Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

S.A. van de Geer, M.M. Voors

Een programma voor het twee-fasen regressiemodel met breukvlak

Afdeling Mathematische Statistiek Notitie MS-N8602 November

A program for the two-phase broken plane regression model

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

Een programma voor het twee-fasen regressiemodel met breukvlak

Sara van de Geer, Michel Voors
Centrum voor Wiskunde en Informatica
Postbus 4079, 1009 AB Amsterdam

Voor een door SARA VAN DE GEER ontwikkeld algoritme voor het bepalen van kleinste kwadraten schatters in een twee-fasen regressie model is een programma in Pascal gemaakt.

1980 Mathematics Subject Classification: 62J02

1. INLEIDING.

We beschouwen het twee-fasen regressie model

$$y_k = \min(\alpha_0 + \alpha_1 x_{k1} + \alpha_2 x_{k2},\ \beta_0 + \beta_1 x_{k1} + \beta_2 x_{k2}) + \epsilon_k\ ,\quad k = 1,2,...,n \tag{1.1}$$

Uit de data $x_1,x_2,\cdots,x_n$ en $y_1,y_2,..,y_n$, waarin $x_i = (1,x_{i1},x_{i2})'$, wil je de kleinste kwadraten schatters $\hat{\alpha},\hat{\beta}$ en σ_ϵ^2 van α,β en de residuele kwadratensom bepalen. Het model is te herschrijven als, $k=1,2,...,n$

$$y_k = \begin{cases} \alpha' x_k & \Leftrightarrow \alpha' x_k \leqslant \beta' x_k \quad \Leftrightarrow (\alpha-\beta)' x_k \leqslant 0 \\ \beta' x_k & \Leftrightarrow \alpha' x_k \geqslant \beta' x_k \quad \Leftrightarrow (\alpha-\beta)' x_k \geqslant 0 \end{cases} \tag{1.2}$$

Bekijk eerst het meer algemene probleem, $k=1,2,...,n$

$$y_k = \begin{cases} \alpha' x_k & \Leftrightarrow \gamma' x_k \leqslant 0 \\ \beta' x_k & \Leftrightarrow \gamma' x_k \geqslant 0 \end{cases} \tag{1.3}$$

waarin $\gamma \in \mathbb{R}^3$. Neem nu alle mogelijke partities van $x_1,x_2,\cdots,x_n$ en bepaal voor elke partitie parameterschattingen (merk op dat het vlak $\gamma' x$ ook een partitie bepaalt). Neem nu de schatting met minimale σ_ϵ^2 als kleinste kwadraten schatting. Omdat je echter niet met een willekeurige γ te maken hebt, maar met $\gamma = \alpha-\beta$, moet je achteraf kijken of de schattingen $\hat{\alpha}$ en $\hat{\beta}$ dezelfde partitie opleveren. Dit is het *controleren van de continuïteitsrestrictie.* Zo kom je tot je uiteindelijke parameterschattingen.

Notitie MS-N8602
Centrum voor Wiskunde en Informatica
Postbus 4079, 1009 AB Amsterdam

2. HET ALGORITHME.
Het genereren van de partities gaat als volgt.
Sorteer de $x = (x_{i1}, x_{i2})$ door de eerste coördinaten in opklimmende grootte achter elkaar te zetten, d.w.z. $x_{11} \leqslant x_{21} \leqslant \ldots \leqslant x_{n1}$. Bij gelijke eerste coördinaten zet je de tweede coordinaten in opklimmende volgorde. Bepaal nu de $m = n(n-1)/2$ hellingen $S_1, S_2, \ldots, S_m$ door

$$S_{k,l} = \frac{x_{k2} - x_{l2}}{x_{k1} - x_{l1}} \quad k < l \tag{2.1}$$

($S_{k,l} = \infty$ als $x_{l1} = x_{k1}$) en sorteer deze hellingen, $S_{(1)} \leqslant S_{(2)} \leqslant \cdots \leqslant S_{(m)}$. De hellingen kunnen worden gerepresenteerd als punten in een graaf. We definiëren nu wanneer twee hellingen in de graaf met elkaar verbonden zijn door een tak. ($S_{(i)} = S_{k,l}$, dan is in de volgende definitie $k = S(i, 2)$ en $l = S(i, 3)$).

DEFINITIE 2.1. $S_{(i)}$ is verbonden met $S_{(j)}$ als :
(1) $i < j$
(a) $S(i, 2) = S(j, 2)$ en voor $i < k < j$, $S(k, 2) \neq S(i, 2)$ en $S(k, 3) \neq S(i, 2)$
(b) $S(i, 3) = S(j, 3)$ en voor $i < k < j$, $S(k, 2) \neq S(i, 3)$ en $S(k, 3) \neq S(i, 3)$

(2) $i > j$
(a) $S(i, 2) = S(j, 2)$ en voor $j < k < i$, $S(k, 2) \neq S(i, 2)$ en $S(k, 3) \neq S(i, 2)$
(b) $S(i, 3) = S(j, 3)$ en voor $j < k < i$, $S(k, 2) \neq S(i, 3)$ en $S(k, 3) \neq S(i, 3)$
De zo gedefinieerde graaf is samenhangend.
Voorbeeld:
Stel dat $S_{(1)} = S_{k,l}$, $S_{(2)} = S_{r,t}$, $S_{(3)} = S_{k,m}$ en $S_{(4)} = S_{k,n}$, $k \neq l \neq r \neq t$. Dan zijn $S_{(1)}$ en $S_{(3)}$ met elkaar verbonden, evenals $S_{(3)}$ en $S_{(4)}$. Echter $S_{(1)}$ en $S_{(4)}$ zijn niet met elkaar verbonden.

Iedere helling correspondeert op éénduidige wijze met een partitie (zie 2.2). Als twee hellingen verbonden zijn, verschillen de bijbehorende partities slechts in één punt. Hiervan maak je gebruik bij het bepalen van de parameterschattingen. We gaan nu een boom in de graaf doorlopen op de volgende manier. Als je uitgaande van een helling $S_{(i)}$ zoekt in de richting van $S_{(1)}$ dan is dat de 'negatieve' richting, terwijl andersom dan natuurlijk niet anders als de 'positieve' richting kan zijn.
Laat $P(i) = 0$, $i = 1,2,\ldots,m$ en $D_1 = D_2 = 0$. Start in een gekozen $i = i_0$. $P(i) := 1$ ($P(i)$ geeft het rangnummer van de i-de helling aan in de boom). Probeer in 'positieve' richting een helling $S_{(j)}$ te vinden die verbonden is met $S_{(i)}$. Als dat niet kan ($j = m$), dan gaan we in 'negatieve' richting. Bij elke helling $S_{(j)}$ gaan we na of we hem al gehad hebben ($P(j) = 0$?). Zo niet dan bekijken we of hij verbonden is met $S_{(i)}$. Als dat zo is gaan we vanaf $S_{(j)}$ verder zoeken ($i := j$). Als $S_{(j)}$ niet verbonden is met $S_{(i)}$ gaan we na of er verwisseling van de hellingen plaatsvindt ($S(i, 2) = S(j, 3)$ of $S(i, 3) = S(j, 2)$). Zo ja, dan moeten we daar bij de volgende helling rekening mee houden. Vinden we dat $S_{(i)}$ met geen enkele andere helling verbonden is die we nog niet gehad hadden ($j = 1$), dan gaan we terug naar het vorige punt in de boom ($i := j_0$ waarvoor geldt dat $P(j_0) := P(i) - 1$), en zoeken van daaruit verder. D_1 en D_2 geven aan of we in 'positieve' richting gaan resp. welke richtingswisseling heeft plaats gevonden.

De partitie $P_{(j)} = (\mathcal{G}_{(j)}, \mathcal{G}_{(j)^c})$ behorend bij helling $S_{(j)} = S_{k,l}$ kan als volgt bepaald worden.
Definieer $\gamma \in \mathbb{R}^3$ door

$$\begin{aligned} \gamma_{2j} &= x_{l1} - x_{k1} \\ \gamma_{1j} &= x_{k2} - x_{l2} \\ \gamma_{0j} &= x_{k1}x_{l2} - x_{k2}x_{l1}. \end{aligned} \tag{2.2}$$

Dan
1 $x_l \in \mathcal{G}_{(j)}$
2 $x_k \in \mathcal{G}_{(j)^c}$

3 Als $m \neq k$, $m \neq l$, dan $x_m \in \mathcal{G}_{(j)}$ als
$\gamma_{2j}x_{m2}+\gamma_{1j}x_{m1}+\gamma_{0j} < 0$ of als
$\gamma_{2j}x_{m2}+\gamma_{1j}x_{m1}+\gamma_{0j} = 0$ én $k < m < l$.

Met behulp van deze partitie kun je parameterschattingen bepalen. Noem $Z = (x_1,x_2,....,x_n)'$ een $n\times 3$-matrix en $C = Z'Z$, $E = Z'Y$, waarin $Y = (y_1,y_2,\ldots,y_n)'$. Verder wordt $\sigma^2 = Y'Y$. Start met helling $S_{(j_0)}$. Noem $D_{j_0} = 0$. Z_{j_0} is een matrix die op de rijen die x_i heeft staan, die in $\mathcal{G}_{j_0}$ zitten. Analoog wordt ook Y_{j_0} gedefinieerd. Bepaal nu

$$C_{j_0} = Z_{j_0}'Z_{j_0}\,, \qquad C_{f_0} = C - C_{j_0}\,, \tag{2.3}$$

en

$$E_{j_0} = Z_{j_0}'Y_{j_0}\,, \qquad E_{f_0} = E - E_{j_0}\,. \tag{2.4}$$

Als $|C_{j_0}| \geqslant k$ en $|C_{f_0}| \geqslant k$ met k een (positieve) scalar in de buurt van 0 dan bepaal je de volgende grootheden

$$D_{j_0} = 1\,, \qquad A_{j_0} = C_{j_0}^{-1}\,, \qquad A_{f_0} = C_{f_0}^{-1} \tag{2.5}$$

en de parameterschattingen

$$\alpha_{j_0} = A_{j_0}E_{j_0}\,, \qquad \alpha_{f_0} = A_{f_0}E_{f_0}\,, \tag{2.6}$$

en

$$\sigma_{j_0}^2 = \sigma^2-(E_{j_0}'\alpha_{j_0} + E_{f_0}'\alpha_{f_0}). \tag{2.7}$$

Als de twee bovengenoemde determinanten niet groot genoeg zijn ($\geqslant k$) dan ga je naar de volgende helling in de boom. Het hierboven beschrevene doe je als je start met het doorlopen van de boom, of als je bij een helling bent aangekomen die verbonden is met een helling $S_{(j)}$, waarvoor $D_j = 0$. Stel nu echter dat voor helling $S_{(j_1)} = S_{k,l}$ de bovenstaande grootheden al zijn bepaald en opgeslagen, en dat de volgende helling $S_{(j_2)} = S_{r,m}$ verbonden is met $S_{(j_1)}$. Dan geldt dat of $r = k$ of $m = l$.
Als $r = k$ dan:
Als $S_{(j_1)}<S_{(j_2)}$ of $\{S_{(j_1)} = S_{(j_2)},\ l<m\}$ dan $\mathcal{G}_{j_2} = \mathcal{G}_{j_1}\cup\{x_m\}$.
Als $S_{(j_1)}>S_{(j_2)}$ of $\{S_{(j_1)} = S_{(j_2)},\ l>m\}$ dan $\mathcal{G}_{j_2} = \mathcal{G}_{j_1}-\{x_l\}$.
Als $m = l$ dan:
Als $S_{(j_1)}<S_{(j_2)}$ of $\{S_{(j_1)} = S_{(j_2)},\ k<r\}$ dan $\mathcal{G}_{j_2} = \mathcal{G}_{j_1}-\{x_r\}$.
Als $S_{(j_1)}>S_{(j_2)}$ of $\{S_{(j_1)} = S_{(j_2)},\ k>r\}$ dan $\mathcal{G}_{j_2} = \mathcal{G}_{j_1}\cup\{x_k\}$.
Als nu $\mathcal{G}_{j_2} = \mathcal{G}_{j_1}\cup\{x_s\}$. met $s = m,k$, dan voer je onderstaande berekeningen uit. Als echter $\mathcal{G}_{j_2} = \mathcal{G}_{j_1}-\{x_s\}$ met $s = r,l$, dan voer je onderstaande berekeningen met de rol van j_1 en j_1^c, en j_2 en j_2^c verwisseld.

Noem eerst $z_s = x_s'$ en bepaal dan

$$C_{j_2} = C_{j_1}+z_s'z_s\,, \qquad C_{f_2} = C - C_{j_2} \tag{2.8}$$

en

$$E_{j_2} = E_{j_1}+z_s'y_s\,, \qquad E_{f_2} = E - E_{j_2} \tag{2.9}$$

Als nu $D_{j_1} = 0$ dan berekenen we de determinanten van C_{j_2} en C_{f_2} direct en als beiden groter zijn dan k bereken je de rest van de grootheden als op de eerstgenoemde wijze. Anders, als $D_{j_1} = 1$ dan berekenen we de determinanten van C_{j_2} en C_{f_2} als volgt

$$|C_{j_2}| = |C_{j_1}|(1+z_sA_{j_1}z_s') \tag{2.10}$$

en

$$|C_{j_2^c}| = |C_{j_1^c}|(1-z_s A_{j^{c_1}} z_s') \tag{2.11}$$

Als beide groter dan k zijn wordt $D_{j_2} = 1$ (als niet: $D_{j_2} = 0$) en je bepaalt

$$A_{j_2} = A_{j_1} - \frac{A_{j_1} z_s' z_s A_{j_1}}{1+z_s A_{j_1} z_s'} \tag{2.12}$$

$$A_{j_2^c} = A_{j_1^c} + \frac{A_{j_1^c} z_s' z_s A_{j_1^c}}{1-z_s A_{j_1^c} z_s'} \tag{2.13}$$

$$\alpha_{j_2} = \alpha_{j_1} + A_{j_2} z_s'(y_s - z_s \alpha_{j_1}) \tag{2.14}$$

$$\alpha_{j_2^c} = \alpha_{j_1^c} + A_{j_2^c} z_s'(y_s - z_s \alpha_{j_1^c}) \tag{2.15}$$

en

$$\sigma_{j_2}^2 = \sigma_{j_1}^2 + \frac{(y_s - z_s\alpha_{j_1})^2}{1+z_s A_{j_1} z_s'} - \frac{(y_s - z_s \alpha_{j_1^c})^2}{1-z_s A_{j_1^c} z_s'}. \tag{2.16}$$

Zoek nu de minimale σ_ϵ^2 en je hebt schattingen voor α en β.

Het is nu dus tijd om de continuïteitsrestrictie te controleren en om als deze niet voldaan is de parameterschattingen te corrigeren. Hoe gaat dat?
Stel dat σ_ϵ^2 minimaal is bij helling $S_{(j_0)} = S_{k,l}$. Definieer $\gamma \in \mathbb{R}^3$ door

$$\begin{aligned} \gamma_{2j_0} &= x_{l1} - x_{k1} \ (>0) \\ \gamma_{1j_0} &= x_{k2} - x_{l2} \\ \gamma_{0j_0} &= x_{k1}x_{l2} - x_{k2}x_{l1}. \end{aligned} \tag{2.17}$$

Dan geldt dat $x \in \mathcal{G}_{j_0} \Leftrightarrow \gamma_{j_0} x \leqslant 0$ en $x \in \mathcal{G}_{j_0^c} \Leftrightarrow \gamma_{j_0} x > 0$. Definieer nu $\hat{\gamma}_{j_0} \in \mathbb{R}^3$ door $\hat{\gamma}_{j_0} = \hat{\beta}_{j_0} - \hat{\alpha}_{j_0}$ als $\hat{\beta}_{2j_0} \geqslant \hat{\alpha}_{2j_0}$ en $\hat{\gamma}_{j_0} = \hat{\alpha}_{j_0} - \hat{\beta}_{j_0}$ als $\hat{\beta}_{2j_0} < \hat{\alpha}_{2j_0}$. Voor alle x_i, $i = 1,2,...,n$ moet nu worden nagegaan of geldt dat $\gamma_{j_0} x_i \leqslant 0 \Leftrightarrow \hat{\gamma}_{j_0} x_i \leqslant 0$. Als aan deze eis voldaan is leveren de parameterschattingen $\hat{\alpha}_{j_0}$ en $\hat{\beta}_{j_0}$ een continu regressievlak. Zo niet, vind dan alle data-punten op de lijn door x_k en x_l. Voor al deze punten bereken je parameterschattingen met restrictie van 'graad 1'. Zij x zo'n punt. Definieer

$$\hat{\delta}_{j_0} = (\hat{\alpha}_{0j_0}, \hat{\alpha}_{1j_0}, \hat{\alpha}_{2j_0}, \hat{\beta}_{0j_0}, \hat{\beta}_{1j_0}, \hat{\beta}_{2j_0})' \tag{2.18}$$

$$C_{j_0} = \begin{bmatrix} A_{j_0} & 0 \\ 0 & B_{j_0} \end{bmatrix} \tag{2.19}$$

$$R_{j_0}^{(1)} = R_{j_0}^{(1)}(x) = (1, x_1, x_2, -1, -x_1, -x_2). \tag{2.20}$$

De schatter met restrictie van de graad 1 $\delta_{j_0}^{R_1}(x) = (\alpha_{0j_0}^{R_1}(x), \cdots, \beta_{2j_0}^{R_1}(x))'$ wordt

$$\delta_{j_0}^{R_1}(x) = \hat{\delta}_{j_0} - C_{j_0} R_{j_0}^{(1)\prime}(R_{j_0}^{(1)} C_{j_0} R_{j_0}^{(1)\prime})^{-1} R_{j_0}^{(1)} \hat{\delta}_{j_0} \tag{2.21}$$

en

$$\sigma_{j_0}^{R_1}(x) = \hat{\sigma}^2 + \hat{\delta}_{j_0} R_{j_0}^{(1)\prime}(R_{j_0}^{(1)} C_{j_0} R_{j_0}^{(1)\prime})^{-1} R_{j_0}^{(1)} \hat{\delta}_{j_0} \tag{2.22}$$

We berekenen $\delta_{j_0}^{R_1}(x)$ en $\sigma_{j_0}^{R_1}(x)$ voor alle x op de lijn $\gamma' x = 0$ en we vervangen $\hat{\sigma}_{j_0}^2$ door $\sigma_{j_0}^{R_1}(x_t) = \min_{x:\gamma x = 0} \sigma_{j_0}^{R_1}(x)$, $\hat{\delta}_{j_0}$ door $\delta_{j_0}^{R_1}(x_t)$ en C_{j_0} door $C_{j_0}(x_t)$ gegeven door

$$C_{j_0}(x_t) = C_{j_0} - \frac{C_{j_0} R_{j_0}(x_t)' R_{j_0}(x_t) C_{j_0}}{R_{j_0}(x_t) C_{j_0} R_{j_0}(x_t)'} \tag{2.23}$$

Bepaal nu opnieuw de kleinste residuele kwadratensom en controleer wederom de

continuïteitsrestrictie.

Steeds moet dus de continuïteitsrestrictie gecheckt worden. Maar wat te doen als bij zekere stap een $\sigma_{j_0}^{R_1}$ het kleinst is en aan de restrictie is niet voldaan. Dan introduceren we restrictie van 'graad 2'. Laat weer $S_{(j_0)} = S_{k,l}$ en laat x_s een punt zijn op de lijn door x_k en x_l ongelijk aan het punt waarop restrictie van de 'graad 1' is toegepast ($x_s \neq x_t$). Schrijf

$$R_{j_0}^{(2)} = (1, x_{s1}, x_{s2}, -1, -x_{s1}, -x_{s2}) \tag{2.24}$$

De parameters met restrictie van de graad 2 worden

$$\delta_{j_0}^{R_2}(x_s, x_t) = \hat{\delta}_{j_0}(x_t) - C_{j_0}(x_t) R_{j_0}^{(2)\prime} (R_{j_0}^{(2)} C_{j_0}(x_t) R_{j_0}^{(2)\prime})^{-1} R_{j_0}^{(2)} \delta_{j_0}(x_t) \tag{2.25}$$

en

$$\sigma_{j_0}^{R_2}(x_s, x_t) = \sigma_{j_0}^{R_1}(x_t) + \delta_{j_0}(x_t) R_{j_0}^{(2)\prime} (R_{j_0}^{(2)} C_{j_0}(x_t) R_{j_0}^{(2)\prime})^{-1} R_{j_0}^{(2)} \delta_{j_0}(x_t) \tag{2.26}$$

Vervang weer $\hat{\sigma}_{j_0}^{R_1}(x_t)$ door $\sigma_{j_0}^{R_2}(x_s, x_t)$ en $\delta_{j_0}^{R_1}(x_t)$ door $\delta_{j_0}^{R_2}(x_s, x_t)$.
Merk verder op dat als de minimale residuele kwadratensom van 'graad 2' is, dat dan automatisch aan de continuïteitsrestrictie voldaan is.

Je hebt nu de kleinste kwadraten schatters gevonden. Omdat $m = n(n-1)/2$ al gauw heel groot is, is het raadzaam om voordat je dit algoritme gaat toepassen eerst te kijken of je n niet kleiner kan maken. Dit kan door gelijke x_m samen te nemen. Stel je hebt $x_m = (1, x_{m1}, x_{m2})$ en $y_{m1}, y_{m2}, \ldots\ldots, y_{mp}$. Neem dan

$$\tilde{x}_m = \sqrt{p}\, x_m \tag{2.27}$$

en

$$\tilde{y}_m = \frac{1}{\sqrt{p}} \sum_{i=1}^{p} y_{mi}. \tag{2.28}$$

Alle hierboven gegeven formules blijven juist met $x_m = \tilde{x}_m$, behalve die waar hellingen berekend worden, en analoge berekeningen. Daar moet steeds $\tilde{x}_{kl}$ vervangen worden door $\tilde{x}_{kl} / \sqrt{p} = \tilde{x}_{kl} / \tilde{x}_{k1}$, $l = 1, 2$.

3. De programma's

Het eerste programma dat genoemd wordt is een programma om data van een twee-fasen regressie model te simuleren. Eerst volgt de source tekst en dan een Example of use.

```
PROGRAM CREATEINPUTFORTWOFASEREGRESSIONPROGRAM(OUTPUT,INPSIM,F);

(* DIT PROGRAMMA SIMULEERT HET TWEEFASE REGRESSIEMODEL OP EENVOUDIGE
   WIJZE. MISSCHIEN OOK NIET HELEMAAL CORRECT, MAAR ALS TEST VOOR HET
   UITEINDELIJKE PROGRAMMA IS HET M.I. GOED GENOEG.
   HET DISCUTABELE ASPECT ZIT IN DE MOGELIJKHEID OM RUIS OP DE DATA AAN
   TE KUNNEN BRENGEN. HIERVOOR WORDT NL. NIET DE GEBRUIKELIJKE NORMALE
   VERDELING GEBRUIKT, MAAR DE UNIFORME VERDELING OP HET INTERVAL (0,U)
   WAAR U=A*Y[I] MET A EEN CONSTANTE DIE DE RELATIEVE GROOTTE VAN DE
   RUIS AANGEEFT *)

LABEL 1, 2;

VAR INPSIM, F                       : TEXT;
    N, COUNT                        : INTEGER;
    A, B                            : ARRAY[1..3] OF REAL;
    START, RANGE, MIN, MAX, SIGMA : REAL;
    ANSWER                          : CHAR;
    NOISE                           : BOOLEAN;



PROCEDURE CONNECT ( VAR F  : TEXT ); EXTERN;

FUNCTION ASELECT(X : REAL) : REAL; FORTRAN;
(* DIT IS DE RANDOM NUMBER GENERATOR UIT DE PASCAL BIBLIOTHEEK
   STATPAS,ID=MATCEN *)


PROCEDURE CALCULATE;
VAR I                 : INTEGER;
    Y1, Y2, Y, X1, X2 : REAL;
BEGIN
   COUNT:=0;
   FOR I:=1 TO N DO
   BEGIN
      RANGE:=MAX-MIN;
      START:=ASELECT(START);
      X1:= RANGE*START+MIN;
      START:=ASELECT(START);
      X2:= RANGE*START+MIN;
      Y1:= A[1] + A[2]*X1 + A[3]*X2;
      Y2:= B[1] + B[2]*X1 + B[3]*X2;
      Y:=Y1;
      IF Y1>Y2 THEN Y:=Y2 ELSE COUNT:=COUNT+1;
      IF NOISE THEN
      BEGIN
         START:=ASELECT(START);
         Y:=Y + ABS(Y)*(START-0.5)*SIGMA;
      END;
      WRITELN(INPSIM,'  ',Y,'   1   ',X1,'   ',X2);
```

```
   END;
   REWRITE(F);
   WRITELN(F,' IN ONE OF THE TWO PARTITIONS ARE ',COUNT:3,' ELEMENTS');
END;


BEGIN
   REWRITE(INPSIM);
   CONNECT(F);
 1:REWRITE(F);
   PAGE(F);
   WRITELN(F);
   WRITELN(F,' ************************************************');
   WRITELN(F,' *                                              *');
   WRITELN(F,' *  CREATION OF TWO-FASE REGRESSION MODEL DATA  *');
   WRITELN(F,' *  ------------------------------------------  *');
   WRITELN(F,' *                                              *');
   WRITELN(F,' *     Y(I) = MIN (ALFA*X(I) , BETA*X(I))       *');
   WRITELN(F,' *     IN WHICH                                 *');
   WRITELN(F,' *       X(I) = (1, X(1), X(2))                 *');
   WRITELN(F,' *       ALFA = (ALFA(1), ALFA(2), ALFA(3))     *');
   WRITELN(F,' *       BETA = (BETA(1), BETA(2), BETA(3))     *');
   WRITELN(F,' *       *    = INPRODUCT                       *');
   WRITELN(F,' *       MIN  = MINIMUM FOR X(1) AND X(2)       *');
   WRITELN(F,' *       MAX  = MAXIMUM FOR X(1) AND X(2)       *');
   WRITELN(F,' *                                              *');
   WRITELN(F,' *       THE X-VALUES ARE GENERATED WITH A      *');
   WRITELN(F,' *       RANDOM NUMBER GENERATOR IN THE INTERVAL *');
   WRITELN(F,' *       (-RANGE, +RANGE)                       *');
   WRITELN(F,' *                                              *');
   WRITELN(F,' ********** PRESS <RETURN> TO CONTINUE **********');
   PAGE(F);
   WRITELN(F);WRITELN(F);
   WRITELN(F,' NUMBER OF POINTS?');
   RESET(F);READ(F,N); REWRITE(F);WRITELN(F);
   WRITELN(F,' BETA(1)    BETA(2)   EN   BETA(3)?');
   RESET(F);READ(F,A[1],A[2],A[3]);REWRITE(F);WRITELN(F);
   WRITELN(F,' ALFA(1)    ALFA(2)   EN   ALFA(3)?');
   RESET(F);READ(F,B[1],B[2],B[3]);REWRITE(F);WRITELN(F);
   WRITELN(F,' INITIAL VALUE FOR RANDOM NUMBER GENERATOR IN (0,1)?');
   REPEAT
   BEGIN
      RESET(F);READ(F,START);
      IF (START < 0) OR (START > 1) THEN
      BEGIN
         REWRITE(F);
         WRITELN(F,' ARE YOU TOO STUPID TO TYPE A NUMBER IN (0,1)?');
         REPEAT
         BEGIN
            RESET(F);READ(F,ANSWER);
            IF (ANSWER <> 'Y') AND (ANSWER <> 'N')
```

```
            THEN
            BEGIN
               REWRITE(F);
               WRITELN(F,' PLEASE ANSWER MY QUESTION, YOU FOOL!');
         WRITELN(F,' ARE YOU TOO STUPID TO TYPE A NUMBER IN (0,1)?');
            END;
         END
         UNTIL (ANSWER = 'Y') OR (ANSWER = 'N');
         IF ANSWER = 'Y' THEN
         BEGIN
            REWRITE(F);
            WRITELN(F,' BYE BYE GENIUS!!!!!');
            GOTO 2;
        *END
         ELSE BEGIN
                 REWRITE(F);
                 WRITELN(F,' I GIVE YOU ANOTHER CHANCE!');
                 WRITELN(F,' PRESS <RETURN> TO CONTINUE');
                 GOTO 1;
               END;
      END;
   END
   UNTIL (START >= 0) AND (START <= 1);
   REWRITE(F);
   WRITELN(F,' MINIMUM OF THE X?');
   RESET(F);READ(F,MIN);
   REWRITE(F);
   WRITELN(F,' MAXIMUM OF THE X?');
   RESET(F);READ(F,MAX);
   REWRITE(F);
   WRITELN(F,' DO YOU WANT TO PUT NOISE ON THE DATA?');
   RESET(F);READ(F,ANSWER);
   IF ANSWER = 'Y'
   THEN BEGIN
           NOISE:=TRUE;
           REWRITE(F);WRITELN(F,' RELATIVE MAGNITUDE OF NOISE?');
           RESET(F);READ(F,SIGMA);
            IF (SIGMA<0) OR (SIGMA>1)
            THEN BEGIN
                    REWRITE(F);
                    WRITELN(F,' PLEASE, TAKE 0 < VALUE < 1 ');
                    WRITELN(F,' PLEASE, RETYPE REL. MAGNITUDE ');
                    RESET(F);READ(F,SIGMA);
                   END;
   END ELSE NOISE:=FALSE;
   CALCULATE;
 2:
END.
```

EXAMPLE OF USE:

```
ATTACH,STATPAS,ID=MATCEN
LIBRARY,STATPAS
PAS3,<PROG>
LGO
```

SCREEN:

```
************************************************
*                                              *
*  CREATION OF TWO-FASE REGRESSION MODEL DATA  *
*  ------------------------------------------  *
*                                              *
*     Y(I) = MIN (ALFA*X(I) , BETA*X(I))       *
*     IN WHICH                                 *
*       X(I) = (1, X(1), X(2))                 *
*       ALFA = (ALFA(1), ALFA(2), ALFA(3))     *
*       BETA = (BETA(1), BETA(2), BETA(3))     *
*       *    = INPRODUCT                       *
*       MIN  = MINIMUM FOR X(1) AND X(2)       *
*       MAX  = MAXIMUM FOR X(1) AND X(2)       *
*                                              *
*       THE X-VALUES ARE GENERATED WITH A      *
*       RANDOM NUMBER GENERATOR IN THE INTERVAL *
*       (-RANGE, +RANGE)                       *
*                                              *
********** PRESS <RETURN> TO CONTINUE **********

NUMBER OF POINTS?
20

BETA(1)    BETA(2)   EN   BETA(3)?
1
3
5

ALFA(1)    ALFA(2)   EN   ALFA(3)?
4
1
2

INITIAL VALUE FOR RANDOM NUMBER GENERATOR IN (0,1)?
0.5
MINIMUM OF THE X?
-20
MAXIMUM OF THE X?
35
```

10

```
DO YOU WANT TO PUT NOISE ON THE DATA?
Y
RELATIVE MAGNITUDE OF NOISE?
0.0025
IN ONE OF THE TWO PARTITIONS ARE   8 ELEMENTS
```

INPSIM :

```
 5.4553853714536E+001   1    1.9122867597285E+001    1.5715493058625E+001
 7.6450570558696E+001   1    2.4180401713306E+001    2.4135084422695E+001
-1.2481745575309E+002   1   -1.6616131293468E+001   -1.5193812374536E+001
-3.7197500703652E+001   1   -5.8457415791158E+000   -4.1320551932608E+000
-4.0114431297105E+001   1    1.7881727376391E+001   -1.8951922685256E+001
-5.5539467530221E+001   1   -1.4801554303411E+001   -2.4269609239980E+000
-3.9703384493954E+001   1    1.0869040248696E+001   -1.4662101048008E+001
 1.3417597822854E+001   1    6.3006356747417E-001    4.3937671276899E+000
-6.5003183510724E+001   1   -1.5044163735428E+001   -4.1741384608878E+000
 1.3021214965812E+001   1    1.3874468535659E+001   -2.4266267849237E+000
 2.6283311998658E+001   1    3.3492086357941E+001   -5.6043871796413E+000
 1.4482659067628E+001   1    2.0311999146510E+001   -4.9146700394408E+000
-7.3947634633267E+001   1    8.3127303405286E-001   -1.5488290747085E+001
 5.1645274832375E+001   1   -5.6953496014364E-001    2.4107404896259E+001
 2.6042093647276E+001   1    2.6692362209743E+001   -2.3251342812335E+000
 2.9423096435630E+000   1    1.5111978433243E+001   -8.0848343948401E+000
 6.3722795739170E+001   1    2.6147639321048E+001    1.6787578209061E+001
-2.6686473782102E+000   1    1.1002885208822E+001   -7.3354606009354E+000
 5.3903308358562E+001   1   -1.1051985229816E+001    3.0477646794189E+001
 7.3245982976566E+001   1    2.7011316935694E+001    2.1117333020436E+001
```

Het volgende programma is om data waarop uiteindelijk het tweefase regressie programma moet werken uit te dunnen door gelijke punten samen te nemen zoals in het vorige hoofdstuk behandeld is.

```
PROGRAM CORRECTDATA(INPUT,OUTPUT,SCREEN);

TYPE CELPTR = ^CEL;
     CEL    = RECORD
                X    : ARRAY[1..3] OF REAL;
                Y    : REAL;
                N    : INTEGER;
                NEXT : CELPTR;
              END;
(* HET RECORD BEVAT X EN Y ALSMEDE N=AANTAL KEER DAT X VOORKOMT EN
   Y=SOM VAN DE Y-WAARDEN BEHOREND BIJ EENZELFDE X *)

VAR INDATA, OUTDATA : INTEGER;
    Y, X2, X3       : REAL;
    DATALIST        : CELPTR;
    SCREEN          : TEXT;


PROCEDURE CONNECT(VAR F : TEXT); EXTERN;


PROCEDURE ADJUSTLIST;
PROCEDURE ADJUSTLIST;
VAR HELP, C : CELPTR;
    STOP    : BOOLEAN;
BEGIN
   HELP:=DATALIST;
   STOP:=FALSE;
   WHILE (HELP<>NIL) AND (NOT STOP) DO
   BEGIN
      IF (HELP^.X[2]=X2) AND (HELP^.X[3]=X3) THEN
      BEGIN
(* DEZE X HEBBEN WE AL GEHAD EN DUS WORDT Y:=Y+YNIEUW EN N:=N+1 *)
         HELP^.Y:=HELP^.Y + Y;
         HELP^.N:=HELP^.N + 1;
         STOP:=TRUE;
      END ELSE HELP:=HELP^.NEXT;
   END;
   IF HELP=NIL THEN
   BEGIN
(* EEN NIEUW PUNT X ZIJN WE TEGENGEKOMEN *)
      NEW(C);
      C^.NEXT:=DATALIST;
      C^.N   :=1;
      C^.X[1]:=1;
```

```
      C^.X[2]:=X2;
      C^.X[3]:=X3;
      C^.Y   :=Y;
      DATALIST:=C;
   END;
END;


PROCEDURE READDATA;
VAR I : INTEGER;
BEGIN
   READLN(INDATA);  (* HET AANTAL WAARNEMINGEN *)
   FOR I:=1 TO INDATA DO
   BEGIN
      READLN(Y,X2,X2,X3);
(* DE INPUT IS Y[I]  1  X2[I]  X3[I]  *)
      ADJUSTLIST;
   END;
END;


PROCEDURE PRINTLIST;
VAR HELP : CELPTR;
BEGIN
   HELP:=DATALIST;
   OUTDATA:=0;
   WHILE HELP<>NIL DO
   BEGIN
      WRITE  ('   ',HELP^.Y/SQRT(HELP^.N));
      WRITE  ('   ',HELP^.X[1]*SQRT(HELP^.N));
      WRITE  ('   ',HELP^.X[2]*SQRT(HELP^.N));
      WRITELN('   ',HELP^.X[3]*SQRT(HELP^.N));
      OUTDATA:=OUTDATA+1;
      HELP:=HELP^.NEXT;
   END;
   CONNECT(SCREEN);
   REWRITE(SCREEN);
   WRITELN(SCREEN,' THERE ARE ',OUTDATA:3,' POINTS LEFT');
END;


BEGIN
   DATALIST:=NIL;
   READDATA;
(* DE VOLGORDE VAN PUNTEN IN DE OUTPUT IS PRECIES OMGEKEERD AAN DIE
    VAN DE INPUT *)
   PRINTLIST;
END.
```

EXAMPLE OF USE:

```
PAS3,<PROG>
LGO
```

INPUT:

```
25
   5.4553853714536E+001   1    1.9122867597285E+001    1.5715493058625E+001
   5.6677853714536E+001   1    1.9122867597285E+001    1.5715493058625E+001
   7.6450570558696E+001   1    2.4180401713306E+001    2.4135084422695E+001
   7.2345570558696E+001   1    2.4180401713306E+001    2.4135084422695E+001
  -1.2481745575309E+002   1   -1.6616131293468E+001   -1.5193812374536E+001
  -3.7197500703652E+001   1   -5.8457415791158E+000   -4.1320551932608E+000
  -4.0114431297105E+001   1    1.7881727376391E+001   -1.8951922685256E+001
  -5.5539467530221E+001   1   -1.4801554303411E+001   -2.4269609239980E+000
  -3.9703384493954E+001   1    1.0869040248696E+001   -1.4662101048008E+001
   1.3417597822854E+001   1    6.3006356747417E-001    4.3937671276899E+000
  -6.5003183510724E+001   1   -1.5044163735428E+001   -4.1741384608878E+000
   1.3021214965812E+001   1    1.3874468535659E+001   -2.4266267849237E+000
   2.6283311998658E+001   1    3.3492086357941E+001   -5.6043871796413E+000
   1.4482659067628E+001   1    2.0311999146510E+001   -4.9146700394408E+000
   1.2222229067628E+001   1    2.0311999146510E+001   -4.9146700394408E+000
  -7.3947634633267E+001   1    8.3127303405286E-001   -1.5488290747085E+001
   5.1645274832375E+001   1   -5.6953496014364E-001    2.4107404896259E+001
   2.6042093647276E+001   1    2.6692362209743E+001   -2.3251342812335E+000
   2.9423096435630E+000   1    1.5111978433243E+001   -8.0848343948401E+000
   6.3722795739170E+001   1    2.6147639321048E+001    1.6787578209061E+001
  -2.6686473782102E+000   1    1.1002885208822E+001   -7.3354606009354E+000
  -2.4444473782102E+000   1    1.1002885208822E+001   -7.3354606009354E+000
  -2.7777473782102E+000   1    1.1002885208822E+001   -7.3354606009354E+000
   5.3903308358562E+001   1   -1.1051985229816E+001    3.0477646794189E+001
   7.3245982976566E+001   1    2.7011316935694E+001    2.1117333020436E+001
```

SCREEN:

```
THERE ARE 20 POINTS LEFT
```

OUTPUT:

7.3245982976566E+001	1.0000000000000E+000	2.7011316935694E+001	2.1117333020436E+001
5.3903308358562E+001	1.0000000000000E+000	-1.1051985229816E+001	3.0477646794189E+001
-4.5557798305618E+000	1.7320508075689E+000	1.9057556211528E+001	-1.2705390457740E+001
6.3722795739170E+001	1.0000000000000E+000	2.6147639321048E+001	1.6787578209061E+001
2.9423096435630E+000	1.0000000000000E+000	1.5111978433243E+001	-8.0848343948401E+000
2.6042093647276E+001	1.0000000000000E+000	2.6692362209743E+001	-2.3251342812335E+000
5.1645274832375E+001	1.0000000000000E+000	-5.6953496014364E-001	2.4107404896259E+001
-7.3947634633267E+001	1.0000000000000E+000	8.3127303405286E-001	-1.5488290747085E+001
1.8883207491268E+001	1.4142135623731E+000	2.8725504671905E+001	-6.9503930243659E+000
2.6283311998658E+001	1.0000000000000E+000	3.3492086357941E+001	-5.6043871796413E+000
1.3021214965812E+001	1.0000000000000E+000	1.3874468535659E+001	-2.4266267849237E+000
-6.5003183510724E+001	1.0000000000000E+000	-1.5044163735428E+001	-4.1741384608878E+000
1.3417597822854E+001	1.0000000000000E+000	6.3006356747417E-001	4.3937671276899E+000
-3.9703384493954E+001	1.0000000000000E+000	1.0869040248696E+001	-1.4662101048008E+001
-5.5539467530221E+001	1.0000000000000E+000	-1.4801554303411E+001	-2.4269609239980E+000
-4.0114431297105E+001	1.0000000000000E+000	1.7881727376391E+001	-1.8951922685256E+001
-3.7197500703652E+001	1.0000000000000E+000	-5.8457415791158E+000	-4.1320551932608E+000
-1.2481745575309E+002	1.0000000000000E+000	-1.6616131293468E+001	-1.5193812374536E+001
1.0521476039850E+002	1.4142135623731E+000	3.4196252046587E+001	3.4132163719595E+001
7.8652694606055E+001	1.4142135623731E+000	2.7043818707545E+001	2.2225063422888E+001

Nu volgt het belangrijkste gedeelte van dit hoofdstuk, nl. het twee-fasen regressie programma.

```
PROGRAM BROKENPLANEREGRESSION( INPUT , OUTPUT );


CONST N      = 20    ; (* HET AANTAL PUNTEN              *)
      M      = 190   ; (* HET AANTAL HELLINGEN N*(N-1)/2 *)
      INF    = 10E+10; (* ONEINDIG GROTE HELLING         *)
      ZERO   = 10E-10; (* NUL                            *)
      MINDET = 10E-10; (* MINIMALE DETERMINANT GROOTTE   *)

(* ALS DE HELLING TUSSEN TWEE PUNTEN ONEINDIG GROOT IS DAN WORDT
   ZIJN WAARDE  INF.
   VANWEGE DE MACHINE PRECISIE ZAL ER EEN ZEKERE ONNAUWKEURIGHEID
   IN DE UITKOMSTEN ZITTEN. DAAROM WORDT ER ALS ER OP NUL GETEST MOET
   WORDEN, HIER GEKEKEN OF HET GETAL IN ABSOLUTE WAARDE KLEINER IS ALS
   ZERO. VOOR MINDET GELDT HETZELFDE.                              *)

TYPE MATRIX    = ARRAY[1..3,1..3] OF REAL;
     NNMATRIX  = ARRAY[1..N,1..N] OF REAL;
     N3MATRIX  = ARRAY[1..N,1..3] OF REAL;
     N3TMATRIX = ARRAY[1..3,1..N] OF REAL;
     VECTOR    = ARRAY[1..N] OF REAL;
     ROW       = ARRAY[1..3] OF REAL;
     ROW6      = ARRAY[1..6] OF REAL;
     MATRIX6   = ARRAY[1..6,1..6] OF REAL;
     SLOPE     = RECORD
                   INDEX   : ARRAY[1..2] OF INTEGER;
                   RANGNR, ELTN, DEGREE, RESTIN : INTEGER;
                   ALFA, BETA, E, EC            : ROW;
                   HELLING, SIGMA, DETC, DETCC  : REAL;
                   A, B, C, CC                  : MATRIX;
                   D                            : BOOLEAN;
                 END;


VAR X, X1, X1C                                    : N3MATRIX;
    Y, Y1, Y1C                                    : VECTOR;
    S                                : ARRAY[1..M] OF SLOPE;
    START, YDIM, YCDIM, HULP, MIN, MININDEX, MAXINDEX,
                        RES1, RES2                : INTEGER;
    ZTZMAT, ZTZMAT2, CT                           : MATRIX;
    SQRSIGMA, ZTZDET, ZTZ2DET                     : REAL;
    DETGRZ, READY                                 : BOOLEAN;
    EE                                            : ROW;
```

```
PROCEDURE LINV1F(A : MATRIX; N, IA : INTEGER; VAR AINV : MATRIX;
                      IDGT : INTEGER; VAR WKAREA : ROW;
                                   VAR IER : INTEGER); FORTRAN;
(* IMSL PROCEDURE OM INVERSE MATRICES TE BEPALEN                *)


PROCEDURE NLINE(I : INTEGER);
VAR K : INTEGER;
BEGIN
   FOR K:=1 TO I DO WRITELN
END;



PROCEDURE MUL3331( A : MATRIX; B : ROW; VAR C : ROW);

(* DEZE PROCEDURE BEREKENT HET PRODUCT VAN EEN (3*3) MATRIX
                                       MET EEN (3*1) MATRIX.

  A[1,1]  A[1,2]  A[1,3]     B[1,1]
  A[2,1]  A[2,2]  A[2,3]  *  B[2,1]
  A[3,1]  A[3,2]  A[3,3]     B[3,1]

                                                              *)

VAR I : INTEGER;
BEGIN
   FOR I:=1 TO 3 DO
      C[I] := A[I,1]*B[1] + A[I,2]*B[2] + A[I,3]*B[3];
END;



PROCEDURE MUL3113( A, B : ROW; VAR C : MATRIX);

(* DEZE PROCEDURE BEREKENT HET PRODUCT VAN EEN (3*1) MATRIX
                                       MET EEN (1*3) MATRIX.

  A[1]
  A[2]  *  B[1]  B[2]  B[3]
  A[3]

                                                              *)

VAR I, J : INTEGER;
BEGIN
   FOR I:=1 TO 3 DO
```

```
      FOR J:=1 TO 3 DO
         C[I,J] := A[I]*B[J];
END;



FUNCTION INPROD( X, Y : ROW ) : REAL;
(* DEZE FUNCTIE BEPAALT HET INPRODUCT VAN TWEE 3*1 VECTOREN *)
VAR I   : INTEGER;
    SOM : REAL;
BEGIN
   SOM:=0;
   FOR I:=1 TO 3 DO SOM := SOM + X[I]*Y[I];
   INPROD:=SOM;
END;



FUNCTION DETERMINANT ( MAT : MATRIX ) : REAL;
(* DEZE FUNCTIE BEPAALT DE DETERMINANT VAN EEN 3*3 MATRIX *)
VAR DET : REAL;
BEGIN
   DET := MAT[1,1]*MAT[2,2]*MAT[3,3] + MAT[1,2]*MAT[2,3]*MAT[3,1]
        + MAT[1,3]*MAT[2,1]*MAT[3,2] - MAT[3,1]*MAT[2,2]*MAT[1,3]
        - MAT[3,2]*MAT[2,3]*MAT[1,1] - MAT[3,3]*MAT[2,1]*MAT[1,2];
   DETERMINANT:=DET;
END;



PROCEDURE LEESDEYENXVECTORENIN;
(* INLEESPROCEDURE VOOR DE DOOR CORRECT GEMAAKTE DATA *)
VAR I : INTEGER;
BEGIN
   FOR I:=1 TO N DO
   BEGIN
      READLN(Y[I],X[I,1],X[I,2],X[I,3]);
   END;
END;



PROCEDURE SORTEERDEXVECTOR;
(* ER IS EEN MATRIX INGELEZEN BESTAANDE UIT :
```

```
        X[1,1]   X[1,1]   X[1,2]
        X[2,1]   X[2,1]   X[2,2]
           .        .        .
           .        .        .
           .        .        .
           .        .        .
        X[N,1]   X[N,1]   X[N,2]

   DIT IS EEN ARRAY[1..N,1..3] OF REAL.
   DE YVECTOR ZIT IN EEN ARRAY[1..N] OF REAL         *)

VAR I, J, MIN                  : INTEGER;
    HELP, HULP1, HULP2, HULP3 : REAL    ;
BEGIN
   FOR I:=1 TO N-1 DO
   BEGIN
      MIN := I;
      FOR J:=I+1 TO N DO
      BEGIN
         IF (X[MIN,2]/X[MIN,1]) > (X[J,2]/X[J,1]) THEN MIN := J;
         IF ((X[MIN,2]/X[MIN,1]) = (X[J,2]/X[J,1])) AND
            ((X[MIN,3]/X[MIN,1]) > (X[J,3]/X[J,1])) THEN MIN:=J;
      END;
      IF MIN <> I THEN
      BEGIN
         HELP     := X[I,1];
         HULP1    := X[I,2];
         HULP2    := X[I,3];
         HULP3    := Y[I];
         X[I,1]   := X[MIN,1];
         X[I,2]   := X[MIN,2];
         X[I,3]   := X[MIN,3];
         Y[I]     := Y[MIN];
         X[MIN,1] := HELP;
         X[MIN,2] := HULP1;
         X[MIN,3] := HULP2;
         Y[MIN]   := HULP3;
      END
   END
END;


PROCEDURE BEREKENDEHELLINGEN;

(* DE HELLINGEN KOMEN IN EEN ARRAY S = ARRAY[1..M] OF RECORDS WAARIN
   M = N*(N-1)/2.

   S[1].HELLING  BEVAT DE HELLING  TUSSEN X[1] EN X[2]
   S[1].INDEX[1] BEVAT DE INDEX 1
   S[1].INDEX[2] BEVAT DE INDEX 2
   S[2].HELLING  BEVAT DE HELLING  TUSSEN X[1] EN X[3]
```

```
   S[2].INDEX[1] BEVAT DE INDEX 1
   S[2].INDEX[2] BEVAT DE INDEX 3
   .
   .
   .
   .
   .
   S[N-1].HELLING  BEVAT DE HELLING  TUSSEN X[1] EN X[N]
   S[N-1].INDEX[1] BEVAT DE INDEX 1
   S[N-1].INDEX[2] BEVAT DE INDEX N
   S[N].HELLING  BEVAT DE HELLING  TUSSEN X[2] EN X[3]
   S[N].INDEX[1] BEVAT DE INDEX 2
   S[N].INDEX[2] BEVAT DE INDEX 3
   .
   .
   .
   .
   S[M].HELLING   BEVAT DE HELLING  TUSSEN X[N-1] EN X[N]
   S[M].INDEX[1] BEVAT DE INDEX N-1
   S[M].INDEX[2] BEVAT DE INDEX N                                    *)

VAR TELLER, K, L : INTEGER;
BEGIN
   TELLER := 1;
   FOR K:=1 TO N-1 DO
   BEGIN
      FOR L:=K+1 TO N DO
      BEGIN
         IF (X[K,2]/X[K,1]) <> (X[L,2]/X[L,1]) THEN
            S[TELLER].HELLING :=  (X[K,3]/X[K,1]-X[L,3]/X[L,1])/
                                  (X[K,2]/X[K,1]-X[L,2]/X[L,1])
         ELSE IF (X[L,3]/X[L,1]) > (X[K,3]/X[K,1])
              THEN S[TELLER].HELLING := INF
              ELSE S[TELLER].HELLING := -INF;
         S[TELLER].INDEX[1] :=  K;
         S[TELLER].INDEX[2] :=  L;
         TELLER := TELLER+1;
      END
   END
END;


PROCEDURE RANGSCHIKDEHELLINGEN;

(*            HELLING  INDEX   INDEX

                S[1]     1       2
                S[2]     1       3
                 .       .       .
                 .       .       .
       S =       .       .       .
```

```
                .       .       .
                .       .       .
              S[M]    N-1      N    *)

VAR I, J, MIN, HULP2, HULP3 : INTEGER;
    HULP1                   : REAL;
BEGIN
   FOR I:=1 TO M-1 DO
   BEGIN
      MIN := I;
      FOR J:=I+1 TO M DO
      BEGIN
         IF S[MIN].HELLING > S[J].HELLING THEN MIN := J;
         IF S[MIN].HELLING = S[J].HELLING THEN
         BEGIN
            IF S[MIN].INDEX[1] > S[J].INDEX[1] THEN MIN:=J;
            IF S[MIN].INDEX[1] = S[J].INDEX[1] THEN
                IF S[MIN].INDEX[2] > S[J].INDEX[2] THEN MIN:=J;
         END;
      END;
      IF MIN <> I THEN
      BEGIN
         HULP1      := S[I].HELLING;
         HULP2      := S[I].INDEX[1];
         HULP3      := S[I].INDEX[2];
         S[I].HELLING    := S[MIN].HELLING;
         S[I].INDEX[1]   := S[MIN].INDEX[1];
         S[I].INDEX[2]   := S[MIN].INDEX[2];
         S[MIN].HELLING := HULP1;
         S[MIN].INDEX[1] := HULP2;
         S[MIN].INDEX[2] := HULP3;
      END
   END
END;



PROCEDURE ZTZ( J : INTEGER );
(* DEZE PROCEDURE BEPAALT DE PARTITIE HOREND BIJ S(J) EN BEREKENT

                              C     C
   Z(J)'*Z(J)      EN    Z(J) '*Z(J)    .

   BEIDE MATRICES MOETEN REGULIER ZIJN                                  *)

VAR P                                                    : VECTOR;
    SO,SOM,SOM1,SOM2,SOM3,SOM4,GAMMA0,GAMMA1,GAMMA2,G    : REAL;
    I, K, L                                              : INTEGER;
    E1, E2, E3                                           : BOOLEAN;
BEGIN
```

```
K:=S[J].INDEX[1];
L:=S[J].INDEX[2];
GAMMA0:=(-X[L,2]*X[K,3]+X[L,3]*X[K,2])/(X[K,1]*X[L,1]);
GAMMA1:=X[K,3]/X[K,1]-X[L,3]/X[L,1];
GAMMA2:=X[L,2]/X[L,1]-X[K,2]/X[K,1];
YDIM:=0;
YCDIM:=0;
FOR I:=1 TO N DO
BEGIN
   P[I]:=0;
   IF I=L THEN P[I]:=1
   ELSE IF I<>K THEN
        BEGIN
           G:=(GAMMA2*X[I,3] + GAMMA1*X[I,2])/X[I,1] + GAMMA0;
           IF G<-ZERO THEN P[I]:=1
           ELSE IF ABS(G)<=ZERO THEN
                BEGIN
                   IF (I>K) AND (I<L) THEN P[I]:=1;
                END;
        END;
   IF P[I]=1 THEN
   BEGIN
           YDIM       := YDIM+1;
           Y1[YDIM]   := Y[I];
           X1[YDIM,1] := X[I,1];
           X1[YDIM,2] := X[I,2];
           X1[YDIM,3] := X[I,3];
        END
   ELSE BEGIN
           YCDIM        := YCDIM+1;
           Y1C[YCDIM]   := Y[I];
           X1C[YCDIM,1] := X[I,1];
           X1C[YCDIM,2] := X[I,2];
           X1C[YCDIM,3] := X[I,3];
        END;
  (* Y1  BEVAT DE Y(I) UIT J(I)
     Y1C BEVAT DE Y(I) UIT COMPLEMENT VAN J(I)
     X1  BEVAT DE X(I) UIT J(I)
     X1C BEVAT DE X(I) UIT COMPLEMENT VAN J(I)
     ALS P(I)=1 DAN ZIT X(I) IN J(I)
     YDIM + YCDIM = N

           *   *   *
           *   *   *
           .   .   .
           .   .   .
     X1 =  *   *   *          HIERIN ZIJN DE * REELE GETALLEN
           0   0   0
           .   .   .
           .   .   .
           0   0   0                                          *)
```

```
   END;
   SO:=0; SOM:=0; SOM1:=0; SOM2:=0; SOM3:=0; SOM4:=0;
   FOR I:=1 TO N DO
   BEGIN
      SO   := SO   + P[I]*SQR(X[I,1]);
      SOM  := SOM  + P[I]*X[I,1]*X[I,2];
      SOM1 := SOM1 + P[I]*X[I,1]*X[I,3];
      SOM2 := SOM2 + P[I]*SQR(X[I,2]);
      SOM3 := SOM3 + P[I]*SQR(X[I,3]);
      SOM4 := SOM4 + P[I]*X[I,2]*X[I,3];
   END;
   ZTZMAT[1,1] := SO;
   ZTZMAT[1,2] := SOM;
   ZTZMAT[1,3] := SOM1;
   ZTZMAT[2,1] := ZTZMAT[1,2];
   ZTZMAT[2,2] := SOM2;
   ZTZMAT[2,3] := SOM4;
   ZTZMAT[3,1] := ZTZMAT[1,3];
   ZTZMAT[3,2] := ZTZMAT[2,3];
   ZTZMAT[3,3] := SOM3;
   FOR K:=1 TO 3 DO
      FOR L:=1 TO 3 DO
          ZTZMAT2[K,L]:=CT[K,L] - ZTZMAT[K,L];
   S[J].ELTN:=YDIM;
END;



PROCEDURE STARTPARAMETERS( I : INTEGER );
(* STARTPARAMETERS BEPAALT RECHTSTREEKS UIT DE DOOR ZTZ BEPAALDE
   PARTITIES DE PARAMETERSCHATTINGEN                              *)
VAR WKAREA          : ROW;
    ZTZINV, ZTZINV2 : MATRIX;
    X1T             : N3TMATRIX;
    I1, I2,SOM      : REAL;
    IDGT, IER, K, L : INTEGER;
BEGIN
   FOR K:=1 TO 3 DO
      FOR L:=1 TO 3 DO
      BEGIN
         S[I].C[K,L]   := ZTZMAT[K,L];
         S[I].CC[K,L]  := ZTZMAT2[K,L];
      END;
   S[I].DETC  := ZTZDET;
   S[I].DETCC := ZTZ2DET;
   IDGT:=0;
   IER :=0;
   LINV1F(ZTZMAT, 3, 3, ZTZINV, IDGT, WKAREA, IER);
   IER :=0;
   LINV1F(ZTZMAT2,3, 3, ZTZINV2,IDGT, WKAREA, IER);
```

```
   FOR K:=1 TO 3 DO
      FOR L:=1 TO 3 DO
      BEGIN
         S[I].A[K,L] := ZTZINV[K,L];
         S[I].B[K,L] :=ZTZINV2[K,L];
      END;
   S[I].D:=TRUE;
   FOR K:=1 TO 3 DO
      FOR L:=1 TO YDIM DO X1T[K,L]:=X1[L,K];
   FOR K:=1 TO 3 DO
   BEGIN
      SOM:=0;
      FOR L:=1 TO YDIM DO SOM:=SOM + X1T[K,L]*Y1[L];
      S[I].E[K]:=SOM;
   END;
   MUL3331( ZTZINV,S[I].E, S[I].ALFA);
   FOR K:=1 TO 3 DO S[I].EC[K] := EE[K] - S[I].E[K];
   MUL3331( ZTZINV2, S[I].EC, S[I].BETA);
   I1 := INPROD( S[I].E , S[I].ALFA);
   I2 := INPROD( S[I].EC, S[I].BETA);
   S[I].SIGMA := SQRSIGMA -I1 -I2;
   S[I].DEGREE:=0;
END;


PROCEDURE PARAMETERS (J2, J1 : INTEGER);
(*   S(J2) = S[R,M]  VOLGT OP  S(J1) = S[K,L]
(*   UIT DE PARAMETERSCHATTINGEN BEHOREND BIJ HELLING S(J1)
(*   WORDEN PARAMETERSCHATTINGEN VOOR S(J2) BEPAALD          *)
VAR I, K, L, M, R, INDEX                  : INTEGER;
    ZSTZS, AZTZA, BZTZB                   : MATRIX;
    H1, H2, SCALAR1, SCALAR2, ZAZT, ZBZT, SIG : REAL;
    Z, AZT, BZT, A2ZT, B2ZT               : ROW;
    JOIN, DETCGRMIN, DETCCGRMIN           : BOOLEAN;
BEGIN
   JOIN:=TRUE;
   K := S[J1].INDEX[1];
   L := S[J1].INDEX[2];
   R := S[J2].INDEX[1];
   M := S[J2].INDEX[2];
   H1:= S[J1].HELLING;
   H2:= S[J2].HELLING;
   IF R=K THEN
   BEGIN
      IF (H1<H2) OR ( (H1=H2) AND (L<M) ) THEN INDEX:=M;
      IF (H1>H2) OR ( (H1=H2) AND (L>M) ) THEN
      BEGIN
         JOIN:=FALSE;
         INDEX:=L;
      END;
```

```
END ELSE IF M=L THEN
         BEGIN
            IF (H1<H2) OR ( (H1=H2) AND (K<R) ) THEN
            BEGIN
               JOIN:=FALSE;
               INDEX:=R;
            END;
            IF (H1>H2) OR ( (H1=H2) AND (K>R) ) THEN INDEX:=K;
         END ELSE WRITELN('  JE MAAKT EEN VERKEERDE STAP!!!');
IF JOIN THEN
BEGIN
 S[J2].ELTN:= S[J1].ELTN + 1;
 IF (S[J2].ELTN < N-2) AND (S[J2].ELTN > 2) THEN
 BEGIN
   IF S[J1].D THEN
   BEGIN
      FOR I:=1 TO 3 DO Z[I]:=X[INDEX,I];
      MUL3113(Z,Z,ZSTZS);
      FOR K:=1 TO 3 DO
      BEGIN
         FOR L:=1 TO 3 DO
         BEGIN
            S[J2].C[K,L] := S[J1].C[K,L] + ZSTZS[K,L];
            S[J2].CC[K,L]:=      CT[K,L] - S[J2].C[K,L];
         END;
         S[J2].E[K] := S[J1].E[K] + Y[INDEX]*Z[K];
         S[J2].EC[K]:=      EE[K] - S[J2].E[K];
      END;
      MUL3331(S[J1].A, Z, AZT);
      MUL3331(S[J1].B, Z, BZT);
      ZAZT := INPROD( Z, AZT);
      ZBZT := INPROD( Z, BZT);
      S[J2].DETC  := S[J1].DETC  *(1 + ZAZT);
      S[J2].DETCC := S[J1].DETCC *(1 - ZBZT);
      DETCGRMIN   := ABS(S[J2].DETC)  >= MINDET;
      DETCCGRMIN  := ABS(S[J2].DETCC) >= MINDET;
      IF DETCGRMIN AND DETCCGRMIN THEN
      BEGIN
         SCALAR1 := Y[INDEX] - INPROD(Z, S[J1].ALFA);
         SCALAR2 := Y[INDEX] - INPROD(Z, S[J1].BETA);
         SIG:=S[J1].SIGMA + SQR(SCALAR1)/(1+ZAZT)
                          - SQR(SCALAR2)/(1-ZBZT);
         S[J2].SIGMA:=SIG;
         S[J2].DEGREE:=0;
         S[J2].D:=TRUE;
         MUL3113(AZT,AZT,AZTZA);
         MUL3113(BZT,BZT,BZTZB);
         FOR K:=1 TO 3 DO
            FOR L:=1 TO 3 DO
            BEGIN
               S[J2].A[K,L] := S[J1].A[K,L] - AZTZA[K,L]/(1+ZAZT);
```

```
            S[J2].B[K,L] := S[J1].B[K,L] + BZTZB[K,L]/(1-ZBZT);
         END;
      MUL3331(S[J2].A, Z, A2ZT);
      MUL3331(S[J2].B, Z, B2ZT);
      FOR K:=1 TO 3 DO
      BEGIN
         S[J2].ALFA[K] := S[J1].ALFA[K] + A2ZT[K]*SCALAR1;
         S[J2].BETA[K] := S[J1].BETA[K] - B2ZT[K]*SCALAR2;
      END;
   END ELSE S[J2].D:=FALSE;
END;
IF NOT S[J1].D THEN
BEGIN
   ZTZ(J2);
   ZTZDET  := DETERMINANT(ZTZMAT );
   ZTZ2DET := DETERMINANT(ZTZMAT2);
   IF ( ABS(ZTZDET) >= MINDET ) AND ( ABS(ZTZ2DET) >= MINDET )
   THEN STARTPARAMETERS(J2)
   ELSE S[J2].D:=FALSE;
END;
END ELSE S[J2].D:= FALSE;
END;
IF NOT JOIN THEN
BEGIN
 S[J2].ELTN:= S[J1].ELTN - 1;
 IF (S[J2].ELTN < N-2) AND (S[J2].ELTN > 2) THEN
 BEGIN
   IF S[J1].D THEN
   BEGIN
      FOR I:=1 TO 3 DO Z[I]:=X[INDEX,I];
      MUL3113(Z,Z,ZSTZS);
      FOR K:=1 TO 3 DO
      BEGIN
         FOR L:=1 TO 3 DO
         BEGIN
            S[J2].CC[K,L]:= S[J1].CC[K,L]+ ZSTZS[K,L];
            S[J2].C[K,L] :=      CT[K,L] - S[J2].CC[K,L];
         END;
         S[J2].EC[K]:= S[J1].EC[K]+ Y[INDEX]*Z[K];
         S[J2].E[K] :=      EE[K] - S[J2].EC[K];
      END;
      MUL3331(S[J1].A, Z, AZT);
      MUL3331(S[J1].B, Z, BZT);
      ZAZT := INPROD( Z, AZT);
      ZBZT := INPROD( Z, BZT);
      S[J2].DETC  := S[J1].DETC  *(1 - ZAZT);
      S[J2].DETCC := S[J1].DETCC *(1 + ZBZT);
      DETCGRMIN   := ABS(S[J2].DETC)  >= MINDET;
      DETCCGRMIN  := ABS(S[J2].DETCC) >= MINDET;
      IF DETCGRMIN AND DETCCGRMIN THEN
      BEGIN
```

```
            SCALAR1 := Y[INDEX] - INPROD(Z, S[J1].ALFA);
            SCALAR2 := Y[INDEX] - INPROD(Z, S[J1].BETA);
            SIG:=S[J1].SIGMA - SQR(SCALAR1)/(1-ZAZT)
                                    + SQR(SCALAR2)/(1+ZBZT);
            S[J2].SIGMA:=SIG;
            S[J2].DEGREE:=0;
            S[J2].D:=TRUE;
            MUL3113(AZT,AZT,AZTZA);
            MUL3113(BZT,BZT,BZTZB);
            FOR K:=1 TO 3 DO
               FOR L:=1 TO 3 DO
               BEGIN
                  S[J2].A[K,L] := S[J1].A[K,L] + AZTZA[K,L]/(1-ZAZT);
                  S[J2].B[K,L] := S[J1].B[K,L] - BZTZB[K,L]/(1+ZBZT);
               END;
            MUL3331(S[J2].A, Z, A2ZT);
            MUL3331(S[J2].B, Z, B2ZT);
            FOR K:=1 TO 3 DO
            BEGIN
               S[J2].ALFA[K] := S[J1].ALFA[K] - A2ZT[K]*SCALAR1;
               S[J2].BETA[K] := S[J1].BETA[K] + B2ZT[K]*SCALAR2;
                        END;
         END ELSE S[J2].D:=FALSE;
      END;
      IF NOT S[J1].D THEN
      BEGIN
         ZTZ(J2);
         ZTZDET  := DETERMINANT(ZTZMAT );
         ZTZ2DET := DETERMINANT(ZTZMAT2);
         IF ( ABS(ZTZDET) >= MINDET ) AND ( ABS(ZTZ2DET) >= MINDET )
         THEN STARTPARAMETERS(J2)
         ELSE S[J2].D:=FALSE;
      END;
    END ELSE S[J2].D:= FALSE;
   END;
END;



PROCEDURE DOORLOOPDEGENERATINGTREE(INUL : INTEGER);
(* DE BOOM WORDT VOLGENS HET GEGEVEN ALGORITME DOORLOPEN *)
LABEL 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16;
VAR D1, D2, I, J, HULP : INTEGER;
BEGIN
   FOR I:=1 TO M DO S[I].RANGNR:=0;
   I:=INUL;
   S[I].RANGNR:=1;
   STARTPARAMETERS(I);
   HULP:=1;
1: IF S[I].RANGNR = M THEN GOTO 2;
```

```
   D1:=0; D2:=0;
   J:=I;
3: IF J = M THEN BEGIN
                    D1:=1;
                    J :=I;
                    IF J=1 THEN GOTO 4;
                    J :=J-1;
                  END
             ELSE J:=J+1;
   GOTO 6;
4: WHILE S[J].RANGNR <> S[I].RANGNR-1 DO J:=J+1;
5: I:=J; GOTO 1;
6: IF S[J].INDEX[1] = S[I].INDEX[1] THEN BEGIN
                                          IF S[J].RANGNR=0 THEN GOTO 7;
                                          GOTO 8;
                                         END
   ELSE GOTO 9;
7: S[J].RANGNR:=HULP+1; PARAMETERS(J,I);
   HULP:=HULP+1; GOTO 5;
8: D2:=1; GOTO 10;
9: IF S[J].INDEX[2] = S[I].INDEX[2] THEN BEGIN
                                          IF S[J].RANGNR=0 THEN GOTO 7;
                                          GOTO 10;
                                         END
   ELSE GOTO 11;
10:IF D1=1 THEN BEGIN
                  IF J=1 THEN GOTO 4;
                  J:=J-1;
                  GOTO 12;
                END
   ELSE IF J=M THEN BEGIN
                      D1:=1;
                      D2:=0;
                      J :=I;
                      IF J=1 THEN GOTO 4;
                      J :=J-1;
                      GOTO 6;
                    END
        ELSE BEGIN
               J:=J+1;
               GOTO 12;
             END;
11:IF S[J].INDEX[1] = S[I].INDEX[2] THEN GOTO 10;
   GOTO 13;
12:IF S[J].INDEX[1+D2] = S[I].INDEX[1+D2] THEN GOTO 14;
   IF D2=1 THEN
   BEGIN
      IF S[J].INDEX[1] = S[I].INDEX[2] THEN GOTO 16;
   END ELSE
       BEGIN
          IF S[J].INDEX[2] = S[I].INDEX[1] THEN GOTO 16;
```

```
        END;
    GOTO 10;
13:IF S[J].INDEX[2] = S[I].INDEX[1] THEN GOTO 8;
    GOTO 15;
14:IF S[J].RANGNR=0 THEN GOTO 7;
16:IF D1=1 THEN BEGIN
                    J:=1;
                    GOTO 4;
                END
    ELSE BEGIN
            D1:=1;
            D2:=0;
            J :=I;
            IF J=1 THEN GOTO 4;
            J :=J-1;
            GOTO 6;
         END;
15:IF D1=1 THEN BEGIN
                    IF J=1 THEN GOTO 4;
                    J:=J-1;
                    GOTO 6;
                END;
    GOTO 3;
2: END; (* OF PROCEDURE *)


PROCEDURE PRINTDEXENYVECTOR;
VAR I : INTEGER;
BEGIN
  WRITE('    X[I,1]     X[I,2]     X[I,3]   ');
  WRITELN('  Y[I]');
  WRITELN;
  FOR I:=1 TO N DO
  BEGIN
    WRITE('  ',X[I,1]:8:4,'   ',X[I,2]:8:4,'   ',X[I,3]:8:4);
    WRITELN('  ',Y[I]:8:4);
  END;
  NLINE(3);
END;



PROCEDURE PRINTOUTPUT(I : INTEGER);
BEGIN
    IF S[I].HELLING = INF THEN
         WRITE('       INF    ',S[I].INDEX[1]:3,'   ')
    ELSE IF S[I].HELLING = -INF THEN
            WRITE('      -INF    ',S[I].INDEX[1]:3,'   ')
         ELSE
       WRITE('   ',S[I].HELLING:9:5,'  ',S[I].INDEX[1]:3,'   ');
```

```
      WRITE(S[I].INDEX[2]:3,'   ',S[I].RANGNR:3,'   ');
      WRITE(S[I].ELTN:2,'  ');
      IF S[I].D THEN
      BEGIN
         WRITE(S[I].ALFA[1]:8,'     ',S[I].ALFA[2]:8,'    ');
         WRITE(S[I].ALFA[3]:8,'     ',S[I].BETA[1]:8,'    ');
         WRITE(S[I].BETA[2]:8,'     ',S[I].BETA[3]:8,'    ');
         WRITELN(S[I].SIGMA:8);
      END ELSE WRITELN;
END;



FUNCTION ZOEKMINIMUM : INTEGER;
(* ZOEKT DE MINIMALE RESIDUELE KWADRATENSOM *)
VAR I, MINDEX : INTEGER;
    MINSIG    : REAL;
BEGIN
   MINSIG:=INF;
   MINDEX:=0;
   FOR I:=1 TO M DO
   BEGIN
      IF S[I].D THEN
      BEGIN
         IF MINSIG > S[I].SIGMA THEN
         BEGIN
            MINSIG:=S[I].SIGMA;
            MINDEX:=I;
         END
      END;
   END;
   ZOEKMINIMUM:=MINDEX
END;



PROCEDURE INITIALISATIE;
VAR SO, SOM, SOM1, SOM2, SOM3, SOM4 : REAL;
    I, J                            : INTEGER;
BEGIN
   SO:=0;SOM:=0;SOM1:=0;SOM2:=0;SOM3:=0;SOM4:=0;
   FOR I:=1 TO N DO
   BEGIN
      SO   := SO   + SQR(X[I,1]);
      SOM  := SOM  + X[I,1]*X[I,2];
      SOM1 := SOM1 + X[I,1]*X[I,3];
      SOM2 := SOM2 + SQR(X[I,2]);
      SOM3 := SOM3 + SQR(X[I,3]);
      SOM4 := SOM4 + X[I,2]*X[I,3];
```

```
    END;
    CT[1,1] := SO;
    CT[1,2] := SOM;
    CT[1,3] := SOM1;
    CT[2,1] := CT[1,2];
    CT[2,2] := SOM2;
    CT[2,3] := SOM4;
    CT[3,1] := CT[1,3];
    CT[3,2] := CT[2,3];
    CT[3,3] := SOM3;
    FOR I:=1 TO 3 DO
    BEGIN
       SOM:=0;
       FOR J:=1 TO N DO SOM := SOM + Y[J]*X[J,I];
       EE[I]:=SOM;
    END;
    SOM:=0;
    FOR I:=1 TO N DO SOM := SOM + Y[I]*Y[I];
    SQRSIGMA:=SOM;
END;



PROCEDURE CALCULATERESTRICTION(HAT, RJ : ROW6;
                CJ : MATRIX6; VAR SQRSIG : REAL;
                 VAR C2 : MATRIX6; VAR RES : ROW6);
VAR H1, H2 : ROW6;
    S1, S2 : REAL;
    K, L   : INTEGER;
BEGIN
    S1:=0;
    S2:=0;
    FOR K:=1 TO 6 DO
    BEGIN
       H1[K]:=0;
       FOR L:=1 TO 6 DO H1[K]:=H1[K] + RJ[L]*CJ[L,K];
       S1:=S1 + RJ[K]*HAT[K];
       S2:=S2 + H1[K]*RJ[K];
    END;
    SQRSIG:=0;
    FOR K:=1 TO 6 DO
    BEGIN
       RES[K]:=0;
       H2[K]:=0;
       SQRSIG:=SQRSIG + SQR(S1)/S2;
       FOR L:=1 TO 6 DO H2[K]:=H2[K] + CJ[K,L]*RJ[L];
       RES[K]:=HAT[K] - H2[K]*S1/S2;
       FOR L:=1 TO 6 DO C2[K,L]:=H1[K]*H1[L]/S2;
    END;
END;
```

```
PROCEDURE FIRSTMODIFY(I : INTEGER; G : ROW);
VAR K, L, M, MININ      : INTEGER;
    DHAT, DRES, DMIN, R : ROW6;
    C, CNEW, CMIN       : MATRIX6;
    SIGMIN, SCALAR1     : REAL;
BEGIN
   FOR K:=1 TO 6 DO
      FOR L:=1 TO 6 DO
      BEGIN
         CNEW[K,L]:=0;
         CMIN[K,L]:=0;
      END;
   MININ:=0;
   SIGMIN:=INF;
   FOR L:=1 TO 6 DO DRES[L]:=0;
   FOR K:=1 TO N DO
   BEGIN
      IF (ABS((G[3]*X[K,3]+G[2]*X[K,2])/X[K,1]+G[1])<=ZERO) THEN
      BEGIN
         FOR L:= 1 TO 3 DO
         BEGIN
            DHAT[L]  := S[I].ALFA[L];
            DHAT[L+3]:= S[I].BETA[L];
            FOR M:=1 TO 3 DO
            BEGIN
               C[L,M]    := S[I].A[L,M];
               C[L+3,M]  := 0;
               C[L,M+3]  := 0;
               C[L+3,M+3]:= S[I].B[L,M];
            END;
            R[L]  := X[K,L];
            R[L+3]:=-X[K,L];
         END;
         SCALAR1:=0;
         CALCULATERESTRICTION(DHAT, R, C, SCALAR1, CNEW, DRES);
         SCALAR1:=SCALAR1+S[I].SIGMA;
         IF SCALAR1 < SIGMIN THEN
         BEGIN
            SIGMIN:=SCALAR1;
            MININ:=K;
            FOR L:=1 TO 6 DO
            BEGIN
               DMIN[L]:=DRES[L];
               FOR M:=1 TO 6 DO CMIN[L,M]:=C[L,M]-CNEW[L,M];
            END;
         END;
      END;
   END;
   IF MININ>0 THEN
   BEGIN
      S[I].SIGMA :=SIGMIN;
```

```
      S[I].DEGREE:=1;
      S[I].RESTIN:=K;
      FOR L:=1 TO 3 DO
      BEGIN
         S[I].ALFA[L]:=DRES[L];
         S[I].BETA[L]:=DRES[L+3];
         FOR M:=1 TO 3 DO
         BEGIN
            S[I].A[L,M]:=CMIN[L  ,M  ];
            S[I].B[L,M]:=CMIN[L+3,M+3];
         END;
      END;
      RES1:=RES1+1;
   END ELSE WRITELN(' VARIANCES IN FIRSTMODIFY TOO LARGE');
END;



PROCEDURE SECMODIFY(INDEX, I, J : INTEGER; G : ROW);
VAR DHAT, DRES, R           : ROW6;
    C, CNEW                 : MATRIX6;
    SIGNEW, H1, RCR, RCRINV : REAL;
    K,L                     : INTEGER;
BEGIN
   FOR K:=1 TO 6 DO
      FOR L:=1 TO 6 DO CNEW[K,L]:=0;
   FOR K:=1 TO 3 DO
   BEGIN
      DHAT[K]  := S[INDEX].ALFA[K];
      DHAT[K+3]:= S[INDEX].BETA[K];
      IF S[INDEX].RESTIN = J THEN
      BEGIN
         R[K]  := X[I,K];
         R[K+3]:=-X[I,K];
       END ELSE
           BEGIN
              R[K]  := X[J,K];
              R[K+3]:=-X[J,K];
            END;
      FOR L:=1 TO 3 DO
      BEGIN
         C[K,L]    :=S[INDEX].A[K,L];
         C[K+3,L+3]:=S[INDEX].B[K,L];
         C[K,L+3]  :=0;
         C[K+3,L]  :=0;
      END;
   END;
   CALCULATERESTRICTION(DHAT, R, C, SIGNEW, CNEW, DRES);
   FOR K:=1 TO 3 DO
   BEGIN
```

```
      S[INDEX].ALFA[K]:= DRES[K];
      S[INDEX].BETA[K]:= DRES[K+3];
   END;
   S[INDEX].SIGMA:= S[INDEX].SIGMA + SIGNEW;
   S[INDEX].DEGREE:=2;
   RES2:=RES2+1;
END;



PROCEDURE CHECKCONTINUITYRESTRICTION(I : INTEGER);
VAR GAM, GAMHAT  : ROW;
    K, L, M      : INTEGER;
    CONT, E1, E2 : BOOLEAN;
   A1,A2 : REAL;
BEGIN
   K:=S[I].INDEX[1];
   L:=S[I].INDEX[2];
   GAM[3]:= X[L,2]/X[L,1] - X[K,2]/X[K,1];
   GAM[2]:= X[K,3]/X[K,1] - X[L,3]/X[L,1];
   GAM[1]:=(X[L,3]*X[K,2] - X[L,2]*X[K,3])/(X[L,1]*X[K,1]);
   WITH S[I] DO
   BEGIN
      IF BETA[3] >= ALFA[3] THEN
      BEGIN
         GAMHAT[3]:= BETA[3] - ALFA[3];
         GAMHAT[2]:= BETA[2] - ALFA[2];
         GAMHAT[1]:= BETA[1] - ALFA[1];
      END ELSE
          BEGIN
             GAMHAT[3]:= ALFA[3] - BETA[3];
             GAMHAT[2]:= ALFA[2] - BETA[2];
             GAMHAT[1]:= ALFA[1] - BETA[1];
          END;
   END;
   CONT:=TRUE;
   FOR M:=1 TO N DO
   BEGIN
      A1:=(GAMHAT[3]*X[M,3] + GAMHAT[2]*X[M,2])/X[M,1] + GAMHAT[1];
      A2:=(GAM[3]*X[M,3] + GAM[2]*X[M,2])/X[M,1] + GAM[1];
      IF (ABS(A1)<=ZERO) OR (ABS(A2)<=ZERO) THEN
      BEGIN
         E1:=TRUE;  E2:=TRUE;
      END ELSE
          BEGIN
             E1:=A1<=0;
             E2:=A2<=0;
          END;
      CONT:=CONT AND ((E1 AND E2) OR ((NOT E1) AND (NOT E2)));
```

```
    END;
    IF CONT THEN READY:=TRUE
    ELSE IF S[I].DEGREE=0 THEN FIRSTMODIFY(I,GAM)
         ELSE SECMODIFY(I,K,L,GAM);
END;


PROCEDURE PRINTMINIMUM( I : INTEGER);
VAR K, L, NUL, TOTI : INTEGER;
    TOT             : REAL;
BEGIN
    NUL:=0;
    WRITELN;WRITELN;
    WITH S[I] DO
    BEGIN
       WRITE('  ALFA  ');
       WRITE('  ',ALFA[1]:12:8,'      ');
       WRITE(ALFA[2]:12:8,'       ');
       WRITELN(ALFA[3]:12:8);
       WRITE('  BETA  ');
       WRITE('  ',BETA[1]:12:8,'      ');
       WRITE(BETA[2]:12:8,'       ');
       WRITELN(BETA[3]:12:8);
       WRITE(' MINIMAL RESIDUAL SUM ');
       WRITELN('OF SQUARES IS  ',SIGMA);
       TOT:=0;
       FOR K:=1 TO N DO TOT:=TOT+SQR(X[K,1]);
       TOTI:=ROUND(TOT);
       WRITELN; WRITELN;
       WRITELN(' DE COVARIANTIEMATRIX VAN (ALPHA, BETA) IS');
       WRITELN;
       FOR K:=1 TO 3 DO
       BEGIN
          FOR L:=1 TO 3 DO WRITE('  ',A[K,L]*SIGMA/TOTI:10:6);
          FOR L:=1 TO 3 DO WRITE('         ',NUL:3);
          WRITELN;
        END;
       FOR K:=1 TO 3 DO
       BEGIN
          FOR L:=1 TO 3 DO WRITE('       ',NUL:3,'  ');
          WRITE('   ');
          FOR L:=1 TO 3 DO WRITE('  ',B[K,L]*SIGMA/TOTI:10:6);
          WRITELN;
        END;
    END;
END;

(* HOOFDPROGRAMMA *)
BEGIN
```

```
PAGE;
WRITELN;
READLN(MININDEX); READLN(MAXINDEX);
IF (MAXINDEX < MININDEX) OR (MININDEX < 0) OR (MAXINDEX > M)
THEN WRITELN(' WRONG INITIAL VALUES HAVE BEEN GIVEN!')
ELSE BEGIN
        LINELIMIT(OUTPUT,MAXINT);
        LEESDEYENXVECTORENIN;
        SORTEERDEXVECTOR;
        PRINTDEXENYVECTOR;
        BEREKENDEHELLINGEN;
        RANGSCHIKDEHELLINGEN;
        INITIALISATIE;
        FOR HULP:=MININDEX TO MAXINDEX DO
        BEGIN
           PAGE;
           WRITELN;
           FOR START:=1 TO M DO S[START].D:= FALSE;
           START:=HULP;
           REPEAT
            BEGIN
              IF START<=M THEN
              BEGIN
                 ZTZ(START);
                 ZTZDET := DETERMINANT(ZTZMAT );
                 ZTZ2DET:= DETERMINANT(ZTZMAT2);
              END;
              START:=START+1;
              DETGRZ:=( ABS(ZTZDET)>=MINDET ) AND
                                ( ABS(ZTZ2DET)>=MINDET );
            END
           UNTIL (START>=M+2) OR DETGRZ;
           IF START=M+2 THEN WRITELN('NIET OPLOSBAAR')
           ELSE
           BEGIN
              DOORLOOPDEGENERATINGTREE(START-1);
              WRITE('     S[I,J]     I     J   RANG ');
              WRITE('#ELTN  ALFA[1]         ALFA[2]');
              WRITE('       ALFA[3]        BETA[1]       ');
              WRITELN('BETA[2]        BETA[3]        SIGMA');
              WRITE('   ------------------------------');
              WRITE('------------------------------');
              WRITE('-----------------------------------');
              WRITELN('-----------------------------------');
              FOR START:=1 TO M DO PRINTOUTPUT(START);
              READY:=FALSE;
              RES1:=0; RES2:=0;
              WHILE NOT READY DO
              BEGIN
                MIN:=ZOEKMINIMUM;
                IF (RES1=0) AND (RES2=0) THEN
```

```
                BEGIN
                 WRITELN; WRITELN;
                 WRITE(' ZONDER CONTINUITEITSRESTRICTIE');
                 WRITE(' ZIJN DE');
                 WRITE(' PARAMETERSCHATTINGEN BEHOREND BIJ');
                 WRITE(' DE PARTITIE MET RANGNR ');
                 WRITELN(S[MIN].RANGNR:3,' :');
                 PRINTMINIMUM(MIN);
                END;
                IF S[MIN].DEGREE = 2 THEN READY:=TRUE
                ELSE CHECKCONTINUITYRESTRICTION(MIN);
              END;
              WRITELN; WRITELN;
              WRITELN; WRITELN;
              WRITE(' NA HET CHECKEN VAN DE ');
              WRITE('CONTINUITEITSRESTRICTIE ZIJN DE ');
              WRITE('PARAMETERSCHATTINGEN BEHOREND BIJ DE ');
              WRITE('PARTITIE MET RANGNR ');
              WRITELN(S[MIN].RANGNR:3,' :');
              PRINTMINIMUM(MIN);
              WRITELN; WRITELN;
              WRITE(' RESTRICTIE VAN DE GRAAD 1 IS ',RES1:2);
              WRITELN(' KEER TOEGEPAST');
              WRITE(' RESTRICTIE VAN DE GRAAD 2 IS ',RES2:2);
              WRITELN(' KEER TOEGEPAST');
              WRITELN;
            END;
        END
    END
END.
```

EXAMPLE OF USE:

```
PAS3,<PROG>
LGO
```

INPUT:

```
8
8
     7.3245982976566E+001     1.0000000000000E+000     2.7011316935694E+001     2.1117333020436E+001
     5.3903308358562E+001     1.0000000000000E+000    -1.1051985229816E+001     3.0477646794189E+001
    -4.5557798305618E+000     1.7320508075689E+000     1.9057556211528E+001    -1.2705390457740E+001
     6.3722795739170E+001     1.0000000000000E+000     2.6147639321048E+001     1.6787578209061E+001
     2.9423096435630E+000     1.0000000000000E+000     1.5111978433243E+001    -8.0848343948401E+000
     2.6042093647276E+001     1.0000000000000E+000     2.6692362209743E+001    -2.3251342812335E+000
     5.1645274832375E+001     1.0000000000000E+000    -5.6953496014364E-001     2.4107404896259E+001
    -7.3947634633267E+001     1.0000000000000E+000     8.3127303405286E-001    -1.5488290747085E+001
     1.8883207491268E+001     1.4142135623731E+000     2.8725504671905E+001    -6.9503930243659E+000
     2.6283311998658E+001     1.0000000000000E+000     3.3492086357941E+001    -5.6043871796413E+000
     1.3021214965812E+001     1.0000000000000E+000     1.3874468535659E+001    -2.4266267849237E+000
    -6.5003183510724E+001     1.0000000000000E+000    -1.5044163735428E+001    -4.1741384608878E+000
     1.3417597822854E+001     1.0000000000000E+000     6.3006356747417E-001     4.3937671276899E+000
    -3.9703384493954E+001     1.0000000000000E+000     1.0869040248696E+001    -1.4662101048008E+001
    -5.5539467530221E+001     1.0000000000000E+000    -1.4801554303411E+001    -2.4269609239980E+000
    -4.0114431297105E+001     1.0000000000000E+000     1.7881727376391E+001    -1.8951922685256E+001
    -3.7197500703652E+001     1.0000000000000E+000    -5.8457415791158E+000    -4.1320551932608E+000
    -1.2481745575309E+002     1.0000000000000E+000    -1.6616131293468E+001    -1.5193812374536E+001
     1.0521476039850E+002     1.4142135623731E+000     3.4196252046587E+001     3.4132163719595E+001
     7.8652694606055E+001     1.4142135623731E+000     2.7043818707545E+001     2.2225063422888E+001
```

OUTPUT:

```
 X[I,1]     X[I,2]     X[I,3]      Y[I]

 1.0000    -16.6161   -15.1938  -124.8175
 1.0000    -15.0442    -4.1741   -65.0032
 1.0000    -14.8016    -2.4270   -55.5395
 1.0000    -11.0520    30.4776    53.9033
 1.0000     -5.8457    -4.1321   -37.1975
 1.0000     -0.5695    24.1074    51.6453
 1.0000      0.6301     4.3938    13.4176
 1.0000      0.8313   -15.4883   -73.9476
 1.0000     10.8690   -14.6621   -39.7034
 1.7321     19.0576   -12.7054    -4.5558
 1.0000     13.8745    -2.4266    13.0212
 1.0000     15.1120    -8.0848     2.9423
 1.0000     17.8817   -18.9519   -40.1144
 1.4142     27.0438    22.2251    78.6527
 1.4142     28.7255    -6.9504    18.8832
 1.4142     34.1963    34.1322   105.2148
 1.0000     26.1476    16.7876    63.7228
 1.0000     26.6924    -2.3251    26.0421
 1.0000     27.0113    21.1173    73.2460
 1.0000     33.4921    -5.6044    26.2833


  S[I,J]     I     J   RANG #ELTN  ALFA[1]        ALFA[2]       ALFA[3]       BETA[1]       BETA[2]       BETA[3]       SIGMA
-------------------------------------------------------------------------------------------------------------------------------
-98.81274    7     8   172    7   -2.2E+001      2.3E+000      3.3E+000      -5.8E-001      1.0E+000      2.3E+000      1.2E+003
```

-35.08704	17	18	188	17	-1.9E+001	2.0E+000	2.7E+000	4.0E+000	1.0E+000	2.0E+000	3.7E+003
-28.26633	6	8	171	6	-1.7E+001	2.6E+000	3.5E+000	-1.3E+000	1.1E+000	2.3E+000	1.1E+003
-17.34893	14	15	189	14	-1.6E+001	2.6E+000	3.3E+000	9.4E+000	8.2E-001	1.9E+000	1.2E+003
-16.43353	6	7	170	7	-1.2E+001	3.0E+000	3.5E+000	-2.7E+000	1.1E+000	2.3E+000	1.2E+003
-10.53369	16	18	186	16	-1.8E+001	2.2E+000	3.0E+000	-1.4E+001	1.6E+000	2.1E+000	2.4E+003
-6.64773	4	5	173	4	1.0E+000	3.0E+000	5.0E+000	-7.8E+000	1.3E+000	2.4E+000	1.9E+003
-4.57225	11	12	1	11	-1.4E+001	2.9E+000	3.3E+000	-4.8E+000	1.2E+000	2.2E+000	1.2E+003
-4.12384	11	13	2	12	-1.5E+001	2.8E+000	3.4E+000	4.2E+000	9.8E-001	2.0E+000	1.0E+003
-4.12323	19	20	190	19							
-3.92349	12	13	3	11	-1.5E+001	2.7E+000	3.4E+000	4.2E+000	9.8E-001	2.0E+000	1.0E+003
-3.86813	4	8	158	5	1.0E+000	3.0E+000	5.0E+000	-1.8E+000	1.1E+000	2.3E+000	7.5E+002
-3.73494	16	17	187	17	-1.8E+001	2.1E+000	2.9E+000	-2.7E+001	2.0E+000	2.2E+000	2.9E+003
-3.38936	6	9	169	8	-1.4E+001	2.8E+000	3.6E+000	5.3E-001	1.0E+000	2.2E+000	9.5E+002
-3.19378	16	20	180	18							
-3.04883	17	20	181	17	-1.8E+001	2.2E+000	3.0E+000	-9.6E+000	1.6E+000	1.9E+000	2.5E+003
-2.71705	6	10	168	9	-1.2E+001	3.0E+000	3.6E+000	-1.2E+000	1.1E+000	2.2E+000	9.9E+002
-2.38333	14	18	185	15	-1.6E+001	2.5E+000	3.3E+000	1.1E+001	7.9E-001	1.9E+000	1.5E+003
-2.33368	6	13	4	10	-1.3E+001	2.8E+000	3.6E+000	4.2E+000	9.8E-001	2.0E+000	7.4E+002
-2.23282	4	7	159	6	-5.3E+000	2.7E+000	4.6E+000	-2.7E+000	1.1E+000	2.3E+000	7.7E+002
-2.05920	4	9	160	7	-4.6E+000	2.8E+000	4.6E+000	2.5E-002	1.1E+000	2.2E+000	4.7E+002
-2.05288	6	12	5	11	-1.2E+001	2.9E+000	3.6E+000	4.2E+000	9.8E-001	2.0E+000	7.4E+002
-1.86111	7	9	161	6	1.0E+000	3.0E+000	5.0E+000	6.0E-001	1.0E+000	2.2E+000	4.3E+002
-1.83703	6	11	6	12	-1.3E+001	2.8E+000	3.6E+000	4.2E+000	9.8E-001	2.0E+000	7.6E+002
-1.71450	4	10	167	8	-3.0E+000	3.0E+000	4.6E+000	-1.3E+000	1.1E+000	2.2E+000	4.7E+002
-1.70837	4	13	164	9	-2.8E+000	2.9E+000	4.7E+000	4.2E+000	9.8E-001	2.0E+000	8.2E+001
-1.70080	5	8	155	4	1.0E+000	3.0E+000	5.0E+000	-5.3E+000	1.2E+000	2.3E+000	1.0E+003
-1.68872	10	13	163	8	-4.6E+000	2.8E+000	4.6E+000	2.6E+000	1.0E+000	2.0E+000	7.0E+001
-1.48372	14	20	182	16	-1.6E+001	2.3E+000	3.3E+000	1.0E+001	1.0E+000	1.7E+000	1.9E+003
-1.47388	4	12	165	10	-2.9E+000	2.9E+000	4.7E+000	4.2E+000	9.8E-001	2.0E+000	8.3E+001
-1.38984	6	15	7	13	-1.4E+001	2.6E+000	3.5E+000	6.0E+000	9.4E-001	1.9E+000	9.9E+002
-1.35324	7	13	162	7	1.0E+000	3.0E+000	5.0E+000	2.9E+000	1.0E+000	2.0E+000	2.7E+001
-1.32005	4	11	166	11	-5.1E+000	2.8E+000	4.5E+000	4.1E+000	9.8E-001	2.0E+000	2.2E+002
-1.13077	7	10	146	8	1.0E+000	3.0E+000	5.0E+000	4.1E+000	9.8E-001	2.0E+000	1.1E+001
-1.12844	4	15	8	12	-7.8E+000	2.6E+000	4.3E+000	5.6E+000	9.6E-001	1.9E+000	5.9E+002
-1.06600	16	19	179	19							
-0.96958	6	18	184	14	-1.5E+001	2.5E+000	3.4E+000	6.5E+000	9.4E-001	1.9E+000	1.4E+003
-0.87230	6	20	183	15	-1.5E+001	2.3E+000	3.4E+000	1.1E+001	9.4E-001	1.7E+000	1.7E+003
-0.86908	4	18	9	13	-9.9E+000	2.4E+000	4.0E+000	6.1E+000	9.5E-001	1.9E+000	1.1E+003
-0.86167	7	12	145	9	4.6E-001	3.0E+000	5.0E+000	4.1E+000	9.8E-001	2.0E+000	1.8E+001
-0.83551	3	8	156	3	1.0E+000	3.0E+000	5.0E+000	-9.6E+000	1.5E+000	2.3E+000	1.5E+003
-0.81003	4	20	10	14	-1.0E+001	2.3E+000	4.0E+000	1.1E+001	9.2E-001	1.7E+000	1.5E+003
-0.71268	2	8	157	2							
-0.62998	5	9	154	5	1.0E+000	3.0E+000	5.0E+000	-3.8E+000	1.2E+000	2.2E+000	8.2E+002
-0.62459	5	13	153	6	1.0E+000	3.0E+000	5.0E+000	-2.3E+000	1.2E+000	2.1E+000	5.1E+002
-0.61172	9	13	152	5	1.0E+000	3.0E+000	5.0E+000	-4.4E+000	1.3E+000	2.2E+000	8.3E+002
-0.60771	4	6	11	15	-1.6E+001	2.3E+000	3.3E+000	1.1E+001	9.1E-001	1.7E+000	1.8E+003
-0.51496	7	11	144	10	-3.9E+000	2.8E+000	4.6E+000	4.1E+000	9.8E-001	2.0E+000	2.1E+002
-0.50561	3	13	151	4	1.0E+000	3.0E+000	5.0E+000	-8.8E+000	1.5E+000	2.3E+000	1.4E+003
-0.48922	4	14	12	16	-1.9E+001	2.2E+000	2.9E+000	1.2E+001	9.0E-001	1.7E+000	2.5E+003
-0.48226	18	20	174	13	-8.8E+000	2.4E+000	4.1E+000	6.0E+000	9.5E-001	1.9E+000	1.2E+003
-0.47662	3	9	150	5	1.0E+000	3.0E+000	5.0E+000	-7.5E+000	1.5E+000	2.2E+000	1.1E+003
-0.47294	7	15	143	11	-8.1E+000	2.6E+000	4.2E+000	4.9E+000	9.7E-001	2.0E+000	5.9E+002
-0.44882	2	13	116	3	1.0E+000	3.0E+000	5.0E+000	-1.2E+001	1.6E+000	2.3E+000	1.8E+003
-0.42615	6	14	13	15	-1.9E+001	2.2E+000	2.9E+000	1.1E+001	9.2E-001	1.7E+000	2.5E+003
-0.40473	2	9	117	4	1.0E+000	3.0E+000	5.0E+000	-1.1E+001	1.6E+000	2.3E+000	1.7E+003
-0.38649	11	15	139	10	-7.0E+000	2.6E+000	4.3E+000	4.7E+000	9.8E-001	2.0E+000	5.6E+002
-0.36802	4	17	178	17	-1.9E+001	2.1E+000	2.8E+000	2.9E+001	7.8E-001	1.1E+000	2.9E+003
-0.30425	7	20	141	12	-9.8E+000	2.4E+000	4.0E+000	5.0E+000	9.8E-001	2.0E+000	1.2E+003
-0.27397	6	17	14	16	-2.0E+001	2.1E+000	2.7E+000	1.2E+001	9.1E-001	1.7E+000	2.9E+003
-0.25780	7	18	142	13	-1.2E+001	2.3E+000	3.8E+000	5.4E+000	9.8E-001	1.9E+000	1.4E+003
-0.24591	4	19	177	18							
-0.20314	8	13	115	2							
-0.19039	3	5	149	6	1.0E+000	3.0E+000	5.0E+000	-5.4E+000	1.4E+000	2.2E+000	9.6E+002
-0.19022	3	10	147	7	1.0E+000	3.0E+000	5.0E+000	-7.0E+000	1.4E+000	2.2E+000	9.1E+002
-0.19013	5	10	148	6	1.0E+000	3.0E+000	5.0E+000	-9.3E+000	1.5E+000	2.3E+000	1.0E+003
-0.18914	3	12	137	8	4.4E-001	3.0E+000	5.0E+000	-8.0E+000	1.4E+000	2.3E+000	8.7E+002
-0.18861	5	12	135	7	3.1E-001	3.0E+000	5.0E+000	-1.0E+001	1.5E+000	2.3E+000	9.7E+002
-0.18237	10	12	136	6	-1.2E+000	2.9E+000	4.9E+000	-8.0E+000	1.5E+000	2.2E+000	1.1E+003

-0.18002	4	16	176	19						
-0.16199	11	20	140	11	-9.7E+000	2.4E+000	4.0E+000	4.7E+000	9.8E-001	2.0E+000	1.2E+003
-0.12968	2	12	118	5	-3.7E+000	3.0E+000	4.7E+000	-1.2E+001	1.6E+000	2.3E+000	1.6E+003
-0.12137	2	10	119	6	1.5E-001	3.0E+000	4.9E+000	-1.4E+001	1.6E+000	2.4E+000	1.3E+003
-0.10894	1	13	114	1							
-0.10841	6	19	15	17	-2.1E+001	2.1E+000	2.5E+000	7.2E+000	9.2E-001	1.9E+000	3.2E+003
-0.07085	3	15	138	9	-7.4E+000	2.6E+000	4.3E+000	-9.3E+000	1.4E+000	2.3E+000	1.4E+003
-0.06579	3	20	131	10	-9.3E+000	2.4E+000	4.1E+000	-9.5E+000	1.4E+000	2.4E+000	2.0E+003
-0.05233	15	20	132	9	-7.0E+000	2.4E+000	4.2E+000	-8.0E+000	1.4E+000	2.3E+000	2.0E+003
-0.03743	5	20	133	8	-7.7E+000	2.4E+000	4.2E+000	-1.0E+001	1.5E+000	2.3E+000	2.1E+003
-0.02992	5	15	134	9	-1.0E+001	2.4E+000	4.0E+000	-1.2E+001	1.4E+000	2.4E+000	2.1E+003
-0.02947	2	20	120	7	-3.7E+000	2.3E+000	4.4E+000	-1.4E+001	1.6E+000	2.4E+000	2.4E+003
-0.02094	2	15	121	8	-8.1E+000	2.3E+000	4.1E+000	-1.6E+001	1.6E+000	2.5E+000	2.4E+003
-0.01688	1	8	113	2							
0.00001	3	11	130	11	-9.4E+000	2.4E+000	4.1E+000	-1.2E+001	1.4E+000	2.5E+000	1.9E+003
0.00112	6	16	16	18							
0.00245	3	18	128	12	-1.1E+001	2.3E+000	3.8E+000	-1.6E+001	1.3E+000	2.7E+000	2.0E+003
0.00458	2	5	122	9	-7.3E+000	2.3E+000	4.2E+000	-1.5E+001	1.6E+000	2.5E+000	2.4E+003
0.00792	11	18	127	11	-1.2E+001	2.3E+000	3.8E+000	-1.1E+001	1.4E+000	2.4E+000	2.2E+003
0.01935	1	9	111	3	1.0E+000	3.0E+000	5.0E+000	-1.7E+001	1.9E+000	2.5E+000	3.7E+003
0.04430	2	18	123	10	-1.0E+001	2.2E+000	3.9E+000	-1.7E+001	1.5E+000	2.6E+000	2.6E+003
0.05553	5	18	124	9	-1.2E+001	2.3E+000	3.8E+000	-1.7E+001	1.5E+000	2.6E+000	2.6E+003
0.06043	2	11	126	11	-9.7E+000	2.2E+000	3.9E+000	-2.1E+001	1.4E+000	2.9E+000	2.2E+003
0.07697	10	20	107	6	-2.9E+001	2.6E+000	3.0E+000	-1.2E+001	1.7E+000	2.3E+000	2.2E+003
0.08231	8	9	112	2							
0.08648	5	11	125	10	-1.1E+001	2.2E+000	3.9E+000	-2.0E+001	1.4E+000	2.9E+000	2.2E+003
0.13495	12	20	108	5	-5.1E+001	2.7E+000	1.7E+000	-1.1E+001	1.7E+000	2.2E+000	1.7E+003
0.15261	14	17	175	15	-1.8E+001	2.1E+000	3.0E+000	1.1E+001	9.2E-001	1.7E+000	2.7E+003
0.19137	1	20	109	4	-3.8E+001	2.3E+000	2.2E+000	-1.7E+001	1.9E+000	2.4E+000	3.7E+003
0.22406	1	12	110	5	-8.5E+000	1.9E+000	3.7E+000	-1.8E+001	1.9E+000	2.5E+000	3.9E+003
0.26005	10	15	106	7	-2.4E+001	2.5E+000	3.3E+000	-1.2E+001	1.6E+000	2.3E+000	2.2E+003
0.27836	1	15	105	6	-6.9E+000	1.9E+000	3.8E+000	-1.8E+001	1.9E+000	2.5E+000	3.8E+003
0.28453	1	10	104	7	7.6E+000	1.5E+000	4.4E+000	-2.3E+001	1.7E+000	2.9E+000	3.1E+003
0.29714	1	18	103	8	6.3E+000	1.4E+000	4.2E+000	-2.4E+001	1.6E+000	3.0E+000	3.0E+003
0.30262	8	20	97	3	-1.6E+001	1.7E+000	2.9E+000	-1.9E+001	1.9E+000	2.5E+000	4.0E+003
0.31934	10	18	102	7	-9.5E+000	1.9E+000	3.6E+000	-1.8E+001	1.9E+000	2.5E+000	3.9E+003
0.40038	9	20	96	2							
0.40586	15	18	101	6	-1.4E+001	1.9E+000	3.3E+000	-1.8E+001	1.9E+000	2.5E+000	3.9E+003
0.41873	1	11	86	9	4.9E+000	1.5E+000	4.1E+000	-2.8E+001	1.4E+000	3.3E+000	2.4E+003
0.44200	3	7	129	13	-9.7E+000	2.2E+000	4.0E+000	-2.4E+001	1.3E+000	3.0E+000	1.7E+003
0.46923	3	17	80	14	-1.8E+001	2.2E+000	2.9E+000	-2.4E+001	1.3E+000	3.0E+000	2.9E+003
0.48570	7	17	81	13	-2.2E+001	2.3E+000	2.7E+000	-1.6E+001	1.3E+000	2.7E+000	2.9E+003
0.49737	12	18	100	5	-3.1E+001	2.3E+000	2.6E+000	-1.7E+001	1.9E+000	2.4E+000	3.7E+003
0.50888	2	17	82	12	-2.3E+001	2.3E+000	2.6E+000	-2.2E+001	1.4E+000	2.9E+000	3.1E+003
0.50899	8	18	98	4	-1.2E+001	1.7E+000	3.1E+000	-1.9E+001	1.9E+000	2.5E+000	4.0E+003
0.51842	8	12	99	5	8.7E+000	1.1E+000	3.7E+000	-1.9E+001	1.9E+000	2.6E+000	4.0E+003
0.53479	3	14	79	15	-2.1E+001	2.2E+000	2.7E+000	-3.0E+001	1.2E+000	3.2E+000	2.9E+003
0.54277	8	15	89	6	8.9E+000	1.1E+000	3.7E+000	-2.0E+001	1.8E+000	2.6E+000	3.9E+003
0.54662	2	7	83	13	-1.8E+001	2.1E+000	3.0E+000	-2.8E+001	1.3E+000	3.2E+000	2.9E+003
0.56309	3	19	78	16	-2.2E+001	2.1E+000	2.5E+000	-3.1E+001	1.1E+000	3.3E+000	3.2E+003
0.58213	2	14	76	14	-2.1E+001	2.2E+000	2.7E+000	-3.1E+001	1.2E+000	3.3E+000	2.9E+003
0.60138	2	19	77	15	-2.2E+001	2.1E+000	2.5E+000	-3.2E+001	1.1E+000	3.3E+000	3.2E+003
0.60964	12	15	90	5	-1.6E-001	1.4E+000	3.5E+000	-1.9E+001	1.9E+000	2.6E+000	4.0E+003
0.61222	7	14	75	13	-2.5E+001	2.3E+000	2.5E+000	-2.3E+001	1.3E+000	3.0E+000	3.1E+003
0.63392	7	19	74	14	-2.6E+001	2.3E+000	2.3E+000	-2.3E+001	1.3E+000	3.0E+000	3.3E+003
0.65387	5	17	84	11	-2.8E+001	2.5E+000	2.5E+000	-2.0E+001	1.4E+000	2.9E+000	3.0E+003
0.68139	3	16	17	17	-2.3E+001	2.1E+000	2.3E+000	-2.3E+001	1.7E+000	3.1E+000	3.5E+003
0.68478	14	19	73	13	-2.7E+001	2.3E+000	2.3E+000	-2.1E+001	1.4E+000	2.9E+000	3.3E+003
0.72172	2	16	18	16	-2.3E+001	2.1E+000	2.3E+000	-2.4E+001	1.7E+000	3.2E+000	3.5E+003
0.74786	1	17	85	10	-1.1E+001	1.7E+000	2.7E+000	-2.8E+001	1.4E+000	3.3E+000	3.0E+003
0.76846	5	19	72	12	-3.1E+001	2.5E+000	2.2E+000	-2.0E+001	1.4E+000	2.9E+000	3.1E+003
0.77967	9	18	93	3	-8.2E+000	1.6E+000	3.2E+000	-1.9E+001	1.9E+000	2.5E+000	4.1E+003
0.79490	5	14	71	13	-3.0E+001	2.5E+000	2.2E+000	-2.2E+001	1.3E+000	2.9E+000	3.1E+003
0.80153	8	10	88	7	1.4E+001	9.5E-001	3.9E+000	-2.5E+001	1.6E+000	3.0E+000	2.5E+003
0.83230	1	19	69	11	-1.5E+001	1.7E+000	2.4E+000	-2.8E+001	1.4E+000	3.3E+000	3.2E+003
0.83826	7	16	19	15	-2.8E+001	2.4E+000	2.2E+000	-7.3E+000	2.8E+000	2.9E+000	2.9E+003
0.85504	13	20	95	1							
0.86486	1	14	70	12	-1.5E+001	1.7E+000	2.4E+000	-2.9E+001	1.3E+000	3.3E+000	3.1E+003

0.94142	5	16	20	14	-3.2E+001	2.5E+000	2.1E+000	-7.9E+000	2.8E+000	2.9E+000	2.6E+003
0.96403	1	16	21	13	-1.7E+001	1.8E+000	2.2E+000	-9.3E+000	3.2E+000	3.1E+000	2.2E+003
1.00142	8	11	87	8	1.2E+001	1.0E+000	3.7E+000	-2.8E+001	1.4E+000	3.2E+000	1.9E+003
1.02705	1	5	22	14	-1.7E+001	1.8E+000	2.2E+000	-1.1E+001	3.1E+000	3.2E+000	2.2E+003
1.03224	9	15	91	4	6.1E+000	1.2E+000	3.5E+000	-1.9E+001	1.9E+000	2.6E+000	4.0E+003
1.13576	1	7	23	15	-1.3E+001	1.6E+000	2.3E+000	-3.3E+001	1.6E+000	3.5E+000	2.2E+003
1.27490	8	17	67	9	1.7E+000	1.1E+000	2.7E+000	-2.8E+001	1.4E+000	3.2E+000	2.3E+003
1.31657	5	7	24	14	-1.0E+001	1.5E+000	2.4E+000	-1.7E+001	2.6E+000	3.2E+000	2.3E+003
1.39823	8	19	68	10	-5.8E-001	1.1E+000	2.4E+000	-2.7E+001	1.5E+000	3.2E+000	2.3E+003
1.55017	9	12	92	5	1.5E+001	8.6E-001	3.6E+000	-2.0E+001	1.8E+000	2.6E+000	3.8E+003
1.56555	11	17	66	8	5.1E-001	1.1E+000	2.7E+000	-2.5E+001	1.6E+000	3.0E+000	2.9E+003
1.59283	10	17	64	7	-1.2E+001	1.6E+000	2.5E+000	-1.9E+001	1.9E+000	2.6E+000	4.1E+003
1.66476	14	16	35	12	-1.8E+001	1.8E+000	2.2E+000	-2.5E+001	1.8E+000	3.3E+000	2.7E+003
1.69700	8	16	36	11	-3.5E+000	1.2E+000	2.2E+000	-2.4E+001	1.8E+000	3.3E+000	1.9E+003
1.70591	8	14	37	12	-3.0E+000	1.1E+000	2.3E+000	-1.6E+001	2.7E+000	3.3E+000	1.5E+003
1.70945	10	11	65	8	-5.9E+000	1.4E+000	2.6E+000	-2.0E+001	1.8E+000	2.7E+000	3.9E+003
1.77736	10	19	59	9	-8.7E+000	1.4E+000	2.4E+000	-2.0E+001	1.9E+000	2.7E+000	3.8E+003
1.79221	11	19	60	8	-1.5E+001	1.6E+000	2.3E+000	-1.9E+001	2.0E+000	2.7E+000	3.9E+003
1.86441	3	6	34	18							
1.88713	13	18	94	2							
1.95387	2	6	33	17	-2.0E+001	2.0E+000	2.4E+000	1.2E+003	8.4E+001	-6.2E+000	3.8E+003
2.05841	9	17	62	6	-2.0E+000	1.2E+000	2.5E+000	-2.0E+001	1.9E+000	2.7E+000	4.0E+003
2.21650	9	19	61	7	-4.4E+000	1.2E+000	2.3E+000	-1.9E+001	2.0E+000	2.7E+000	3.8E+003
2.25382	12	17	63	5	-1.3E+001	1.6E+000	2.5E+000	-1.9E+001	1.9E+000	2.6E+000	4.1E+003
2.38820	10	16	58	10	-1.2E+001	1.5E+000	2.2E+000	-1.8E+001	2.3E+000	3.0E+000	2.7E+003
2.44919	1	6	26	16	-1.1E+001	1.5E+000	2.4E+000	1.0E+003	7.4E+001	-5.1E+000	2.0E+003
2.45410	12	19	54	6	-1.6E+001	1.7E+000	2.3E+000	-1.9E+001	2.0E+000	2.6E+000	3.8E+003
2.57732	11	16	57	9	-1.8E+001	1.7E+000	2.2E+000	-1.8E+001	2.3E+000	2.9E+000	2.6E+003
2.83879	10	14	38	11	-9.9E+000	1.4E+000	2.3E+000	-1.3E+001	3.0E+000	3.3E+000	1.4E+003
2.91459	9	16	56	8	-8.1E+000	1.4E+000	2.2E+000	-1.8E+001	2.3E+000	3.0E+000	2.5E+003
3.45670	11	14	39	10	-1.4E+001	1.6E+000	2.3E+000	-1.4E+001	2.9E+000	3.3E+000	1.3E+003
3.55298	12	16	55	7	-1.9E+001	1.8E+000	2.2E+000	-1.7E+001	2.4E+000	2.9E+000	2.5E+003
3.68043	9	14	40	9	-5.0E+000	1.2E+000	2.2E+000	-1.4E+001	2.8E+000	3.3E+000	1.3E+003
3.71891	15	17	52	4	-3.3E+001	2.2E+000	2.4E+000	-1.9E+001	2.0E+000	2.6E+000	4.0E+003
3.88577	15	19	53	5	-3.6E+001	2.3E+000	2.3E+000	-1.9E+001	2.1E+000	2.6E+000	3.7E+003
4.07113	9	11	41	10	-1.3E+000	1.1E+000	2.2E+000	-1.4E+001	2.9E+000	3.3E+000	1.3E+003
4.32372	13	17	51	3	4.0E+000	1.0E+000	2.0E+000	-1.9E+001	1.9E+000	2.6E+000	4.0E+003
4.38894	13	19	48	4	4.0E+000	1.0E+000	2.0E+000	-1.9E+001	2.0E+000	2.7E+000	3.7E+003
5.01316	17	19	49	3	4.0E+000	1.0E+000	2.0E+000	-1.9E+001	2.0E+000	2.6E+000	3.9E+003
5.35223	5	6	25	15	-8.7E+000	1.4E+000	2.4E+000	-1.2E+000	3.7E+000	3.3E+000	2.1E+003
5.77600	13	15	47	5	1.2E+000	1.1E+000	2.0E+000	-1.9E+001	2.0E+000	2.7E+000	3.7E+003
5.93393	12	14	43	8	-1.2E+001	1.5E+000	2.3E+000	-1.4E+001	2.8E+000	3.3E+000	1.2E+003
6.84065	13	16	45	6	2.2E-001	1.1E+000	2.0E+000	-1.8E+001	2.3E+000	3.0E+000	2.3E+003
7.01012	1	2	27	17	-1.5E+001	1.7E+000	2.4E+000	1.0E+003	7.5E+001	-5.2E+000	2.3E+003
7.03572	1	3	28	18							
7.20161	2	3	29	17	-1.4E+001	1.6E+000	2.4E+000	1.0E+003	7.3E+001	-5.0E+000	2.1E+003
7.50950	15	16	46	5	9.9E-001	1.1E+000	2.0E+000	-1.8E+001	2.3E+000	3.0E+000	2.3E+003
8.20817	1	4	32	19							
8.67992	2	4	30	18							
8.77557	3	4	31	17	-9.6E+000	1.4E+000	2.4E+000	1.0E+000	3.0E+000	5.0E+000	2.0E+003
27.93191	13	14	44	7	4.1E+000	9.8E-001	2.0E+000	-1.5E+001	2.8E+000	3.4E+000	1.0E+003
54.73976	9	10	42	11	1.5E+000	9.8E-001	2.2E+000	-1.7E+001	2.6E+000	3.3E+000	1.1E+003
73.49779	18	19	50	2							

ZONDER CONTINUITEITSRESTRICTIE ZIJN DE PARAMETERSCHATTINGEN BEHOREND BIJ DE PARTITIE MET RANGNR 146 :

ALFA	1.02862009	3.00107982	5.00187771
BETA	4.14299213	0.98112650	1.98984448

MINIMAL RESIDUAL SUM OF SQUARES IS 1.1316094014897E+001

DE COVARIANTIEMATRIX VAN (ALPHA, BETA) IS

0.206341	0.002736	0.016890	0	0	0
0.002736	0.000350	0.000319	0	0	0
0.016890	0.000319	0.001774	0	0	0

0	0	0	0.144953	-0.004886	-0.003090
0	0	0	-0.004886	0.000244	0.000069
0	0	0	-0.003090	0.000069	0.000193

NA HET CHECKEN VAN DE CONTINUITEITSRESTRICTIE ZIJN DE PARAMETERSCHATTINGEN BEHOREND BIJ DE PARTITIE MET RANGNR 146 :

ALFA	1.02862009	3.00107982	5.00187771
BETA	4.14299213	0.98112650	1.98984448

MINIMAL RESIDUAL SUM OF SQUARES IS 1.1316094014897E+001

DE COVARIANTIEMATRIX VAN (ALPHA, BETA) IS

0.206341	0.002736	0.016890	0	0	0
0.002736	0.000350	0.000319	0	0	0
0.016890	0.000319	0.001774	0	0	0
0	0	0	0.144953	-0.004886	-0.003090
0	0	0	-0.004886	0.000244	0.000069
0	0	0	-0.003090	0.000069	0.000193

RESTRICTIE VAN DE GRAAD 1 IS 0 KEER TOEGEPAST
RESTRICTIE VAN DE GRAAD 2 IS 0 KEER TOEGEPAST