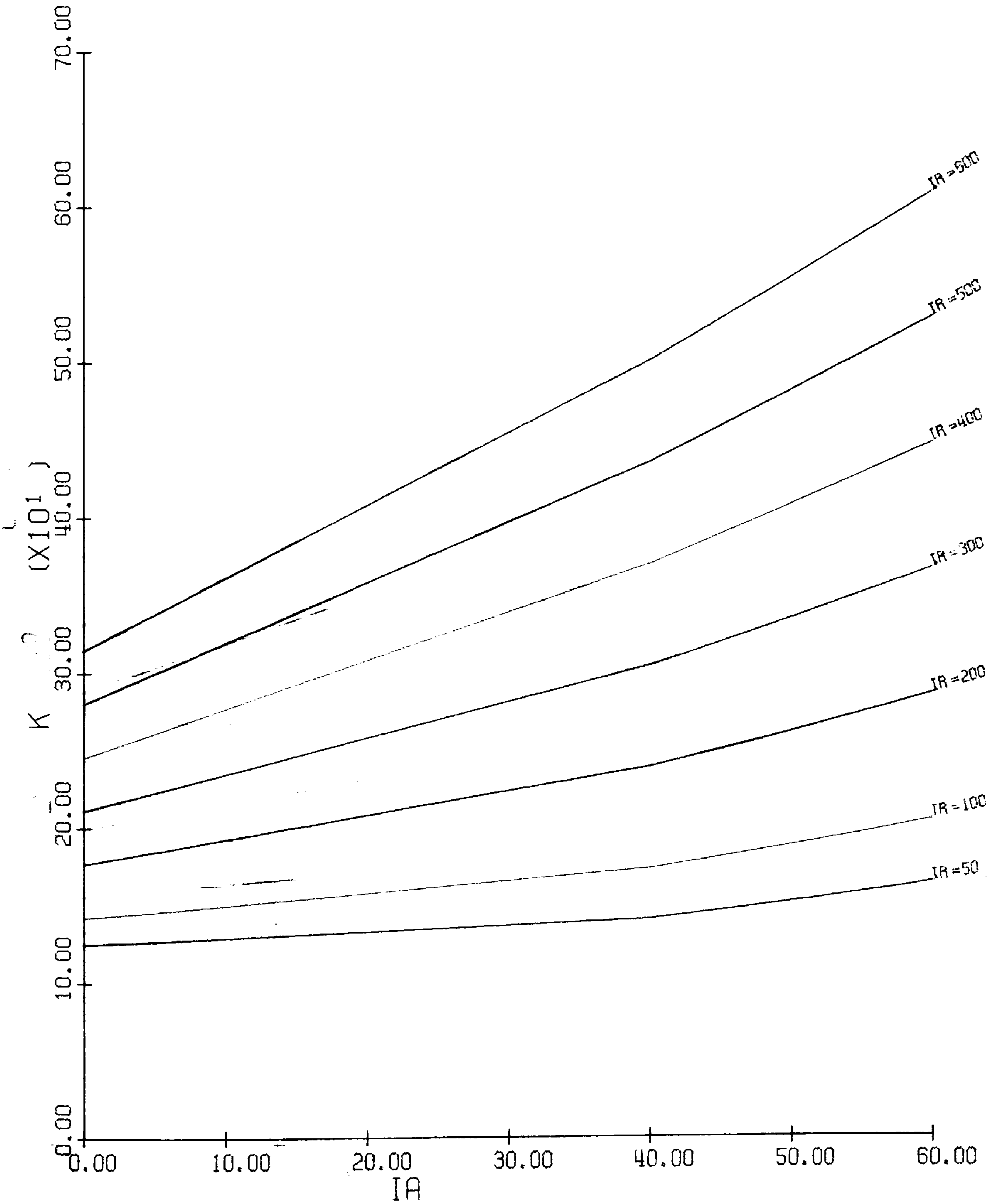
IR ☐ 50 ⦶ 100 △ 200 + 300 × 400 ◇ 500 ↑ 600



VOORBEELD 8

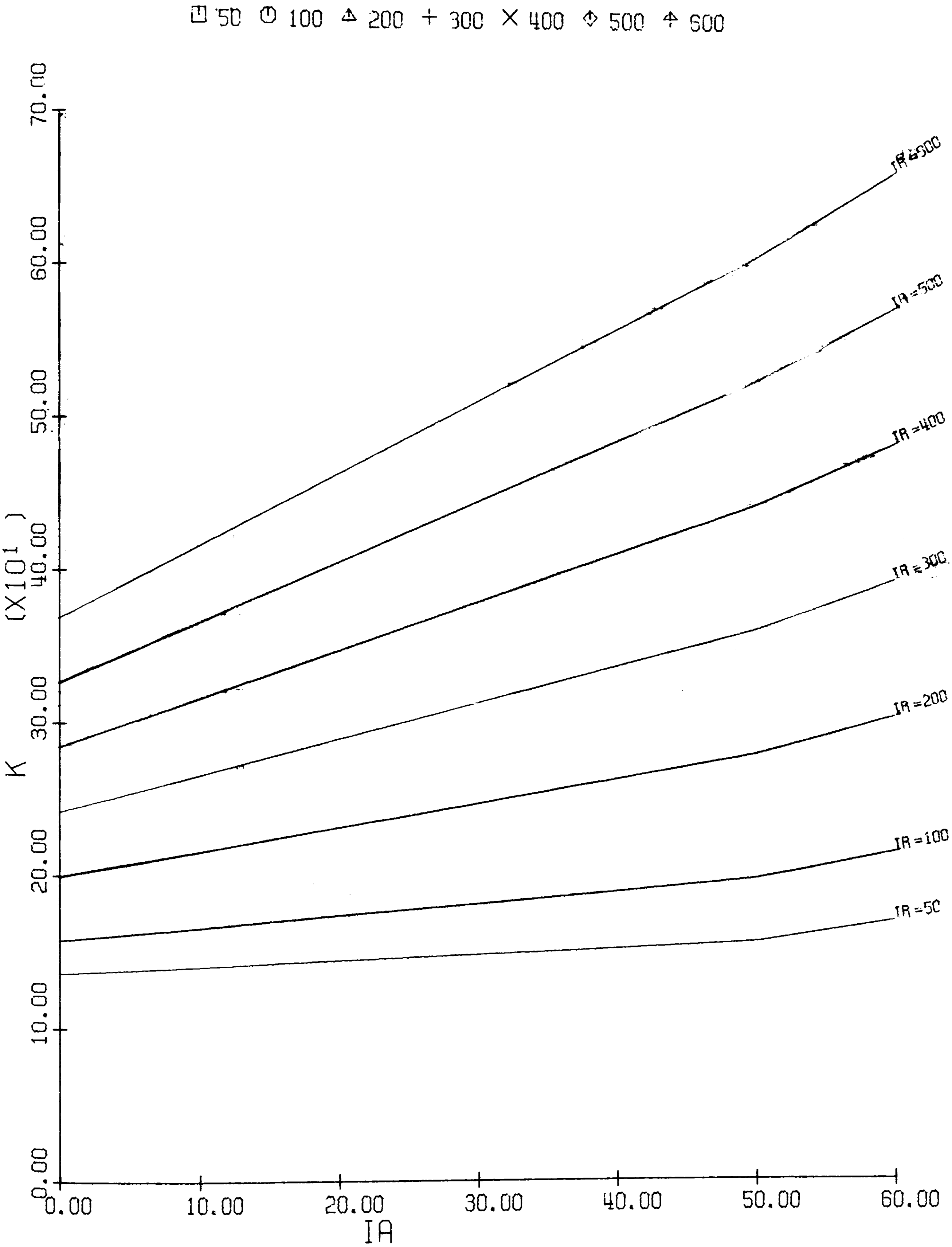
IK = 40

IR □ 50 ○ 100 △ 200 + 300 × 400 ◇ 500 ↑ 600



VOORBEELD A

IK = 50



Bladsy	Reël	
(iii)	23, 24, 25, 26, 27	58 moet wees: 59
20	13 en 14	moet lees: ..... van A_{i-1} wat gevorm word deur die ..... kolomme van A weg .....
22	15	moet lees: .... rang van A is,
30	7	moet lees: .... -metode en
30	8	Die moet wees: die.
35	17	moet lees: r^i is 'n ..... van $\|\|f(x)\|\|$, ekwivalent is.
37	4	moet lees: ..... h-vektor bereken word. Die algoritme ....
39	1	Die moet wees: die
69	17	moet lees: ..... en dus geen grafiek
83	11	moet lees: ..... Suid-Afrika 1969.
84	17	moet lees: ..... New Jersey, 1968.
85	3	A rapidly convergent descent .....
85	4	moet wees: Journal, 6 (1963), 163-168.
85	9	moet wees: ..... Computer. J. Inst. Maths. Applic. 19 (1977), 231-237.
85	11	(1971)
85	14	Vol. 2 No. 2 (1965), 431-441
C13	13	skrap: word
C27	17	skrap: gepas word.
C34	3	moet lees: ..... data nie, slegs die ...