

Computermodellering van de foutenstructuur van 3D-meetmachines

Citation for published version (APA):

Theuws, F. C. C. J. M. (1986). *Computermodellering van de foutenstructuur van 3D-meetmachines*. (TH Eindhoven. Afd. Werktuigbouwkunde, Vakgroep Produktietechnologie : WPB; Vol. WPA0355). Technische Universiteit Eindhoven.

Document status and date:

Gepubliceerd: 01/01/1986

Document Version:

Uitgevers PDF, ook bekend als Version of Record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Computermodellering van de
foutenstructuur van 3D-meetmachines.

Auteur : F.C.C.J.M. Theuws.

WPA-rapport nr. 0355, dec. 1986.

Verslag Onderzoekopdracht WPA

Afstudeerhoogleraar: Prof. Dr. Ir. A.C.H. van der Wolf.

Begeleiders : Dr. Ir. P.H.J. Schellekens,
Ir. J.W.M.C. Teeuwsen.

VOORWOORD

In dit verslag wordt het ontwerp van een computerprogramma besproken. Het programma kan ter ondersteuning dienen bij de afname van 3-Dimensionale meetmachines. Het is ontwikkeld als stageopdracht binnen het laboratorium van Lengtemeting, vakgroep WPA van de afdeling Werktuigbouwkunde aan de TU Eindhoven. Mijn directe begeleider was Ir. J. Teeuwsen en de supervisie over het geheel lag bij Dr. Ir. P. Schellekens. Beiden wil ik bedanken voor hun betrokkenheid en adviezen. Verder wil ik speciaal Frits Theuws bedanken voor zijn steun en programmatechnische adviezen.

Frank Theuws.

Inhoud.

<u>Hoofdstuk</u>	<u>Titel</u>	<u>blz.</u>
Inhoud.....		2
Hoofdstuk 1.	Inleiding.....	3
Hoofdstuk 2.	Geometrische fouten-analyse.....	4
2.1.	Inleiding.....	4
2.2.	Schematische voorstelling van een 3D-meetmachine.....	5
2.3.	De geometrische foutenbronnen.....	7
2.4.	De fouten-analyse.....	10
2.4.1.	Verplaatsingsfouten.....	10
2.4.2.	Draaiingsfouten.....	10
2.4.3.	Haaksheidsfouten.....	13
2.4.4.	De totale meetfout.....	14
Hoofdstuk 3.	De programmeertaal.....	17
3.1.	De keuze van de taal.....	17
3.2.	Eisen aan de PC voor Turbo-Pascal en Turbo-Graphics.....	17
3.3.	Het opstarten van Turbo-Pascal.....	18
3.4.	Het maken van een programma.....	19
3.5.	Vaak voorkomende instructies en tips.....	20
3.6.	Foutmeldingen.....	21
Hoofdstuk 4.	Het programma.....	25
Hoofdstuk 5.	Conclusies.....	30
Literatuurlijst.....		31

1. Inleiding.

Er is in de industrie op het moment een trend zichtbaar naar een betere bewaking van de produktkwaliteit tijdens de fabricage. Hierbij neemt men de maatvoering van het produkt vaak als belangrijkste kwaliteitseigenschap. Door de toenemende complexiteit van de produkten neemt het meten met conventionele middelen echter meer en meer tijd in beslag. Hierdoor is het meten van ieder produkt afzonderlijk een bijna onmogelijke zaak geworden. Een 3-Dimensionale (3-D) meetmachine kan hier uitkomst bieden. Hiermee is het mogelijk om ook zeer complexe produkten snel en nauwkeurig te meten. Op die manier kan men elk produkt controleren om zo een snelle terugkoppeling naar het productieproces te krijgen. Tevens kunnen eventueel nog corrigerende maatregelen genomen worden om ervoor te zorgen dat het produkt bij latere verwerking niet wordt afgekeurd.

Een belangrijk probleem bij 3-D meetmachines is echter de controle op de nauwkeurigheid die door de fabrikant is opgegeven. Omdat hier internationaal nog geen vaste afspraken over zijn gemaakt kan men dit op verschillende manieren doen. De meest rigoreuze manier is alle fouten die in de drie geleidingen van een 3-D meetmachine kunnen voorkomen op te sporen met behulp van bijv. een laserinterferometer. Zo kan men per geleiding een idee krijgen van de geometrische fouten. Om de effecten van iedere geleidingsfout op de meetnauwkeurigheid te bepalen dient men een mathematisch model van de machine op te stellen. Door Ir. J. Teeuwsen is, in het kader van een promotieonderzoek, een model ontwikkeld dat alle 3-D meetmachines beschrijft die opgebouwd zijn als een ketting van drie geleidingen.

Toch is ook de handmatige bepaling van de voor een specifieke machine geldende foutentabel met behulp van dit model een complexe en tijdrovende bezigheid. Daarom is besloten dit zoveel mogelijk te automatiseren met behulp van een computer. Hieruit is uiteindelijk een programma ontstaan (CAFAM) dat de gebruiker in staat stelt om in zeer korte tijd een overzicht te krijgen van alle voor zijn machine geldende fouten en, wat veel belangrijker is, hoe ze doorwerken naar de taster. Verder bestaat de mogelijkheid om fouten in de geleidingen, gemeten met daartoe geeigende meetinstrumenten, in te voeren in het programmapakket en een afschatting te maken van de maximale fout in de machine. Het geheel wordt tenslotte op overzichtelijke wijze op papier gezet zodat men snel kan zien of de meetmachine aan zijn specificaties voldoet en waar eventuele fouten hun oorzaak vinden.

2. Geometrische fouten-analyse

2.1 Inleiding.

Het doel waar naar gestreefd wordt, is een zodanige fouten-analyse op te zetten dat deze zowel voor 3DM's als voor werktuigmachines bruikbaar is. De navolgende analyse beperkt zich tot 3DM's, maar ook werktuigmachines zijn op een analoge manier te analyseren (1).

Bij het ontwerpen van 3DM's zijn verschillende oplossingen bedacht voor de constructie van machines die in 3 coördinaten een redelijk meetbereik hebben, stabiel zijn en een hoge meetnauwkeurigheid opleveren. De meest voorkomende configuraties zijn schematisch weergegeven in figuur 2.1. De 3DM's zijn in te delen in 2 groepen:

- 3DM's die bestaan uit een kinematische ketting van 3 geleidingen (stilstaande meettafel).
- 3DM's bestaande uit een combinatie van 2 kinematische kettingen van respectievelijk 1 en 2 geleidingen (bewegende meettafel).

Omdat de eerste groep verreweg de grootste is zal hier een model gegeven worden waarmee een fouten-analyse van deze groep meetmachines mogelijk is. Op eenzelfde wijze is een model voor de tweede groep op te zetten.

BOUWVORM	ASSENORIËTERING				
PORTAAL CONSTRUKTIE					
VERTIKALE MEETASINBOUW					
HORIZONTALE MEETASINBOUW					

a: Stilstaande meettafel

b: Bewegende meettafel

Figuur 2.1: De 2 groepen van 3D-meetmachines.

2.2 Schematische voorstelling van een 3D-meetmachine

In het algemeen kunnen 3D-meetmachines met een stilstaande meettafel voorgesteld worden door een ketting van vectoren. Hierbij wordt onderscheid gemaakt tussen geleidingsvectoren, verbindingsvectoren en een tastervector. Een geleidingsvector geeft de positie van de geleiding aan ten opzichte van het beginpunt van de geleiding, een verbindingsvector geeft de afstand en richting aan van de verbinding tussen twee geleidingen en tenslotte geeft de tastervector aan waar de taster zich bevindt ten opzichte van het eindpunt van de derde geleiding. Elke machine is weer te geven in 7 vectoren te weten: 3 geleidingsvectoren ($\underline{A_u}, \underline{A_v}, \underline{A_w}$), 3 verbindingsvectoren ($\underline{A_{ou}}, \underline{A_{uv}}, \underline{A_{vw}}$) en 1 tastervector ($\underline{A_t}$). Samen vormen deze 7 vectoren (tabel 2.1) de ideale meetvector $\underline{A_{mi}}$ (fig.2.2), waarvoor geldt:

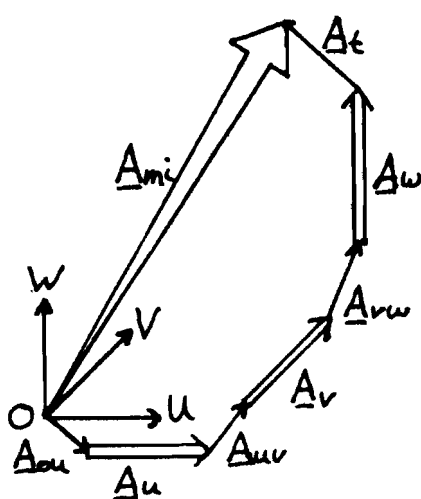
$$\underline{A_{mi}} = \underline{A_{ou}} + \underline{A_u} + \underline{A_{uv}} + \underline{A_v} + \underline{A_{vw}} + \underline{A_w} + \underline{A_t} \quad (2.1)$$

$\underline{A_{mi}}$ ligt in het orthogonale coördinatensysteem UVW (CSU) en geeft het ideale geval weer, waarbij geen geometrische fouten in de geleidingen van de meetmachine voorkomen.

Om de abstracte weergave van de vectorenketting in figuur 2.2 te verduidelijken is in figuur 2.3 een C-type 3DM weergegeven met de daarbijbehorende ketting van vectoren. Uit vergelijking 2.1 en tabel 2.1 volgt:

$$\underline{A_{mi}} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} \quad (2.2)$$

ongeacht de waarden die de componenten van de verbindingsvectoren en de tastervector bezitten.



- \longrightarrow : eenheidsvector van CSU
- \longrightarrow : verbindings- of tastervector
- \Longrightarrow : geleidingsvector
- \Longrightarrow : meetvector

Figuur 2.2: Abstracte weergave van een 3DM zonder geometrische foutenbronnen.

Tabel 2.1: De 7 vectoren die tezamen de kinematische ketting van vectoren vormen.

VECTOR-TYPE : Mathematische beschrijving	
geleiding	: $\underline{A}_u = \begin{bmatrix} U-U_t \\ 0 \\ 0 \end{bmatrix}$, $\underline{A}_v = \begin{bmatrix} 0 \\ V-V_t \\ 0 \end{bmatrix}$, $\underline{A}_w = \begin{bmatrix} 0 \\ 0 \\ W-W_t \end{bmatrix}$
verbinding	: $\underline{A}_{ou} = \begin{bmatrix} -U_{uv}-U_{vw} \\ -V_{uv}-V_{vw} \\ -W_{uv}-W_{vw} \end{bmatrix}$, $\underline{A}_{uv} = \begin{bmatrix} U_{uv} \\ V_{uv} \\ W_{uv} \end{bmatrix}$, $\underline{A}_{vw} = \begin{bmatrix} U_{vw} \\ V_{vw} \\ W_{vw} \end{bmatrix}$
taster	: $\underline{A}_t = \begin{bmatrix} U_t \\ V_t \\ W_t \end{bmatrix}$

Het CSU-nulpunt wordt gedefinieerd als dat punt waarvoor geldt:

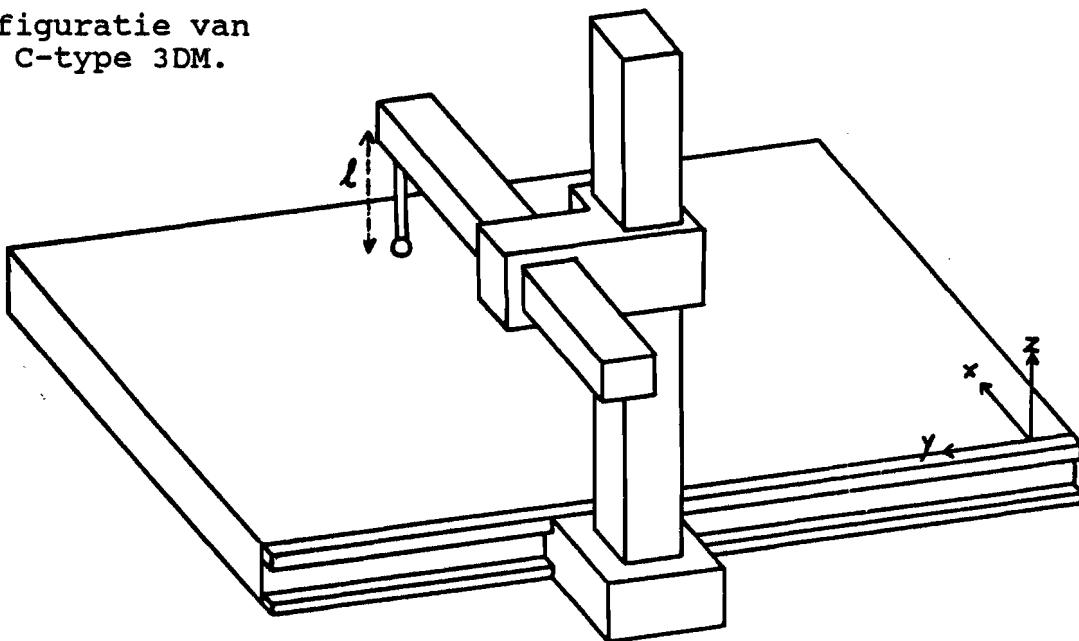
$$U = V = W = 0 \tag{2.3}$$

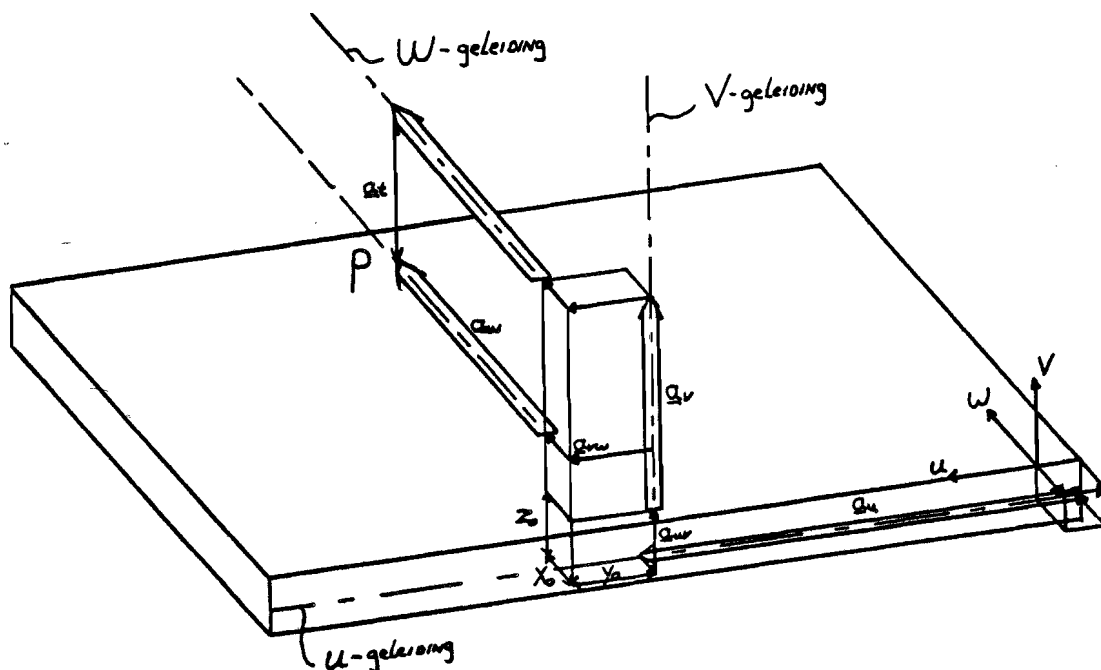
$$\underline{A}_t = \underline{0}$$

De verbindingsvector \underline{A}_{ou} bepaalt de ligging van het startpunt van de eerste geleidingsvector ten opzichte van het CSU-nulpunt. Dit startpunt ligt op de u-geleidings-as en is vrij te kiezen. Na de keuze van dit startpunt is dit evenals het CSU-nulpunt een vast punt in de ruimte. Het startpunt wordt zodanig gekozen dat het dezelfde u-component heeft als het machine-nulpunt, dat door de fabrikant is vastgelegd.

Figuur 2.3.a:

Configuratie van een C-type 3DM.





Figuur 2.3.b: De schematische weergave van de kinematische ketting van vectoren voor de C-type 3DM van fig.2.3.a voor $\underline{A}_t = \underline{0}$ en voor $\underline{A}_t \neq \underline{0}$.

De verbindingsvectoren \underline{A}_{uv} en \underline{A}_{vw} zijn constante vectoren, die bepaald worden door de configuratie van de meetmachine. Ook de tastervector \underline{A}_t is een constante vector die bepaald wordt door de gebruikte taster tijdens de metingen met de 3DM. De geleidingsvectoren \underline{A}_u , \underline{A}_v en \underline{A}_w zijn variabel en afhankelijk van de positie van de geleidingen. Het meetvolume van de meetmachine wordt bepaald door de bereiken van de u-, v- en w-geleiding. Ieder punt in dit meetvolume wordt, bij gebruik van een bepaalde taster, eenduidig weergegeven door een bepaalde combinatie van geleidingsvectoren.

2.3 De geometrische foutenbronnen

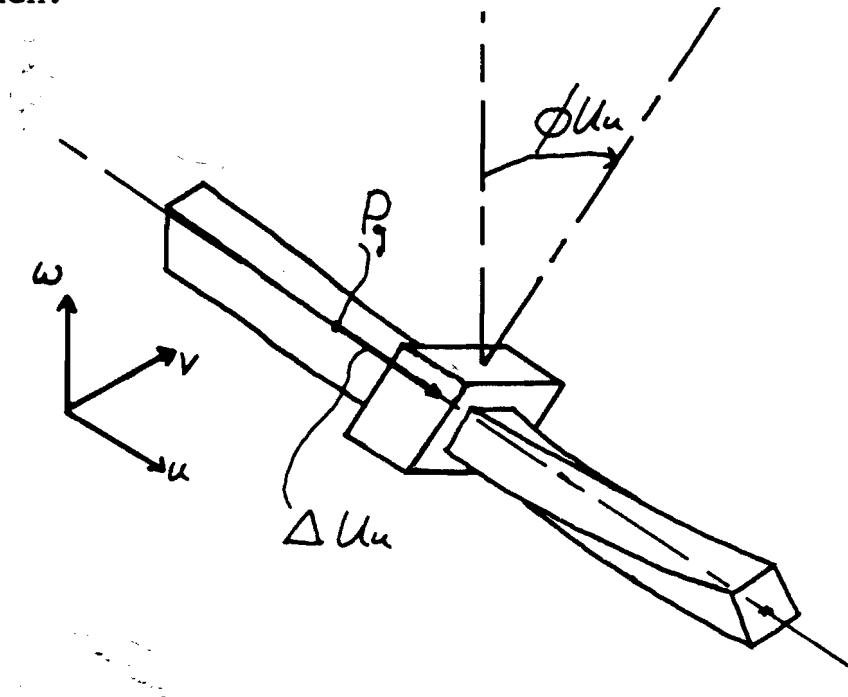
In principe wordt de plaats en stand van een object in de ruimte bepaald door 6 vrijheidsgraden, namelijk 3 translaties en 3 rotaties. Vanuit dit oogpunt heeft een geleiding 6 mogelijke foutenbronnen waardoor de slede op een andere positie of in een andere stand kan staan dan gewenst is. Per geleiding zijn dit 3 verplaatsingsfouten: ΔU_i , ΔV_i en ΔW_i en 3 draaiingsfouten: ϕU_i , ϕV_i en ϕW_i , met $i = u, v$ of w . In tabel 2.2 wordt de gehanteerde foutensymboliek verklaard. De translatiefout van een geleiding in de bewegingsrichting van de betreffende geleiding heet de lineariteitsfout. In figuur 2.4.a wordt als voorbeeld voor de u-geleiding de lineariteitsfout ΔU_u aangegeven.

Tabel 2.2: De symboliek voor de geometrische foutenbronnen.

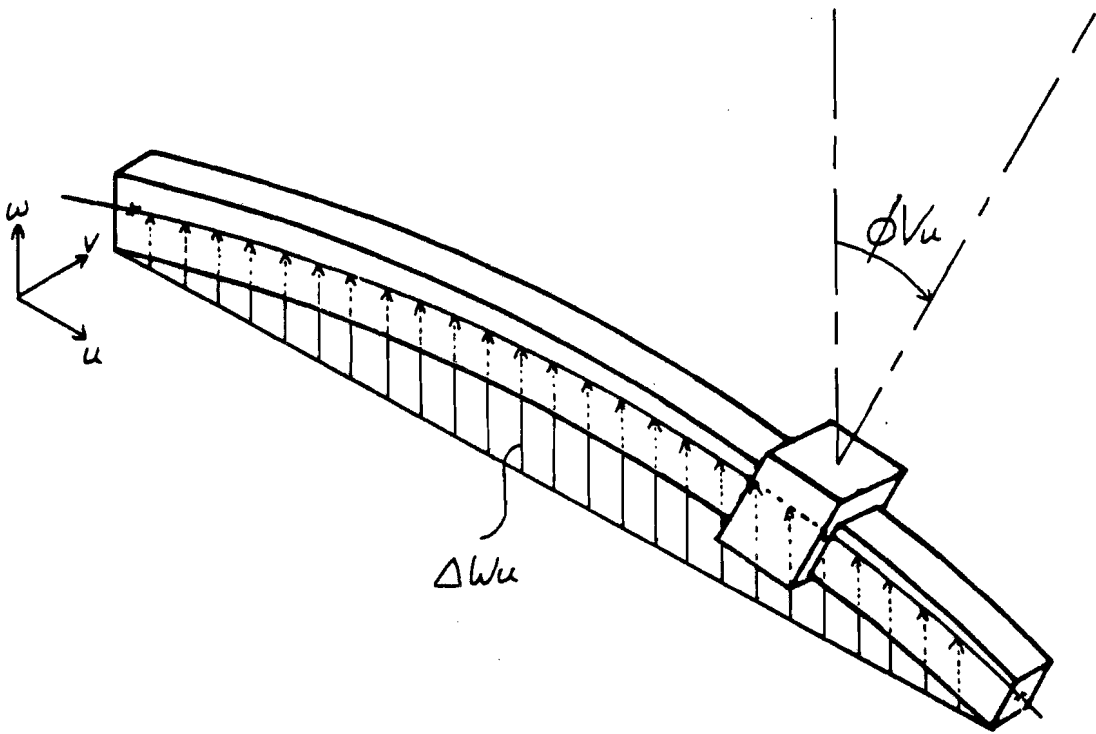
1-ste SYMBOOL	2-de SYMBOOL	INDEX
Foutbron	Foutbepaling	Geleiding
Δ : verplaatsing	U, V, W : verplaatsingsrichting	u, v, w
ϕ : draaiing	U, V, W : draaiings-as	u, v, w
θ : haaksheid	U, V : referentie-as	v, w

De translatiefouten van een geleiding in de richting van een van de andere twee geleidingen zijn de rechtheidsfouten van de desbetreffende geleiding. In figuur 2.4.b&c wordt de rechtheidsfout van de u-geleiding in de richting van de w-geleiding, oftewel de ΔW_u -fout, en de rechtheidsfout in de richting van de v-geleiding, dus de ΔV_u -fout aangegeven.

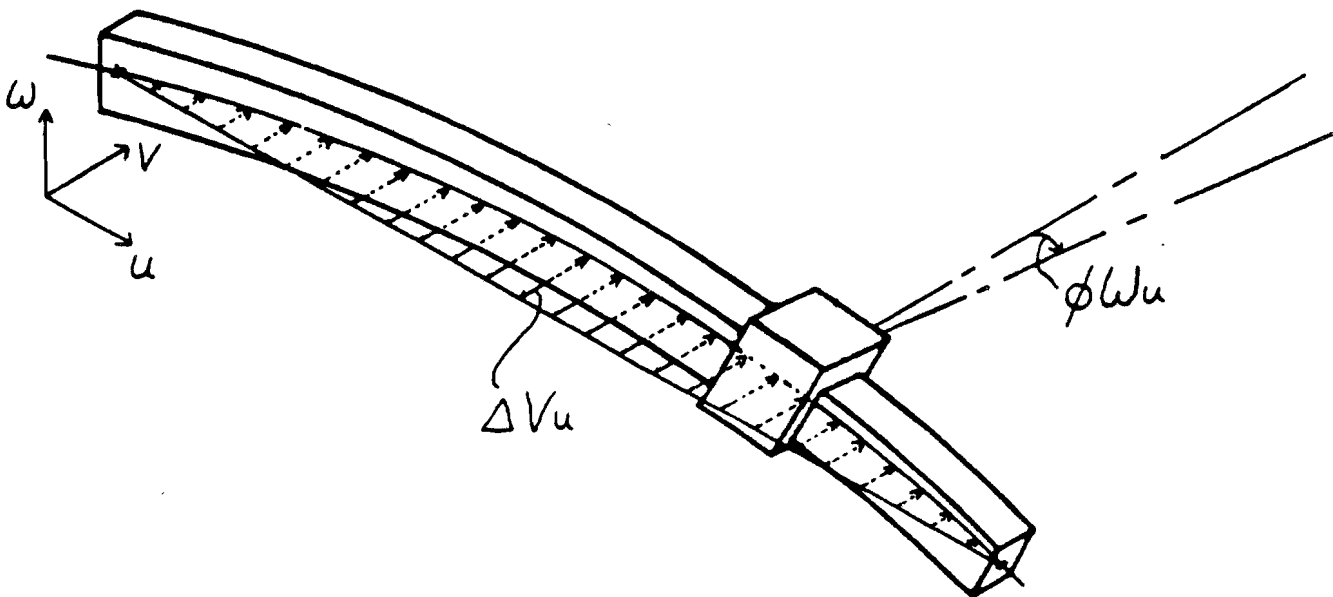
Bij de draaiingsfouten is de rotatiefout die fout waarbij de geleiding om zijn eigen as roteert (fig.2.4.a, ϕU_u). Vervolgens zijn er nog twee draaiingen van een geleiding om een van de andere twee geleidingen aan te geven, de zogenaamde kantelfouten van een geleiding (fig.2.4.b&c, ϕV_u en ϕW_u). Een 3DM bezit 3 geleidingen waardoor er nog 3 haaksheidsfouten ϕU_v , ϕU_w en ϕV_w gedefinieerd moeten worden. Dit omdat de 3 geleidingen nooit exact haaks staan, d.w.z. geen orthogonaal stelsel vormen. In het totaal zijn er in een 3DM dus $3 \cdot 6 + 3 = 21$ mogelijke geometrische foutenbronnen.



Figuur 2.4.a: Schematische weergave van de lineariteitsfout (ΔU_u) en de rotatiefout (ϕU_u) van de u-geleiding.



Figuur 2.4.b: Schematische weergave van de rechtheidsfout ΔW_u en de kantelfout ϕV_u van de u-geleiding.



Figuur 2.4.c: Schematische weergave van de rechtheidsfout ΔV_u en de kantelfout ϕW_u van de u-geleiding.

2.4 De fouten-analyse

In deze paragraaf blijkt ten eerste dat alle soorten van fouten, die veroorzaakt worden door de geometrische foutenbronnen, te vertalen zijn naar correctie-vectoren en -matrices in het CSU. Ten tweede blijkt dat de waargenomen meetvector (\underline{Amw}) gelijk is aan de ketting van vectoren, die de ideale meetvector (\underline{Ami}) vormt, waaraan deze correctie-vectoren en -matrices op een juiste wijze zijn toegevoegd.

FOUTBRON	GELEIDING	CORRECTIE-VECTOR/MATRIX
Verpl.	u, v, w	$\Delta \underline{Au} = \begin{bmatrix} \Delta Uu \\ \Delta Vu \\ \Delta Wu \end{bmatrix}$, $\Delta \underline{Av} = \begin{bmatrix} \Delta Uv \\ \Delta Vv \\ \Delta Wv \end{bmatrix}$, $\Delta \underline{Aw} = \begin{bmatrix} \Delta Uw \\ \Delta Vw \\ \Delta Ww \end{bmatrix}$
Haaks.	v, w	$\Theta \underline{Av} = \begin{bmatrix} \bar{v} \cdot \Theta Uv \\ 0 \\ 0 \end{bmatrix}$, $\Theta \underline{Aw} = \begin{bmatrix} \bar{w} \cdot \Theta Uw \\ \bar{w} \cdot \Theta Vw \\ 0 \end{bmatrix}$
Draaiing	u, v, w	$\phi_i = \begin{bmatrix} 1 & -\phi_{wi} & \phi_{vi} \\ \phi_{wi} & 1 & -\phi_{ui} \\ -\phi_{vi} & \phi_{ui} & 1 \end{bmatrix}$, $i = u, v \text{ of } w$

Tabel 2.3: Mathematische beschrijving van de correctie-vectoren en -matrices.

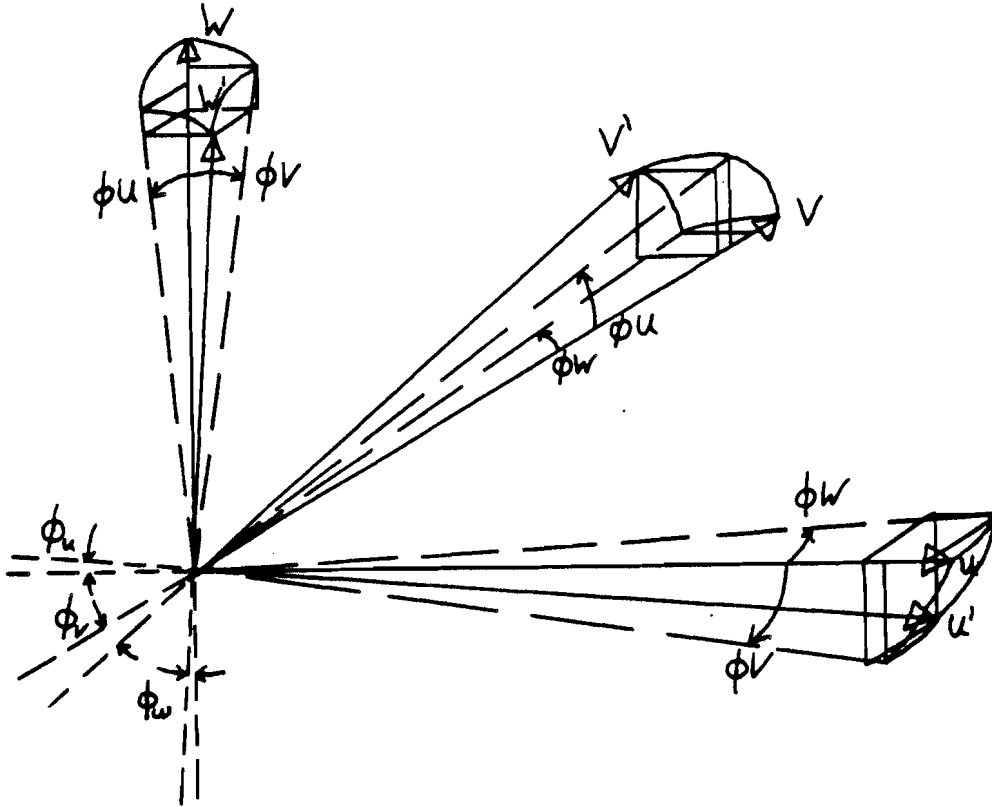
2.4.1 Verplaatsingsfouten

Verplaatsingsfouten van een geleiding worden één op één doorgegeven naar de taster. Deze fouten worden dan ook gecorrigeerd door aan de u-, v- en w-geleidingsvector respectievelijk de correctie-vectoren $\Delta \underline{Au}$, $\Delta \underline{Av}$ en $\Delta \underline{Aw}$ toe te voegen (tabel 2.3). De component van de correctie-vectoren parallel aan de geleidings-vector is de lineariteitsfout, terwijl de andere twee componenten de rechtheidsfouten aangeven.

2.4.2 Draaiingsfouten

De fouten die ontstaan door draaiingen worden d.m.v. de drie draaiings-correctie-matrices ϕ_u , ϕ_v en ϕ_w verdisconteerd. Beschouw hiertoe twee orthogonale coördinatensystemen CSU en CSU', waarbij CSU' t.o.v. CSU om ϕ_u , ϕ_v en ϕ_w gedraaid is (fig.2.5). Daar de grootte-orde van de hoekverdraaiingen boogsecondes zijn, geldt:

$$\begin{aligned} \cos \phi_i &= 1 \\ \sin \phi_i &= \phi_i \\ \tan \phi_i &= \phi_i \end{aligned} \tag{2.4}$$



Figuur 2.5: Het coördinatensysteem \underline{CSU}' dat t.o.v. \underline{CSU} gedraaid is. De hoeken tussen respectievelijk \underline{U} - en \underline{U}' -as, \underline{V} - en \underline{V}' -as en \underline{W} - en \underline{W}' -as zijn ϕ_U , ϕ_V en ϕ_W , D is het draaipunt.

De transformatie van \underline{CSU} naar \underline{CSU}' wordt, met behulp van vergelijking 2.4, in matrixvorm gegeven door:

$$\underline{U}' = \begin{bmatrix} 1 & -\phi_W & \phi_V \\ \phi_W & 1 & -\phi_U \\ -\phi_V & \phi_U & 1 \end{bmatrix} * \underline{U} = \phi * \underline{U} \quad (2.5)$$

met:

$$\underline{U}' = \begin{bmatrix} U' \\ V' \\ W' \end{bmatrix}, \quad \underline{U} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} \quad (2.6)$$

De verplaatsingsfout ($\Delta \underline{Ud}$) tengevolge van de draaiingen is afhankelijk van de gedraaide hoeken en van de armen waarop die hoeken werken, hiervoor geldt:

$$\Delta \underline{Ud} = \underline{U}' - \underline{U} = [\phi - \underline{I}] * \underline{U} \quad (2.7)$$

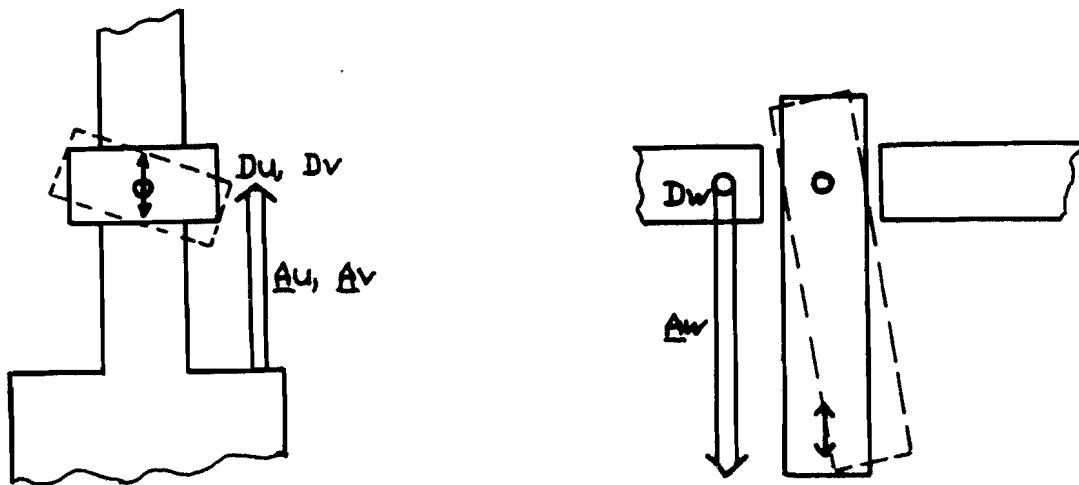
Afhankelijk van de plaats van het draaipunt in de kinematische ketting kunnen zo meerdere correctie-vectoren opgesteld worden.

De draaipunten D_u , D_v en D_w van respectievelijk de u-, v- en w-geleiding bevinden zich op:

- het eindpunt van respectievelijk de u- en v-geleidingsvector voor D_u en D_v .
- het beginpunt van de w-geleidingsvector voor D_w .

Dit verschil in plaats tussen D_u , D_v en D_w volgt uit de configuratie van de geleidingen, dit is in figuur 2.6 schematisch weergegeven. Voor de u- en v-geleiding geldt dat het draaipunt zich verplaatst met de geleidingsvector. Dit heeft tot gevolg dat de bij de geleiding behorende draaiings-correctie-matrix de geleidingsvector niet draait. De w-geleidingsvector wordt daarentegen wel gedraaid door de draaiings-correctie-matrix, daar het draaipunt zich niet met de geleidingsvector mee verplaatst. Draaiings-correctie-matrices zijn altijd werkzaam op vectoren die ten opzichte van het betreffende draaipunt in rangorde hoger liggen in de kinematische ketting.

De draaiings-correctie-matrices zijn commutatief, d.w.z. dat de totale foutvector ten gevolge van draaiingen van de drie geleidingen gelijk is aan de som van de draaiingsfouten van de 3 geleidingen afzonderlijk. Oftewel, de totale foutvector in een bepaald punt in het meetvolume, is onafhankelijk van de gevolgde weg naar dit punt.



a: Draaipunt u- en v-geleiding.

b: Draaipunt w-geleiding.

Figuur 2.6: De ligging van de draaipunten.

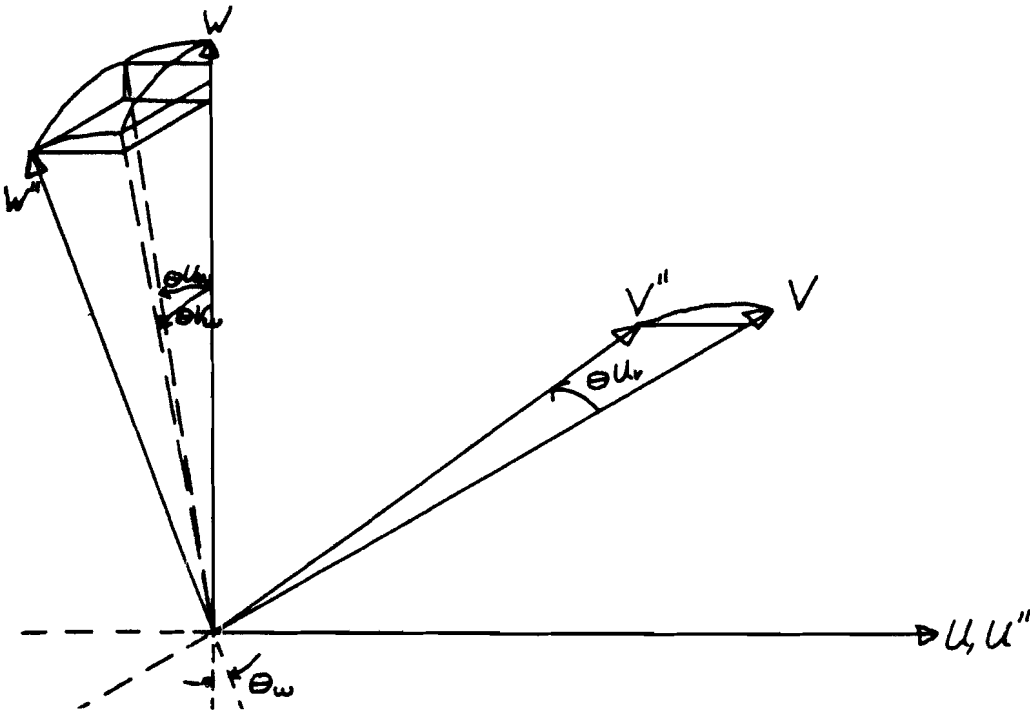
2.4.3 Haaksheidsfouten

Haaksheidsfouten worden in rekening gebracht door aan de v- en w-geleidingsvectoren respectievelijk de haaksheids-correctie-vectoren θ_{Av} en θ_{Aw} (tabel 2.3) toe te voegen. Beschouw hiertoe het orthonormale stelsel \underline{CSU} en het niet orthonormale stelsel \underline{CSU}'' (fig.2.7). Per definitie wordt nu het \underline{CSU}'' t.o.v. het \underline{CSU} zodanig genomen dat de U'' - en U -as en het $U''V''$ -en UV -vlak samenvallen. Analooq aan de in de literatuur gangbare haaksheids-hoek definitie, wordt ook hier de hoek positief genomen als de hoek tussen de geleidingen groter is dan 90 graden. Met behulp van vergelijking 2.4 volgt voor de transformatie van \underline{CSU} naar \underline{CSU}'' :

$$\begin{aligned} U'' &= U - V*\theta_{Uv} - W*\theta_{Uw} \\ V'' &= \quad \quad V - W*\theta_{Vw} \\ W'' &= \quad \quad \quad W \end{aligned} \quad (2.8)$$

De verplaatsingsfout (ΔU_h) tengevolge van het niet haaks zijn van de drie geleidingen is:

$$\Delta U_h = \begin{bmatrix} U'' - U \\ V'' - V \\ W'' - W \end{bmatrix} = \begin{bmatrix} -V*\theta_{Uv} - W*\theta_{Uw} \\ 0 - W*\theta_{Vw} \\ 0 - 0 \end{bmatrix} = \theta_{Av} + \theta_{Aw} \quad (2.9)$$



Figuur 2.7: Het niet haakse \underline{CSU} t.o.v. het \underline{CSU} .

Hieruit is af te leiden dat het verplaatsen van de eerste geleiding t.g.v. de gekozen definitie geen aanleiding geeft tot haaksheidsfouten. Door de tweede geleiding te verplaatsen vanuit het CSU-nulpunt ontstaat een haaksheids-correctie-vector $\theta \underline{A}_v$ en door de derde geleiding te verplaatsen een correctie-vector $\theta \underline{A}_w$.

2.4.4 De totale meetfout

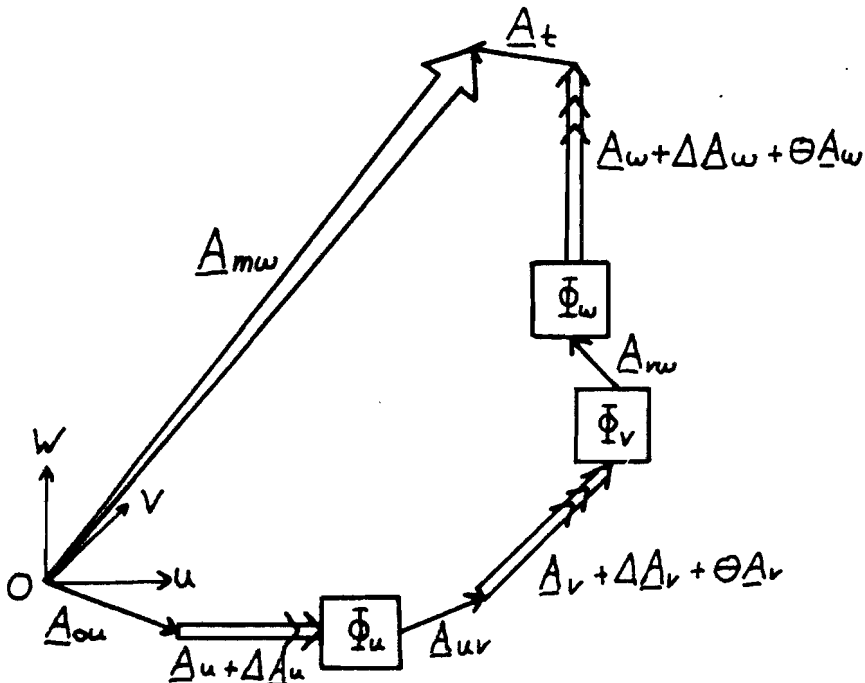
De door de 3DM waargenomen meetvector (\underline{A}_{mw}) bestaat uit de kinematische ketting van vectoren, die de ideale meetvector (\underline{A}_{mi}) vormt, waaraan op de juiste plaats de besproken correctie-vectoren en correctie-matrices zijn toegevoegd. De waargenomen meetvector is schematisch weergegeven in figuur 2.8 en wordt gegeven door:

$$\underline{A}_{mw} = \underline{A}_{ou} + \underline{A}_u + \Delta \underline{A}_u + \phi_u * (\underline{A}_{uv} + \underline{A}_v + \Delta \underline{A}_v + \theta \underline{A}_v + \phi_v * (\underline{A}_{vw} + \phi_w * (\underline{A}_w + \Delta \underline{A}_w + \theta \underline{A}_w + \underline{A}_t))) \quad (2.10)$$

De meetfout ($\Delta \underline{U}_m$) is de waargenomen meetvector (vgl.2.1) minus de ideale meetvector (vgl.2.10), oftewel:

$$\Delta \underline{U}_m = \underline{A}_{mw} - \underline{A}_{mi} \quad (2.11)$$

In tabel 2.4 wordt de uitwerking van vergelijking 2.11 gegeven waarbij tweede-orde effecten verwaarloosd zijn. De verplaatsingsfouten ΔU_i , ΔV_i en ΔW_i en de draaiingsfouten ϕ_{U_i} , ϕ_{V_i} en ϕ_{W_i} zijn allen functies die afhangen van de desbetreffende machinecoördinaat (= geleidings-coördinaat) U, V en W en de tastercoördinaten U_t , V_t en W_t , zoals aangegeven in de eerste kolom. De fouten zijn gerangschikt naar de componenten (u-, v- en w-richting) van de meetfout en naar geometrische foutbron.



Figuur 2.8: Schematische voorstelling van de vector \underline{A}_{mw} .

Tabel 2.4: De componenten ΔU_{mu} , ΔU_{mv} en ΔU_{mw} van de meetfoutvector ΔU_m . In de 3 rechterkolommen staan vermenigvuldigingsfactoren, d.w.z. bijvoorbeeld: de bijdrage in de u-component van de meetfout (ΔU_{mu}) door rotatie ϕ_{Vv} is $[W+W_{vw}] * \phi_V(V-Vt)$.

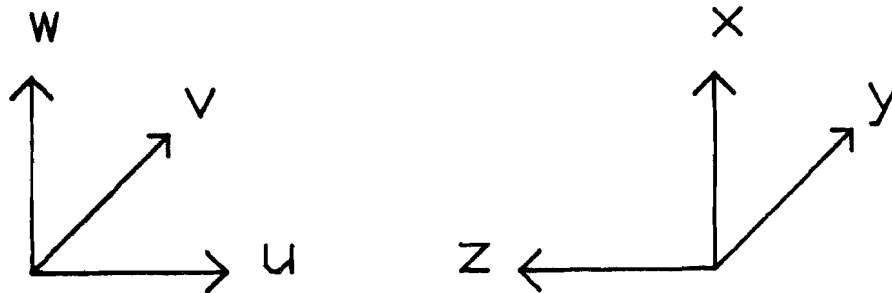
FOUTBRON	ΔU_{mu}	ΔU_{mv}	ΔU_{mw}
LINEAR.: $\Delta U_u = \Delta U(U-Ut)$ $\Delta V_v = \Delta V(V-Vt)$ $\Delta W_w = \Delta W(W-Wt)$	1	1	1
RECHTH.: $\Delta U_v = \Delta U(V-Vt)$ $\Delta U_w = \Delta U(W-Wt)$ $\Delta V_u = \Delta V(U-Ut)$ $\Delta V_w = \Delta V(W-Wt)$ $\Delta W_u = \Delta W(U-Ut)$ $\Delta W_v = \Delta W(V-Vt)$	1 1	1 1	1 1
KANTEL.: $\phi_{Uv} = \phi_U(V-Vt)$ $\phi_{Uw} = \phi_U(W-Wt)$ $\phi_{Vu} = \phi_V(U-Ut)$ $\phi_{Vw} = \phi_V(W-Wt)$ $\phi_{Wu} = \phi_W(U-Ut)$ $\phi_{Wv} = \phi_W(V-Vt)$	$d(W+W_{uv}+W_{vw})$ $d(W)$ $d(-V-V_{uv}-V_{vw})$ $d(-V_{vw}-Vt)$	$d(-W-W_{vw})$ $d(-W)$ $d(U_{uv}+U_{vw}+Ut)$ $d(U_{vw}+Ut)$	$d(V_{vw}+Vt)$ $d(Vt)$ $d(-U_{uv}-U_{vw}-Ut)$ $d(-Ut)$
ROTATIE: $\phi_{Uu} = \phi_U(U-Ut)$ $\phi_{Vv} = \phi_V(V-Vt)$ $\phi_{Ww} = \phi_W(W-Wt)$	$d(W+W_{vw})$ $d(-Vt)$	$d(-W-W_{uv}-W_{vw})$ $d(Ut)$	$d(V+V_{uv}+V_{vw})$ $d(-U_{vw}-Ut)$
HAAKSH.: θ_{Uv} θ_{Uw} θ_{Vw}	-V -V	-W	

In de tabel zien we bij de kantel- respectievelijk rotatiefouten een extra factor die, afhankelijk van de keuze van het UVW-assenkruis en de ligging van het machinenulpunt, +1 of -1 kan zijn. Deze factor, in de tabel vernoemd als "d", is gedefinieerd als:

$$d = E * T \quad (2.12)$$

Hierin zijn "E" en "T" op zich ook weer factoren die alleen +1 of -1 kunnen zijn. De factor "E" geeft aan of het gekozen UVW-assenkruis een rechtsdraaiend (E=1) dan wel linksdraaiend (E=-1) systeem is. De factor "T" beschrijft de overgang van het UVW-assenkruis naar het XYZ-assenkruis. Wanneer de richting van een as van het gekozen UVW-assenkruis tegengesteld is aan de corresponderende richting van het XYZ-assenkruis zullen de tekens met betrekking tot de kantelfouten omgekeerd zijn. Op deze manier kan men alle drie de richtingen van het UVW-assenkruis vergelijken met die van het XYZ-assenkruis. Dit levert drie richtingsfactoren T_u , T_v en T_w op.

Door deze factoren, T_u , T_v en T_w , met elkaar te vermenigvuldigen kan men tot een factor T komen die de overgang van het UVW-assenkruis naar het XYZ-assenkruis eenduidig beschrijft. In figuur 2.9 is een voorbeeld gegeven van het hetgeen hierboven beschreven is.



UVW = rechtsdraaiend $\implies e = 1$.

U tegengesteld aan Z $\implies T_u = -1$.

V gelijk aan Y $\implies T_v = 1$.

W gelijk aan X $\implies T_w = 1$.

$\implies T = -1 * 1 * 1 = -1$.

Totaal : $d = E * T = 1 * -1 = -1$.

Figuur 2.9 Bepaling van de factor d voor de kantelfouten.

Het blijkt dat de 21 mogelijke foutenbronnen die in een 3DM kunnen voorkomen 30 mogelijke fouten opleveren. Met behulp van deze fouten-analyse kan nu iedere meetfout, die resulteert uit een fout in de geometrie van de geleidingen, binnen het meet-volume worden bepaald. Hoe de 21 geometrische foutenbronnen gemeten worden, wordt hier verder buiten beschouwing gelaten. Verder zijn alle andere, niet minder belangrijke, foutenbronnen zoals bijvoorbeeld trillingen, temperatuurgradienten en aantastfouten nog niet verdisconteerd, alhoewel hun invloed vrijwel zeker niet te verwaarlozen is.

3. De programmeertaal.

3.1 De keuze van de taal.

De keuze van de programmeertaal bepaalt in hoge mate de verdere mogelijkheden van het programma. Het is daarom zaak een verantwoorde keuze te maken. Daartoe dient men eerst een pakket van eisen te formuleren waaraan de taal moet voldoen. Bij de keuze van de taal is uitgegaan van de volgende eisen :

- relatief eenvoudig te leren.
- gebruikersvriendelijk.
- gestructureerde opbouw van het programma i.v.m. latere wijzigingen c.q. toevoegingen.
- grafische mogelijkheden.
- geschikt voor Philips PC.

In principe komen de volgende talen in aanmerking: Basic, Fortran, Pascal, Turbo-Pascal en Modula. Uiteindelijk is gekozen voor Turbo-Pascal en wel om de volgende redenen:

- a) Goede handleiding beschikbaar.
- b) Grafische mogelijkheden aanwezig (Turbo-Graphics).
- c) Ten opzichte van Pascal en Modula zeer snelle compiler.
- d) Ten opzichte van Basic blijven de programma's veel overzichtelijker.
- e) Eenvoudiger ten opzichte van Fortran.

Het belangrijkste nadeel van Turbo-Pascal is echter dat maximaal 64 Kbyte in een keer gecompileerd en gerund kan worden. Dit valt echter te omzeilen en zodoende kan men programma's van welhaast onbeperkte lengte maken. Hoe dit gebeurt komt later in het verslag ter sprake.

3.2 Eisen aan de PC voor Turbo-Pascal en Turbo-Graphics.

In principe kan men Turbo-Pascal op iedere IBM-compatible PC gebruiken (2), voor Turbo-Graphics komt hier echter nog een voorwaarde bij (3). Zo dient de PC voorzien te zijn van grafische mogelijkheden in de vorm van een IBM kaart (640x200 pixels), of een Hercules kaart (720x350 pixels). Voor de Philips PC's betekent dit dat Turbo-Graphics draait op de P3200, P3102, P3100+grafische kaart. Op de P3100 is het zonder grafische kaart niet mogelijk om Turbo-Graphics te gebruiken, men kan echter wel gewoon met Turbo-Pascal werken.

3.3 Het opstarten van Turbo-Pascal.

Nadat de PC opgestart is met een IBM MS-DOS versie dient Turbo-Pascal geïnstalleerd te worden voor de computer waar men op wil werken m.b.v. het commando "TINST". Van het getoonde menu moet men "S(creentype)" kiezen en het nummer van het gebruikte grafische scherm ingeven. Bij gebruik van een hard-disk is het aan te bevelen het gehele Turbo-pakket onder een directory te plaatsen. Dan moet men ook de optie "M(sg file path)" benutten om aan te geven waar zich de file Turbo.Msg bevindt. Het grafische pakket Turbo-Graphics kan men installeren met behulp van de instructie "TGINST". Hierbij is het noodzakelijk om aan te geven welke grafische kaart in de hard-ware aanwezig is. Wanneer het gehele pakket geïnstalleerd is, kan men Turbo-Pascal eenvoudigweg opstarten vanuit diskettestation B, waar de zich datadiskette bevindt, door het commando "A:Turbo" in te geven. Het zich hierna aandienende hoofdmenu bestaat uit de volgende items:

- L: veranderen van de drive A -> B -> A
- W: aangeven van de workfile die van de diskette gehaald moet worden en waar men verder mee wil werken.
- E: inschakelen editor om programma's te schrijven of te veranderen.
- C: compileren van een gemaakt programma.
- R: compileren en runnen van een programma.
- S: save van een programma op de ingeschakelde drive (te veranderen met L).
- D: overzicht van alle files die op de ingeschakelde drive staan.
- Q: stoppen met Turbo-Pascal en terug naar het systeem.
- O: options, hier kan men aangeven wat voor soort file men wil aanmaken.

3.4 Het maken van een programma.

Het maken van een nieuw programma gebeurt door vanaf het menu, wat hiervoor besproken is, een "E" in te geven. Bij de vraag "Workfile:" geeft men de naam in van het programma dat men wil gaan schrijven. Indien nog geen file aanwezig is met dezelfde naam zal de computer dit melden met: "New file". In principe bestaat ieder programma uit de volgende structuur:

```
Program NAAM;

Var a,b,c:real;
Var d,e,f:integer;
Var g,h,i:char;
Var j,k,l:boolean;

BEGIN
    .....;
    .....;
    .....;
    BEGIN
        .....;
        .....;
    END;
    .....;
    .....;
END.
```

Bij het programmeren gelden enige algemene regels zoals:

- sluit een regel altijd af met een ";", dit geldt echter niet na een "THEN" en voor een "ELSE" in een "IF...THEN...ELSE..." instructie.
- declareer bovenaan in het programma alle in het programma voorkomende variabelen behalve die welke uitsluitend in procedures gebruikt worden.
- probeer een overzichtelijk programma te creeren door tussen een "BEGIN" en "END" een paar kolommen in te springen.
- sluit het programma altijd af met "END." i.p.v. "END;".
- controleer alvorens men het programma compileert of het aantal "BEGINs" overeenstemt met het aantal "ENDs".

3.5 Vaak voorkomende instructies en tips.

a) Het maken van een automatisch opstartende file.

Het komt vaak voor dat men enige standaard programma's iedere keer in het geheugen wil laden wanneer men de PC aanzet. Het is mogelijk dit automatisch te doen onder voorwaarde dat het te laden programma van het type COM, EXE of BAT is. In principe zijn hiertoe twee mogelijkheden:

- 1) M.b.v. het EDLIN-commando.
- 2) M.b.v. een pascal-programma.

Ad 1) Hiertoe heeft men een diskette nodig waar het programma EDLIN.COM op staat. Hieronder is de procedure gegeven om op deze manier een automatisch opstartende file aan te maken.

<u>Op computerscherm</u>	<u>Uw acties</u>
A>	Edlin autoexec.bat <CR>
-----	-----
write-protect error writing drive A.	verwissel de diskette met Uw eigen diskette.
-----	-----
Abort, Retry, Ignore	R(etry)
-----	-----
*	i
-----	-----
*i *1	naam1 <CR>
-----	-----
*i *1 naam1 *2	naam2 <CR>
-----	-----
*i *1 naam1 *2 naam2 *3	Ctrl-C
-----	-----
*i * naam1 * naam2 * ^C	end <CR>
-----	-----
*	

Ad 2) Men maakt eenvoudigweg een Turbo-Pascal file dat men "AUTOEXEC.BAT" als naam geeft. Hierin geeft men na elkaar de namen van de files die automatisch opgestart dienen te worden. Alle normale regels voor een pascal-programma zijn hier niet van toepassing, alleen de namen is voldoende.

Voorbeeld: naam1
naam2

Wanneer men nu de PC opnieuw opstart door bijv. Ctrl-Alt-Del te drukken dan starten de files naam1 en naam2 automatisch mee op.

b) Het uitprinten van een file.

```
Toets in: COPY NAAM.TYPE PRN
          vb. COPY REKEN.PAS PRN
```

c) Het uitprinten van tekst vanuit het programma.

```
Toets in: Writeln(LST,'.....');
```

d) Het invoeren van een letter in het programma zonder dat <CR> hoeft te worden gebruikt.

```
Toets in: Read(kbd,letter);
```

Letter is hier een variabele die gedeclareerd is als een character.

e) Een kleine letter veranderen in een hoofdletter.

```
Toets in: Letter:=upcase(Letter);
```

3.6 Foutmeldingen.

Bij het compileren van een programma kijkt de computer of er fouten gemaakt zijn door de programmeur. Is dit het geval dan onderbreekt hij zijn compilertaak en vertelt de gebruiker welk foutnummer hij ontdekt heeft en waar in het programma de fout ontdekt werd. De betekenis van de foutnummers staat achter in de Turbo-Pascal manual. Meestal gaat het om syntactische fouten die vrij gemakkelijk op te lossen zijn zoals een ";" vergeten of het gebruik van een variabele die niet gedeclareerd is. Er zijn echter een tweetal foutmeldingen die de programmeur voor problemen kunnen stellen. Dit zijn de volgende:

- 1) MEMORY OVERFLOW
- 2) HEAP-STACK COLLISION

In het navolgende zal besproken worden waar de fouten vandaan komen en hoe ze op te lossen of te omzeilen zijn.

1) MEMORY OVERFLOW

Deze fout kan twee oorzaken hebben, ten eerste kan het zo zijn dat U de maximale lengte dat een programma in Turbo-Pascal mag hebben, namelijk 64 Kbyte, overschrijdt. Ten tweede is er de mogelijkheid dat U een Chain-file gebruikt dat groter is dan U eerder heeft gedeclareerd, maar daarover later meer. Heeft U nog geen Chain-files dan zijn er twee mogelijkheden om toch een groter programma dan 64 Kbyte te maken. Dit zijn:

- a) Overlay procedures
- b) Chainen van aparte blokken.

Ook een combinatie van beide methoden behoort tot de mogelijkheden.

Ad a) Overlay procedures.

Wanneer men in een programma een aantal procedures heeft gemaakt die apart opgeroepen worden en elkaar niet oproepen dan kan men die procedures in een apart blok zetten. Het programma roept dan de overlay procedure pas van de diskette wanneer dat nodig is en reserveert slechts zoveel ruimte in het geheugen als nodig voor de grootste overlay procedure. Men kan binnen de overlay procedure weer een ander cluster van overlay procedures definiëren. In schema ziet een en ander als volgt uit:

PROGRAMMA	
----- Program heading -----	OVERLAY FILE .000 -----
Variabelen declaratie -----	Overlay procedure 1 -----
Overlay ruimte -----	Overlay procedure 2 -----
-----	-----
Hoofdprogramma -----	Overlay procedure 3 -----

Op de diskette wordt het eerste overlay blok weggeschreven onder naam.000. Alle volgende blokken worden op dezelfde manier weggezet waarbij het nummer iedere keer wordt opgehoogd, dus naam.001, naam.002, etc.

Ad b) Chainen van aparte blokken.

Men kan een heel groot programma opdelen in blokken die kleiner zijn dan de maximale 64 Kbyte en deze als een ketting na elkaar afwerken. Mits men bepaalde regels in acht neemt kan men ook de waarden van de variabelen overnemen naar het volgende blok. Op die manier kan men dus een programma schrijven van vrijwel onbeperkte lengte.

Daartoe dient men echter altijd aan het volgende te voldoen:

- 1) Voor overname van variabelen van het ene naar het andere blok dient men de variabelen in beide blokken onder dezelfde naam en in dezelfde volgorde te declareren. Wanneer men in het ene blok standaardfiles in het geheugen laadt mbv het "\$include"-statement dient men diezelfde files ook in het andere blok te laden, zelfs wanneer men die daar niet meer nodig heeft. Doet men dit niet dan loopt het programma vast of geeft de variabelen in het tweede blok waarden die volledig random zijn.
- 2) Een CHN-file kan alleen aangestuurd worden vanuit een ander CHN-file of vanuit een COM-file.

Houdt men deze twee regels in het oog dan is de procedure voor het aan elkaar ketenen van programmablokken als volgt:

- Zet voor de END. instructie in het tweede, derde, etc. blok de volgende commando's:

```
Assign(variabele, 'NAAM.CHN');
Chain(variabele);
```

Variabele is gedeclareerd als een file.
- Kies bij Options voor CHN-file.
- Compileer het tweede, derde, etc. blok tot een CHN versie en noteer bij ieder blok de code en datagrootte. Deze getallen zijn een hexadecimale weergave van het zojuist gecompileerde programmablok. (De CHN-files worden automatisch op diskette gezet.)
- Kies bij Options voor COM-file.
- Voer nu tevens bij code en data getallen in die in ieder geval groter zijn dan de grootste getallen die U heeft genoteerd bij het compileren van de CHN-files. Doet U dit niet dan kan later een MEMORY OVERFLOW foutmelding komen omdat de computer bij de uitvoering van een Chain opdracht een groter blok in zijn geheugen moet laden dan waarvoor ruimte was gereserveerd.
- Compileer nu het eerste blok tot een COM-file.

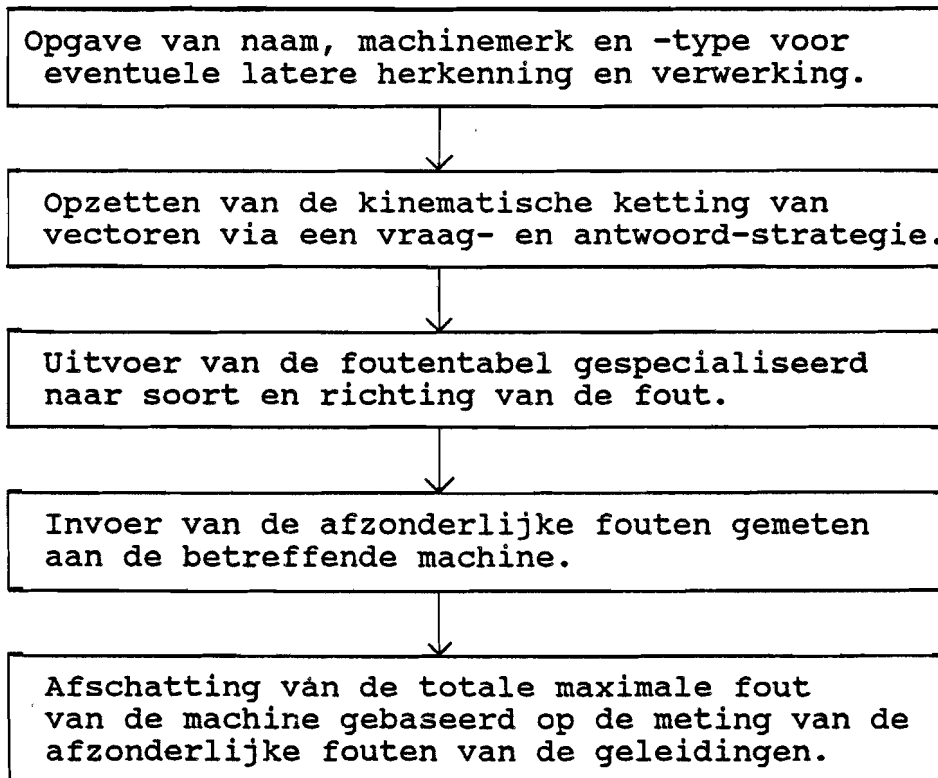
Men kan nu de reeks blokken opstarten door vanuit het systeem (dus bij "A>") de naam van het eerste programmablok in te geven gevolgd door een <CR>.

2) HEAP-STACK COLLISION

Dit is de tweede foutmelding die nogal wat moeilijkheden kan oproepen. Een frekvent voorkomende oorzaak hiervoor is dat grafische schermen te vaak weggeschreven en weer teruggehaald worden. De computer zet namelijk het scherm niet alleen op disk maar ook bovenop de stack. Men kan een opstapeling van veel ruimte kostende schermen in het geheugen voorkomen door vlak voor men het scherm wegschrijft een pijltje te zetten op de plaats in het geheugen waar de computer op dat moment is. Dit gebeurt met de instructie MARK(variabele). Zo gauw men het scherm weer teruggehaald heeft kan alles in het geheugen wat na het pijltje staat weggegooid worden m.b.v. de instructie RELEASE(variabele). De hier gebruikte variabele moet gedeclareerd worden als een pointerinteger (^integer). Omdat werkelijk alles wat NA het pijltje komt in het geheugen wordt gewist, is het zaak om erg zorgvuldig met deze instructiecombinatie om te springen.

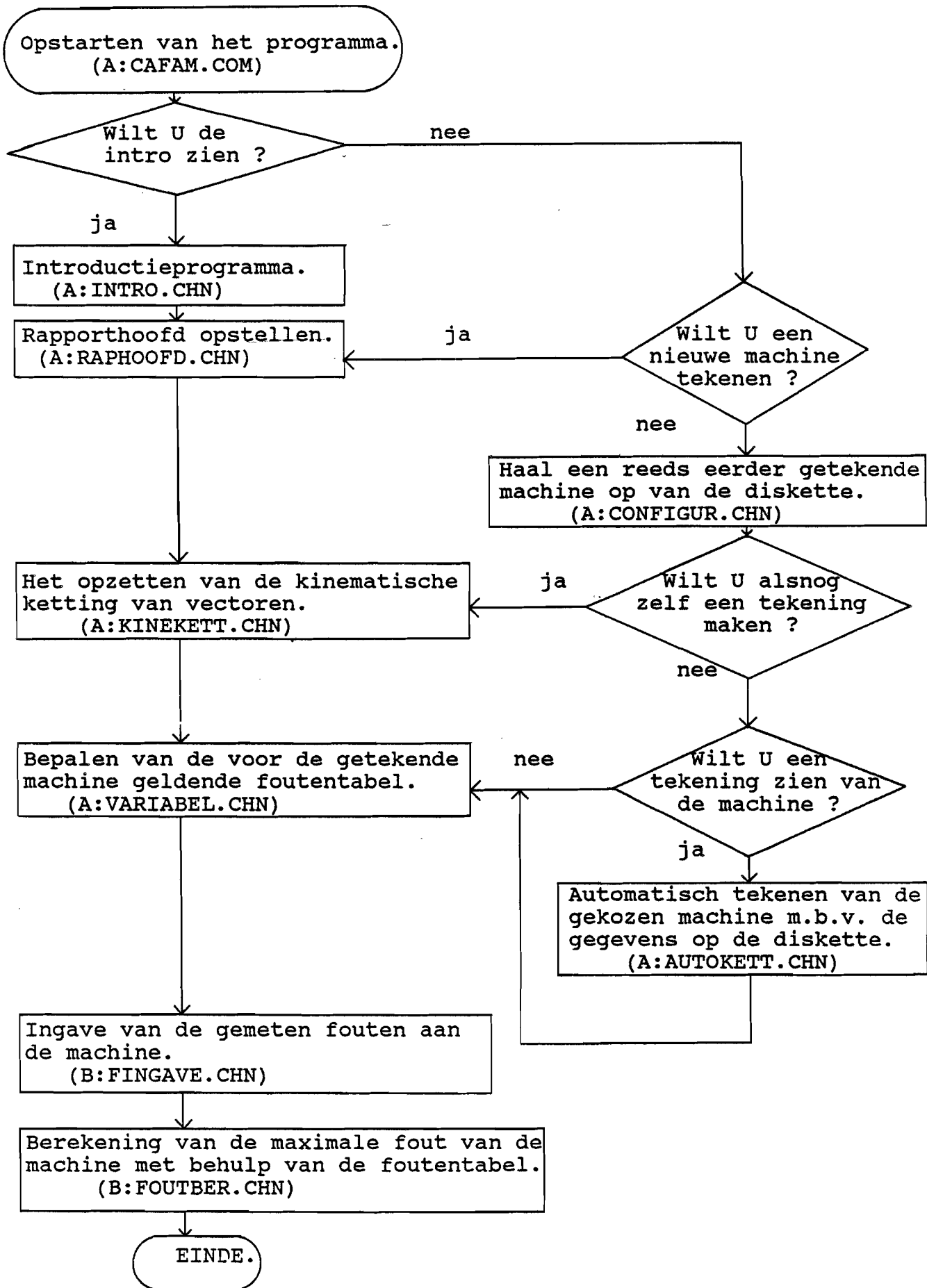
4. Het programma.

Van het gegeven model kan met behulp van de gekozen programmeertaal een programma worden geschreven. Daartoe zal men op de eerste plaats een flow-chart moeten bezitten dat de eisen aan het programma bloksgewijze weergeeft. Hieronder wordt het flow-chart gegeven dat voor dit programma als leidraad is gebruikt.



Vanuit dit eisenpakket is een programma geschreven dat in grote lijnen dezelfde blokkenstructuur heeft als het bovenstaande flow-chart. Omdat het een ondoenlijke zaak is alle afzonderlijke stappen en sprongen van het programma in flow-chart vorm weer te geven zal hier volstaan worden met een overzicht van de functionele blokken in het programma en hun onderlinge samenhang.

In feite bestaat het totale programma uit een aantal losse programmablokken die met behulp van het CHAIN-commando aan elkaar gekoppeld zijn. De toevoeging .CHN achter de naam van een programmablok betekent dan ook niets meer of minder dan dat het een gecompileerd programma betreft dat aan een vorig programma is gekoppeld. In het onderstaande flow-chart is de structuur van het programma weergegeven. In de blokken staan de hoofdfuncties en de namen van de programmablokken vermeld zoals die op de diskette voorkomen. De hoofdletters voor de namen geven aan op welke diskette het programma zich bevindt.

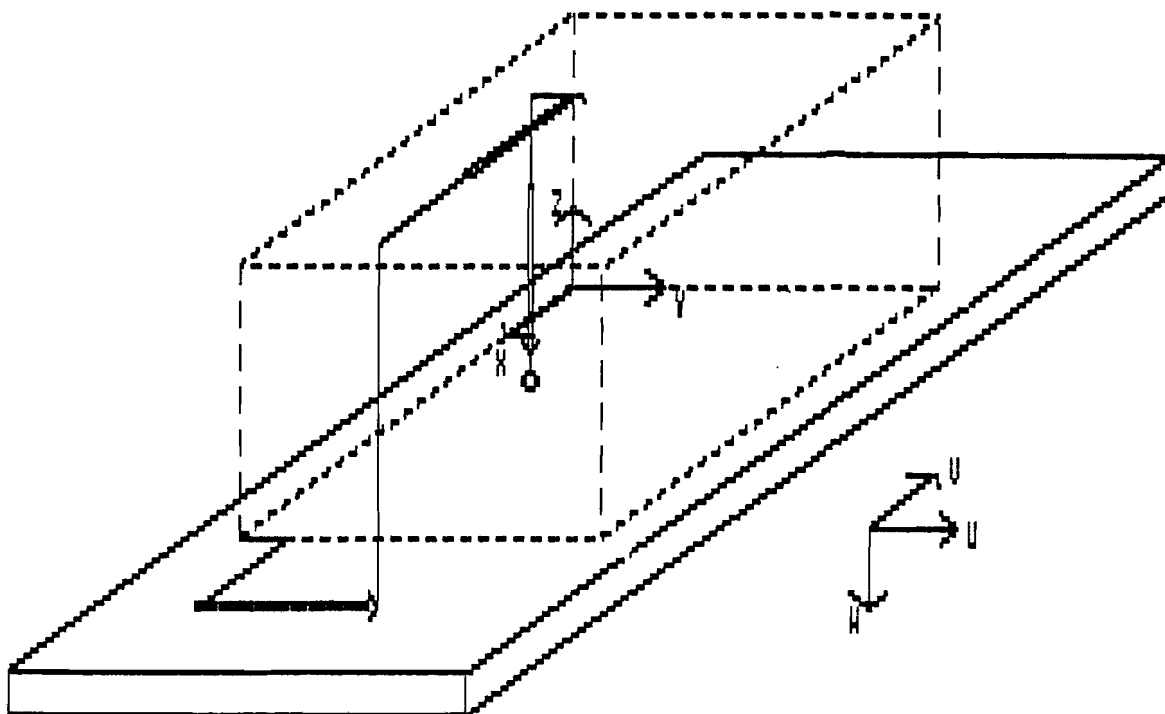


Aan de hand van dit schema is het mogelijk om vrij eenvoudig te bepalen in welk specifiek programmablok een item van het eisenpakket wordt afgehandeld en waar eventuele veranderingen c.g. aanvullingen kunnen worden aangebracht. Het feitelijke programma, dat als bijlage is toegevoegd, is voorzien van functioneel commentaar waarmee de opbouw en structuur van het geheel voor eenieder relatief eenvoudig kan worden doorgrond. Onderstaand is een voorbeeld gegeven van een analyse van een 3D-meetmachine van het portaal-type met behulp van het CAFAM-pakket waarbij slechts twee foutenbronnen gemeten zijn.

```

*****
*
* OPERATOR          MERK          TYPE          DATUM
* FRANK THEUWS     VOORBEELD     PORTAAL      1986-12-02
*
*
*****

```



ASBENOEMING : 1-ste geleiding --> U = Y
 2-de geleiding --> U = XB-X
 3-de geleiding --> W = ZB-Z

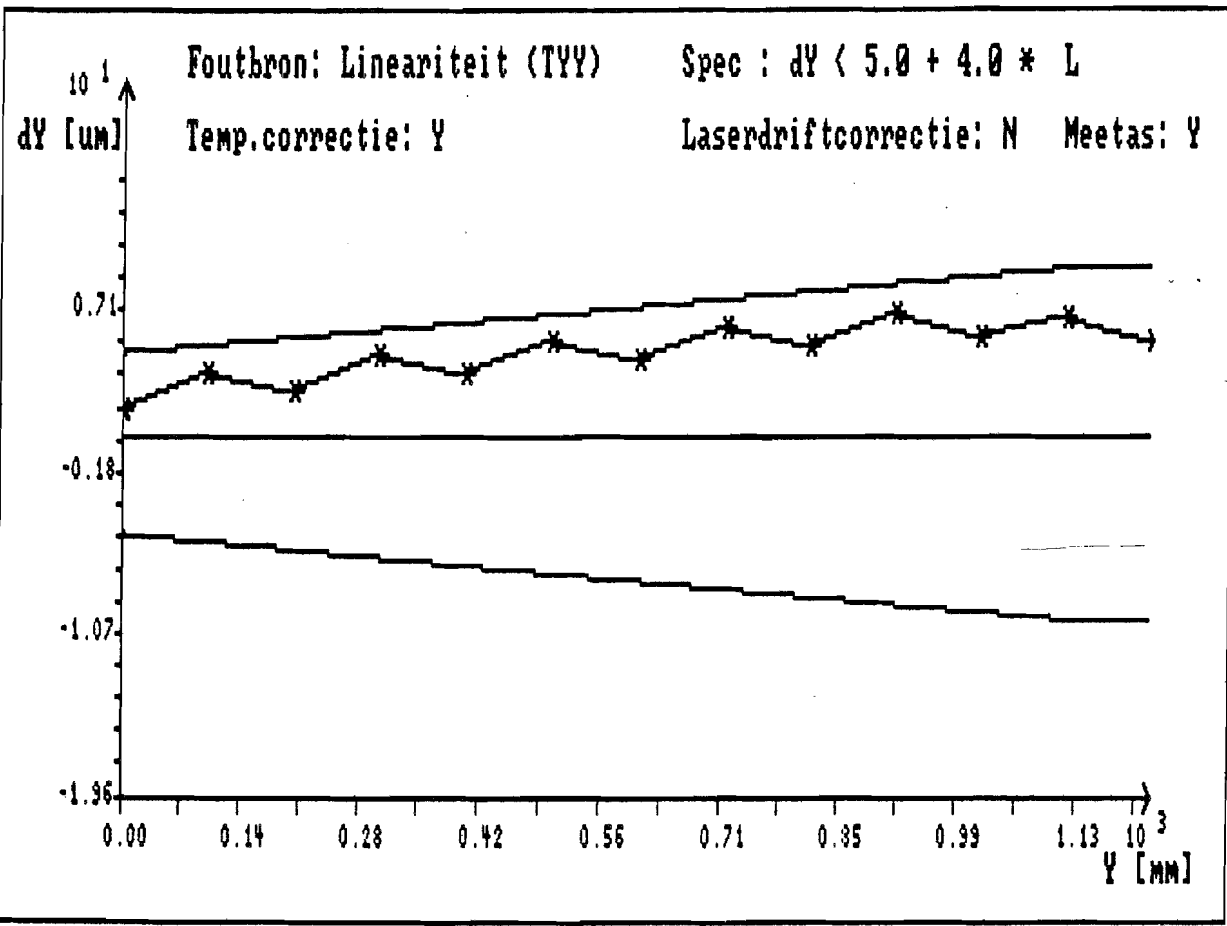
MEETBEREIK : X-bereik --> XB = 550.0 mm
 Y-bereik --> YB = 1200.0 mm
 Z-bereik --> ZB = 450.0 mm

VERBINDINGSVECTOREN: Uuv --> Yyx = 0.0 mm
 Uuv --> Xyx = -150.0 mm
 Wuv --> Zyx = 600.0 mm
 Uvw --> Yxz = -160.0 mm
 Uvw --> Xxz = 0.0 mm
 Wvw --> Zxz = -150.0 mm

TASTERCOMPONENTEN : Y-component --> Yt = 0.0 mm
 X-component --> Xt = 0.0 mm
 Z-component --> Zt = -50.0 mm

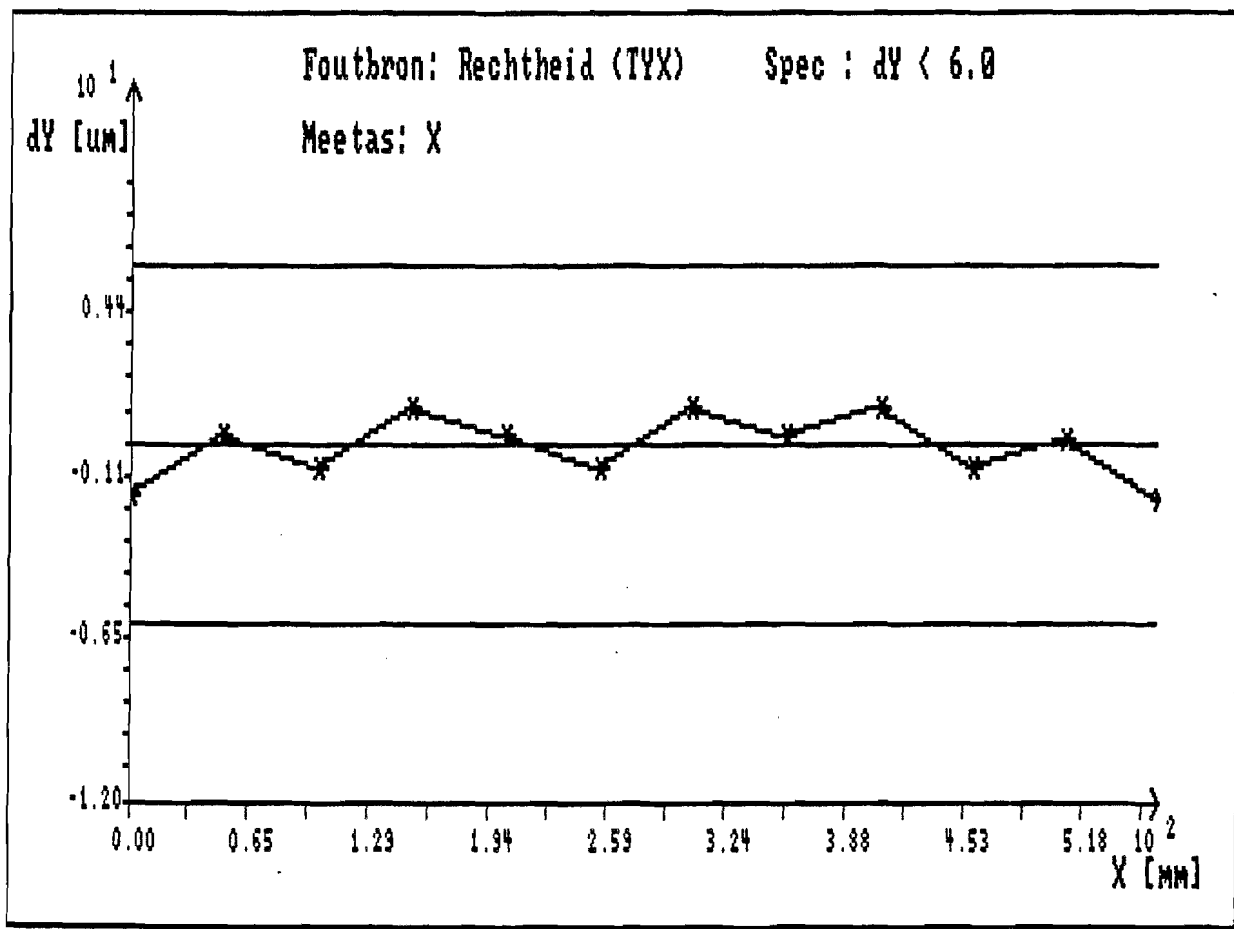
AS-COORDINATEN : X-as --> X" = X-Xt
 Y-as --> Y" = Y-Yt
 Z-as --> Z" = Z-Zt

FOUISOORT	dy	dx	dz
LINEARITEIT			
TYy=TY(y) TXx=TX(x) TZz=TZ(z")	TYy	TXx	TZz
RECHTHEID			
TYx=TY(x) TYz=TY(z") TXy=TX(y) TXz=TX(z") TZy=TZ(y) TZx=TZ(x)	TYx TYz	TXy TXz	TZy TZx
HAAKSHEID			
SYx SYz SXz	-SYx* [X] -SYz* [Z]	-SXz* [Z]	
KANTEL			
RYx=RY(x) RYz=RY(z") RXy=RX(y) RXz=RX(z") RZy=RZ(y) RZx=RZ(x)	-RXy* [Z-ZB+Zyx+Zxz] -RXz* [Z-ZB] RZy* [X-XB+Xyx+Xxz] RZx* [Xxz+Xt]	RYx* [Z-ZB+Zxz] RYz* [Z-ZB] -RZy* [Yyx+Yxz+Yt] -RZx* [Yxz+Yt]	-RYx* [Xxz+Xt] -RYz* [Xt] RXy* [Yyx+Yxz+Yt] RXz* [Yt]
ROTATIE			
RYy=RY(y) RXx=RX(x) RZz=RZ(z")	-RXx* [Z-ZB+Zxz] RZz* [Xt]	RYy* [Z-ZB+Zyx+Zxz] -RZz* [Yt]	-RYy* [X-XB+Xyx+Xxz] RXx* [Yxz+Yt]



Foutbron : IYY Temperatuurcorrectie: Y Laserdriftcorrectie: N
 X-coördinaat : 0.00 Z-coördinaat : 0.00

Machinewaarde	IYY
0.00	2.00
100.00	3.99
200.00	2.96
300.00	4.91
400.00	3.84
500.00	5.75
600.00	4.64
700.00	6.51
800.00	5.36
900.00	7.19
1000.00	6.00
1100.00	6.79
1200.00	5.56



Foutbron : TYX Y-coördinaat : 0.00 Z-coördinaat : 0.00

Machinewaarde	TYX
0.00	-1.63
50.00	0.36
100.00	-0.64
150.00	1.35
200.00	0.34
250.00	-0.66
300.00	1.33
350.00	0.32
400.00	1.32
450.00	-0.69
500.00	0.30
550.00	-1.71

```

#####
#                                     #
#                   CAFAM FOUTENBEREKENING                   #
#                   -----                   #
#   De fout in X-richting   :   0.0000   um.                 #
#   De fout in Y-richting   :   7.3894   um.                 #
#   De fout in Z-richting   :   0.0000   um.                 #
#   TOTALE MAXIMALE FOUT   :   7.3894   um.                 #
#                                                                 #
#####

```


5. Conclusies.

Met behulp van het hier gepresenteerde programma is het mogelijk om snel een inzicht te krijgen in de geometrische foutenstructuur van een 3D-meetmachine. Men kan de resultaten die voortvloeien uit dit programma gebruiken om die foutenbronnen te extraheren die de grootste invloed hebben op de nauwkeurigheid van de machine en deze dan kwantitatief bepalen. Ingevoerd in het programma kan zo een relatief snelle afschatting gegeven worden van de maximale onnauwkeurigheid van de machine. Hierbij wordt echter alleen aandacht geschonken aan de geometrische fouten en worden zaken als temperatuursinvloeden, trillingen, massakrachten en statische vervormingen buiten beschouwing gelaten. Het verdient echter aanbeveling om ook deze invloeden in dit pakket op te nemen. In de nabije toekomst zal nog een blok aan het bestaande pakket worden toegevoegd dat de problematiek van de testobjecten poogt te vereenvoudigen. Resumerend kan geconcludeerd worden dat het programma een goede aanzet vormt in de richting van het computerondersteund controleren van 3D-meetmachines.

Literatuurlijst.

- (1) Collegediktaat B.O. Productiemiddelen, deel Ir. J. Teeuwsen, nr. nog niet bekend.
- (2) Turbo-Pascal reference manual, version 3.0.
- (3) Turbo-Graphics Toolbox owners handbook.

BIJLAGEN

Computermodellering van de
foutenstructuur van 3D-meetmachines.

Auteur : F.C.C.J.M. Theuws.

WPA-rapport nr. 0355, dec. 1986.

Verslag Onderzoekopdracht WPA

Afstudeerhoogleraar: Prof. Dr. Ir. A.C.H. van der Wolf.

Begeleiders : Dr. Ir. P.H.J. Schellekens,
Ir. J.W.M.C. Teeuwsen.

Inhoud.

<u>Programmanaam</u>	<u>blz.</u>
Cafam.....	1
Intro.....	2
Rapporthoofd.....	7
Kinekett.....	10
Variabel.....	38
Fouteningave.....	52
Foutenberekening.....	83
Configuratie.....	87
Autokinekett.....	90

```
PROGRAM CAFAM;
```

```
Var eerste:file;           {hulpvariabele om de volgende file te koppelen}  
Var intronee:integer;      {variabele die de keuze aangeeft}
```

```
BEGIN
```

```
  clrscr;  
  gotoxy(12,8);  
  write('Maak Uw keuze uit de volgende mogelijkheden : ');  
  gotoxy(25,12);  
  write('1: Het totale programma.');
```

gotoxy(25,14);
write('2: Het opzetten v/d kinematische ketting van vectoren.');

gotoxy(25,16);
write('3: Foutenanalyse van een reeds bekende machine.');

gotoxy(60,8);
readln(intronee); {lees de keuze in}

```
  case intronee of  
    1: BEGIN assign(eerste,'intro.chn'); chain(eerste); END;  
    2: BEGIN assign(eerste,'raphoofd.chn'); chain(eerste); END;  
  END;  
  assign(eerste,'configur.chn'); {koppel het volgende blok afhankelijk}  
  chain(eerste);                {van de keuze van de operator}
```

```
END.
```

```

{$C-}
PROGRAM INTRO;

{$I a:typedef.sys}           {Include de standaard grafische files van de}
{$I a:graphix.sys}           {Turbo-systeem diskette}
{$I a:kernel.sys}
{$I a:windows.sys}

Var tweede:file;             {hulpvariabele om het volgende blok te koppelen}

PROCEDURE DELAY(n: real);    {deze procedure genereert een wachttijd maar}
var i:real;                  {kan afgebroken worden door een CR in te geven}
    j:integer;
    ch:char;
    quit:boolean;

BEGIN
    i:=0;
    ch:= ' ';
    repeat
        i:=i+1;
        quit:=false;
        if keypressed then
            BEGIN
                read(kbd,ch);
                quit:=(ch='^C');
                if (ch='^[') and keypressed then
                    BEGIN
                        read(kbd,ch);
                        quit:=(ch='^O');
                        ch:= ' ';
                    END;
            END;
        if quit then
            BEGIN
                leavegraphic;
                halt;
            END;
        until (ch='^M') or (i)>=n);
    END;

PROCEDURE SELECTIBM;        {zet de world en het window op IBM grootte}
BEGIN
    SelectWorld(2);
    SelectWindow(11);
END;

PROCEDURE DEFINEWINDOWIBM(i,X1,Y1,X2,Y2:integer); {Definieer een werkwindow}
BEGIN
    DefineWindow(i,Trunc(X1/79*XMax61b+0.001),Trunc(Y1/199*YMax61b+0.001),
                Trunc(X2/79*XMax61b+0.5),Trunc(Y2/199*YMax61b+0.5));
END;

```

```

PROCEDURE INTRO;
var i,k:integer;                                (dit zijn telvariabelen)
BEGIN
  SetHeaderOff;
  DefineWindowIBM(1,5,0,10,40);                 (window voor letter C)
  DefineWindowIBM(2,5,41,10,80);               (window voor letter A)
  DefineWindowIBM(3,5,81,10,120);              (window voor letter F)
  DefineWindowIBM(4,5,121,10,160);             (window voor letter A)
  DefineWindowIBM(5,5,161,10,199);             (window voor letter M)
  DefineWindowIBM(6,11,0,79,40);                (window voor oputer)
  DefineWindowIBM(7,11,41,79,80);              (window voor ided)
  DefineWindowIBM(8,11,81,79,120);             (window voor aults)
  DefineWindowIBM(9,11,121,79,160);            (window voor nalysis of 3-D)
  DefineWindowIBM(10,11,161,79,199);          (window voor easuringmachines)
  DefineWindowIBM(12,15,5,60,40);              (window voor CAFAM)
  DefineWindowIBM(15,12,50,68,180);           (window voor tekening 3D machine)
  SelectWindow(1);
  SelectIBM;
  DrawTextW(55,20,6,'C');                       (zet COMPUTER op het scherm)
  DrawTextW(90,30,3,'OMPUTER');
  StoreWindow(1);
  delay(900);
  SelectWindow(2);
  SelectIBM;
  DrawTextW(55,60,6,'A');                       (zet AIDED op het scherm)
  DrawTextW(90,70,3,'IDED');
  StoreWindow(2);
  delay(900);
  SelectWindow(3);
  SelectIBM;
  DrawTextW(55,100,6,'F');                       (zet FAULTS op het scherm)
  DrawTextW(90,110,3,'AULTS');
  StoreWindow(3);
  delay(900);
  SelectWindow(4);
  SelectIBM;
  DrawTextW(55,140,6,'A');                       (zet ANALYSIS of 3D op het scherm)
  DrawTextW(90,150,3,'NALYSIS of 3D');
  StoreWindow(4);
  delay(900);
  SelectWindow(5);
  SelectIBM;
  DrawTextW(55,180,6,'M');                       (zet MEASURINGMACHINES op het scherm)
  DrawTextW(90,190,3,'EASURINGMACHINES');
  StoreWindow(5);
  delay(4000);                                    (wacht 4 sec)
  for i:=6 to 10 do
    BEGIN
      selectwindow(i);                            (veeg alles na de eerste letters uit)
      setbackground(0);                          (zodat alleen CAFAM blijft staan)
    END;

```

```

selectwindow(1);           {verplaats de C 10 hor.}
movehor(10,false);
selectwindow(2);           {verplaats de A 20 hor en 40 omhoog}
movehor(20,false);
movever(-40,false);
selectwindow(3);           {verplaats de F 30 hor en 80 omhoog}
movehor(30,false);
movever(-80,false);
selectwindow(4);           {verplaats de A 40 hor en 120 omhoog}
movehor(40,false);
movever(-120,false);
selectwindow(5);           {verplaats de M 50 hor en 160 omhoog}
movehor(50,false);
movever(-160,false);
for k:=1 to 8 do
  BEGIN
    for i:=1 to 5 do
      BEGIN
        selectwindow(i);     {inverteer de opeenvolgende letters}
        invertwindow;
        delay(70);
      END;
      delay(70);
    END;
  for i:=1 to 4 do
    BEGIN
      selectwindow(12);      {inverteer alle letters samen}
      invertwindow;
      delay(300);
    END;
  delay(2000);
  SelectWindow(15);         {teken de 3D meetmachine}
  SetBackground(0);
  DrawBorder;
  BEGIN
    SetAspect(1);           {Zet de hor-vert verhouding van een cirkel op 1}
    DrawLine(50,180,50,190);
    DrawLine(50,190,300,190);
    Drawline(300,190,300,180);
    Drawline(300,180,50,180);
    Drawline(50,180,170,140);
    Drawline(170,140,180,140);
    Drawline(180,140,240,120);
    Drawline(170,140,170,50);
    Drawline(180,140,180,70);
    Drawline(240,120,240,70);
    Drawline(410,70,410,140);
    Drawline(410,140,420,140);
    Drawline(420,50,420,140);
    Drawline(300,180,600,80);
    Drawline(480,120,480,30);
    Drawline(420,50,480,30);
    Drawline(170,50,230,30);
    Drawline(300,190,600,90);
    Drawline(600,90,600,80);
  END;

```



```

Drawline(240,116,288,100);
Drawline(390,80,410,80);
Drawline(480,80,600,80);
Drawline(170,50,260,50);
Drawline(180,70,260,70);
Drawline(230,30,260,30);
Drawline(260,25,260,100);
Drawline(260,100,320,100);
Drawline(320,100,395,75);
Drawline(320,100,320,25);
Drawline(320,25,395,4);
Drawline(335,4,395,4);
Drawline(260,25,335,4);
Drawline(260,25,320,25);
Drawline(395,4,395,30);
Drawline(395,70,395,75);
Drawline(330,50,390,30);
Drawline(330,50,330,70);
Drawline(390,30,480,30);
Drawline(330,50,420,50);
Drawline(330,70,410,70);
Drawline(313,100,313,125);
Drawline(343,92,343,125);
Drawline(313,125,343,125);
Drawline(328,125,328,135);
Drawcircle(328,135,0.05);
END;
delay(4000);                {wacht 4 sec}
SelectScreen(2);           {ga in het RAM scherm}
ClearScreen;
InvertScreen;              {inverteer dit scherm}
SetColorBlack;            {zet de tekenkleur zwart}
SelectIBM;
DrawTextW(40,95,3,'Door');
DrawTextW(40,140,4,'F.THEUWS en J.TEEUNSEN');
CopyScreen;                {zet het RAM scherm op het zichtbare scherm}
SelectScreen(1);
delay(3000);
DefineWindowIBM(1,1,128,28,158);    {maak een window rond de namen}
StoreWindow(1);
DefineWindowIBM(2,31,125,78,157);
StoreWindow(2);
for i:=1 to 20 do drawline(170+(3*i),180-i,540+(3*i),180-i);    {teken het }
for i:=1 to 11 do drawline(170,191-i,540,191-i);                {modderbad}
for i:=1 to 11 do drawline(540,191-i,600,171-i);
setcolorwhite;
DrawtextW(300,182,2,'MODDER');
setcolorblack;
delay(1000);

```

```

for i:=1 to 16 do
  BEGIN
    SelectWindow(2);           {kies het window met en J.TEEUWSEN erin}
    MoveVer(2,false);         {verplaats het 2 omlaag}
    ReDefineWindow(2,X1RefG1b,Y1RefG1b,X2RefG1b,Y2RefG1b-2);
    DELAY(600);               {maak het vert 2 kleiner aan de onderzijde}
  END;
for i:=1 to 10 do
  BEGIN
    selectwindow(1); {laat het window met F.THEUWS een paar keer knippen}
    invertwindow;
    delay(100);
  END;
  delay(1000);
  GotoXY(1,25);              {maak de onderste regel op het scherm zuiver}
  ClrEol;
  GotoXY(17,25);
  SetColorWhite;            {schrijf de volgende tekst in het wit}
  Write('Copyright (C) 1986 Frank Theuws.');
```

-----}

```

BEGIN
  InitGraphic;                {ga in de grafische mode}
  DefineWorld(2,0,199,639,0); {kies de world en het window gelijk}
  DefineWindow(11,0,0,XMaxG1b,YMaxG1b); {aan de resolutie v/d grafische kaart}
  INTRO;                      {doe de intro}
  LeaveGraphic;               {ga uit de grafische mode}
  assign(tweede,'raphoofd.chn'); {koppel raphoofd}
  chain(tweede);
END.
```

PROGRAM RAPPORTHOOFD;

```
($I a:typedef.sys)      (Include de grafische files van de Turbo-diskette)
($I a:graphix.sys)      (hoewel ze hier niet nodig zijn moeten ze geladen)
($I a:kernel.sys)       (worden om de variabelen in ieder blok op dezelfde)
($I a:windows.sys)      (plaats te krijgen zodat ze overgenomen kunnen worden)
```

```
Type Letters=packed array[1..20] of char;  (definieer letters als een rij van)
Type Dag=packed array[1..16] of char;      (maximaal 20 karakters en dag als )
                                           (een rij van maximaal 16 karakters)
```

```
Var uber,vber,wber:real;
Var startpunt,twegel,np,uas,vas,was:char;
Var draaiing:real;
Var uuv,uvw,vuv,vvm,wuv,wvm:real;
Var ucomp,vcomp,wcomp:real;
Var verb1,verb2,verb3:char;
Var verb4,verb5,verb6:char;              (de bovenstaande variabelen worden)
Var merk,tiepe:string[20];              (hier niet gebruikt doch moeten wel genoemd)
Var merkk,tiepee:letters;              (worden om een goede aansluiting te krijgen)
Var derde:file;                          (van de blokken)
                                           (merk, tiepe, etc worden hier wel gebruikt)
```

PROCEDURE RAPPORT;

```
Var naam:letters;                      (naam van de operator)
Var datum:dag;                          (datum van de meting)
Var keuring,afdruk:char;                (keuzevariabelen voor herstel fouten en printen)
Var t:integer;                           (hulpvariabele voor tellen)
```

BEGIN

```
keuring:='N';
while (keuring='N') do                  (blok herhalen totdat het goedgekeurd wordt)
  BEGIN                                  (hier gaan we na of alles goed is ingetypt)
    ClrScr;                               (maak scherm zuiver)
    Gotoxy(7,5);
    Write(' Toets de naam v/d operator in.      ');
    Readln(naam);                          (naam wordt opgeslagen onder naam v/h type letters)
    Gotoxy(7,7);
    Write(' Toets het merk v/d te meten machine in. ');
    Readln(merkk);
    Gotoxy(7,9);
    Write(' Toets het type v/d te meten machine in. ');
    Readln(tiepee);
    Gotoxy(7,11);
    Write(' Toets achtereenvolgens jaar-maand-dag in. ');
    Readln(datum);                          (datum wordt opgeslagen onder datum.dag)
    Writeln;
    ClrScr;
    for t:=1 to 80 do Write('*');Writeln;      (maak bovenregel kader)
    Writeln('*',' ':78,'*');Writeln('*',' ':78,'*');
    Writeln('* ','OPERATOR',' ':12,'MERK',' ':16,'TYPE',' ':16,'DATUM',
             ' ':11,'*');
    Writeln('*',' ':78,'*');
    Write('* ',naam);
    Write(merkk);
```

```

Write(tiepee);
Writeln(datum,'*');          (zet de ingegeven namen op het scherm)
Writeln('*',' ':78,'*');Writeln('*',' ':78,'*');
for t:=1 to 80 do Write('*');
Writeln;Writeln;
Writeln(' Is hetgeen u ingegeven hebt correct ? (Y/N)');
Read(kbd,keuring);
keurings:=upcase(keuring);
Writeln;
END;
Writeln;Writeln(' Wilt u dit afgedrukt hebben ? (Y/N)');
Read(kbd,afdruk);
afdruk:=upcase(afdruk);
Writeln;
if (afdruk='Y') then      (print het ingegeven blok als men een afdruk wil)
BEGIN
write(LST,'-----');
writeln(LST,'-----');
write(LST,'-----');
writeln(LST,'-----');
writeln(LST);
write(LST,'   CCCCCCCC   AAAAA   FFFFFFFF   AAAAA');
writeln(LST,'   MM     MM   ');
write(LST,'   CC     AA   AA   FF     AA   AA');
writeln(LST,'   MMMM   MMMM ');
write(LST,'   CC     AA   AA   FF     AA   AA');
writeln(LST,'   MM MMMMMM MM ');
write(LST,'   CC     AA   AA   FFFFFF   AA   AA');
writeln(LST,'   MM   MM   MM ');
write(LST,'   CC     AAAAAAAAAA   FF     AAAAAAAAAA');
writeln(LST,'   MM     MM ');
write(LST,'   CC     AA   AA   FF     AA   AA');
writeln(LST,'   MM     MM ');
write(LST,'   CC     AA   AA   FF     AA   AA');
writeln(LST,'   MM     MM ');
write(LST,'   CCCCCCCC   AA   AA   FF     AA   AA');
writeln(LST,'   MM     MM ');
writeln(LST);
writeln(LST);
write(LST,'   Computer   Aided   Faults ');
writeln(LST,'   Analysis of 3-D Measuringmachines');
writeln(LST);
write(LST,'-----');
writeln(LST,'-----');
write(LST,'-----');
writeln(LST,'-----');
writeln(LST); writeln(LST);
writeln(LST); writeln(LST);
writeln(LST); writeln(LST);
for t:=1 to 80 do Write(LST,'*');Writeln(LST); {maak bovenregel kader}
Writeln(LST,'*',' ':78,'*');Writeln(LST,'*',' ':78,'*');
Writeln(LST,'* ', 'OPERATOR', ' ':12, 'MERK', ' ':16, 'TYPE', ' ':16, 'DATUM',
', ' ':11, '*');
Writeln(LST,'*',' ':78,'*');
Write(LST,'* ',naam);

```

```

Write(LST,merkk);
Write(LST,tiepee);
Writeln(LST,datum,'*');           {zet de ingegeven namen op de printer}
Writeln(LST,'*',':78,*');Writeln(LST,'*',':78,*');
for t:=1 to 80 do Write(LST,'*');
Writeln(LST);writeln(LST);writeln(LST);writeln(LST);writeln(LST);
Writeln(LST);writeln(LST);writeln(LST);writeln(LST);
END;
END;

BEGIN                               {hier start het feitelijke programma}
RAPPORT;                             {voer procedure rapport uit}
merk:=merkk;                          {geef het ingevoerde merk en type mee aan}
tiepe:=tiepee;                        {de variabelen merk en tiepe voor latere opslag op disk}
assign(derde,'kinekett.chn');{ken aan de hulpvariabele derde de naam toe van}
chain(derde);                          {het volgende blok en koppel dit blok hier aan dit blok}
END.

```

PROGRAM KINEKETT;

```
($I a:typedef.sys)      (include deze files van de Turbo-Graphics diskette)
($I a:graphix.sys)
($I a:kernel.sys)
($I a:windows.sys)
```

```
Type machine=record      (dit record wordt later gebruikt)
    uber,vber,wber:real;  (om de machinegegevens naar disk)
    startpunt,twegel,np,uas,vas,was:char; (te schrijven)
    draaiing:real;
    uuv,uvw,vuv,vvw,wuv,wvw:real;
    ucomp,vcomp,wcomp:real;
    verb1,verb2,verb3:char;
    verb4,verb5,verb6:char;
    merk,tiepe:string[20];
end;
```

```
Var uber,vber,wber:real;      (bereiken van de u, v en w-as)
Var startpunt,twegel,np,uas,vas,was:char; (np=nulpunt,uas=x/y/z,twegel=y/n)
Var draaiing:real;
Var uuv,uvw,vuv,vvw,wuv,wvw:real; (de componenten v/d verbindingsvectoren)
Var ucomp,vcomp,wcomp:real;      (dit zijn de tastercomponenten)
Var verb1,verb2,verb3:char;      (verbindingscomponenten tussen u en v-as)
Var verb4,verb5,verb6:char;      (verbindingscomponenten tussen v en w-as)
Var merk,tiepe:string[20];      (merk en type overgenomen van het vorige blok)
```

```
Var ht:^integer;              (pointer die de plaats in het geheugen aangeeft)
Var xb,yb,xe,ye,keus:real;    (begin en eindpunten in de TEKEN procedure)
Var richting:char;            (richting=h/s/v)
Var hs,vs,ss:real;            (deze geven het bereik en de schermfactoren)
Var unul,vnul,wnul:real;      (de nul vectorcomponenten)
Var v,s,d,vhulp,shulp,dhulp:real; (d=hor.richting=1 of -1,s=schuin,v=verticaal)
Var kleiner:char;              (keuzevariabele voor het verkleinen van de tekening)
Var k,i,t,q,l:integer; (k=schaalfactor voor verkleinen,i en t zitten in for-to)
Var a,b,c,g,uverpl,vverpl,wverpl:real; (zie proc.meettafel2)
Var x1,y1,x2,y2,x3,y3,x4,y4,x5,y5,x6,y6,x7,y7:real;
Var pyl,cr:char;              (pyl geeft richting hoedje aan op de vectoren)
Var nr,f:integer;              (zie proc. verbind en proc.fout)
Var vb:char;
Var vector,vraag:boolean;     (vector:zie proc.teken,vraag:zie proc.verbind)
Var htast,vtast,wtast:real;   (zie proc.tastvector)
Var zeker:char;               (keuzevariabele voor het verlaten van het programma)
Var config:machine;           (variabele met daarin alle specifieke data van de machine)
Var datafile:file of machine; (file op disk met daarin blokken machine)
Var variabel:file;             (hulpvariabele voor de koppeling met het volgende blok)
```

```

PROCEDURE FOUT(f:integer);forward;      (declaratie van procedures die eerder)
PROCEDURE TOTAAL1;forward;              (opgeroepen worden dan ze in het programma)
PROCEDURE TOTAAL2;forward;              (vermeld zijn)
PROCEDURE TOTAAL3;forward;
PROCEDURE TOTAAL4A;forward;
PROCEDURE TOTAAL4B;forward;
PROCEDURE TOTAAL4C;forward;
PROCEDURE TOTAAL5;forward;
PROCEDURE TOTAAL6;forward;
PROCEDURE TOTAAL7;forward;
PROCEDURE TOTAAL8;forward;
PROCEDURE VERKLEIN;forward;

```

```

OVERLAY PROCEDURE TEKST;                (nu volgen een hele hoop tekstprocedures)
BEGIN                                    (die in een overlayblok gezet zijn omdat ze nooit)
  clrscr;                                (tegelijk opgeroepen worden, dit om ruimte te besparen)
  for t:=1 to 80 do write(' ');
  write(' ', ' ':78, ' ');
  write(' ', ' ':78, ' ');
  write(' Dit is een programma dat U in staat stelt uw 3D-meetmachine ');
  write(' te tekenen als een ketting van vectoren.Het begint met ');
  write(' het tekenen van een meetvolume met daarbij de vraag ');
  write(' waar U de eerste geleiding wilt laten beginnen.Daarna volgen ');
  write(' de definities van de richtingen van de tweede en de derde ');
  write(' geleiding.Steeds ziet U rechts onderaan op het scherm het ');
  write(' door U gekozen assenkruis staan.De geleidingsvectoren hebben ');
  write(' een constante grootte maar de verbindingsvectoren moeten ');
  write(' door U zelf worden ingegeven.De volgorde kunt U zelf bepalen. ');
  write(' Een vraag waarbij een of meerdere letters moet worden ingegeven ');
  write(' mag niet met <CR> worden afgesloten,doch de invoer van getallen ');
  write(' dient wel met <CR> afgesloten te worden.Steeds wanneer het menu ');
  write(' op het scherm staat kunt U hulp krijgen door een H in te toetsen');
  write(' of het programma verlaten door een Q in te geven. ');
  write(' ', ' ':78, ' ');
  write(' ', ' ':78, ' ');
  write(' Druk op <CR> voor start programma. ');
  write(' ', ' ':78, ' ');
  for t:=1 to 80 do write(' ');
  writeln;
  read;
END;

```

OVERLAY PROCEDURE STARTINFO;

BEGIN

```
mark(ht);                                (zet een pijltje in het geheugen)
savescreen('scherm');                    (zet het huidige scherm op disk onder 'scherm')
leavegraphic;                             (ga uit de grafische mode)
clearscreen;                              (maak het scherm vrij)
for t:=1 to 80 do write('*');            (schrijf een kader)
write('*',' ':78,'*');write('*',' ':78,'*');    (en de tekst)
write('* U wordt gevraagd aan te geven waar in uw geval *');
write('* het beginpunt ligt van de eerste geleiding. *');
write('* Dit hoeft niet altijd de X-geleiding te zijn. *');
write('* U moet de weg naar de eerste volgen via de *');
write('* geleidingen en met de eerste geleiding wordt die *');
write('* geleiding bedoeld waarop U normaalgesproken Uw weg *');
write('* naar de taster begint.Het kan mogelijk zijn dat *');
write('* de eerste geleiding niet aan de voorkant van de *');
write('* machine ligt.In dat geval wordt U geacht zodanig *');
write('* naar de machine te kijken dat de eerste geleiding *');
write('* wel voor ligt.Ook kan de tekening op het beeld een *');
write('* onjuiste voorstelling geven van de werkelijke verhoudingen *');
write('* van de machinematen.Zo hoeft de langste zijde van *');
write('* de meetafel op de tekening niet overeen te stemmen *');
write('* met het werkelijke grootste meetbereik. *');
write('*',' ':78,'*');
write('*',' ':78,'*');
write('* Druk op <CR> voor terug naar het programma. *');
write('*',' ':78,'*');
for t:=1 to 80 do write('*');
writeln;
read;
entergraphic;                             (ga terug in de grafische mode)
loadscreen('scherm');                    (haal het scherm terug van de diskette)
release(ht);                              (gooi alles na het pijltje in het geheugen weg)
TOTAAL;
```

END;


```

OVERLAY PROCEDURE VASINFO;
BEGIN
  mark(ht);
  savescreen('scherM');
  leavegraphic;
  clearscreen;
  for t:=1 to 80 do write('*');
  write('*',' ':78,'*');write('*',' ':78,'*');
  write('* Met de V-as wordt hier de tweede geleiding bedoeld. *');
  write('* Die tweede geleiding kan zowel in horizontale (A/B) *');
  write('* als in verticale richting lopen.Bedenk steeds dat we *');
  write('* in de richting van de taster moeten en tevens dat de *');
  write('* positieve of negatieve richting van de machinecoördinaat *');
  write('* niets te maken heeft met de richting van de gevraagde *');
  write('* geleiding. *');
  write('* Voorbeeld: 1) Bij een portaalmachine ligt *');
  write('* de tweede geleiding vaak boven en wijst naar achteren, *');
  write('* dus zouden we hier keuze A moeten maken. *');
  write('* 2) Bij een C-type gaat de tweede *');
  write('* geleiding vaak omhoog en zouden we voor alternatief C *');
  write('* moeten kiezen. *');
  write('*',' ':78,'*');
  write('*',' ':78,'*');
  write('* Druk op <CR> voor terug naar het programma. *');
  write('*',' ':78,'*');
  write('*',' ':78,'*');
  for t:=1 to 80 do write('*');
  writeln;
  read;
  entergraphic;
  loadscreen('scherM');
  release(ht);
  TOTAAL2;
END;

```

OVERLAY PROCEDURE WASINFO;

(voor uitleg zie procedure STARINFO)

BEGIN

```
mark(ht);
savescreen('scherma');
leavegraphic;
clearscreen;
gotoxy(1,3);
for t:=1 to 80 do write(' ');
write(' ', ' ':78, ' '); write(' ', ' ':78, ' ');
write(' U wordt gevraagd de richting van de derde geleiding ');
write(' (= W-geleiding) op te geven. Met de derde geleiding ');
write(' wordt bedoeld de laatste geleiding voordat U aan de ');
write(' taster bent. Dus na het eindpunt van de derde geleiding ');
write(' is het enige wat ons nog van de tastkogel afhoudt, de ');
write(' tastvector. De derde geleiding hoeft niet noodzakelijk ');
write(' de Z-richting te zijn op Uw machine. ');
write(' Voorbeeld 1) Bij een portaalmachine is de richting ');
write(' van de derde geleiding omlaag priemend in het meetvolume. ');
write(' 2) Bij een C-type machine is de richting ');
write(' van de derde geleiding horizontaal priemend in het ');
write(' meetvolume. ');
write(' ', ' ':78, ' '); write(' ', ' ':78, ' ');
write(' Druk op <CR> voor terug naar het programma. ');
write(' ', ' ':78, ' ');
for t:=1 to 80 do write(' ');
writeln;
read;
entergraphic;
loadscreen('scherma');
release(ht);
TOTAAL3;
END;
```

```

OVERLAY PROCEDURE VERBIND1INFO;          (voor uitleg zie procedure STARTINFO)
BEGIN
mark(ht);
savescreen('schem');
leavegraphic;
clearscreen;
for t:=1 to 80 do write('*');
write('*',' ':78,'*');
write('*Tussen het eindpunt van de eerste geleiding en het beginpunt *');
write('*van de tweede geleiding zit altijd een bindingsvector.Deze *');
write('*bindingsvector kunnen we splitsen in drie componenten U,V en W. *');
write('*Daarom wordt U gevraagd de drie componenten in te geven en zo *');
write('*de bindingsvector samen te stellen.Om de werkelijke configuratie*');
write('*van de machine zo goed mogelijk te benaderen wordt U allereerst *');
write('*gevraagd de richtingsvolgorde van de componenten in te geven. *');
write('*U moet wel steeds goed in de gaten houden dat een positief *');
write('*ingegeven bindingsvector in de richting gaat van de *');
write('*betreffende geleidingsrichting. *');
write('* Voorbeeld: Na de vraag " richtingsvolgorde =" geeft U *');
write('* WUV in.Dit houdt in dat de bindingsvector opgebouwd *');
write('* wordt uit componenten in respectievelijk W,U en V *');
write('* richting en wel in die volgorde.Wanneer U de definitie *');
write('* van de W-richting bijv. omlaag gekozen heeft,betekent dit *');
write('* dat een positieve W-component van de bindingsvector *');
write('* ook omlaag zal wijzen. *');
write('*',' ':78,'*');
write('* Druk op <CR> voor terug naar het programma. *');
write('*',' ':78,'*');
for t:=1 to 80 do write('*');
writeln;
read;
entergraphic;
loadscreen('schem');
release(ht);
TOTAALS;
END;

```

```

OVERLAY PROCEDURE VERBIND2INFO;          (voor uitleg zie procedure STARTINFO)
BEGIN
mark(ht);
savescreen('scher#');
leavegraphic;
clearscreen;
for t:=1 to 80 do write('*');
write('*',' ':78,'*');
write('*Tussen het eindpunt van de tweede geleiding en het beginpunt *');
write('*van de derde geleiding zit altijd een verbindingsvector.Deze *');
write('*verbindingsvector kunnen we splitsen in drie componenten U,V en W. *');
write('*Daarom wordt U gevraagd de drie componenten in te geven en zo *');
write('*de verbindingsvector samen te stellen.Om de werkelijke configuratie*');
write('*van de machine zo goed mogelijk te benaderen wordt U allereerst *');
write('*gevraagd de richtingsvolgorde van de componenten in te geven. *');
write('*U moet wel steeds goed in de gaten houden dat een positief *');
write('*ingegeven verbindingsvector in de richting gaat van de *');
write('*betreffende geleidingsrichting. *');
write('* Voorbeeld: Na de vraag " richtingsvolgorde =" geeft U *');
write('* MUV in.Dit houdt in dat de verbindingsvector opgebouwd *');
write('* wordt uit componenten in respectievelijk W,U en V *');
write('* richting en wel in die volgorde.Wanneer U de definitie *');
write('* van de W-richting bijv. omlaag gekozen heeft,betekent dit *');
write('* dat een positieve W-component van de verbindingsvector *');
write('* ook omlaag zal wijzen. *');
write('*',' ':78,'*');
write('* Druk op <CR> voor terug naar het programma. *');
write('*',' ':78,'*');
for t:=1 to 80 do write('*');
writeln;
read;
entergraphic;
loadscreen('scher#');
release(ht);
TOTAAL6;
END;

```

```

OVERLAY PROCEDURE MANULPINFO;          (voor uitleg zie procedure STARTINFO)
BEGIN
  mark(ht);
  savescreen('schem');
  leavegraphic;
  clearscreen;
  gotoxy(1,5);
  for t:=1 to 80 do write('*');
  write('*',' ':78,'*');write('*',' ':78,'*');
  write('* U wordt gevraagd het werkelijke machinenuipunt in           *');
  write('* te geven.Hiermee wordt bedoeld het punt waar op Uw           *');
  write('* machine de coördinaten X,Y en Z allen nul zijn.           *');
  write('* Omdat dit punt natuurlijk altijd op een van de           *');
  write('* hoekpunten van het meetvolume ligt kunt U kiezen uit           *');
  write('* acht mogelijkheden nl. A t/m G en K.           *');
  write('*',' ':78,'*'); write('*',' ':78,'*');
  write('*          Druk op <CR> voor terug naar het programma.           *');
  write('*',' ':78,'*');write('*',' ':78,'*'); for t:=1 to 80 do write('*');
  writeln;
  read;
  entergraphic;
  loadscreen('schem');
  release(ht);
  TOTAAL7;
END;

```

```

OVERLAY PROCEDURE XYZINFO;          (voor uitleg zie procedure STARTINFO)
BEGIN
  mark(ht);
  savescreen('schem');
  leavegraphic;
  clearscreen;
  gotoxy(1,4);
  for t:=1 to 80 do write('*');
  write('*',' ':78,'*');write('*',' ':78,'*');
  write('* Er is reeds een assenkruis gelegd in het goede           *');
  write('* machinenuipunt.Het enige wat U nu nog moet doen           *');
  write('* is aangeven hoe de X,Y en Z richtingen lopen op           *');
  write('* Uw machine.Daartoe benoemt U alleen de horizontale           *');
  write('* geleiding.De andere twee worden automatisch benoemd           *');
  write('* omdat het assenkruis altijd rechtsdraaiend is en ze           *');
  write('* zodoende eenduidig zijn vastgelegd.Omdat het assenkruis           *');
  write('* altijd in het meetvolume priemt,kan de toegekende           *');
  write('* as alleen positief zijn zodat -X,-Y en -Z niet           *');
  write('* ingevoerd kunnen en mogen worden.           *');
  write('*',' ':78,'*');write('*',' ':78,'*');
  write('*          Druk op <CR> voor terug naar het programma.           *');
  write('*',' ':78,'*');
  for t:=1 to 80 do write('*');
  writeln; read;
  entergraphic;
  loadscreen('schem');
  release(ht);
  TOTAAL8;
END;

```



```

PROCEDURE MEETTAFEL;      {teken de eerste meettafel waarop het meetvolume staat}
BEGIN
  drawline(100,1275,750,1275);
  drawline(750,1275,750,1375);
  drawline(750,1375,100,1375);
  drawline(100,1375,100,1275);
  drawline(100,1275,700,675);
  drawline(700,675,1350,675);
  drawline(1350,675,750,1275);
  drawline(1350,675,1350,775);
  drawline(1350,775,750,1375);
END;

```

```

PROCEDURE MEETVOLUME1;   {teken letters op de hoekpunten voor de keuze}
BEGIN                   {van het startpunt en geef pijltjes voor de U-richtingen aan}
  setaspect(1);         {geef lengte-breedte verhouding 1 op voor cirkel}
  drawtextw(60,625,1,'A');      {teken op de vier hoekpunten de cirkels}
  drawcircle(100,625,0.05);     {en bijbehorende keuzeletters}
  drawtextw(790,625,1,'C');
  drawcircle(750,625,0.05);
  drawtextw(60,1275,1,'B');
  drawcircle(100,1275,0.05);
  drawtextw(790,1275,1,'D');
  drawcircle(750,1275,0.05);
  drawline(100,625,200,625);     {teken pijltje bij punt A}
  drawline(200,625,195,610);
  drawline(200,625,185,640);
  drawline(650,625,750,625);     {teken pijltje bij punt C}
  drawline(650,625,665,610);
  drawline(650,625,665,640);
  drawline(200,1275,185,1290);   {teken pijltje bij punt B}
  drawline(200,1275,185,1260);
  drawline(650,1275,665,1260);   {teken pijltje bij punt D}
  drawline(650,1275,665,1290);
  drawtextw(220,590,1,'U');      {teken de vier U's in de hoekpunten}
  drawtextw(220,1240,1,'U');
  drawtextw(630,590,1,'U');
  drawtextw(630,1240,1,'U');
END;

```

```

PROCEDURE MEETVOLUME2;      {teken het meetvolume op de tafel als stippellijnen}
BEGIN
  setlinestyle(2);          {zet de lijn op stippelen}
  drawline(100,1275,100,625); {teken het meetvolume als stippellijnen}
  drawline(100,625,750,625);
  drawline(750,625,750,1275);
  drawline(100,625,700,25);
  drawline(700,25,1350,25);
  drawline(1350,25,750,625);
  drawline(1350,25,1350,675);
  drawline(700,25,700,675);
  drawline(100,1275,750,1275);
  drawline(750,1275,1350,675);
  drawline(1350,675,700,675);
  drawline(100,1275,700,675);
  setlinestyle(0);        {zet de lijn weer terug op vol}
END;

```

```

PROCEDURE MEETVOLUME3;      {tektent letters op alle 8 hoekpunten v/h meet}
                              {volume}
BEGIN
  drawtextw(60,625,1,'A');
  setaspect(1);
  drawcircle(100,625,0.05);
  drawtextw(790,625,1,'C');
  drawcircle(750,625,0.05);
  drawtextw(60,1275,1,'B');
  drawcircle(100,1275,0.05);
  drawtextw(790,1275,1,'D');
  drawcircle(750,1275,0.05);
  drawtextw(640,20,1,'E');
  drawcircle(700,25,0.05);
  drawtextw(700,735,1,'F');
  drawcircle(700,675,0.05);
  drawtextw(1390,25,1,'G');
  drawcircle(1350,25,0.05);
  drawtextw(1390,675,1,'K');
  drawcircle(1350,675,0.05);
END;

```

```

PROCEDURE TEKEN (keus:real;vector:boolean;richting:char);
BEGIN
  if vector=true then      {true=gel.vector,false=verbindingsvector}
  BEGIN
    if richting='H' then   {teken de hor.geleidingsvector}
    BEGIN
      xe:=xb+(d*325);     {bepaal de eindpunten tov de beginpunten}
      ye:=yb;
      drawline(xb,yb-8,xb,yb+8);
      drawline(xb,yb-8,xe-(d*12),ye-8);      {teken vector}
      drawline(xb,yb+8,xe-(d*12),ye+8);
      drawline(xe,ye,xe-(d*20),ye-30);      {pijl aan de vector}
      drawline(xe,ye,xe-(d*20),ye+30);
    END;
  END;

```



```

if richting='V' then                (teken de verticale geleidingsvector)
  BEGIN
    xe:=xb;
    ye:=yb-(v#400);
    drawline(xb-5,yb,xb+5,yb);
    drawline(xb-5,yb,xb-5,ye);
    drawline(xb+5,yb,xe+5,ye);
    drawline(xe,ye,xe-20,ye+(v#20));
    drawline(xe,ye,xe+20,ye+(v#20));
  END;
if richting='S' then                (teken de 'schuine' geleidingsvector)
  BEGIN
    xe:=xb+(s#200);
    ye:=yb-(s#200);
    drawline(xb-10,yb,xb+10,yb);
    if s=1 then BEGIN drawline(xb-10,yb,xe-10,ye);
                      drawline(xb+10,yb,xe,ye+10); END
    else BEGIN drawline(xb-10,yb,xe-3,ye-10);
               drawline(xb+10,yb,xe+10,ye); END;
    drawline(xe,ye,xe-(s#35),ye);
    drawline(xe,ye,xe+(s#8),ye+(s#30));
  END;
END
else
  BEGIN
    if richting = 'H' then          (teken de verbindingsvectoren)
      BEGIN
        xe:= xb+(d#keus#hs);
        ye:=yb;
      END;
    if richting = 'V' then
      BEGIN
        ye:= yb-(v#keus#vs);
        xe:=xb;
      END;
    if richting = 'S' then
      BEGIN
        xe:= xb+(s#keus#ss#0.70710678);
        ye:= yb-(s#keus#ss#0.70710678);
      END;
    drawline(xb,yb,xe,ye);
  END;
  xb:=xe;                          (maak de beginpunten voor de volgende lijn)
  yb:=ye;                          (gelijk aan de huidige eindpunten)
END;

PROCEDURE HOED(PYL:CHAR);          (tekent een hoedje aan de vector)
  BEGIN
    if pyl='H' then                 (teken pijl aan horizontale vector)
      BEGIN
        drawline(xe,ye,xe-(d#((k/2)+1)#15),ye-(d#((k/2)+1)#15));
        drawline(xe,ye,xe-(d#((k/2)+1)#15),ye+(d#((k/2)+1)#15));
      END;
    END;

```

```

if pyl='S' then                                {teken pijl aan schuine vector}
  BEGIN
    drawline(xe,ye,xe+(s*((k/2)+1)*3),ye+(s*((k/2)+1)*15));
    drawline(xe,ye,xe-(s*((k/2)+1)*22),ye);
  END;
if pyl='V' then                                {teken pijl aan verticale vector}
  BEGIN
    drawline(xe,ye,xe-(v*((k/2)+1)*15),ye+(v*((k/2)+1)*15));
    drawline(xe,ye,xe+(v*((k/2)+1)*15),ye+(v*((k/2)+1)*15));
  END;
END;

PROCEDURE SCHRIJF;                             {maakt een regel onder op het scherm leeg}
BEGIN                                          {en gaat alvast met de cursor onder staan}
  gotoxy(8,25);
  write(' ');
  gotoxy(9,25);
END;

PROCEDURE TWEESSEN;                            {tekent alle mogelijkheden voor de tweede as}
BEGIN
  drawline(3,300,1400,300);
  drawline(350,0,350,300);
  drawline(700,0,700,300);
  drawline(1050,0,1050,300);
  hs:=0.65;                                    {geef de scherfactoren beginwaarden omdat ze hier}
  vs:=0.65;                                    {nag niet bekend zijn}
  ss:=0.85;
  for i:=1 to 4 do                             {teken de vier mogelijkheden afhankelijk}
    BEGIN                                       {van de keuze v/h startpunt}
      xb:=-175+i*350;
      yb:=150;
      TEKEN(100,false,'H');
      HOED('H');
      drawtextw(xe+(d*20),ye,1,'U');
      xb:=-175+i*350;
      if i=1 then BEGIN s:=1;TEKEN(80,false,'S');HOED('S');
        drawtextw(xe+30,ye-30,1,'V');END;
      if i=2 then BEGIN s:=-1;TEKEN(80,false,'S');HOED('S');
        drawtextw(xe-30,ye+30,1,'V');END;
      if i=3 then BEGIN v:=1;TEKEN(120,false,'V');HOED('V');
        drawtextw(xe-30,ye-20,1,'V');END;
      if i=4 then BEGIN v:=-1;TEKEN(120,false,'V');HOED('V');
        drawtextw(xe-30,ye+10,1,'V');END;
    END;
  drawtextw(30,250,2,'A');                     {geef bij de vier mogelijkheden een}
  drawtextw(380,250,2,'B');                   {keuzeletter aan}
  drawtextw(730,250,2,'C');
  drawtextw(1080,250,2,'D');
END;

```

```

PROCEDURE DRIEASSEN;           {tekent de twee overgebleven mogelijkheden}
BEGIN                           {om de derde geleiding te kiezen}
  drawline(3,300,700,300);      {teken een nieuw kader}
  drawline(350,0,350,300);
  drawline(700,0,700,300);
  for i:=1 to 2 do              {teken de twee mogelijkheden voor de derde geleiding}
    BEGIN
      xb:=-175+i*350;
      yb:=150;
      TEKEN(100,false,'H');HOED('H');
      if d=1 then drawtextw(xe+20,ye,1,'U') else drawtextw(xe-20,ye,1,'U');
      xb:=-175+i*350;
      if twegel='N' then        {als tweede geleiding schuin loopt}
        BEGIN                  {teken dan derde geleiding verticaal}
          TEKEN(80,false,'S');HOED('S');
          if s=1 then drawtextw(xe+30,ye-30,1,'V')
            else drawtextw(xe-30,ye+30,1,'V');
          xb:=-175+i*350;yb:=150;
          if i=1 then BEGIN v:=1;TEKEN(120,false,'V');HOED('V');
            drawtextw(xe-30,ye-20,1,'W');END;
          if i=2 then BEGIN v:=-1;TEKEN(120,false,'V');HOED('V');
            drawtextw(xe-30,ye+10,1,'W');END;
        END
      else                      {als tweede geleiding verticaal loopt}
        BEGIN                  {teken dan derde geleiding schuin}
          TEKEN(120,false,'V');HOED('V');
          if v=1 then drawtextw(xe-30,ye-20,1,'V')
            else drawtextw(xe-30,ye+10,1,'V');
          xb:=-175+i*350;yb:=150;
          if i=1 then BEGIN s:=1;TEKEN(80,false,'S');HOED('S');
            drawtextw(xe+30,ye-30,1,'W');END;
          if i=2 then BEGIN s:=-1;TEKEN(80,false,'S');HOED('S');
            drawtextw(xe-30,ye+30,1,'W');END;
        END;
      END;
    END;
  drawtextw(30,250,2,'A');      {geef de keuzemogelijkheden een letter}
  drawtextw(380,250,2,'B');
END;

```

```

PROCEDURE UVWKRUIS;           {tekent het gekozen UVW model}
BEGIN
  xb:=1225;yb:=1250;TEKEN((65*((k/2)+1)/hs),false,'H');HOED('H');
  drawtextw(xe+(d*20),ye,1,'U');
  xb:=1225;yb:=1250;TEKEN((68*((k/2)+1)/ss),false,'S');HOED('S');
  if twegel='N' then drawtextw(xe+(s*30),ye-(s*30),1,'V')
    else drawtextw(xe+(s*30),ye-(s*30),1,'W');
  xb:=1225;yb:=1250;TEKEN((78*((k/2)+1)/vs),false,'V');HOED('V');
  if twegel='N' then BEGIN
    if v=1 then drawtextw(xe-30,ye-20,1,'W') else drawtextw(xe-30,ye+10,1,'W');
    END
    else BEGIN
    if v=1 then drawtextw(xe-30,ye-20,1,'V') else drawtextw(xe-30,ye+10,1,'V');
    END;
  END;
END;

```

```

PROCEDURE START;                                {bepaalt het startpunt van de}
BEGIN                                           {kinematische ketting van vectoren}
  if (startpunt='A') or (startpunt='B') or (startpunt='a')
    or (startpunt='b') then xb:=100 else xb:=750;
  if (startpunt='A') or (startpunt='C') or (startpunt='a')
    or (startpunt='c') then yb:=625 else yb:=1275;
  setlinestyle(2);
  if (startpunt='A') or (startpunt='a') then drawline(100,625,100,1275);
  if (startpunt='C') or (startpunt='c') then drawline(750,625,750,1275);
  setlinestyle(0);
END;

PROCEDURE TASTVECTORU;                          {vraagt naar de U-component van de tastvector}
BEGIN
  gotoxy(60,23);                               {maak twee regels schoon op het scherm}
  write(' ');
  gotoxy(60,24);
  write(' ');
  SCHRIJF;
  write('U-component v/d tastvector = ');
  {$I-} read(ucomp);                           {lees de ingevoerde waarde en check}
  if ioresult(<>0) then FOUT(13); {$I+}        {of geen verkeerde of te grote}
  if ucomp>5E4 then FOUT(13);                  {ingegeven wordt. Is dit zo, ga dan}
END;                                           {naar procedure FOUT(13)}

PROCEDURE TASTVECTORV;                          {vraagt naar de V-component van de tastvector}
BEGIN
  SCHRIJF;
  write('V-component v/d tastvector = ');
  {$I-} read(vcomp);
  if ioresult(<>0) then FOUT(14); {$I+}        {check ook hier naar juistheid van}
  if vcomp>5E4 then FOUT(14);                 {de variabele en ga zondig naar FOUT(14)}
END;

PROCEDURE TASTVECTORW;                          {vraagt naar de W-component van de tastvector}
BEGIN
  SCHRIJF;
  write('W-component v/d tastvector = ');
  {$I-} read(wcomp);
  if ioresult(<>0) then FOUT(15); {$I+}        {idem als bij tastvectoru}
  if wcomp>5E4 then FOUT(15);
END;

PROCEDURE TASTTEKEN;                            {tekent de tastvector aan het einde van de}
BEGIN                                           {kinematische ketting}
  htast:=ucomp#hs;                             {de lengte van de vector in horizontale richting}
  if twegel='N' then                            {als de V-geleiding schuin loopt, dan is}
    BEGIN                                       {vtast de lengte in schuine richting en}
      vtast:=vcomp#ss;                         {wtast in verticale richting}
      wtast:=wcomp#vs;
    END
END

```

```

else                                     {als de V-geleiding verticaal loopt, dan}
  BEGIN                                  {wordt de schuine lengte bepaald door de}
    v tast:=wcomp#ss;                   {w-component en de verticale door de}
    w tast:=vcomp#vs;                   {v-component}
  END;
xe:=xb+(d#htast)+(s#0.7071#vtast);      {bepaal het eindpunt van de tast-}
ye:=yb-(v#wtast)-(s#0.7071#vtast);      {vector t.o.v het eindpunt van de}
drawline(xb,yb,xe,ye);                  {derde geleiding en teken deze vector}
drawcircle(xe,ye,0.05);                 {met een cirkel rond zijn eindpunt}
END;

PROCEDURE VERBIND (vb:char;nr:integer;vraag:boolean); {teken de verbindings-}
BEGIN                                          {stukken tussen de geleidingen}
  if (vb='u') or (vb='U') then               {als om een u-verbinding gevraagd word}
    BEGIN
      if vraag=true then                    {stel de vraag hoe groot de verb. vector is}
        BEGIN
          SCHRIJF;
          if nr=1 then BEGIN write('Lengte Uuv-vector in mm = ');
                           ($I-) read(uuv); ($I+);
                           END
          else BEGIN write('Lengte Uvw-vector in mm = ');
                  ($I-) read(uvw); ($I+);
                  END;
        END;
      END;                                  {teken de verbindingsvector}
      if nr=1 then TEKEN(uuv,false,'H') else TEKEN(uvw,false,'H');
    END;
  if (vb='v') or (vb='V') then             {als om een v-verbinding gevraagd word}
    BEGIN
      if vraag=true then
        BEGIN
          SCHRIJF;
          if nr=1 then BEGIN write('Lengte Vuv-vector in mm = ');
                           ($I-) read(vuv); ($I+);
                           END
          else BEGIN write('Lengte Vvw-vector in mm = ');
                  ($I-) read(vvw); ($I+);
                  END;
        END;
      END;
      if twegel='N' then                    {als de v-geleiding schuin loopt teken dan de}
        BEGIN                                {v-verbindingvectoren ook schuin}
          if nr=1 then TEKEN(vuv,false,'S') else TEKEN(vvw,false,'S');
        END
      else
        BEGIN                                {als de v-geleiding verticaal loopt teken dan}
          {de v-verbindingvectoren ook verticaal}
          if nr=1 then TEKEN(vuv,false,'V') else TEKEN(vvw,false,'V');
        END;
      END;
    END;
  END;

```

```

if (vb='w') or (vb='W') then      (als naar de w-verbinding gevraagd word)
  BEGIN
    if vraag=true then
      BEGIN
        SCHRIJF;
        if nr=1 then BEGIN write('Lengte Wuv-vector in mm = ');
                          ($I-) read(wuv); ($I+)
                          END
        else              BEGIN write('Lengte Wvw-vector in mm = ');
                          ($I-) read(wvw); ($I+)
                          END;
      END;
    if twegel='N' then      (als de v-geleiding schuin loopt teken dan)
      BEGIN                (de w-verbindingvector verticaal)
        if nr=1 then TEKEN(wuv,false,'V') else TEKEN(wvw,false,'V');
      END
    else                    (als de v-geleiding verticaal loopt teken dan)
      BEGIN                (de w-verbindingvector schuin)
        if nr=1 then TEKEN(wuv,false,'S') else TEKEN(wvw,false,'S');
      END;
    END;
  END;

```

```
END;
```

```

PROCEDURE NUL;                (teken de vector Aou van het nulpunt naar )
  BEGIN                      (het begin v/d U-geleiding)
    START;
    TEKEN(unul,false,'H');
    if twegel='N' then TEKEN(vnul,false,'S') else TEKEN(vnul,false,'V');
    if twegel='N' then TEKEN(wnul,false,'V') else TEKEN(wnul,false,'S');
  END;

```

```

PROCEDURE KETTING;          (tekent de ingevoerde vectorenketting)
  BEGIN
    TEKEN(ye,true,'H');
    VERBIND(verb1,1,false);
    VERBIND(verb2,1,false);
    VERBIND(verb3,1,false);
    if twegel='N' then TEKEN(wuv,true,'S') else TEKEN(wuv,true,'V');
    VERBIND(verb4,2,false);
    VERBIND(verb5,2,false);
    VERBIND(verb6,2,false);
    if twegel='N' then TEKEN(wvw,true,'V') else TEKEN(wvw,true,'S');
    drawline(xb,yb,xb+(d#htast)+(s#0.7071#vtast),yb-(v#wtast)-
             (s#0.7071#vtast));
    drawcircle(xb+(d#htast)+(s#0.7071#vtast),yb-(v#wtast)-
              (s#0.7071#vtast),0.05);
  END;

```

```

PROCEDURE MEETTAFEL2;          {tekent een grote meettafel onder het meetvolume}
BEGIN                          {om duidelijk te laten zien waar de geleidingen liggen}
  unul:=-uuv-uvw;vnul:=-vuv-vvw;wnul:=-wuv-wvw; {#nul is de offset component}
  if (v=-1) and (twegel='N') then wnul:=wnul-wber; {die nodig is om uit te}
  if (v=-1) and (twegel='Y') then vnul:=vnul-wber; {komen in een hoekpunt}
  if ((startpunt='A') or (startpunt='C') or (startpunt='a') {van het meet-}
    or (startpunt='c')) and (v=-1) then {volume als alle}
    BEGIN                      {geleidingsvectoren nul zijn}
      if (v=-1) and (twegel='N') then wnul:=wnul+(650/vs);
      if (v=-1) and (twegel='Y') then vnul:=vnul+(650/vs);
    END;
  if twegel='N' then
    BEGIN
      if wnul<0 then BEGIN a:=0;b:=1;c:=0;g:=wnul; END;
      if wnul>0 then BEGIN a:=0;b:=1;c:=1;g:=0; END
      else BEGIN a:=0;b:=2;c:=1;g:=0; END;
    END
  else
    BEGIN
      if vnul<0 then BEGIN a:=1;b:=0;c:=0;g:=vnul; END;
      if vnul>0 then BEGIN a:=1;b:=0;c:=1;g:=0; END
      else BEGIN a:=2;b:=0;c:=1;g:=0; END;
    END;
  uverpl:=abs(unul*hs);
  vverpl:=0.7071*ss*vnul;
  wverpl:=0.7071*ss*wnul;
  x1:=(100-uverpl+(a#wverpl)+(b#vverpl)); {dit zij de hoekpunten van de}
  y1:=(1275-(a#wverpl)-(b#vverpl)); {meettafel die onder het meetvolume}
  x2:=750+uverpl+(a#wverpl)+(b#vverpl); {getekend wordt en geen enkele}
  y2:=y1; {functie heeft behalve het aangeven}
  x3:=x2; {van de ligging van de geleidingen}
  y3:=y2+(c#100)-(g#2*vs); {t.o.v. het meetvolume}
  x4:=x1;
  y4:=y3;
  x5:=700-uverpl-(a#wverpl)-(b#vverpl);
  y5:=675+(a#wverpl)+(b#vverpl);
  x6:=1350+uverpl-(a#wverpl)-(b#vverpl);
  y6:=y5;
  x7:=x6;
  y7:=y6+(c#100)-(g#2*vs);
  drawline(x1,y1,x2,y2); {teken de fictieve meettafel}
  drawline(x2,y2,x3,y3);
  drawline(x3,y3,x4,y4);
  drawline(x4,y4,x1,y1);
  drawline(x1,y1,x5,y5);
  drawline(x5,y5,x6,y6);
  drawline(x6,y6,x2,y2);
  drawline(x6,y6,x7,y7);
  drawline(x7,y7,x3,y3);
END;

```

```

PROCEDURE STOP;                (deze procedure maakt het mogelijk)
BEGIN                          (het programma vroegtijdig te beëindigen)
  SCHRIJF;
  write('Weet U het zeker ? (Y/N) ');
  {%-} read(kbd,zeker); {%-}
  zeker:=upcase(zeker);
  if zeker='Y' then BEGIN clearscreen; leavegraphic; halt; END;
  exit;                          (stop het programma en ga terug naar het systeem)
END;

```

```

PROCEDURE XYZASSEN;           (bepaal het startpunt voor het xyz assenkruis)
BEGIN
  if (np='A') or (np='a') then BEGIN xb:=100;yb:=625; END;
  if (np='B') or (np='b') then BEGIN xb:=100;yb:=1275; END;
  if (np='C') or (np='c') then BEGIN xb:=750;yb:=625; END;
  if (np='D') or (np='d') then BEGIN xb:=750;yb:=1275; END;
  if (np='E') or (np='e') then BEGIN xb:=700;yb:=25; END;
  if (np='F') or (np='f') then BEGIN xb:=700;yb:=675; END;
  if (np='G') or (np='g') then BEGIN xb:=1350;yb:=25; END;
  if (np='K') or (np='k') then BEGIN xb:=1350;yb:=675; END;
  if (np='A') or (np='a') or (np='B') or (np='b') or (np='E')
    or (np='e') or (np='F') or (np='f') then d:=1 else d:=-1;
  if (np='A') or (np='a') or (np='B') or (np='b') or (np='C')
    or (np='c') or (np='D') or (np='d') then s:=1 else s:=-1;
  if (np='B') or (np='b') or (np='D') or (np='d') or (np='F')
    or (np='f') or (np='K') or (np='k') then v:=1 else v:=-1;
  draaiing:=(d*v*s);           (draaiing geeft aan hoe het assenkruis in de)
END;                          (ruimte ligt. Dit hebben we later nodig bij)
                               (het bepalen van het teken in de foutentabel)

```

```

PROCEDURE VERKLEIN;          (verkleint het window en daarmee de tekening)
BEGIN
  SCHRIJF;
  write('Verkleinen of vergroten (V/G) of nulpunt vastleggen (N) ?');
  {%-} read(kbd,kleiner); {%-}
  if (kleiner='q') or (kleiner='Q') then STOP;
  if (kleiner='v') or (kleiner='V') or (kleiner='g') or (kleiner='G') then
    BEGIN
      if k>1 then setclippingoff;           (teken ook buiten het window)
      if (kleiner='v') or (kleiner='V') then k:=k+1 else k:=k-1;
      if k<0 then BEGIN SCHRIJF;
        write(' ');
        write(' ');
        SCHRIJF;
        write('U kunt niet groter tekenen !');
        {%-} read(kbd,kleiner); {%-}
        k:=k+2;
      END;
    END;

```



```

if k>6 then BEGIN SCHRIJF;
                write(' ');
                write(' ');
                SCHRIJF;
                write('U kunt niet kleiner tekenen !');
                ($I-) read(kbd,kleiner);($I+)
                k:=k-2;
            END;
clearscreen;          (maak het scherm vrij)
redefinewindow(1,6*k,15*k,79-(6*k),199-(15*k)); (verander het window)
selectworld(1);      (kies toch dezelfde coordinatengrootte)
selectwindow(1);    (voor het window)
MEETVOLUME2;        (teken de gehele tekening in het nieuwe window)
MEETTAFEL2;START;NUL;KETING;UVMKRUIS;MENU;
VERKLEIN; (herhaal deze procedure totdat geen V of 6 ingegeven wordt)
END;
if (kleiner='H') or (kleiner='h') then VERKLEININFO;
SCHRIJF;
write(' ');
MEETTAFEL2;        (teken de meetafel opnieuw omdat eventuele tekst)
END;                (hiervan een stuk kan hebben uitgewist)

```

PROCEDURE TOTAAL1;

```

BEGIN                (HIER BEGINT HET EIGENLIJKE PROGRAMMA)
MENU;                (zet het menu op het scherm)
SCHRIJF;
write(' Kies beginpunt A,B,C of D ');
($I-) read(kbd,startpunt);($I+) (lees in waar het startpunt ligt)
if (startpunt='q') or (startpunt='D') then STOP;
if (startpunt='C') or (startpunt='D') or (startpunt='c') (bepaal de
or (startpunt='d') then d:=-1; (richting=d van de u-as)
if (startpunt='A') or (startpunt='a') or (startpunt='B') (rechts=1 en
or (startpunt='b') then d:=1; (links=-1)
if (startpunt='h') or (startpunt='H') then STARTINFO;
if (startpunt<>'A') and (startpunt<>'a') and (startpunt<>'B') and
(startpunt<>'b') and (startpunt<>'C') and (startpunt<>'c') and
(startpunt<>'D') and (startpunt<>'d') then FOUT(1);
END;

```

PROCEDURE TUSSEN12;

```

BEGIN
setcolorblack;
MEETVOLUME1;        (veeg het meetvolume weer weg)
MEETVOLUME2;
setcolorwhite;
SCHRIJF;
MEETTAFEL;          (teken de meetafel opnieuw)
TWEESSEN;          (teken de keuzes voor de tweede geleiding)
END;

```

PROCEDURE TOTAAL2;

```
BEGIN
  SCHRIJF;
  write('Kies V-as (A,B,C of D) ');
  { $I- } read(kbd,vas); { $I+ }
  if (vas='Q') or (vas='q') then STOP;
  if (vas='h') or (vas='H') then VASINFO;
  if (vas('>')'A') and (vas('>')'a') and (vas('>')'B') and (vas('>')'b') and (vas('>')'C')
    and (vas('>')'c') and (vas('>')'D') and (vas('>')'d') then FOUT(2);
END;
```

PROCEDURE TUSSEN23;

```
BEGIN
  setcolorblack;
  TWEEASSEN; (veeg die keuzes weer weg)
  setcolorwhite;
  if (vas='A') or (vas='a') then BEGIN twegel:='N'; s:=1; END;
  if (vas='B') or (vas='b') then BEGIN twegel:='N'; s:=-1; END;
  if (vas='C') or (vas='c') then BEGIN twegel:='Y'; v:=1; END;
  if (vas='D') or (vas='d') then BEGIN twegel:='Y'; v:=-1; END;
  DRIEASSEN; (teken de keuzes voor de derde geleiding)
  (hier zijn de richting van de v-geleiding bepaald)
  (wanneer de richting naar achteren of omhoog is)
  (noemen we hem positief, resp. s=1 of v=1, wanneer)
  (de richting naar vooren of omlaag is noemen we)
  (hem negatief, resp. s=-1 of v=-1)
END;
```

PROCEDURE TOTAAL3;

```
BEGIN
  SCHRIJF;
  write('Kies W-as (A of B) ');
  { $I- } read(kbd,was); { $I+ }
  if (was='Q') or (was='q') then STOP;
  if (was='h') or (was='H') then WASINFO;
  if (was('>')'A') and (was('>')'a') and (was('>')'B') and (was('>')'b') then FOUT(3);
END;
```

PROCEDURE TUSSEN34;

```
BEGIN
  setcolorblack;
  DRIEASSEN; (veeg die keuzes weer weg)
  setcolorwhite;
  gotoxy(60,23);
  write(' ');
  gotoxy(60,24);
  write(' ');
  if (twegel='N') and ((was='A') or (was='a')) then v:=1;
  if (twegel='N') and ((was='B') or (was='b')) then v:=-1;
  if (twegel='Y') and ((was='A') or (was='a')) then s:=1;
  if (twegel='Y') and ((was='B') or (was='b')) then s:=-1;
  (hier is de waarde van v of s bepaald afhankelijk van de gekozen v-as)
  (en van de richting van de w-as.)
  UVWKRUIS; (teken het gekozen UVW assenkruis)
END;
```

```

PROCEDURE TOTAAL4A;
BEGIN
  SCHRIJF;write('Bereik U-as in mm = ');
  {$I-}read(uber); {lees het U-bereik in}
  if ioreult<>0 then FOUT(10); {$I+} {als geen getal of een getal kleiner}
  if (uber<0) or (uber>5E4) then FOUT(10); {dan 0 of groter dan 50000 geef}
  END; {dan een foutmelding}

```

```

PROCEDURE TOTAAL4B;
BEGIN
  SCHRIJF;write('Bereik V-as in mm = ');
  {$I-}read(vber); {bepaal het bereik in vert. of schuine}
  if ioreult<>0 then FOUT(11); {$I+} {richting v/d tweede geleiding}
  if (vber<0) or (vber>5E4) then FOUT(11);{bij foute ingave geef foutmelding}
  END;

```

```

PROCEDURE TOTAAL4C;
BEGIN
  SCHRIJF;write('Bereik W-as in mm = '); {evenzo voor de derde geleiding}
  {$I-}read(wber);
  if ioreult<>0 then FOUT(12); {$I+}
  if (wber<0) or (wber>5E4) then FOUT(12);
  END;

```

```

PROCEDURE TUSSEN45;
BEGIN
  hs:=650/uber; {bepaal de schermfactor in hor. richting}
  if twegel='N' then ss:=850/vber else vs:=650/vber; {richting afh.v/d keuze}
  if twegel='N' then vs:=650/wber else ss:=850/wber; {v/d 2e geleiding}
  START; {bepaal de startwaarden van x en y}
  TEKEN(ys,true,'H'); {teken de U-geleidingsvector}
  END;

```

```

PROCEDURE TOTAAL5;
BEGIN
  MENU; {roep het menu weer op}
  SCHRIJF;
  write('Richtingsvolgorde = ');
  {$I-} read(kbd,verb1); {$I+}
  if (verb1='h') or (verb1='H') then VERBIND1INFO
  else {$I-} read(kbd,verb2,verb3); {$I+}
  if (verb1='Q') or (verb1='q') then STOP;
  if((verb1<>'U') and (verb1<>'u') and (verb1<>'V') and (verb1<>'v') and
  (verb1<>'W') and (verb1<>'w'))
  or ((verb2<>'U') and (verb2<>'u') and (verb2<>'V') and (verb2<>'v')
  and (verb2<>'W') and (verb2<>'w')) or ((verb3<>'U') and (verb3<>'u')
  and (verb3<>'V') and (verb3<>'v') and (verb3<>'W') and (verb3<>'w'))
  then FOUT(5);
  END;

```

```

PROCEDURE TUSSEN5;
BEGIN
  VERBIND(verb1,1,true);      (teken de eerste 3 verbindingsvectoren met)
  VERBIND(verb2,1,true);      (eerst de vraag naar de grootte)
  VERBIND(verb3,1,true);
  if twegel='N' then TEKEN(wuv,true,'S') else TEKEN(wuv,true,'V');
END;

```

```

PROCEDURE TOTAAL6;
BEGIN
  SCHRIJF;
  write('Richtingsvolgorde = ');
  {$I-} read(kbd,verb4); {$I+}
  if (verb4='Q') or (verb4='q') then STOP;
  if (verb4='h') or (verb4='H') then VERBIND2INFO
  else {$I-} read(kbd,verb5,verb6); {$I+}
  if ((verb4<>'U') and (verb4<>'u') and (verb4<>'V') and (verb4<>'v') and
      (verb4<>'W') and (verb4<>'w')) or
      ((verb5<>'U') and (verb5<>'u') and (verb5<>'V') and (verb5<>'v') and
      (verb5<>'W') and (verb5<>'w')) or ((verb6<>'U') and (verb6<>'u') and
      (verb6<>'V') and (verb6<>'v') and (verb6<>'W') and (verb6<>'w'))
  then FOUT(6);
END;

```

```

PROCEDURE TUSSEN6;
BEGIN
  VERBIND(verb4,2,true);      (teken de laatste 3 verbindingsvectoren met)
  VERBIND(verb5,2,true);      (eerst de vraag naar de grootte)
  VERBIND(verb6,2,true);
  if twegel='N' then TEKEN(wvw,true,'V') else TEKEN(wvw,true,'S');
  TASTVECTORU;                (bepaal de u-component v/d taster)
  TASTVECTORV;                (bepaal de v-component v/d taster)
  TASTVECTORW;                (bepaal de w-component v/d taster)
  TASTTEKEN;                  (teken de gekozen taster)
  SCHRIJF;write('Druk <CR> voor fysische situatie');
  {$I-} read(cr);
  if (cr='Q') or (cr='q') then STOP; {$I+}      (laat de tekening staan tot)
  SCHRIJF;START;              (return ingetoetst wordt)
  setcolorblack;
  MEETAFEL;                   (veeg de tafel en de vectorenketting weer uit)
  KETTING;
  setcolorwhite;
  setclippingon;              (teken niet buiten het window)
  MEETVOLUME2;
  MEETAFEL2;START;NUL;KETTING; (teken het meetvolume en de vectorenketting)
  MENU;
  VERKLEIN;                   (verklein tot de juiste grootte)
  setcolorblack;
  START;NUL;KETTING;         (veeg de vectorenketting uit)
  setcolorwhite;
  MEETVOLUME2;                (teken het meetvolume opnieuw en zet hier)
  MEETVOLUME3;                (de mogelijke nulpunten in)
END;

```

```

PROCEDURE TOTAAL7;
BEGIN
  SCHRIJF;
  write('Het machinenulpunt ligt in: ');
  {$I-} read(kbd,np); {$I+}                (vraag naar het ulpunt)
  if (np='Q') or (np='q') then STOP;
  if (np='h') or (np='H') then MANULPINFO;
  if (np('>'A') and (np('>'a') and (np('>'B') and (np('>'b') and (np('>'E')
    and (np('>'e') and (np('>'F') and (np('>'f') and (np('>'C') and (np('>'c')
    and (np('>'D') and (np('>'d') and (np('>'G') and (np('>'g') and (np('>'k')
    and (np('>'K') then FOUT(7);
END;

```

```

PROCEDURE TUSSEN78;
BEGIN
  setcolorblack;
  SCHRIJF;
  MEETVOLUME3;                (veeg de mogelijke letters weer uit de tekening)
  setcolorwhite;
  MEETVOLUME2;
  dhulp:=d;                    (bewaars de d,s en v waarden van de vectorenketting)
  shulp:=s;                    (in dhulp,shulp en vhulp)
  vhulp:=v;
  XYZASSEN;
  TEKEN((65*((k/2)+1)/hs),false,'H');HOED('H');    (teken het XYZassenkruis)
  XYZASSEN;                    (in het juiste hoekpunt en)
  TEKEN((68*((k/2)+1)/ss),false,'S');HOED('S');    (gebruik daarbij de )
  XYZASSEN;                    (schaalfactor k)
  TEKEN((78*((k/2)+1)/vs),false,'V');HOED('V');
  d:=dhulp;                    (geef d,s en v de oude waarden terug)
  s:=shulp;
  v:=vhulp;
  START;NUL;KETTING;         (teken de vectorenketting)
END;

```

```

PROCEDURE TOTAAL8;
BEGIN
  SCHRIJF;
  write('Benoem de U-as.(X,Y of Z) : ');
  {$I-} read(kbd,uas); {$I+}
  if (uas='Q') or (uas='q') then STOP;
  SCHRIJF;                    (bepaal waar de U-geleiding voor staat)

```

```

if (uas='x') or (uas='X') then      (en leidt hieruit af waar de V en W)
BEGIN                                (geleiding voor staan omdat het een)
  XYZASSEN;                          (rechtsdraaiend assenkruis moet zijn)
  TEKEN((65*((k/2)+1)/hs),false,'H');
  drawtextw(xe+(d*20),ye+30,1,'X');
  XYZASSEN;
  TEKEN((68*((k/2)+1)/ss),false,'S');      (als d*v*s=1 en de U-as)
  if (d*v*s)=1 then drawtextw(xe-15,ye-(s*50),1,'Y') (=X dan is de Y-as)
    else drawtextw(xe-15,ye-(s*50),1,'Z'); (schuin en de Z-as)
  XYZASSEN;                            (verticaal en vice-versa)
  TEKEN((78*((k/2)+1)/vs),false,'V');
  if (d*v*s)=1 then drawtextw(xe-35,ye-(v*30),1,'Z')
    else drawtextw(xe-35,ye-(v*30),1,'Y');
END;
if (uas='y') or (uas='Y') then
BEGIN
  XYZASSEN;
  TEKEN((65*((k/2)+1)/hs),false,'H');
  drawtextw(xe+(d*20),ye+30,1,'Y');
  XYZASSEN;
  TEKEN((68*((k/2)+1)/ss),false,'S');      (als d*v*s=1 en de u-as)
  if (d*v*s)=1 then drawtextw(xe-15,ye-(s*50),1,'Z') (=Y dan is de Z-as)
    else drawtextw(xe-15,ye-(s*50),1,'X'); (schuin en de X-as)
  XYZASSEN;                            (verticaal en vice-versa)
  TEKEN((78*((k/2)+1)/vs),false,'V');
  if (d*v*s)=1 then drawtextw(xe-35,ye-(v*30),1,'X')
    else drawtextw(xe-35,ye-(v*30),1,'Z');
END;
if (uas='z') or (uas='Z') then
BEGIN
  XYZASSEN;
  TEKEN((65*((k/2)+1)/hs),false,'H');
  drawtextw(xe+(d*20),ye+30,1,'Z');
  XYZASSEN;
  TEKEN((68*((k/2)+1)/ss),false,'S');      (als d*v*s=1 en de u-as)
  if (d*v*s)=1 then drawtextw(xe-15,ye-(s*50),1,'X') (=Z dan is de X-as)
    else drawtextw(xe-15,ye-(s*50),1,'Y'); (schuin en de Y-as)
  XYZASSEN;                            (verticaal en vice-versa)
  TEKEN((78*((k/2)+1)/vs),false,'V');
  if (d*v*s)=1 then drawtextw(xe-35,ye-(v*30),1,'Y')
    else drawtextw(xe-35,ye-(v*30),1,'X');
END;
if (uas='h') or (uas='H') then XYZINFO;
if (uas<>'X') and (uas<>'x') and (uas<>'Y') and (uas<>'y') and
  (uas<>'Z') and (uas<>'z') then FOUT(8); (wordt iets anders ingetoetst)
END;                                (dan x, y of z geef dan een foutmelding)

```

```

PROCEDURE TUSSEN89;
BEGIN
  gotoxy(60,23);
  write(' ');
  gotoxy(60,24);
  write(' ');
  SCHRIJF;
  write('Druk Shft PrtSc');           {druk de tekening af}
  delay(1500);                       {even wachten en bovenstaande}
  SCHRIJF;                           {writeopdracht uitvoegen zodat deze niet op papier komt}
  delay(3000);
  SCHRIJF;
  write('Toets <CR> voor foutentabel.');
```

read; {wacht op een return en}

leavegraphic; {verlaat de grafische mode}

END;

```

PROCEDURE FOUT;
BEGIN
  SCHRIJF;
  write('Invoer niet correct.Toets <CR>');
```

read;

if f=1 then TOTAAL1; {keer afhankelijk van het meegevoerde nummer}

if f=2 then TOTAAL2; {terug naar de subroutine waar de fout optrad}

if f=3 then TOTAAL3;

if f=5 then TOTAAL5;

if f=6 then TOTAAL6;

if f=7 then TOTAAL7;

if f=8 then TOTAAL8;

if f=10 then TOTAAL4A;

if f=11 then TOTAAL4B;

if f=12 then TOTAAL4C;

if f=13 then TASTVECTORU;

if f=14 then TASTVECTORV;

if f=15 then TASTVECTORW;

END;

```

BEGIN
  TEKST;                                (laat tekst zien)
  k:=0;                                  (zet k op 0 dus gebruik het hele scherm als window)
  INITIAL;                               (ga in de graphische mode)
  selectwindow(1);                       (kies het scherm als window)
  MEETTAFEL;                             (teken de meettafel)
  MEETVOLUME1;                           (teken het meetvolume met de startpunten)
  MEETVOLUME2;
  TOTAAL1;                               (doorloop alle hoofdprocedures)
  TUSSEN12;
  TOTAAL2;
  TUSSEN23;
  TOTAAL3;
  TUSSEN34;
  TOTAAL4A;
  TOTAAL4B;
  TOTAAL4C;
  TUSSEN45;
  TOTAAL5;
  TUSSEN56;
  TOTAAL6;
  TUSSEN67;
  TOTAAL7;
  TUSSEN78;
  TOTAAL8;
  TUSSEN89;
  config.uber:=uber;                    (schrijf alle relevante kenmerken v/d)
  config.vber:=vber;                    (machine in de variabele config)
  config.wber:=wber;
  config.startpunt:=startpunt;
  config.twegel:=twegel;
  config.np:=np;
  config.uas:=uas;
  config.vas:=vas;
  config.was:=was;
  config.draaiing:=draaiing;
  config.uuv:=uuv;
  config.uvw:=uvw;
  config.vuv:=vuv;
  config.vvw:=vwv;
  config.wuv:=wuv;
  config.wvw:=wvw;
  config.ucomp:=ucomp;
  config.vcomp:=vcomp;
  config.wcomp:=wcomp;
  config.verb1:=verb1;
  config.verb2:=verb2;
  config.verb3:=verb3;
  config.verb4:=verb4;
  config.verb5:=verb5;
  config.verb6:=verb6;
  config.merk:=merk;
  config.tiepe:=tiepe;
  assign(datafile,'B:'+tiepe+'.FI6');   (schrijf de configuratie naar disk)
  rewrite(datafile);                    (B onder de typenaam.FI6)

```



```
write(datafile,config);  
close(datafile);  
assign(variabel,'variabel.chn');    (koppel het volgende blok aan dit blok)  
chain(variabel);  
END.
```

```

PROGRAM VARIABEL;

($I a:typedef.sys)      (include de grafische files ondanks dat we in deze)
($I a:graphix.sys)      (file alleen met tekst werken.Dit is nodig om de)
($I a:kernel.sys)       (variabelen die we over willen nemen van de )
($I a:windows.sys)      (tekenfile op de juiste plaats te krijgen)

Var uber,vber,wber:real;           (bereiken van de machine)
Var startpunt,twegel,np,uas,vas,was:char; (gegevens over het assenkruis)
Var draaiing:real;
Var uuv,uvm,vuv,vvm,uuv,wvw:real; (de componenten van de verbindingsvectoren)
Var ucomp,vcomp,wcomp:real;       (deze bovenstaande variabelen kennen we al)
Var u,v,w:char;
Var volgorde,twegelnr:real;
Var u1,v1,w1,u2,v2,w2,u3,v3,w3:string[4]; (u#=u-as in x,y,z,idem voor v en w)
Var paramu,paramv,paramw:string[4];
Var param1,paramv1,paramw1:string[4];
Var xber,yber,zber:real;           (bereik v/d x,y,z geleiding)
Var vbru,vbrv,vbrw:char;         (zet wel of geen -teken bij de)
Var e:char;                       (omzetting uvw --> xyz)
Var xturn,yturn,zturn,eturn:real; (bepaal de rotatiefactor van xyz)
Var uvtturn,uwtturn,vwtturn,uvrot,uwrot,vwrot:real;
Var uturn,vturn,wturn:real;
Var uvr,umr,vmr:char;
Var afstv,afstw:integer;
Var ruvv,ruvw,ruvw,rumw,rvum1,rvum2,rvum,rvwm,rwu1,rwu2,rwu:string[21];
Var rwu,ruuv1,ruuv2,ruuv,rvvw,rwu:string[21];(deze strings komen in de tabel)
Var ut,vt,wt:string[2];          (deze geven aan welke tastercomponenten nul zijn)
Var afdruk:char;
Var fingave:file;

OVERLAY PROCEDURE TABEL;          (geeft een overzicht van de kenmerkende)
BEGIN                               (afmetingen van de machine)
  clrscr;
  gotoxy(12,1);
  writeln('ASBENOEMING      :');      (geef een tabel voor de benoeming)
  gotoxy(33,1);
  write('1-ste geleiding -->  U = ',paramu);
  gotoxy(33,2);
  write('2-de geleiding -->  V = ',paramv);
  gotoxy(33,3);
  write('3-de geleiding -->  W = ',paramw);
  gotoxy(12,5);
  write('MEETBEREIK      :');      (geef een tabel voor het meetbereik)
  gotoxy(40,5);
  write('X-bereik -->  XB = ',xber:8:1,' mm');
  gotoxy(40,6);
  write('Y-bereik -->  YB = ',yber:8:1,' mm');
  gotoxy(40,7);
  write('Z-bereik -->  ZB = ',zber:8:1,' mm');

```

```

gotoxy(12,9);
writeln('VERBINDINGSVECTOREN:'); {geef een tabel voor de verbindingsvectoren}
gotoxy(45,9);
write('Uuv --> ',upcase(u),u,v,' = ',uuv:8:1,' mm');
gotoxy(45,10);
write('Vuv --> ',upcase(v),u,v,' = ',vuv:8:1,' mm');
gotoxy(45,11);
write('Wuv --> ',upcase(w),u,v,' = ',wuv:8:1,' mm');
gotoxy(45,13);
writeln('Uvw --> ',upcase(u),v,w,' = ',uvw:8:1,' mm');
gotoxy(45,14);
writeln('Vvw --> ',upcase(v),v,w,' = ',vvw:8:1,' mm');
gotoxy(45,15);
writeln('Wvw --> ',upcase(w),v,w,' = ',wvw:8:1,' mm');
gotoxy(12,17);
write('TASTERCOMPONENTEN :'); {geef een tabel met de tastercomponenten}
gotoxy(37,17);
write(upcase(u),'-component --> ',upcase(u),'t = ',ucomp:8:1,' mm');
gotoxy(37,18);
write(upcase(v),'-component --> ',upcase(v),'t = ',vcomp:8:1,' mm');
gotoxy(37,19);
writeln(upcase(w),'-component --> ',upcase(w),'t = ',wcomp:8:1,' mm');
writeln;
write(' ':11);
write('AS-COORDINATEN :');
writeln(' ':12,'X-as --> X" = X-Xt');
writeln(' ':43,'Y-as --> Y" = Y-Yt');
writeln(' ':43,'Z-as --> Z" = Z-Zt');
gotoxy(25,24);
write('Druk <CR> voor verder');
read; {wacht op een return}
clrscr;
write('FOUTSDOORT !', ' ':9,'d',u,' ':11,'i', ' ':10,'d',v,' ':10,'i');
write(' ':9,'d',w,' ':10,'i');
write('-----|-----|-----');
write('-----!');
write('LINEARITEIT!', ' ':22,'i', ' ':22,'i', ' ':21,'i');
write(' ':11,'i', ' ':22,'i', ' ':22,'i', ' ':21,'i');
write(' T',upcase(u),u,'=T',upcase(u),' (' ,u,ut,'i');
write(' ':9,'T',upcase(u),u,' ':10,'i', ' ':22,'i', ' ':21,'i');
write(' T',upcase(v),v,'=T',upcase(v),' (' ,v,vt,'i');
write(' ':22,'i', ' ':10,'T',upcase(v),v,' ':9,'i', ' ':21,'i');
write(' T',upcase(w),w,'=T',upcase(w),' (' ,w,wt,'i', ' ':22,'i', ' ':22,'i');
write(' ':9,'T',upcase(w),w,' ':9,'i');
write('-----|-----|-----');
write('-----!');

```

```

write(' RECHTHEID !', ' ':22, '!', ' ':22, '!', ' ':21, '!');
write(' ':11, '!', ' ':22, '!', ' ':22, '!', ' ':21, '!');
write(' T', uppercase(u), v, '=T', uppercase(u), '(', v, vt, '!');
write(' ':9, 'T', uppercase(u), v, ' ':10, '!', ' ':22, '!', ' ':21, '!');
write(' T', uppercase(u), w, '=T', uppercase(u), '(', w, wt, '!');
write(' ':9, 'T', uppercase(u), w, ' ':10, '!', ' ':22, '!', ' ':21, '!');
write(' T', uppercase(v), u, '=T', uppercase(v), '(', u, ut, '!');
write(' ':22, '!', ' ':10, 'T', uppercase(v), u, ' ':9, '!', ' ':21, '!');
write(' T', uppercase(v), w, '=T', uppercase(v), '(', w, wt, '!');
write(' ':22, '!', ' ':10, 'T', uppercase(v), w, ' ':9, '!', ' ':21, '!');
write(' T', uppercase(w), u, '=T', uppercase(w), '(', u, ut, '!');
write(' ':22, '!', ' ':22, '!', ' ':9, 'T', uppercase(w), u, ' ':9, '!');
write(' T', uppercase(w), v, '=T', uppercase(w), '(', v, vt, '!');
write(' ':22, '!', ' ':22, '!', ' ':9, 'T', uppercase(w), v, ' ':9, '!');
write('-----|-----|-----|-----');
write('-----!');

```

```

write(' HAAKSHEID !', ' ':22, '!', ' ':22, '!', ' ':21, '!');
write(' ':11, '!', ' ':22, '!', ' ':22, '!', ' ':21, '!');
write(' S', uppercase(u), v, ' ':4, '!');
write(' ':1, '-S', uppercase(u), v, ' [', uppercase(v), '!');
write(' ':12, '!', ' ':22, '!', ' ':21, '!');
write(' S', uppercase(u), w, ' ':4, '!');
write(' ':1, '-S', uppercase(u), w, ' [', uppercase(w), '!');
write(' ':12, '!', ' ':22, '!', ' ':21, '!');
write(' S', uppercase(v), w, ' ':4, '!', ' ':22, '!');
write(' ':1, '-S', uppercase(v), w, ' [', uppercase(w), '!');
write(' ':12, '!', ' ':21, '!');
write('-----|-----|-----|-----');
write('-----!');

```

```

write(' Druk <CR> voor verder');
read;
clrscr;
write(' FOUTSOORT !', ' ':9, 'd', u, ' ':11, '!', ' ':10, 'd', v, ' ':10, '!');
write(' ':9, 'd', w, ' ':10, '!');
write('-----|-----|-----|-----');
write('-----!');
write(' KANTEL !', ' ':22, '!', ' ':22, '!', ' ':21, '!');
write(' ':11, '!', ' ':22, '!', ' ':22, '!', ' ':21, '!');

```

(# RUV #)

```

write(' R', uppercase(u), v, '=R', uppercase(u), '(', v, vt, '!');
write(' ':22, '!');
ruvv:=uvr+'-R'+uppercase(u)+v+' [ '+paramw+' '+uppercase(w)+v+w+' ]';
if pos('--', ruvv) <> 0 then BEGIN delete(ruvv,1,2); insert(' ', ruvv,1) END;
ruvw:=uwr+'R'+uppercase(u)+v+' [ '+uppercase(v)+v+w+' '+uppercase(v)+t+' ]';
write(ruvv, ' ':8-afstw, '!', ' ');
write(ruvw, ' ':6, '!');
write(' ':11, '!', ' ':22, '!', ' ':22, '!', ' ':21, '!');

```

```

      (* RUM *)
write(' R',upcase(u),w,'=R',upcase(u),'(,w,wt,')');
write(' ':22,');
ruvw:=uvr+'-R'+upcase(u)+w+'* ['+paramw+']';
if pos('--',ruvw) <>0 then BEGIN delete(ruvw,1,2); insert(' ',ruvw,1); END;
ruwm:=uwr+'-R'+upcase(u)+w+'* ['+upcase(v)+'t'];
write(ruvw,' ':12-afstw,');
write(ruwm,' ':10,');
write(' ':11,',' ':22,',' ':22,',' ':21,');

```

```

      (* RVU *)
write(' R',upcase(v),u,'=R',upcase(v),'(,u,ut,')');
write(uvr,' R',upcase(v),u,'* ['');
write(paramw,'+',upcase(w),u,v,'+',upcase(w),v,w,']');
write(' ':4-afstw,',' ':22,');
rvuw1:=vwr+'-R'+upcase(v)+u+'* ['+upcase(u)+v+'+'];
rvuw2:=upcase(u)+v+w+'+'+upcase(u)+'t'];
rvuw:=rvuw1+rvuw2;
if pos('--',rvuw) <>0 then BEGIN delete(rvuw,1,2); insert(' ',rvuw,1); END;
write(rvuw,' ',');
write(' ':11,',' ':22,',' ':22,',' ':21,');

```

```

      (* RVW *)
write(' R',upcase(v),w,'=R',upcase(v),'(,w,wt,')');
write(uvr,' R',upcase(v),w,'* ['',paramw,']');
write(' ':12-afstw,',' ':22,');
rvww:=vwr+'-R'+upcase(v)+w+'* ['+upcase(u)+'t'];
if pos('--',rvww) <>0 then BEGIN delete(rvww,1,2); insert(' ',rvww,1); END;
write(rvww,' ':10,');
write(' ':11,',' ':22,',' ':22,',' ':21,');

```

```

      (* RWU *)
write(' R',upcase(w),u,'=R',upcase(w),'(,u,ut,')');
rwu1:=uwr+'-R'+upcase(w)+u+'* ['+paramv+'+'];
rwu2:=upcase(v)+u+v+'+'+upcase(v)+v+w+'+'];
rwuu:=rwu1+rwu2;
if pos('--',rwuu) <>0 then BEGIN delete(rwuu,1,2); insert(' ',rwuu,1);END;
write(rwuu);
write(' ':4-afstv,');
write(vwr,' R',upcase(w),u,'* ['',upcase(u),u,v,'+',upcase(u),v,w,'+');
write(upcase(u),'t]',',' ':3,',' ':21,');
write(' ':11,',' ':22,',' ':22,',' ':21,');

```

```

      (* RWV *)
write(' R',upcase(w),v,'=R',upcase(w),'(,v,vt,')');
rwvu:=uwr+'-R'+upcase(w)+v+'* ['+upcase(v)+v+w+'+'+upcase(v)+'t'];
if pos('--',rwvu) <>0 then BEGIN delete(rwvu,1,2); insert(' ',rwvu,1) END;
write(rwvu,' ':7,');
write(vwr,' R',upcase(w),v,'* ['',upcase(u),v,w,'+',upcase(u),'t]');
write(' ':7,',' ':21,');
write(' ':11,',' ':22,',' ':22,',' ':21,');

```

```

write('-----|-----|-----|-----');
write('-----|');
write(' ROTATIE  ', ' ':22, '!', ' ':22, '!', ' ':21, '!');
write(' ':11, '!', ' ':22, '!', ' ':22, '!', ' ':21, '!');

      (* RUU *)
write(' R', upcase(u), u, '=R', upcase(u), '( ', u, ut, '!');
write(' ':22, '!');
ruuv1:=uvr+'-R'+upcase(u)+u+'# ['+paramw+'+';
ruuv2:=upcase(w)+u+v+'+'+upcase(w)+v+w+']';
ruuv:=ruuv1+ruuv2;
if pos('--', ruuv) <> 0 then BEGIN delete(ruuv, 1, 2); insert(' ', ruuv, 1) END;
write(ruuv);
write(' ': (4-afstw), '! ');
write(uvr, 'R', upcase(u), u, '# ['+paramv, '+', upcase(v), u, v, '+', upcase(v));
write(v, w, ']', '!': (4-afstw));

      (* RVV *)
write(' R', upcase(v), v, '=R', upcase(v), '( ', v, vt, '! ');
write(uvr, 'R', upcase(v), v, '# ['+paramw, '+', upcase(w), v, w, ']');
write(' ': (8-afstw), '! ', ' ':22, '!');
rvvw:=vmr+'-R'+upcase(v)+v+'# ['+upcase(u)+v+w+'+'+upcase(u)+'t]';
if pos('--', rvvw) <> 0 then BEGIN delete(rvvw, 1, 2); insert(' ', rvvw, 1) END;
write(rvvw, ' ':6, '!');

      (* RWW *)
write(' R', upcase(w), w, '=R', upcase(w), '( ', w, wt, '!');
rwwu:=uwr+'-R'+upcase(w)+w+'# ['+upcase(v)+'t]';
if pos('--', rwwu) <> 0 then BEGIN delete(rwwu, 1, 2); insert(' ', rwwu, 1) END;
write(rwwu, ' ':11, '! ');
write(vwr, 'R', upcase(w), w, '# ['+upcase(u), 't]', ' ':11, '!', ' ':21, '!');
write('-----|-----|-----|-----');
write('-----|');
write('      Wilt U een afdruk van de foutentabel ? (Y/N) ');
read(afdruk);
afdruk:=upcase(afdruk);
END;

```

```

(doe hetzelfde als in het vorige blok maar dan op de printer)
OVERLAY PROCEDURE TABELPR;
BEGIN
  writeln(LST);writeln(LST);writeln(LST);writeln(LST);(zet een hoop spaties om)
  writeln(LST);writeln(LST);writeln(LST);writeln(LST); (het formaat goed te)
  writeln(LST);writeln(LST);writeln(LST);writeln(LST);
  write(LST,' ':11); (maken op de printer)
  write(LST,'ASBENOEMING :');
  write(LST,' ');
  writeln(LST,'1-ste geleiding --> U = ',paramu1);
  write(LST,' ':32);
  writeln(LST,'2-de geleiding --> V = ',paramv1);
  write(LST,' ':32);
  writeln(LST,'3-de geleiding --> W = ',paramw1);
  writeln(LST);
  write(LST,' ':11);
  write(LST,'MEETBEREIK :');
  write(LST,' ':8);
  writeln(LST,'X-bereik --> XB = ',xber:8:1,' mm');
  write(LST,' ':39);
  writeln(LST,'Y-bereik --> YB = ',yber:8:1,' mm');
  write(LST,' ':39);
  writeln(LST,'Z-bereik --> ZB = ',zber:8:1,' mm');
  writeln(LST);
  write(LST,' ':11);
  write(LST,'VERBINDINGSVECTOREN:');
  write(LST,' ':13);
  writeln(LST,'Uuv --> ',upcase(u),u,v,' = ',uuv:8:1,' mm');
  write(LST,' ':44);
  writeln(LST,'Vuv --> ',upcase(v),u,v,' = ',vuv:8:1,' mm');
  write(LST,' ':44);
  writeln(LST,'Wuv --> ',upcase(w),u,v,' = ',wuv:8:1,' mm');
  writeln(LST);
  write(LST,' ':44);
  writeln(LST,'Uvw --> ',upcase(u),v,w,' = ',uvw:8:1,' mm');
  write(LST,' ':44);
  writeln(LST,'Vvw --> ',upcase(v),v,w,' = ',vvw:8:1,' mm');
  write(LST,' ':44);
  writeln(LST,'Wvw --> ',upcase(w),v,w,' = ',wvw:8:1,' mm');
  writeln(LST);
  write(LST,' ':11);
  write(LST,'TASTERCOMPONENTEN :');
  write(LST,' ':5);
  writeln(LST,upcase(u),'-component --> ',upcase(u),'t = ',ucomp:8:1,' mm');
  write(LST,' ':36);
  writeln(LST,upcase(v),'-component --> ',upcase(v),'t = ',vcomp:8:1,' mm');
  write(LST,' ':36);
  writeln(LST,upcase(w),'-component --> ',upcase(w),'t = ',wcomp:8:1,' mm');
  writeln(LST);
  write(LST,' ':11);
  write(LST,'AS-COORDINATEN :');
  writeln(LST,' ':12,'X-as --> X" = X-Xt');
  writeln(LST,' ':43,'Y-as --> Y" = Y-Yt');
  writeln(LST,' ':43,'Z-as --> Z" = Z-Zt');
  writeln(LST);writeln(LST);

```

```

write(LST,' FOUTSOORT ',':9,'d',u,':11','l',':10,'d',v,':10,')';
writeln(LST,':9,'d',w,':10,');
write(LST,'-----|-----|-----');
writeln(LST,'');
writeln(LST,' LINEARITEIT',':22,',' ':22,',' ':21,');
writeln(LST,':11,',' ':22,',' ':22,',' ':21,');
write(LST,' T',uppercase(u),u,'=T',uppercase(u),','(,u,ut,');
writeln(LST,':9,'T',uppercase(u),u,':10,',' ':22,',' ':21,');
write(LST,' T',uppercase(v),v,'=T',uppercase(v),','(,v,vt,');
writeln(LST,':22,',' ':10,'T',uppercase(v),v,':9,',' ':21,');
write(LST,' T',uppercase(w),w,'=T',uppercase(w),','(,w,wt,');
write(LST,':22,',' ':22,');
writeln(LST,':9,'T',uppercase(w),w,':9,');
write(LST,'-----|-----|-----');
writeln(LST,'');

```

```

writeln(LST,' RECHTHEID ',':22,',' ':22,',' ':21,');
writeln(LST,':11,',' ':22,',' ':22,',' ':21,');
write(LST,' T',uppercase(u),v,'=T',uppercase(u),','(,v,vt,');
writeln(LST,':9,'T',uppercase(u),v,':10,',' ':22,',' ':21,');
write(LST,' T',uppercase(u),w,'=T',uppercase(u),','(,w,wt,');
writeln(LST,':9,'T',uppercase(u),w,':10,',' ':22,',' ':21,');
write(LST,' T',uppercase(v),u,'=T',uppercase(v),','(,u,ut,');
writeln(LST,':22,',' ':10,'T',uppercase(v),u,':9,',' ':21,');
write(LST,' T',uppercase(v),w,'=T',uppercase(v),','(,w,wt,');
writeln(LST,':22,',' ':10,'T',uppercase(v),w,':9,',' ':21,');
write(LST,' T',uppercase(w),u,'=T',uppercase(w),','(,u,ut,');
writeln(LST,':22,',' ':22,',' ':9,'T',uppercase(w),u,':9,');
write(LST,' T',uppercase(w),v,'=T',uppercase(w),','(,v,vt,');
writeln(LST,':22,',' ':22,',' ':9,'T',uppercase(w),v,':9,');
write(LST,'-----|-----|-----');
writeln(LST,'');

```

```

writeln(LST,' HAAKSHEID ',':22,',' ':22,',' ':21,');
writeln(LST,':11,',' ':22,',' ':22,',' ':21,');
write(LST,' S',uppercase(u),v,':4,');
write(LST,':1,'-S',uppercase(u),v,'# [',uppercase(v),']');
writeln(LST,':12,',' ':22,',' ':21,');
write(LST,' S',uppercase(u),w,':4,');
write(LST,':1,'-S',uppercase(u),w,'# [',uppercase(w),']');
writeln(LST,':12,',' ':22,',' ':21,');
write(LST,' S',uppercase(v),w,':4,',' ':22,');
write(LST,':1,'-S',uppercase(v),w,'# [',uppercase(w),']');
writeln(LST,':12,',' ':21,');
write(LST,'-----|-----|-----');
writeln(LST,'');

```



```

writeln(LST,' KANTEL      ',':22,',';':22,',';':21,');
writeln(LST,' ':11,',';':22,',';':22,',';':21,');

      (*      1      *)
write(LST,' R',upcase(u),v,'=R',upcase(u),'(,v,vt,');
write(LST,' ':22,');
ruvv:=uvr+'-R'+upcase(u)+v+'* ['+paramw+'+upcase(w)+v+w'];
if pos('--',ruvv) <>0 then BEGIN delete(ruvv,1,2); insert(' ',ruvv,1) END;
ruvw:=uwr+'R'+upcase(u)+v+'* ['+upcase(v)+v+w+'+upcase(v)+t'];
write(LST,ruvv,' :(8-afstw),',';');
writeln(LST,ruvw,' ':6,');
writeln(LST,' ':11,',';':22,',';':22,',';':21,');

      (*      2      *)
write(LST,' R',upcase(u),w,'=R',upcase(u),'(,w,wt,');
write(LST,' ':22,');
ruvw:=uvr+'-R'+upcase(u)+w+'* ['+paramw+''];
if pos('--',ruvw) <>0 then BEGIN delete(ruvw,1,2); insert(' ',ruvw,1); END;
ruww:=uwr+'R'+upcase(u)+w+'* ['+upcase(v)+t'];
write(LST,ruvw,' :(12-afstw),',';');
writeln(LST,ruww,' ':10,');
writeln(LST,' ':11,',';':22,',';':22,',';':21,');

      (*      3      *)
write(LST,' R',upcase(v),u,'=R',upcase(v),'(,u,ut,');
write(LST,uvr,'R',upcase(v),u,'* ['');
write(LST,paramw,'+',upcase(w),u,v,'+',upcase(w),v,w,')');
write(LST,' :(4-afstw),',';':22,');
rvuw1:=vwr+'-R'+upcase(v)+u+'* ['+upcase(u)+u+v+'+'];
rvuw2:=upcase(u)+v+w+'+upcase(u)+t'];
rvuw:=rvuw1+rvuw2;
if pos('--',rvuw) <>0 then BEGIN delete(rvuw,1,2); insert(' ',rvuw,1); END;
writeln(LST,rvuw,' ',');
writeln(LST,' ':11,',';':22,',';':22,',';':21,');

      (*      4      *)
write(LST,' R',upcase(v),w,'=R',upcase(v),'(,w,wt,');
write(LST,uvr,'R',upcase(v),w,'* ['+paramw,']');
write(LST,' :(12-afstw),',';':22,');
rvww:=vwr+'-R'+upcase(v)+w+'* ['+upcase(u)+t'];
if pos('--',rvww) <>0 then BEGIN delete(rvww,1,2); insert(' ',rvww,1); END;
writeln(LST,rvww,' ':10,');
writeln(LST,' ':11,',';':22,',';':22,',';':21,');

      (*      5      *)
write(LST,' R',upcase(w),u,'=R',upcase(w),'(,u,ut,');
rvuu1:=vwr+'-R'+upcase(w)+u+'* ['+paramv+'+'];
rvuu2:=upcase(v)+u+v+'+upcase(v)+v+w+''];
rvuu:=rvuu1+rvuu2;
if pos('--',rvuu) <>0 then BEGIN delete(rvuu,1,2); insert(' ',rvuu,1);END;
write(LST,rvuu);
write(LST,' :(4-afstv),',';');
write(LST,vwr,'R',upcase(w),u,'* ['+upcase(u),u,v,'+',upcase(u),v,w,')');
writeln(LST,upcase(u),t'],' ':3,',';':21,');
writeln(LST,' ':11,',';':22,',';':22,',';':21,');

```

```

      (* 6 *)
write(LST,' R',upcase(w),v,'=R',upcase(w),'(',v,vt,')');
rwwu:=uwr+'-R'+upcase(w)+v+'# ['+upcase(v)+v+w+'+'+upcase(v)+'t]';
if pos('--',rwwu) <>0 then BEGIN delete(rwwu,1,2); insert(' ',rwwu,1) END;
write(LST,rwwu,' ':7,')');
write(LST,vwr,'R',upcase(w),v,'# ['+upcase(u),v,w,'+',upcase(u),'t]');
writeln(LST,' ':7,')');
writeln(LST,' ':11,')');
writeln(LST,' ':11,')');
writeln(LST,' ':22,')');
writeln(LST,' ':22,')');
writeln(LST,' ':21,')');
writeln(LST,' ':21,')');

write(LST,'-----|-----|-----|-----');
writeln(LST,'-----|');
writeln(LST,' ROTATIE ',':22,')');
writeln(LST,' ':11,')');
writeln(LST,' ':22,')');
writeln(LST,' ':22,')');
writeln(LST,' ':21,')');

      (* 1 *)
write(LST,' R',upcase(u),u,'=R',upcase(u),'(',u,ut,')');
write(LST,' ':22,')');
ruuv1:=uwr+'-R'+upcase(u)+u+'# ['+paramw+'+';
ruuv2:=upcase(w)+u+v+'+'+upcase(w)+v+w+']';
ruuv:=ruuv1+ruuv2;
if pos('--',ruuv) <>0 then BEGIN delete(ruuv,1,2); insert(' ',ruuv,1) END;
write(LST,ruuv);
write(LST,' ':4-afstw,')');
write(LST,uwr,'R',upcase(u),u,'# ['+paramv,'+',upcase(v),u,v,'+',upcase(v)];
writeln(LST,v,w,']',' ':4-afstv);

      (* 2 *)
write(LST,' R',upcase(v),v,'=R',upcase(v),'(',v,vt,')');
write(LST,uwr,'R',upcase(v),v,'# ['+paramw,'+',upcase(w),v,w,']');
write(LST,' ':8-afstw,')');
rvvw:=vwr+'-R'+upcase(v)+v+'# ['+upcase(u)+v+w+'+'+upcase(u)+'t]';
if pos('--',rvvw) <>0 then BEGIN delete(rvvw,1,2); insert(' ',rvvw,1) END;
writeln(LST,rvvw,' ':6,')');

      (* 3 *)
write(LST,' R',upcase(w),w,'=R',upcase(w),'(',w,wt,')');
rwwu:=uwr+'-R'+upcase(w)+w+'# ['+upcase(v)+'t]';
if pos('--',rwwu) <>0 then BEGIN delete(rwwu,1,2); insert(' ',rwwu,1) END;
write(LST,rwwu,' ':11,')');
writeln(LST,vwr,'R',upcase(w),w,'# ['+upcase(u),'t]',' ':11,')');
write(LST,'-----|-----|-----|-----');
writeln(LST,'-----|');
writeln(LST);writeln(LST);writeln(LST);writeln(LST);
writeln(LST);writeln(LST);

```

END;

```

BEGIN                                (HIER BEGINT HET PROGRAMMA)
startpunt:=upcase(startpunt);
np:=upcase(np);
twegel:=upcase(twegel);              (upcase = maak van iedere letter een)
uas:=upcase(uas);                    (hoofdletter)
vas:=upcase(vas);
was:=upcase(was);
if twegel='N' then twegelnr:=1 else twegelnr:=-1; {bepaal of v-geleiding om-}
volgorde:=draaiing*twegelnr;        (hoog gaat of niet kijk naar de draaiing van)
vbru:=' ';                            (het xyz assenkruis.Geef de tekens een waarde)
vbrv:=' ';
vbrw:=' ';

```

```

{in het nu volgende gedeelte wordt bekeken of er op grond van de keuze van}
{het startpunt gecombineerd met de keuze van het nulpunt een omdraaiing van}
{tekens moet plaatsvinden omdat bijv. de u-as in werkelijkheid X-XB is}
{twegel geeft aan hoe de tweede geleiding loopt. Y=verticaal,N=horizontaal}

```

```

if ((startpunt='A') or (startpunt='B')) and ((np='A') or (np='B')
or (np='E') or (np='F')) then BEGIN u1:='X'; u2:='Y'; u3:='Z'; END
else BEGIN
  if ((startpunt='C') or (startpunt='D')) and ((np='C') or (np='D') or
(np='G') or (np='K')) then BEGIN u1:='X'; u2:='Y'; u3:='Z'; END
  else BEGIN u1:='X-XB'; u2:='Y-YB'; u3:='Z-ZB'; vbru:='-'; END;
  END;      (bepaal of de richting v/d u-as en de x/y/z as overeenstemmen)

```

```

(hier kijken we hoe de v-as is gekozen t.o.v. het gekozen nulpunt)

```

```

if twegel='N' then      (als de v-as horizontaal loopt kijk dan hoe de hor.)
  BEGIN                (component in het xyz-assenkruis loopt)
    if ((np='A') or (np='B') or (np='C') or (np='D')) and (vas='A')
      then BEGIN v1:='X'; v2:='Y'; v3:='Z'; END
    else BEGIN
      if ((np='E') or (np='F') or (np='G') or (np='K')) and (vas='B')
        then BEGIN v1:='X'; v2:='Y'; v3:='Z'; END
      else BEGIN v1:='X-XB'; v2:='Y-YB'; v3:='Z-ZB'; vbrv:='-'; END;
      END;
    END;
if twegel='Y' then      ( als de v-as verticaal loopt kijk dan hoe de vert.)
  BEGIN                (component in het xyz-assenkruis loopt tov de gekozen vas)
    if ((np='A') or (np='C') or
(np='E') or (np='G')) and (vas='D') then
      BEGIN v1:='X'; v2:='Y'; v3:='Z'; END
    else
      BEGIN
        if ((np='B') or (np='D') or
(np='F') or (np='K')) and (vas='C') then
          BEGIN v1:='X'; v2:='Y'; v3:='Z'; END
        else BEGIN v1:='X-XB'; v2:='Y-YB'; v3:='Z-ZB'; vbrv:='-'; END;
        END;
      END;
  END;

```

{hier kijken we hoe de w-as is gekozen t.o.v het gekozen nulpunt}

```
if twegel='Y' then {als de v-as verticaal loopt kijk dan hoe de hor.comp.}
BEGIN {van het xyz-kruis loopt tov de gekozen was}
  if ((np='A') or (np='B') or (np='C') or (np='D')) and (was='A')
    then BEGIN w1:='X'; w2:='Y'; w3:='Z'; END
  else BEGIN
    if ((np='E') or (np='F') or (np='G') or (np='K')) and (was='B')
      then BEGIN w1:='X'; w2:='Y'; w3:='Z'; END
    else BEGIN w1:='X-XB'; w2:='Y-YB'; w3:='Z-ZB'; vbrw:='-';END;
  END;
END;
if twegel='N' then {als de v-as hor.loopt kijk dan hoe de vert.comp.}
BEGIN {van het xyz-kruis loopt tov de gekozen was}
  if ((np='A') or (np='C') or
      (np='E') or (np='G')) and (was='B') then
    BEGIN w1:='X'; w2:='Y'; w3:='Z'; END
  else
    BEGIN
      if ((np='B') or (np='D') or
          (np='F') or (np='K')) and (was='A') then
        BEGIN w1:='X'; w2:='Y'; w3:='Z'; END
      else BEGIN w1:='X-XB'; w2:='Y-YB'; w3:='Z-ZB'; vbrw:='-';END;
    END;
END;
if uas='X' then BEGIN {als de u-as benoemd word met X bepaal dan de }
  paramu:=u1; {andere twee afh.van de draaiing en twegel}
  if volgorde=1 then BEGIN paramv:=v2; paramw:=w3; END
  else BEGIN paramv:=v3; paramw:=w2; END;
END;
if uas='Y' then BEGIN
  paramu:=u2;
  if volgorde=1 then BEGIN paramv:=v3; paramw:=w1; END
  else BEGIN paramv:=v1; paramw:=w3; END;
END;
if uas='Z' then BEGIN
  paramu:=u3;
  if volgorde=1 then BEGIN paramv:=v1; paramw:=w2; END
  else BEGIN paramv:=v2; paramw:=w1; END;
END;
{als een bereik in de benoeming zit keer de tekens om}
{van de ingegeven waarde van de verbindingsvectoren}
if vbru='- ' then BEGIN uuv:=-1$uuv; uvw:=-1$uvw; ucomp:=-1$ucomp; END;
if vbrv='- ' then BEGIN vuv:=-1$vuv; vvw:=-1$vwv; vcomp:=-1$vcomp; END;
if vbrw='- ' then BEGIN wuv:=-1$wuv; wvw:=-1$wvw; wcomp:=-1$wcomp; END;
```

```

if paramu='X-XB' then paramul:='XB-X';
if paramu='Y-YB' then paramul:='YB-Y';
if paramu='Z-ZB' then paramul:='ZB-Z';
if paramv='X-XB' then paramvl:='XB-X';
if paramv='Y-YB' then paramvl:='YB-Y';
if paramv='Z-ZB' then paramvl:='ZB-Z';
if paramw='X-XB' then paramwl:='XB-X';
if paramw='Y-YB' then paramwl:='YB-Y';
if paramw='Z-ZB' then paramwl:='ZB-Z';

```

```

if paramu='X' then paramul:='X';
if paramu='Y' then paramul:='Y';
if paramu='Z' then paramul:='Z';
if paramv='X' then paramvl:='X';
if paramv='Y' then paramvl:='Y';
if paramv='Z' then paramvl:='Z';
if paramw='X' then paramwl:='X';
if paramw='Y' then paramwl:='Y';
if paramw='Z' then paramwl:='Z';

```

```

if (paramu='X-XB') or (paramu='X') then xber:=uber; {zet de bereiken in u,v,w}
if (paramu='Y-YB') or (paramu='Y') then yber:=uber; {richting om naar x,y,z-}
if (paramu='Z-ZB') or (paramu='Z') then zber:=uber; {richting}
if (paramv='X-XB') or (paramv='X') then xber:=vber;
if (paramv='Y-YB') or (paramv='Y') then yber:=vber;
if (paramv='Z-ZB') or (paramv='Z') then zber:=vber;
if (paramw='X-XB') or (paramw='X') then xber:=wber;
if (paramw='Y-YB') or (paramw='Y') then yber:=wber;
if (paramw='Z-ZB') or (paramw='Z') then zber:=wber;
if (paramu='X-XB') or (paramu='X') then u:='x'; {vergeet het bereik en kijk}
if (paramu='Y-YB') or (paramu='Y') then u:='y'; {alleen of het x,y of z is}
if (paramu='Z-ZB') or (paramu='Z') then u:='z';
if (paramv='X-XB') or (paramv='X') then v:='x';
if (paramv='Y-YB') or (paramv='Y') then v:='y';
if (paramv='Z-ZB') or (paramv='Z') then v:='z';
if (paramw='X-XB') or (paramw='X') then w:='x';
if (paramw='Y-YB') or (paramw='Y') then w:='y';
if (paramw='Z-ZB') or (paramw='Z') then w:='z';

```

{bepaal of het aantal spaties in de tabel met of zonder -bereik is}

```

if (paramv='X') or (paramv='Y') or (paramv='Z') then afstv:=0
else afstv:=3;
if (paramw='X') or (paramw='Y') or (paramw='Z') then afstw:=0
else afstw:=3;

```

```

if ucomp(>0) then ut:='')' else ut:='')';
if vcomp(>0) then vt:='')' else vt:='')';
if wcomp(>0) then wt:='')' else wt:='')';

```

(bepaal de e-factor afhankelijk van het startpunt en geleidingsrichtingen)
 (voor het UVW-assenkruis)

```

if ((startpunt='A')or(startpunt='B')) and (vas='A')and(was='A') then e:=' ' ;
if ((startpunt='A')or(startpunt='B')) and (vas='B')and(was='B') then e:=' ' ;
if ((startpunt='A')or(startpunt='B')) and (vas='C')and(was='B') then e:=' ' ;
if ((startpunt='A')or(startpunt='B')) and (vas='D')and(was='A') then e:=' ' ;
if ((startpunt='C')or(startpunt='D')) and (vas='A')and(was='B') then e:=' ' ;
if ((startpunt='C')or(startpunt='D')) and (vas='B')and(was='A') then e:=' ' ;
if ((startpunt='C')or(startpunt='D')) and (vas='C')and(was='A') then e:=' ' ;
if ((startpunt='C')or(startpunt='D')) and (vas='D')and(was='B') then e:=' ' ;
if ((startpunt='A')or(startpunt='B')) and (vas='A')and(was='B') then e:='- ' ;
if ((startpunt='A')or(startpunt='B')) and (vas='B')and(was='A') then e:='- ' ;
if ((startpunt='A')or(startpunt='B')) and (vas='C')and(was='A') then e:='- ' ;
if ((startpunt='A')or(startpunt='B')) and (vas='D')and(was='B') then e:='- ' ;
if ((startpunt='C')or(startpunt='D')) and (vas='A')and(was='A') then e:='- ' ;
if ((startpunt='C')or(startpunt='D')) and (vas='B')and(was='B') then e:='- ' ;
if ((startpunt='C')or(startpunt='D')) and (vas='C')and(was='B') then e:='- ' ;
if ((startpunt='C')or(startpunt='D')) and (vas='D')and(was='A') then e:='- ' ;

```

```

if vbru='- ' then xturn:=-1 else xturn:=1; (bepaal of in u,v en w-richting)
if vbrv='- ' then yturn:=-1 else yturn:=1; (een bereik voorkomt en bepaal zo)
if vbrw='- ' then zturn:=-1 else zturn:=1; (of de draaiing bij overschakeling)
if vbru='- ' then uturn:=-1 else uturn:=1; (van UVW naar XYZ een + of - erbij)
if vbrv='- ' then vturn:=-1 else vturn:=1; (krijgt)
if vbrw='- ' then wturn:=-1 else wturn:=1;

```

(bepaal de T-factor)

```

uvtturn:=xturn*yturn#wturn; (extra term die voor de kantelfouten komt die)
                                (draaien om de U of V-as en een bijdrage in de)
                                (V of U-richting leveren)
uwturn:=xturn*zturn#vturn; (extra term die voor de kantelfouten komt die)
                                (draaien om de U of W-as en een bijdrage in de)
                                (W of U-richting leveren)
vwturn:=yturn*zturn#uturn; (extra term die voor de kantelfouten komt die)
                                (draaien om de V of W-as en een bijdrage in de)
                                (W of V-richting leveren)

```

```

if e='- ' then eturn:=-1 else eturn:=1;
uvrot:=uvtturn#eturn; (totaal teken voor rotaties van U en V-as in V)
                                (en U-richting)
uwrot:=uwturn#eturn; (totaal teken voor rotaties van U en W-as in W)
                                (en U-richting)
vwrot:=vwturn#eturn; (totaal teken voor rotaties van V en W-as in W)
                                (en V-richting)

```

```

(al deze variabelen uvrot, uwrot en vwrot zijn gelijk aan de d-factor. )
if uvrot=1 then uvr:=' ' else uvr:='- ' ; (schrijf de 1-en en 0-en om)
if uwrot=1 then uwr:=' ' else uwr:='- ' ; (naar '+'-en en '-'-en)
if vwrot=1 then vwr:=' ' else vwr:='- ' ;

```

```

TABEL; (zet de foutentabel op het scherm)
if afdruk='Y' then TABELPR; (en eventueel op papier)

```

{geef de verbindingsvectoren weer de waarde in het UVW-assenstelsel}

```
if vbru='- ' then BEGIN uuv:=-1#uuv; uvw:=-1#uvw; ucomp:=-1#ucomp; END;  
if vbrv='- ' then BEGIN vuv:=-1#vuv; vvw:=-1#vwv; vcomp:=-1#vcomp; END;  
if vbrw='- ' then BEGIN wuv:=-1#wuv; wvw:=-1#wvw; wcomp:=-1#wcomp; END;
```

```
assign(fingave,'B:fingave.chn');      {roep het blok fouteningave van disk B}  
chain(fingave);                       {koppel het aan dit blok}  
END.
```

{ \$C- }

PROGRAM FOUTENINGAVE;

```
($I A:typedef.sys)      (Include deze grafische files van de Turbo-diskette)
($I A:graphix.sys)
($I A:kernel.sys)
($I A:windows.sys)
($I A:findwrl.d.hgh)    (Include deze files om de grafieken te tekenen)
($I A:axis.hgh)
($I A:polygon.hgh)
```

```
Var uber,vber,wber:real;           (Dit zijn algemene variabelen)
Var startpunt,twegel,np,uas,vas,was:char; (overgenomen van het vorige)
Var draaiing:real;                 (programmablok.)
Var uuv,uvm,vuv,vvm,muv,mvm:real;
Var ucomp,vcomp,wcomp:real;
Var u,v,w:char;
```

```
Var TXX,TYY,TZZ:real;             (Dit zijn variabelen die naar)
Var TYX,TZX,TXY,TZY,TeX,TyZ:real; (het volgende blok gaan.)
Var hhfuv,hhfuv,hhfvm:real;
Var RYX,RZX,RXY,RZY,RXZ,RYZ:real;
Var RXX,RYY,RZZ:real;
Var deltax,deltay,deltaz:real;
```

```
Var randstap,meetas:char;         (Dit zijn variabelen die in)
Var beginwaarde,eindwaarde,stapgrootte:real; (practisch iedere procedure)
Var machinewaarde,meetfout:real; (voorkomen.)
Var mfa:PlotArray;
Var puntenaantal:integer;
Var hulp:real;
Var coord1,coord2:char;
Var coorda,coordb:real;
Var heenterug:char;
Var i:integer;
Var linxx,linyy,linzz:PlotArray;
Var rechtheidyx,rechtheidzx,rechtheidxy,rechtheidzy:PlotArray;
Var rechtheidxz,rechtheidyz:PlotArray;
Var hellingyx,hellingzx,hellingxy,hellingzy,hellingxz,hellingyz:real;
Var foutber:file;
```



```

($ VERKLARING VAN DE ALGEMENE VARIABELEN $)
($ $)
($ uber: bereik in de richting van de U-vector $)
($ vber: bereik in de richting van de V-vector $)
($ wber: bereik in de richting van de W-vector $)
($ startpunt: punt waar de ketting van vectoren gestart is $)
($ twegel: geeft aan of de tweede geleiding omhoog(=y) of omlaag(=n) gaat $)
($ np: keuze van het machinenuipunt (a t/m g of k) $)
($ uas: benoeming van de eerste geleiding (x/y of z) $)
($ vas: keuze van de richting van de tweede geleiding $)
($ was: keuze van de richting van de derde geleiding $)
($ draaiing: geeft aan hoe het xyz assenkruis draait in uvw termen $)
($ uuv,vuv,wuv: componenten van de verbindingsvector tussen u en v $)
($ uvw,vvw,wvw: componenten van de verbindingsvector tussen v en w $)
($ ucomp,vcomp,wcomp: componenten van de tastervector $)
($ u: geeft aan of de eerste geleiding x,y of z is zonder het bereik erbij $)
($ v: geeft aan of de tweede geleiding x,y of z is zonder het bereik erbij $)
($ w: geeft aan of de derde geleiding x,y of z is zonder het bereik erbij $)
($ rota: geeft aan of de factor voor de kantelformules 1 of -1 is $)
($ $)
($ randstap: keuze om met stapgrootte (S) of random (R) te meten $)
($ beginwaarde: waar begint men met meten $)
($ eindwaarde: waar stoppen de metingen $)
($ stapgrootte: hoeveel mm zit er tussen twee metingen $)
($ machinewaarde: waarde die de meetlineaal van de machine uitgeeft $)
($ meetfout: echte waarde minus de machinewaarde $)
($ mfa: in dit array staan alle ingegeven fouten en de machinewaarden $)
($ puntenaantal: het aantal meetpunten $)
($ hulp: wordt alleen gebruikt om tijdelijk de worldcoördinaat op te slaan $)
($ coorda,coordb: deze slaan de coördinaten op van de andere twee assen $)
($ coord1,coord2: deze geven aan welke twee assen niet gemeten worden $)
($ heenterug: bepaalt of men heen en terug (y) of alleen heen (N) wil meten $)
($ i :lopende variabele voor for-to loops te maken $)
($ linxx,linyy,linzz: hierin komen de meetgegevens voor elke as afzonderlijk $)
($ rechtheidyx: meetgegevens over de rechtheidsfout van de x-geleiding $)
($ in y-richting $)
($ idem voor rechtheidzx,rechtheidxy,rechtheidzy,rechtheidxz en rechtheidyz $)
($ hhfuv: de berekende haaksheidsfout tussen de eerste en de tweede geleid. $)
($ hhfuw: de berekende haaksheidsfout tussen de eerste en de derde geleid. $)
($ hhfvw: de berekende haaksheidsfout tussen de tweede en de derde geleid. $)
($ TXX,TYY,TZZ: maximale lineariteitsfouten $)
($ TYX,TZX,TXY,TZY,TXZ,TYZ: maximale rechtheidsfouten $)
($ RYX,RZX,RXY,RZY,RXZ,RYZ: maximale kantelfouten $)
($ RXX,RYY,RZZ: maximale rotatiefouten $)
($ deltax,deltay,deltaz: schatting van de fout in x,y en z richting $)
($ deltaxkwadr,deltaykwadr,deltazkwadr: fout in het kwadraat $)
($ $)
($ EINDE $)

```

OVERLAY PROCEDURE LINEARITEITSFOUT;

```

Var begintemp,eindtemp,tsprong:real;
Var tempcorr:char;
Var specsa:PlotArray;
Var minmeetfout,hellmeetfout:real;
Var specxa,specxb,specya,specyb,specza,speczb:real;
Var lasdrccorr:char;
Var mfaoud:PlotArray;
Var laserdrift:real;
Var eigenlinuutz,uitzetting:real;
Var linuutz:integer;

```

```

(*          VARIABELEN IN PROCEDURE LINEARITEITSFOUT          *)
(*          *)
(* begintemp: temperatuur aan het begin van de metingen      *)
(* eindtemp: temperatuur aan het eind van de metingen        *)
(* tsprong: berekende temperatuurverhoging tussen twee metingen *)
(* tempcorr: wel (y) of geen (n) temperatuurcorrectie toepassen *)
(* specsa: array waarin de punten van de specificatielijnen zitten *)
(* minmeetfout: specificatie van de minimale meetfout bij lineariteit *)
(* hellmeetfout: specificatie van de toename van de meetfout per meter *)
(* specxa,specxb: de door de fabrikant opgegeven specificatie voor de x-as *)
(* specya,specyb: de door de fabrikant opgegeven specificatie voor de y-as *)
(* specza,speczb: de door de fabrikant opgegeven specificatie voor de z-as *)
(* lasdrccorr: wel(y) of geen (n) laserdriftcorrectie toepassen *)
(* mfaoud: bewaart de waarde van het eerste meetpunt tijdens correctie *)
(* laserdrift: correctieterm voor de laserdrift *)
(* eigenlinuutz: eventuele zelf ingevoerde lin.uitzettingscoefficient *)
(* uitzetting: lin.uitzettingscoefficient waarmee gecompenseerd word *)
(* linuutz: keuze uit 2 vaste of een zelf ingegeven lin.uitzettingscoeff. *)
(*          *)
(*          EINDE          *)

```

BEGIN

```

clearscreen;
gotoxy(22,2);
writeln('LINEARITEITSFOUTEN');
gotoxy(12,12);
write('Wilt U Random of met vaste Stapgrootte meten ? (R/S) ');
readln(randstap);
randstap:=upcase(randstap);
if randstap='S' then
  BEGIN
    clearscreen;
    gotoxy(12,6);
    write('Welke meetas wilt U controleren ? (X/Y/Z) ');
    readln(meetas);
    meetas:=upcase(meetas);
    if meetas='X' then BEGIN coord1:='Y'; coord2:='Z'; END;
    if meetas='Y' then BEGIN coord1:='X'; coord2:='Z'; END;
    if meetas='Z' then BEGIN coord1:='X'; coord2:='Y'; END;
  END;

```

```

gotoxy(12,8);
write('Geef de ',coord1,'-coördinaat in : ');
read(coorda);
gotoxy(12,10);
write('Geef de ',coord2,'-coördinaat in : ');
read(coordb);
clearscreen;
gotoxy(8,4);
write('Lineariteitsfout: T',meetas,meetas,' ':8);
write(coord1,'= ',coorda:6:2,' ':5,coord2,'= ',coordb:6:2);
gotoxy(12,8);
write('Wat is de startwaarde van de metingen ? ');
readln(beginwaarde);
gotoxy(12,10);
write('Wat is de eindwaarde van de metingen ? ');
readln(eindwaarde);
gotoxy(12,12);
write('Wat is de gewenste stapgrootte ? ');
readln(stapgrootte);
gotoxy(12,14);
write('Wilt U heen en terug meten ? (Y/N) ');
read(heenterug);
heenterug:=upcase(heenterug);
gotoxy(12,16);
write('Wat is de begintemperatuur ? (graden celcius) ');
read(begintemp);
punteanaantal:=trunc(((eindwaarde-beginwaarde)/stapgrootte)+1);
for i:=1 to punteanaantal do
  BEGIN
    clearscreen;
    gotoxy(12,5);
    write('LET OP : de machinewaarde is in mm, de meetfout is in um. ');
    mfa[i,1]:=beginwaarde+((i-1)*stapgrootte);
    gotoxy(12,15);
    writeln('De machinewaarde = ',(beginwaarde+((i-1)*stapgrootte))
      :6:2);
    gotoxy(12,17);
    write('De meetfout = ');
    readln(meetfout);
    mfa[i,2]:=meetfout;(zet de meetfout in de tweede rij van het array)
  END;
if heenterug='Y' then
  BEGIN
    for i:=punteanaantal+1 to (2*punteanaantal)-1 do
      BEGIN
        clearscreen;
        gotoxy(12,5);
        write('LET OP : de machinewaarde is in mm, de meetfout is in
          um. ');
        mfa[i,1]:=mfa[punteanaantal,1]-((i-punteanaantal)*stapgrootte);
        gotoxy(12,15);
        write('De machinewaarde = ');
        writeln(mfa[punteanaantal,1]-((i-punteanaantal)*stapgrootte)
          :6:2);
      END;
    end;
  END;

```

```

        gotoxy(12,17);
        write('De meetfout =      ');
        readln(meetfout);
        mfa[i,2]:=meetfout;
    END;
END;
END;

if randstap='R' then
BEGIN
    clearscreen;
    gotoxy(12,10);
    write('Welke meetas wilt U controleren ? (X/Y/Z) ');
    readln(meetas);
    meetas:=upcase(meetas);
    if meetas='X' then BEGIN coord1:='Y'; coord2:='Z'; END;
    if meetas='Y' then BEGIN coord1:='X'; coord2:='Z'; END;
    if meetas='Z' then BEGIN coord1:='X'; coord2:='Y'; END;
    gotoxy(12,12);
    write('Geef de ',coord1,'-coördinaat in : ');
    read(coorda);
    gotoxy(12,14);
    write('Geef de ',coord2,'-coördinaat in : ');
    read(coordb);
    clearscreen;
    gotoxy(8,4);
    write('Lineariteitsfout: T',meetas,meetas,' ':8);
    write(coord1,'= ',coorda:6:2,' ':5,coord2,'= ',coordb:6:2);
    gotoxy(12,8);
    write('Hoeveel meetpunten wilt U invoeren ? ');
    read(puntenaantal);
    gotoxy(12,10);
    write('Wat is de begintemperatuur ? (graden celcius) ');
    read(begintemp);
    for i:=1 to puntenaantal do
        BEGIN
            clearscreen;
            gotoxy(12,5);
            write('LET OP : -machinewaarden in oplopende grootte invoeren. ');
            gotoxy(12,7);
            write('      : -de machinewaarde is in mm, de meetfout is in um. ');
            gotoxy(12,13);
            write('Puntnummer : ',i);
            gotoxy(12,15);
            write('De machinewaarde = ');
            read(machinewaarde);
            mfa[i,1]:=machinewaarde;
            gotoxy(12,17);
            write('De meetfout =      ');
            readln(meetfout);
            mfa[i,2]:=meetfout;
            eindwaarde:=mfa[puntenaantal,1];
            beginwaarde:=mfa[1,1];
        END;
    END;
END;

```

```

clearscreen;
gotoxy(9,10);
write('Wat is de eindtemperatuur ? ');
read(eindtemp);

(* corrigeer voor laserdrift *)

gotoxy(9,12);
write('Wilt U corrigeren voor laserdrift ? (Y/N) ');
read(lasdrcorr);
lasdrcorr:=upcase(lasdrcorr);
if lasdrcorr='Y' then
  BEGIN
    if (heenterug<>'Y') or (randstap='R') then
      BEGIN
        clearscreen;
        gotoxy(9,12);
        writeln('U kunt geen laserdriftcorrectie toepassen omdat U of');
        gotoxy(9,14);
        writeln('random of niet heen en terug gemeten hebt. ');
        delay(4000);
        lasdrcorr:='N';
        clearscreen;
      END
    else
      BEGIN
        laserdrift:=mfaoud[1,2]-mfa[1,2];
        for i:=1 to puntenaantal do mfa[i,2]:=mfa[i,2]+laserdrift;
      END;
    END;
gotoxy(9,14);
write('Wilt U corrigeren voor temperatuurdrift v/d machine ? (Y/N) ');
read(tempcorr);
tempcorr:=upcase(tempcorr);

(* corrigeer voor temperatuurdrift. *)

if tempcorr='Y' then
  BEGIN
    clearscreen;
    gotoxy(9,10);
    write('Maak een keuze uit de volgende uitzettingscoefficienten :');
    gotoxy(16,12);
    write('1 Staal :          12E-6          M/M/graad ');
    gotoxy(16,14);
    write('2 Glas :           8E-6           M/M/graad ');
    gotoxy(16,16);
    write('3 ??? :              M/M/graad ');
    gotoxy(67,10);
    read(linuitz);
  END;

```

```

case linuitz of
  1:uitzetting:=12E-3;
  2:uitzetting:=8E-3;
  3:BEGIN
    gotoxy(12,20);
    write('Toets de juiste uitzettingscoefficient in. ');
    gotoxy(38,16);
    read(eigenlinuitz);
    uitzetting:=1000#eigenlinuitz;
  END;
END;
if heenterug='N' then
  BEGIN
    tsprong:=(eindtemp-begintemp)/(punteanaantal-1);
    for i:=2 to punteanaantal do
      mfa[i,2]:=mfa[i,2]-(i-1)#tsprong#uitzetting#mfa[i,1];
    END;
  if heenterug='Y' then
    BEGIN
      tsprong:=(eindtemp-begintemp)/((2#punteanaantal)-2);
      for i:=2 to ((2#punteanaantal)-1) do
        mfa[i,2]:=mfa[i,2]-(i-1)#tsprong#uitzetting#mfa[i,1];
      END;
    END;
  END;
(* bepaal het gemiddelde bij heen en terug meten *)
if heenterug='Y' then
  BEGIN
    mfaoud[1,2]:=mfa[1,2];
    for i:=1 to punteanaantal-1 do
      BEGIN mfa[i,2]:=(mfa[i,2]+mfa[(2#punteanaantal)-i,2])/2; END;
    END;
  clearscreen;
  gotoxy(9,10);
  write('Geef de specificaties van de machine in op de volgende manier :');
  gotoxy(9,12);
  write('Meetfout = A + B * L [um]          A = [um] B=[um/m] L=[m] ');
  gotoxy(9,14);
  write('A = ');
  read(minmeetfout);
  gotoxy(9,16);
  write('B = ');
  read(hellmeetfout);
  for i:=1 to 11 do
    BEGIN
      specsai,1]:=((eindwaarde-beginwaarde)/10)#(i-1)+beginwaarde;
      specsai,2]:=minmeetfout+((i-1)#((eindwaarde-beginwaarde)/10000)#hell-
        meetfout);
    END;
  for i:=12 to 22 do
    BEGIN
      specsai,1]:=specsai(i-11),1];
      specsai,2]:=-1#(minmeetfout+((i-12)#((eindwaarde-beginwaarde)/10000)#
        hellmeetfout));
    END;
  END;

```

```

clearscreen;
gotoxy(12,2);
writeln('MEETAS = ',meetas);
writeln;
for i:=1 to puntenaantal do
  BEGIN
    write(' machinewaarde = ',mfa[i,1]:8:2);
    writeln(' meetfout = ',mfaf[i,2]:8:2);
  END;
writeln;
write('Specificatie: Max. fout = ',minmeetfout:4:1,' + ');
writeln('hellmeetfout:4:1,' * L [um]');
gotoxy(12,25);
write('Druk <CR> voor grafiek');
read;

clearscreen;
findworld(3,specs,22,1,2); (zoek een world zodat de specs hierin vallen)
with world[3] do
  BEGIN
    hulp:=Y1; (keer de gevonden world om zodat het nulpunt)
    Y1:=Y2; (linksonder komt te liggen)
    Y2:=hulp;
  END;
selectworld(3); (kies de berekende world)
selectwindow(3); (kies hetzelfde window)
drawborder;
setlinestyle(0); (zet de lijn op vol)
drawaxis(8,5,4,10,3,13,0,0,true); (teken een assenkruis)
drawpolygon(mfa,1,-punteaantal,7,2,0); (teken de meetpunten)
drawaxis(0,0,4,10,3,13,0,0,false); (teken hetzelfde assenkruis)
drawpolygon(specs,1,-11,0,1,0); (teken de bovenste specslijn)
drawaxis(0,0,4,10,3,13,0,0,false); (teken hetzelfde assenkruis)
drawpolygon(specs,12,-22,0,1,0); (teken de onderste specslijn)
gotoxy(73,24);
write(meetas,' [um]');
gotoxy(2,4);
write('d',meetas,' [um]');
gotoxy(13,2);
writeln('Foutbron: Lineariteit (T',meetas,meetas,')');
gotoxy(70,4);
writeln('Meetas: ',meetas);
gotoxy(45,2);
write('Spec: d',meetas,' <',minmeetfout:4:1,' +',hellmeetfout:4:1,' * L');
gotoxy(13,4);
write('Temp.correctie: ',tempcorr);
gotoxy(45,4);
write('Laserdriftcorrectie: ',lasdrccorr);
gotoxy(12,24);
write('Druk Shft PrtSc');
delay(2000);
gotoxy(12,24);
write(' ');

```

```

delay(2500);
writeln(LST);writeln(LST);           {schrijf de meetwaarden onder de}
write(LST,'Foutbron : T',meetas,meetas);      {grafiek op de printer}
write(LST,'  Temperatuurcorrectie: ',tempcorr);
writeln(LST,'  Laserdriftcorrectie: ',lasdrccorr);
writeln(LST);
write(LST,coord1,'-coördinaat : ',coorda:8:2);
writeln(LST,'  ':5,coord2,'-coördinaat : ',coordb:8:2);
write(LST,'-----');
writeln(LST,'-----');
writeln(LST);
writeln(LST,'  Machinewaarde           T',meetas,meetas);

```

```

for i:=1 to puntenaantal do
  BEGIN
  writeln(LST,mfa[i,1]:12:2,' ':17,mfa[i,2]:12:2);
  if meetas='X' then           {zet de meetserie in een nieuw array}
    BEGIN                     {en bepaal de maximaal gemeten lineariteitsfout}
      linxx[i,1]:=mfa[i,1];
      linxx[i,2]:=mfa[i,2];
      specxa:=minmeetfout;
      specxb:=hellmeetfout;
      if abs(linxx[i,2]) > abs(TXX) then TXX:=linxx[i,2];
    END;
  if meetas='Y' then
    BEGIN
      linyy[i,1]:=mfa[i,1];
      linyy[i,2]:=mfa[i,2];
      specya:=minmeetfout;
      specyb:=hellmeetfout;
      if abs(linyy[i,2]) > abs(TYY) then TYY:=linyy[i,2];
    END;
  if meetas='Z' then
    BEGIN
      linzz[i,1]:=mfa[i,1];
      linzz[i,2]:=mfa[i,2];
      specza:=minmeetfout;
      speczb:=hellmeetfout;
      if abs(linzz[i,2]) > abs(TZZ) then TZZ:=linzz[i,2];
    END;
  END;
  writeln(LST);
  writeln(LST);
  gotoxy(12,24);
  write('Druk <CR> voor terug naar hoofdmenu.');
```

```

  read;
END;

```



```

if meetas='X' then BEGIN coord1:='Y'; coord2:='Z'; END;
if meetas='Y' then BEGIN coord1:='X'; coord2:='Z'; END;
if meetas='Z' then BEGIN coord1:='X'; coord2:='Y'; END;
gotoxy(12,14);
write('Geef de ',coord1,'-coördinaat in : ');
read(coorda);
gotoxy(12,16);
write('Geef de ',coord2,'-coördinaat in : ');
read(coordb);
clearscreen;
gotoxy(8,4);
write('Rechtheidsfout: T',meetrichting,meetas,' ':8);
write(coord1,'= ',coorda:6:2,' ':5,coord2,'= ',coordb:6:2);
gotoxy(12,10);
write('Wat is de startwaarde van de metingen ? ');
readln(beginwaarde);
gotoxy(12,12);
write('Wat is de eindwaarde van de metingen ? ');
readln(eindwaarde);
gotoxy(12,14);
write('Wat is de gewenste stapgrootte ? ');
readln(stapgrootte);
gotoxy(12,16);
write('Wilt U heen en terug meten ? (Y/N) ');
read(heenterug);
heenterug:=upcase(heenterug);
puntenaantal:=trunc(((eindwaarde-beginwaarde)/stapgrootte)+1);
for i:=1 to puntenaantal do
  BEGIN
    clearscreen;
    gotoxy(10,5);
    write('LET OP : de machinewaarde is in mm, de meetfout is in um');
    mfa[i,1]:=beginwaarde+((i-1)*stapgrootte);
    gotoxy(12,15);
    write('De machinewaarde = ',(beginwaarde+((i-1)*stapgrootte)):6:2);
    gotoxy(12,17);
    write('De meetfout = ');
    readln(meetfout);
    mfa[i,2]:=meetfout;
  END;
if heenterug='Y' then
  BEGIN
    for i:=puntenaantal+1 to (2*puntenaantal)-1 do
      BEGIN
        clearscreen;
        gotoxy(12,5);
        write('LET OP : de machinewaarde is in mm, de meetfout is in um. ');
        mfa[i,1]:=mfa[puntenaantal,1]-((i-puntenaantal)*stapgrootte);
        gotoxy(12,15);
        write('De machinewaarde = ');
        writeln(mfa[puntenaantal,1]-((i-puntenaantal)*stapgrootte)
          :6:2);
        gotoxy(12,17);

```

```

        write('De meetfout =      ');
        readln(meetfout);
        mfa[i,2]:=meetfout;
    END;
END;
END;

if randstap='R' then
BEGIN
    clearscreen;
    gotoxy(12,10);
    write('Welke meetas wilt U controleren ? (X/Y/Z) ');
    readln(meetas);
    meetas:=upcase(meetas);
    gotoxy(12,12);
    write('In welke richting gaat de fout ? (X/Y/Z) ');
    read(meetrichting);
    meetrichting:=upcase(meetrichting);
    if meetas='X' then BEGIN coord1='Y'; coord2='Z'; END;
    if meetas='Y' then BEGIN coord1='X'; coord2='Z'; END;
    if meetas='Z' then BEGIN coord1='X'; coord2='Y'; END;
    gotoxy(12,14);
    write('Geef de ',coord1,'-coördinaat in : ');
    read(coorda);
    gotoxy(12,16);
    write('Geef de ',coord2,'-coördinaat in : ');
    read(coordb);
    clearscreen;
    gotoxy(8,4);
    write('Rechtheidsfout: T',meetrichting,meetas,' ':8);
    write(coord1,'= ',coorda:6:2,' ':5,coord2,'= ',coordb:6:2);
    gotoxy(12,10);
    write('Hoeveel meetpunten wilt U invoeren ? ');
    read(puntenaantal);
    for i:=1 to puntenaantal do
        BEGIN
            clearscreen;
            gotoxy(12,5);
            write('LET OP :- machinewaarden in oplopende grootte invoeren.');
```

```

if heenterug='Y' then
  BEGIN
    for i:=1 to puntenaantal-1 do
      BEGIN mfa[i,2]:=(mfa[i,2]+mfa[(2*puntenaantal)-i,2])/2; END;
    END;
  clearscreen;
  gotoxy(9,12);
  write('Geef de specificaties van de machine in op de volgende manier :');
  gotoxy(9,14);
  write('Rechtheidsfout < A           A = [um] ');
  gotoxy(9,16);
  write('A = ');
  read(minmeetfout);
  for i:=1 to 11 do
    BEGIN
      specsa[i,1]:=((eindwaarde-beginwaarde)/10)*(i-1)+beginwaarde;
      specsa[i,2]:=minmeetfout;
    END;
  for i:=12 to 22 do
    BEGIN
      specsa[i,1]:=specsa[(i-11),1];
      specsa[i,2]:=-1*(minmeetfout);
    END;

    clearscreen;
    gotoxy(12,2);
    writeln('MEETAS = ',meetas);
    gotoxy(12,4);
    writeln('MEETRICHTING = ',meetrichting);
    writeln;
    for i:=1 to puntenaantal do
      BEGIN
        write(' machinewaarde = ',mfa[i,1]:8:2);
        writeln(' meetfout = ',mfa[i,2]:8:2);
      END;
    writeln;
    write('Specificatie: Max. rechtheidsfout = ',minmeetfout:4:1,' [um]');
    gotoxy(12,25);
    write('Druk <CR> voor grafiek');
    read;
  clearscreen;
  findworld(3,specsa,22,1,2);
  with world[3] do
    BEGIN
      hulp:=Y1;
      Y1:=Y2;
      Y2:=hulp;
    END;
  selectworld(3);
  selectwindow(3);
  drawborder;
  setlinestyle(0);
  drawaxis(8,5,4,10,3,13,0,0,true);
  drawpolygon(mfa,1,-puntenaantal,7,2,0);
  drawaxis(0,0,4,10,3,13,0,0,false);

```

```

drawpolygon(specsa,1,-11,0,1,0);
drawaxis(0,0,4,10,3,13,0,0,false);
drawpolygon(specsa,12,-22,0,1,0);
gotoxy(73,24);
write(meetas,' [mm]');
gotoxy(2,4);
write('d',meetrichting,' [um]');
gotoxy(20,2);
writeln('Foutbron: Rechtheid');
gotoxy(20,4);
writeln('Meetas: ',meetas);
gotoxy(45,2);
write('Spec : d',meetrichting,' <',minmeetfout:4:1);
gotoxy(7,25);
write('Druk <CR> voor gecorrigeerde versie.');
```

```

(*****)
```

```

(* bepaal de best paasende rechte mbv lineaire regressie*)
```

```

xtot:=0;
ytot:=0;
teller:=0;
noemer:=0;
for i:=1 to puntenaantal do
  BEGIN
    xtot:=xtot+mfa[i,1];
    ytot:=ytot+mfa[i,2];
  END;
xgem:=xtot/puntenaantal;
ygem:=ytot/puntenaantal;
for i:=1 to puntenaantal do
  BEGIN
    teller:=((mfa[i,1]-xgem)*(mfa[i,2]-ygem))+teller;
    noemer:=((mfa[i,1]-xgem)*(mfa[i,1]-xgem))+noemer;
  END;
helling:=teller/noemer;
nulwaarde:=ygem-(helling*xgem);

(* y=nulwaarde+helling*x=a+b*x *)
```

```

for i:=1 to puntenaantal do
  BEGIN
    mfa[i,2]:=mfa[i,2]-(nulwaarde+(mfa[i,1]*helling));
  END;
```

```

(*****)
```

```

clearscreen;
findworld(3,specsa,22,1,2);
with world[3] do
  BEGIN
    hulp:=Y1;
    Y1:=Y2;
    Y2:=hulp;
  END;
```

```

selectworld(3);
selectwindow(3);
drawborder;
setlinestyle(0);
drawaxis(8,5,4,10,3,13,0,0,true);
drawpolygon(mfa,1,-punteanaantal,7,2,0);
drawaxis(0,0,4,10,3,13,0,0,false);
drawpolygon(specsa,1,-11,0,1,0);
drawaxis(0,0,4,10,3,13,0,0,false);
drawpolygon(specsa,12,-22,0,1,0);
gotoxy(73,24);
write(meetas,' [mm]');
gotoxy(2,4);
write('d',meetrichting,' [um]');
gotoxy(20,2);
writeln('Foutbron: Rechtheid (T',meetrichting,meetas,')');
gotoxy(20,4);
writeln('Meetas: ',meetas);
gotoxy(50,2);
write('Spec : d',meetrichting,' <',minmeetfout:4:1);
gotoxy(12,24);
write('Druk Shft PrtSc');
delay(2000);
gotoxy(12,24);
write(' ');
delay(2500);
writeln(LST);writeln(LST);
write(LST,'Foutbron : T',meetrichting,meetas);
write(LST,' ':5,coord1,'-coördinaat : ',coorda:8:2);
writeln(LST,' ':5,coord2,'-coördinaat : ',coordb:8:2);
write(LST,'-----');
writeln(LST,'-----');
writeln(LST);
writeln(LST,' Machinewaarde T',meetas,meetas);

```

```

for i:= 1 to puntenaantal do
  BEGIN
  writeln(LST,mfali,1):12:2,' ':17,mfali,2):12:2);
  if meetas='X' then
    BEGIN
    if meetrichting='Y' then (schrijf de meetserie in een)
      BEGIN (nieuw array en bepaal de maximale)
        rechtheidyx[i,1]:=mfali,1); (gemeten rechtheidsfout)
        rechtheidyx[i,2]:=mfali,2);
        specyx:=minmeetfout;
        hellingyx:=helling;
        nulwaardeyx:=nulwaarde;
        if abs(rechtheidyx[i,2])>abs(TYX) then TYX:=rechtheidyx[i,2];
      END;
    END;
  END;

```

```

if meetrichting='Z' then
  BEGIN
    rechtheidzx[i,1]:=mfa[i,1];
    rechtheidzx[i,2]:=mfa[i,2];
    speczx:=minmeetfout;
    hellingzx:=helling;
    nulwaardezx:=nulwaarde;
    if abs(rechtheidzx[i,2])>abs(TZX) then TZX:=rechtheidzx[i,2];
  END;
END;
if meetas='Y' then
  BEGIN
    if meetrichting='X' then
      BEGIN
        rechtheidxy[i,1]:=mfa[i,1];
        rechtheidxy[i,2]:=mfa[i,2];
        specxy:=minmeetfout;
        hellingxy:=helling;
        nulwaardexy:=nulwaarde;
        if abs(rechtheidxy[i,2])>abs(TXY) then TXY:=rechtheidxy[i,2];
      END;
    if meetrichting='Z' then
      BEGIN
        rechtheidzy[i,1]:=mfa[i,1];
        rechtheidzy[i,2]:=mfa[i,2];
        speczy:=minmeetfout;
        hellingzy:=helling;
        nulwaardezy:=nulwaarde;
        if abs(rechtheidzy[i,2])>abs(TZY) then TZY:=rechtheidzy[i,2];
      END;
    END;
if meetas='Z' then
  BEGIN
    if meetrichting='X' then
      BEGIN
        rechtheidxz[i,1]:=mfa[i,1];
        rechtheidxz[i,2]:=mfa[i,2];
        specxz:=minmeetfout;
        hellingxz:=helling;
        nulwaardexz:=nulwaarde;
        if abs(rechtheidxz[i,2])>abs(TXZ) then TXZ:=rechtheidxz[i,2];
      END;
    if meetrichting='Y' then
      BEGIN
        rechtheidyz[i,1]:=mfa[i,1];
        rechtheidyz[i,2]:=mfa[i,2];
        specyz:=minmeetfout;
        hellingyz:=helling;
        nulwaardeyz:=nulwaarde;
        if abs(rechtheidyz[i,2])>abs(TYZ) then TYZ:=rechtheidyz[i,2];
      END;
    END;
END;
END;

```

```

writeln(LST);
writeln(LST);
gotoxy(12,24);
write('Druk <CR> voor terug naar hoofdmenu.');
```

```

read;
END;
```

OVERLAY PROCEDURE HAAKSHEIDSFOUT;

BEGIN

```

clearscreen;
gotoxy(22,2);
writeln('HAAKSHEIDSFOUTEN');
clearscreen;
gotoxy(12,10);
write('We onderscheiden drie haaksheidsfouten nl : ');
gotoxy(15,12);
write('1: S',upcase(u),v);
gotoxy(15,14);
write('2: S',upcase(u),w);
gotoxy(15,16);
write('3: S',upcase(v),w);
gotoxy(12,20);
write('Fout 1 en 2 hebben de ',upcase(u),'-geleiding als referentie-as');
gotoxy(12,21);
write('en fout 3 heeft de ',upcase(v),'-geleiding als referentie-as.');
```

```

gotoxy(12,22);
write('De haaksheidsfout is de hoek tussen de assen minus 90 graden.');
```

```

gotoxy(12,24);
write('Toets <CR> voor haaksheidsfouten ');
read;
```

```

if (hellingxy=0) or (hellingyx=0) or (hellingxz=0) or (hellingzx=0)
  or (hellingyz=0) or (hellingzy=0) then
  BEGIN
    clearscreen;
    gotoxy(12,10);
    write('U heeft nog niet alle rechtheidsfouten gemeten.');
```

```

    gotoxy(12,12);
    write('Toets <CR> en meet eerst alle rechtheidsfouten.');
```

```

    read;
    exit;
  END;
```



```

if ((u='y') or (u='z')) and ((w='z') or (w='y')) then
hhfuv:=-hellingyz-hellingzy;
if ((v='x') or (v='y')) and ((w='y') or (w='x')) then
hhfvw:=-hellingyx-hellingxy;
if ((v='x') or (v='z')) and ((w='z') or (w='x')) then
hhfvw:=-hellingxz-hellingzx;
if ((v='y') or (v='z')) and ((w='z') or (w='y')) then
hhfvw:=-hellingyz-hellingzy;

hhfuv:=hhfuv/4.8481368E-3;           {maak van um/mm hier boogseconden}
hhfuv:=hhfuv/4.8481368E-3;
hhfvw:=hhfvw/4.8481368E-3;

```

```

clearscreen;
gotoxy(12,12);
write('S',upcase(u),v,' = ',hhfuv:6:4,' boogseconde. ');
gotoxy(12,14);
write('S',upcase(u),w,' = ',hhfuv:6:4,' boogseconde. ');
gotoxy(12,16);
write('S',upcase(v),w,' = ',hhfvw:6:4,' boogseconde. ');
gotoxy(12,24);
write('Druk <CR> voor terug naar hoofdaenu. ');
read;
END;

```

OVERLAY PROCEDURE KANTELFOUT;

```

Var kantelas:char;
Var boventemp, ondertemp, sensorafst:real;
Var RotYX, RotZX, RotXY, RotZY, RotXZ, RotYZ:PlotArray;

```

```

(*           VARIABELEN IN PROCEDURE KANTELFOUT           *)
(*           *)
(* kantelas: as waarom de betreffende meetas kantelt      *)
(* boventemp: temperatuur bovenop het portaal of een andere balk *)
(* ondertemp: temperatuur onder op hetzelfde portaal of balk *)
(* sensorafst: afstand tussen de temperatuursensoren      *)
(* RotYX: array waarin alle meetgegevens zitten van de kantelfout van *)
(*           de x-geleiding om de y-as                      *)
(* idem voor RotZX, RotXY, RotZY, RotXZ, RotYZ            *)
(*           *)
(*           EINDE                                           *)

```

```

BEGIN
clearscreen;
gotoxy(22,2);
writeln('KANTELFOUTEN');
gotoxy(12,12);
write('Wilt U Random of met vaste Stapgrootte meten ? (R/S) ');
readln(randstap);
randstap:=upcase(randstap);

```

```

if randstap='S' then
  BEGIN
    clearscreen;
    gotoxy(12,6);
    write('Welke meetas wilt U controleren ? (X/Y/Z) ');
    readln(meetas);
    meetas:=upcase(meetas);
    if meetas='X' then BEGIN coord1:='Y'; coord2:='Z'; END;
    if meetas='Y' then BEGIN coord1:='X'; coord2:='Z'; END;
    if meetas='Z' then BEGIN coord1:='X'; coord2:='Y'; END;
    gotoxy(12,8);
    write('Om welke as kantelt de machine ? (X/Y/Z) ');
    read(kantelas);
    kantelas:=upcase(kantelas);
    gotoxy(12,10);
    write('Geef de ',coord1,'-coördinaat in : ');
    read(coorda);
    gotoxy(12,12);
    write('Geef de ',coord2,'-coördinaat in : ');
    read(coordb);
    clearscreen;
    gotoxy(8,4);
    write('Kantelfout : R',kantelas,meetas,' ':8);
    write(coord1,'= ',coorda:6:2,' ':5,coord2,'= ',coordb:6:2);
    gotoxy(12,10);
    write('Wat is de startwaarde van de metingen ? ');
    readln(beginwaarde);
    gotoxy(12,12);
    write('Wat is de eindwaarde van de metingen ? ');
    readln(eindwaarde);
    gotoxy(12,14);
    write('Wat is de gewenste stapgrootte ? ');
    readln(stapgrootte);
    gotoxy(12,16);
    write('Wilt U heen en terug meten ? (Y/N) ');
    read(heenterug);
    heenterug:=upcase(heenterug);
    gotoxy(12,18);
    write('Wat is de boventemperatuur ? (graden celcius) ');
    read(boventemp);
    gotoxy(12,20);
    write('Wat is de ondertemperatuur ? (graden celcius) ');
    read(ondertemp);
    gotoxy(12,22);
    write('Wat is de afstand tussen de temperatuursensoren ? (mm) ');
    read(sensorafst);
    puntenaantal:=trunc(((eindwaarde-beginwaarde)/stapgrootte)+1);
    for i:=1 to puntenaantal do
      BEGIN
        clearscreen;
        gotoxy(12,5);
        write('LET OP : de machinewaarde is in mm,');
        write(' de meetfout is in bgsec. ');
        mfa[i,1]:=beginwaarde+((i-1)*stapgrootte);
      END
    end
  end

```

```

if randstap='S' then
  BEGIN
    clearscreen;
    gotoxy(12,6);
    write('Welke meetas wilt U controleren ? (X/Y/Z) ');
    readln(meetas);
    meetas:=upcase(meetas);
    if meetas='X' then BEGIN coord1:='Y'; coord2:='Z'; END;
    if meetas='Y' then BEGIN coord1:='X'; coord2:='Z'; END;
    if meetas='Z' then BEGIN coord1:='X'; coord2:='Y'; END;
    gotoxy(12,8);
    write('Om welke as kantelt de machine ? (X/Y/Z) ');
    read(kantelas);
    kantelas:=upcase(kantelas);
    gotoxy(12,10);
    write('Geef de ',coord1,'-coördinaat in : ');
    read(coorda);
    gotoxy(12,12);
    write('Geef de ',coord2,'-coördinaat in : ');
    read(coordb);
    clearscreen;
    gotoxy(8,4);
    write('Kantelfout : R',kantelas,meetas,' :8);
    write(coord1,'= ',coorda:6:2,' ':5,coord2,'= ',coordb:6:2);
    gotoxy(12,10);
    write('Wat is de startwaarde van de metingen ? ');
    readln(beginwaarde);
    gotoxy(12,12);
    write('Wat is de eindwaarde van de metingen ? ');
    readln(eindwaarde);
    gotoxy(12,14);
    write('Wat is de gewenste stapgrootte ? ');
    readln(stapgrootte);
    gotoxy(12,16);
    write('Wilt U heen en terug meten ? (Y/N) ');
    read(heenterug);
    heenterug:=upcase(heenterug);
    gotoxy(12,18);
    write('Wat is de boventemperatuur ? (graden celcius) ');
    read(boventemp);
    gotoxy(12,20);
    write('Wat is de ondertemperatuur ? (graden celcius) ');
    read(ondertemp);
    gotoxy(12,22);
    write('Wat is de afstand tussen de temperatuursensoren ? (mm) ');
    read(sensorafst);
    puntenaantal:=trunc(((eindwaarde-beginwaarde)/stapgrootte)+1);
    for i:=1 to puntenaantal do
      BEGIN
        clearscreen;
        gotoxy(12,5);
        write('LET OP : de machinewaarde is in mm,');
        write(' de meetfout is in bgsec. ');
        afali,11:=beginwaarde+(i-1)*stapgrootte);

```

```

gotoxy(12,15);
writeln('De machinewaarde = ',(beginwaarde+((i-1)*stapgrootte)
:6:2);
gotoxy(12,17);
write('De meetfout = ');
readln(meetfout);
mfa[i,2]:=meetfout;
END;
if heenterug='Y' then
BEGIN
for i:=punteanaantal+1 to (2*punteanaantal)-1 do
BEGIN
clearscreen;
gotoxy(12,5);
write('LET OP : de machinewaarde is in mm,');
write(' de meetfout is in bgsec. ');
mfa[i,1]:=mfa[punteanaantal,1]-((i-punteanaantal)*stapgrootte);
gotoxy(12,15);
write('De machinewaarde = ');
writeln(mfa[punteanaantal,1]-((i-punteanaantal)*stapgrootte)
:6:2);
gotoxy(12,17);
write('De meetfout = ');
readln(meetfout);
mfa[i,2]:=meetfout;
END;
END;
END;
if randstap='R' then
BEGIN
clearscreen;
gotoxy(12,6);
write('Welke meetas wilt U controleren ? (X/Y/Z) ');
readln(meetas);
meetas:=upcase(meetas);
if meetas='X' then BEGIN coord1:='Y'; coord2:='Z'; END;
if meetas='Y' then BEGIN coord1:='X'; coord2:='Z'; END;
if meetas='Z' then BEGIN coord1:='X'; coord2:='Y'; END;
gotoxy(12,8);
write('Om welke as kantelt de machine ? (X/Y/Z) ');
read(kantelas);
kantelas:=upcase(kantelas);
gotoxy(12,10);
write('Geef de ',coord1,'-coördinaat in : ');
read(coorda);
gotoxy(12,12);
write('Geef de ',coord2,'-coördinaat in : ');
read(coordb);
clearscreen;
gotoxy(8,4);
write('Kantelfout : R',kantelas,meetas,' ':8);
write(coord1,'= ',coorda:6:2,' ':5,coord2,'= ',coordb:6:2);
gotoxy(12,10);

```

```

write('Hoeveel meetpunten wilt U invoeren ? ');
read(puntenaantal);
gotoxy(12,12);
write('Wat is de boventemperatuur ? (graden celcius) ');
read(boventemp);
gotoxy(12,14);
write('Wat is de ondertemperatuur ? (graden celcius) ');
read(ondertemp);
gotoxy(12,16);
write('Wat is de afstand tussen de temperatuursensoren ? (mm) ');
read(sensorafst);
for i:=1 to puntenaantal do
  BEGIN
    clearscreen;
    gotoxy(12,5);
    write('LET OP: -de machinewaarden in oplopende grootte invoeren.');
```

```

    gotoxy(12,7);
    write('      : -de machinewaarde is in mm,');
    write(' de meetfout is in bgsec.');
```

```

    gotoxy(12,13);
    write('Puntnummer : ',i);
    gotoxy(12,15);
    write('De machinewaarde = ');
    read(machinewaarde);
    mfa[i,1]:=machinewaarde;
    gotoxy(12,17);
    write('De meetfout = ');
    readln(meetfout);
    mfa[i,2]:=meetfout;
    eindwaarde:=mfa[puntenaantal,1];
    beginwaarde:=mfa[1,1];
  END;
END;
if heenterug='Y' then
  BEGIN
    for i:=1 to puntenaantal-1 do
      BEGIN
        mfa[i,2]:=(mfa[i,2]+mfa[(2*puntenaantal)-i,2])/2;
      END;
    END;
  END;

clearscreen;
findworld(3,mfa,puntenaantal,1,1.5);
with world[3] do
  BEGIN
    hulp:=Y1;
    Y1:=Y2;
    Y2:=hulp;
  END;
selectworld(3);
selectwindow(3);
drawborder;
setlinestyle(0);
drawaxis(8,5,4,10,3,13,0,0,true);
drawpolygon(mfa,1,-puntenaantal,7,2,0);

```

```

gotoxy(73,24);
write(meetas, ' [mm]');
gotoxy(2,4);
write('R',kantelas,meetas, ' [bgsec]');
gotoxy(20,2);
writeln('Foutbron: Kantel (R',kantelas,meetas,')');
gotoxy(20,4);
writeln('Meetas: ',meetas);
gotoxy(50,2);
write('Kantelas : ',kantelas);
gotoxy(12,24);
write('Druk Shft PrtSc');
delay(2000);
gotoxy(12,24);
write(' ');
delay(2500);
writeln(LST);writeln(LST);
write(LST,'Foutbron : R',kantelas,meetas,':5);
write(LST,coord1,'-coördinaat : ',coorda:8:2);
writeln(LST,':5,coord2,'-coördinaat : ',coordb:8:2);
writeln(LST,'Boventemperatuur (Celcius) : ',boventemp:6:2);
writeln(LST,'Ondertemperatuur (Celcius) : ',ondertemp:6:2);
writeln(LST,'Sensorafstand (mm) : ',sensorafst:6:2);
write(LST,'-----');
writeln(LST,'-----');
writeln(LST);
writeln(LST,' Machinewaarde T',kantelas,meetas);

for i:=1 to puntenaantal do
  BEGIN
    writeln(LST,mfa[i,1]:12:2,' ':17,mfa[i,2]:12:2);
    if meetas='X' then {schrijf de meetserie in een nieuw array}
      BEGIN {en bepaal de maximale kantelfout}
        if kantelas='Y' then
          BEGIN
            RotYX[i,1]:=mfa[i,1];
            RotYX[i,2]:=mfa[i,2];
            if abs(RotYX[i,2]) > abs(RYX) then RYX:=RotYX[i,2];
          END;
        if kantelas='Z' then
          BEGIN
            RotZX[i,1]:=mfa[i,1];
            RotZX[i,2]:=mfa[i,2];
            if abs(RotZX[i,2]) > abs(RZX) then RZX:=RotZX[i,2];
          END;
        END;
    if meetas='Y' then
      BEGIN
        if kantelas='X' then
          BEGIN
            RotXY[i,1]:=mfa[i,1];
            RotXY[i,2]:=mfa[i,2];
            if abs(RotXY[i,2]) > abs(RXY) then RXY:=RotXY[i,2];
          END;
        END;
      END;
  END;

```

```

    if kantelas='Z' then
      BEGIN
        RotZY[i,1]:=mfa[i,1];
        RotZY[i,2]:=mfa[i,2];
        if abs(RotZY[i,2]) > abs(RZY) then RZY:=RotZY[i,2];
      END;
    END;
  if meetas='Z' then
    BEGIN
      if kantelas='X' then
        BEGIN
          RotXZ[i,1]:=mfa[i,1];
          RotXZ[i,2]:=mfa[i,2];
          if abs(RotXZ[i,2]) > abs(RXZ) then RXZ:=RotXZ[i,2];
        END;
      if kantelas='Y' then
        BEGIN
          RotYZ[i,1]:=mfa[i,1];
          RotYZ[i,2]:=mfa[i,2];
          if abs(RotYZ[i,2]) > abs(RYZ) then RYZ:=RotYZ[i,2];
        END;
      END;
    END;
  writeln(LST);
  writeln(LST);
  gotoxy(12,24);
  write('Druk <CR> voor terug naar hoofdmenu. ');
  read;
END;

```

OVERLAY PROCEDURE ROTATIEFOUT;

Var RotXX, RotYY, RotZZ: PlotArray;

```

(*          VARIABELEN IN PROCEDURE ROTATIEFOUT          *)
(*                                                    *)
(* RotXX: array waarin alle gegevens zitten over de rotatiefout v/d x-as *)
(* idem voor RotYY en RotZZ                               *)
(*                                                    *)
(*          EINDE                                         *)

```

BEGIN

```

  clearscreen;
  gotoxy(22,2);
  writeln('ROTATIEFOUTEN');
  gotoxy(12,12);
  write('Wilt U Random of met vaste Stapgrootte meten ? (R/S) ');
  readln(randstap);
  randstap:=upcase(randstap);

```

```

if randstap='S' then
  BEGIN
    clearscreen;
    gotoxy(12,6);
    write('Welke meetas wilt U controleren ? (X/Y/Z)   ');
    readln(meetas);
    meetas:=upcase(meetas);
    if meetas='X' then BEGIN coord1:='Y'; coord2:='Z'; END;
    if meetas='Y' then BEGIN coord1:='X'; coord2:='Z'; END;
    if meetas='Z' then BEGIN coord1:='X'; coord2:='Y'; END;
    gotoxy(12,8);
    write('Geef de ',coord1,'-coördinaat in : ');
    read(coorda);
    gotoxy(12,10);
    write('Geef de ',coord2,'-coördinaat in : ');
    read(coordb);
    clearscreen;
    gotoxy(8,4);
    write('Rotatiefout : R',meetas,meetas,' ':8);
    write(coord1,'= ',coorda:6:2,' ':5,coord2,'= ',coordb:6:2);
    gotoxy(12,10);
    write('Wat is de startwaarde van de metingen ?   ');
    readln(beginwaarde);
    gotoxy(12,12);
    write('Wat is de eindwaarde van de metingen ?   ');
    readln(eindwaarde);
    gotoxy(12,14);
    write('Wat is de gewenste stapgrootte ?         ');
    readln(stapgrootte);
    gotoxy(12,16);
    write('Wilt U heen en terug meten ? (Y/N)       ');
    read(heenterug);
    heenterug:=upcase(heenterug);
    puntenaantal:=trunc(((eindwaarde-beginwaarde)/stapgrootte)+1);
    for i:=1 to puntenaantal do
      BEGIN
        clearscreen;
        gotoxy(12,5);
        write('LET OP : de machinewaarde is in mm,');
        write(' de meetfout is in bgsec. ');
        mfa[i,1]:=beginwaarde+((i-1)*stapgrootte);
        gotoxy(12,15);
        writeln('De machinewaarde = ',(beginwaarde+((i-1)*stapgrootte))
          :6:2);
        gotoxy(12,17);
        write('De meetfout =   ');
        readln(meetfout);
        mfa[i,2]:=meetfout;
      END;
    end;
  end;

```



```

if heenterug='Y' then
  BEGIN
    for i:=punteanaantal+1 to (2*punteanaantal)-1 do
      BEGIN
        clearscreen;
        gotoxy(12,5);
        write('LET OP : de machinewaarde is in mm,');
        write(' de meetfout is in bgsec.');
```

mfa[i,1]:=mfa[punteanaantal,1]-((i-punteanaantal)*stapgrootte);

```

        gotoxy(12,15);
        write('De machinewaarde = ');
        writeln(mfa[punteanaantal,1]-((i-punteanaantal)*stapgrootte)
          :6:2);
        gotoxy(12,17);
        write('De meetfout =      ');
        readln(meetfout);
        mfa[i,2]:=meetfout;
      END;
    END;
  END;

if randstap='R' then
  BEGIN
    clearscreen;
    gotoxy(12,6);
    write('Welke meetas wilt U controleren ? (X/Y/Z)      ');
    readln(meetas);
    meetas:=upcase(meetas);
    if meetas='X' then BEGIN coord1:='Y'; coord2:='Z'; END;
    if meetas='Y' then BEGIN coord1:='X'; coord2:='Z'; END;
    if meetas='Z' then BEGIN coord1:='X'; coord2:='Y'; END;
    gotoxy(12,8);
    write('Geef de ',coord1,'-coördinaat in : ');
    read(coorda);
    gotoxy(12,10);
    write('Geef de ',coord2,'-coördinaat in : ');
    read(coordb);
    clearscreen;
    gotoxy(8,4);
    write('Rotatiefout : R',meetas,meetas,' ':8);
    write(coord1,'= ',coorda:6:2,' ':5,coord2,'= ',coordb:6:2);
    gotoxy(12,10);
    write('Hoeveel meetpunten wilt U invoeren ?      ');
    read(punteanaantal);
    for i:=1 to punteanaantal do
      BEGIN
        clearscreen;
        gotoxy(12,5);
        write('LET OP : -de machinewaarden in oplopende grootte invoeren.');
```

-gotoxy(12,7);

```

        write('      : -de machinewaarde is in mm,');
        write(' de meetfout is in bgsec.');
```

gotoxy(12,13);

```

        write('Puntnummer : ',i);
        gotoxy(12,15);
```

```

        write('De machinewaarde = ');
        read(machinewaarde);
        mfa[i,1]:=machinewaarde;
        gotoxy(12,17);
        write('De meetfout = ');
        readln(meetfout);
        mfa[i,2]:=meetfout;
        eindwaarde:=mfa[puntenaantal,1];
        beginwaarde:=mfa[1,1];
    END;
END;
if heenterug='Y' then
    BEGIN
        for i:=1 to puntenaantal-1 do
            BEGIN
                mfa[i,2]:=(mfa[i,2]+mfa[(2#puntenaantal)-i,2])/2;
            END;
        END;

clearscreen;
findworld(3,mfa,puntenaantal,1,1.5);
with world[3] do
    BEGIN
        hulp:=Y1;
        Y1:=Y2;
        Y2:=hulp;
    END;
selectworld(3);
selectwindow(3);
drawborder;
setlinestyle(0);
drawaxis(8,5,4,10,3,13,0,0,true);
drawpolygon(mfa,1,-puntenaantal,7,2,0);
gotoxy(73,24);
write(meetas, ' [mm]');
gotoxy(2,4);
write('R',meetas,meetas, ' [bgsec]');
gotoxy(20,2);
writeln('Foutbron: Rotatie (R',meetas,meetas,')');
gotoxy(20,4);
writeln('Meetas: ',meetas);
gotoxy(12,24);
write('Druk Shft PrtSc');
delay(2000);
gotoxy(12,24);
write(' ');
delay(2500);
writeln(LST);writeln(LST);
write(LST,'Foutbron : R',meetas,meetas);
write(LST,' ':5,coord1,'-coördinaat : ',coorda:8:2);
writeln(LST,' ':5,coord2,'-coördinaat : ',coordb:8:2);
write(LST,'-----');
writeln(LST,'-----');
writeln(LST);
writeln(LST,' Machinewaarde T',meetas,meetas);

```

```

for i:=1 to puntenaantal do
  BEGIN
    writeln(LST,mfa[i,1]:12:2,' ':17,mfa[i,2]:12:2);
    if meetas='X' then          {schrijf de meetserie in een nieuw array}
      BEGIN                    {en bepaal de maximale rotatiefout}
        RotXX[i,1]:=mfa[i,1];
        RotXX[i,2]:=mfa[i,2];
        if abs(RotXX[i,2]) > abs(RXX) then RXX:=RotXX[i,2];
      END;
    if meetas='Y' then
      BEGIN
        RotYY[i,1]:=mfa[i,1];
        RotYY[i,2]:=mfa[i,2];
        if abs(RotYY[i,2]) > abs(RYY) then RYY:=RotYY[i,2];
      END;
    if meetas='Z' then
      BEGIN
        RotZZ[i,1]:=mfa[i,1];
        RotZZ[i,2]:=mfa[i,2];
        if abs(RotZZ[i,2]) > abs(RZZ) then RZZ:=RotZZ[i,2];
      END;
    END;
  writeln(LST);
  writeln(LST);
  gotoxy(12,24);
  write('Druk <CR> voor terug naar hoofdmenu.');
```

```

  read;
END;
```

```
OVERLAY PROCEDURE MAXMEETFOUT;
```

```
Var rekenen:char;
```

```
BEGIN
```

```

  RXX:=4.8481368E-3#RXX;          (ga van bgsec naar um/mm)
  RYY:=4.8481368E-3#RYY;
  RZZ:=4.8481368E-3#RZZ;
  RYX:=4.8481368E-3#RYX;
  RZX:=4.8481368E-3#RZX;
  RXY:=4.8481368E-3#RXY;
  RZY:=4.8481368E-3#RZY;
  RXZ:=4.8481368E-3#RXZ;
  RYZ:=4.8481368E-3#RYZ;
  hhfuv:=4.8481368E-3#hhfuv;
  hhfuw:=4.8481368E-3#hhfuw;
  hhfvw:=4.8481368E-3#hhfvw;
```

```

clearscreen;
gotoxy(9,4);
write('De volgende fouten in de machine heeft U nog niet gemeten :');
gotoxy(1,6);
if TXX=0 then writeln('          TXX');
if TYY=0 then writeln('          TYY');
if TZZ=0 then writeln('          TZZ');
if TYX=0 then writeln('          TYX');
if TZX=0 then writeln('          TZX');
if TXY=0 then writeln('          TXY');
if TZY=0 then writeln('          TZY');
if TXZ=0 then writeln('          TXZ');
if TVZ=0 then writeln('          TVZ');
if RYX=0 then writeln('          RYX');
if RZX=0 then writeln('          RZX');
if RXY=0 then writeln('          RXY');
if RZY=0 then writeln('          RZY');
if RXZ=0 then writeln('          RXZ');
if RYZ=0 then writeln('          RYZ');
if RXX=0 then writeln('          RXX');
if RYY=0 then writeln('          RYY');
if RZZ=0 then writeln('          RZZ');
gotoxy(9,25);
write('Wilt U toch met de reeds gemeten fouten rekenen ? (Y/N) ');
read(rekenen);
rekenen:=upcase(rekenen);
if rekenen='Y' then
  BEGIN
    assign(foutber,'B:foutber.chn');{koppel het volgende blok aan dit blok}
    chain(foutber);
  END;
END;

```

```

function menuchoice(oldchoice:integer):integer;      (in deze functie wordt de)
const prompt:string[6]='=> ';                       (keuze van de foutsoort via)
      title='CAFAM FOUTENBEREKENING';               (een menu bepaald)
      n=6;
      menuitems=array [0..6] of string[20]=         (onderdelen van het menu)
      ('Quit',
       'Lineariteitsfouten',
       'Rechtheidsfouten',
       'Haaksheidsfouten',
       'Kantelfouten',
       'Rotatiefouten',
       'Meetfoutberekening');

```

```

Var x,y,i,j:integer;
    ch:char;
    quit:boolean;

```

```

begin
  quit:=false;
  x:=40-length(title) div 2;           {x-coördinaat van de titel}
  gotoxy(x,1);
  write(title);                         {schrijf de titel op het scherm}
  gotoxy(5,24);
  write('Maak Uw keuze met SPACE/BACKSPACE of een nummer, toets dan RETURN.');
```

{y=8-(6/2)=5}

```

  y:=8-n div 2;
  x:=20;
  for i:=0 to n do
    begin
      gotoxy(x-4,y+i#2);
      write(i:1,' -- ',menuitems[i]); {zet de menu mogelijkheden op het scherm}
    end;
  i:=oldchoice;                         {maak i initieel nul}
  gotoxy(x-10,y+i#2);
  write(prompt,i:1,' -- ',menuitems[i]);{schrijf:==> '0 -- Quit' op het scherm}
  gotoxy(50,y+i#2);
  write('<==');                           {schrijf hierachter '<=='}
  while ch<> ^M do                       {doe het volgende zolang geen <CR> ingetoetst wordt}
    begin
      ch='.';                             {geef de variabele ch voor de zekerheid de waarde '.'}
      if keypressed then
        begin
          read(kbd,ch);
          j:=0;
        end;
      if ch= ^H then                     {als het ingetoetste karakter BACKSPACE is}
        begin                           {doe dan het volgende}
          gotoxy(x-10,y+i#2);           {veeg het pijltje voor de vorige keuze weg}
          write(' ');
          gotoxy(50,y+i#2);             {veeg het pijltje na de vorige keuze weg}
          write(' ');
          i:=i-1;                       {maak de keuze een lager}
          if i=-1 then i:=n;            {als de keuze daardoor lager gaat}
          gotoxy(x-10,y+i#2);           {dan nul ga dan naar keuze nummer zes}
          write(prompt,i:1,' -- ',menuitems[i]); {schrijf voor en na de nieuwe}
          gotoxy(50,y+i#2);             {een pijltje}
          write('<==');
        end;
      if ch= ' ' then                   {als het ingetoetste karakter een SPACE is}
        begin
          gotoxy(x-10,y+i#2);           {veeg het pijltje voor en na de vorige}
          write(' ');                   {keuze weg}
          gotoxy(50,y+i#2);
          write(' ');
          i:=i+1;                       {hoog de huidige keuze met een op}
          if i>n then i:=0;             {als de keuze hoger wordt dan het}
          gotoxy(x-10,y+i#2);           {aantal items ga dan naar keuze nul}
          write(prompt,i:1,' -- ',menuitems[i]); {schrijf voor en na de nieuwe}
          gotoxy(50,y+i#2);             {keuze een pijltje}
          write('<==');
        end;
    end;
end;
```

```

if (ord(ch)-48 in [0..n]) and (ord(ch)-48 <>i) then
  begin
    gotoxy(x-10,y+i#2);      {als het ingetikte een getalwaarde representeerd}
    write(' ');              {die tussen nul en zes ligt en niet gelijk}
    write(' ');              {is aan de huidige keuze doe dan het volgende}
    gotoxy(50,y+i#2);       {veeg de pijltjes bij de vorige keuze weer uit}
    write(' ');
    i:=ord(ch)-48;          {geef de nieuwe keuze de waarde van het ingegeven}
    gotoxy(x-10,y+i#2);      {getal}
    write(prompt,i:1,' -- ',menuitems[i]); {zet voor en na deze nieuwe}
    gotoxy(50,y+i#2);       {keuze een pijltje}
    write('<=>');
  end;
if ch='^C' then quit:=true;
if (ch='^[') then           {als het ingetikte karakter de ESC-toets is}
  if keypressed then       {lees het volgende karakter en maak}
    begin                  {de variabele quit true als het een nul is}
      read(kbd,ch);
      if ch='0' then quit:=true;
    end
  else quit:=true;         {als geen toets ingedrukt wordt maak dan}
if quit then               {quit ook true}
  begin                    {als quit true is ga dan uit de grafische}
    LeaveGraphic;          {mode en ga terug naar het systeem}
    halt;
  end;
end;
menuchoice:=i;            {maak de uitkomst van deze subroutine gelijk aan de keuze}
end;

```

```

Var choice:integer;

```

```

Begin
  {hier begint het hoofdprogramma}
  InitGraphic;             {ga in de grafische mode}
  SetBreakOn;              {zorg dat het programma stopt bij een fout}
  SetMessageOn;           {en dat het foutnummer gemeld wordt}
  SetHeaderOn;            {zet de header aan bij eventuele grafieken}
  DefineWorld(2,0,199,639,0); {definieer een coördinatensysteem}
  DefineWindow(10,0,0,XMax61b,YMax61b); {definieer een window}
  choice:=0;              {maak de beginkeuze gelijk aan nul}

  deltax:=0; deltax:=0; deltax:=0; {maak alle variabelen initieel nul}
  TXX:=0; TYY:=0; TZZ:=0;          {zodat geen waarden uit vorige metingen}
  TYX:=0; TZX:=0; TXY:=0; TZY:=0; TXZ:=0; TYZ:=0;          {worden meegenomen}
  hhfuv:=0; hhfuv:=0; hhfvw:=0;
  RYX:=0; RZX:=0; RXY:=0; RZY:=0; RXZ:=0; RYZ:=0;
  RXX:=0; RYY:=0; RZZ:=0;
  hellingyx:=0; hellingzx:=0; hellingxy:=0; hellingzy:=0;
  hellingxz:=0; hellingyz:=0;

```

```

repeat                                     {blijf het onderstaande herhalen totdat de}
  ClearScreen;                             {keuze op Quit is gevallen}
  ResetWindowStack;                         {maak alles opnieuw in orde voor het menu}
  ResetWindows;
  ResetWorlds;
  DefineWorld(2,0,199,639,0);
  DefineWindow(10,0,0,XmaxGlb,YMaxGlb);
  SelectWorld(2);
  SelectWindow(10);
  SetAspect(1.0);
  choice:=menuchoice(choice);               {haal de keuze uit de menufunctie}
  ClearScreen;
  case choice of                             {kies de juiste procedure}
    1:LINEARITEITSFOOT;
    2:RECHTHEIDSFOOT;
    3:HAAKSHEIDSFOOT;
    4:KANTELFOUT;
    5:ROTATIEFOOT;
    6:MAXMEETFOUT;
  end;
until choice=0;
LeaveGraphic;
End.

```

PROGRAM FOUTENBEREKENING;

```
{ $I A:typedef.sys }           {Include deze files van de Turbo-diskette}
{ $I A:graphix.sys }
{ $I A:kernel.sys }
{ $I A:windows.sys }
```

```
Var uber,vber,wber:real;      { Deze variabelen nemen we over }
Var startpunt,twegel,np,uas,vas,was:char; { van de vorige blokken.      }
Var draaiing:real;
Var uuv,uvw,vuv,vvw,muv,mvw:real;
Var ucomp,vcomp,wcomp:real;
Var u,v,w:char;
```

```
Var TXX,TYY,TZZ:real;
Var TYX,TZX,TXY,TZY,TeX,TyZ:real;
Var hhfuv,hhfuv,hhfvm:real;
Var RYX,RZX,RXY,RZY,RXZ,RYZ:real;
Var RXX,RYY,RZZ:real;
Var deltax,deltay,deltaz:real;
Var deltaxkwadr,deltaykwadr,deltazkwadr:real;
Var termvuu,termuu,termvuu:real;
```

BEGIN

```
u:=upcase(u);                 {maak hoofdletters van de uvw variabelen}
v:=upcase(v);
w:=upcase(w);
```

{bepaal de grootste arm in absolute waarde de bij kantelfouten meewerken}

```
if abs(wber+wuv+mvw)>abs(muv+mvw) then termvuu:=abs(wber+wuv+mvw)
                                     else termvuu:=abs(muv+mvw);
if abs(vber+vuv+vvw)>abs(vuv+vvw) then termuu:=abs(vber+vuv+vvw)
                                     else termuu:=abs(vuv+vvw);
if abs(wber+mvw)>abs(mvw) then termvuu:=abs(wber+mvw)
                                     else termvuu:=abs(mvw);
```

{hieronder wordt iedere de som van iedere kolom uit de foutentabel}
{bepaald door de componenten kwadratisch op te tellen}

if (u='X') and (v='Y') then

BEGIN

```
deltaxkwadr:=sqr(TXX)+sqr(TXY)+sqr(TXZ)+sqr(RYX*(termvuu))
              +sqr(wber*RYZ)+sqr(RZX*(termuu))+
              sqr(RZY*(vvw+vcomp))+sqr(RYY*(termvuu))+
              sqr(RZZ*vcomp)+sqr(hhfuv*vber)+sqr(hhfvm*wber);
deltaykwadr:=sqr(TYY)+sqr(TYX)+sqr(TYZ)+sqr(RXY*(termvuu))+
              +sqr(wber*RXZ)+sqr(RZX*(uuv+uvw+ucomp))+
              +sqr(RZY*(uvw+ucomp))+sqr(RXX*(termvuu))
              +sqr(RZZ*ucomp)+sqr(hhfvm*wber);
```



```

deltazkwadr:=sqr(TZZ)+sqr(TZX)+sqr(TZY)+sqr(RXY*(vbw+vcomp))+
sqr(RXZ*vcomp)+sqr(RYX*(uuv+uvw+ucomp))+
sqr(ucomp#RYZ)+sqr(RXX*(ter#vuu))+
sqr(RYY*(uvw+ucomp));
END;
if (u='X') and (v='Z') then
BEGIN
deltaxkwadr:=sqr(TXX)+sqr(TXY)+sqr(TXZ)+sqr(RXZ*(ter#vuu))
+(wber#RZY)+sqr(RYX*(ter#vuu))+
sqr(RYZ*(vbw+vcomp))+sqr(RZZ*(ter#vuu))+
sqr(RYY*vcomp)+(hhfuv#wber)+sqr(hhfuv#wber);
deltazkwadr:=sqr(TZZ)+sqr(TZX)+sqr(TZY)+sqr(RXZ*(ter#vuu))+
sqr(wber#RXY)+sqr(RYX*(uuv+uvw+ucomp))+
sqr(RYZ*(uvw+ucomp))+sqr(RXX*(ter#vuu))
+sqr(RYY*ucomp)+sqr(hhfuv#wber);
deltaykwadr:=sqr(TYY)+sqr(TYX)+sqr(TYZ)+sqr(RXZ*(vbw+vcomp))+
sqr(RXY*vcomp)+sqr(RXZ*(uuv+uvw+ucomp))+
sqr(ucomp#RZY)+sqr(RXX*(ter#vuu))+
sqr(RZZ*(uvw+ucomp));
END;
if (u='Y') and (v='X') then
BEGIN
deltaykwadr:=sqr(TYY)+sqr(TYX)+sqr(TYZ)+sqr(RXY*(ter#vuu))
+sqr(wber#RXZ)+sqr(RZY*(ter#vuu))+
sqr(RXZ*(vbw+vcomp))+sqr(RXX*(ter#vuu))+
sqr(RZZ*vcomp)+sqr(hhfuv#wber)+sqr(hhfuv#wber);
deltaxkwadr:=sqr(TXX)+sqr(TXY)+sqr(TXZ)+sqr(RYX*(ter#vuu))+
sqr(wber#RZY)+sqr(RZY*(uuv+uvw+ucomp))+
sqr(RXZ*(uvw+ucomp))+sqr(RYY*(ter#vuu))
+sqr(RZZ*ucomp)+sqr(hhfuv#wber);
deltazkwadr:=sqr(TZZ)+sqr(TZX)+sqr(TZY)+sqr(RYX*(vbw+vcomp))+
sqr(RYZ*vcomp)+sqr(RYX*(uuv+uvw+ucomp))+
sqr(ucomp#RXZ)+sqr(RYY*(ter#vuu))+
sqr(RXX*(uvw+ucomp));
END;
if (u='Y') and (v='Z') then
BEGIN
deltaykwadr:=sqr(TYY)+sqr(TYZ)+sqr(TYX)+sqr(RZY*(ter#vuu))
+sqr(wber#RXZ)+sqr(RYX*(ter#vuu))+
sqr(RXZ*(vbw+vcomp))+sqr(RZZ*(ter#vuu))+
sqr(RXX*vcomp)+sqr(hhfuv#wber)+sqr(hhfuv#wber);
deltazkwadr:=sqr(TZZ)+sqr(TZY)+sqr(TZX)+sqr(RYZ*(ter#vuu))+
sqr(wber#RXY)+sqr(RYX*(uuv+uvw+ucomp))+
sqr(RXZ*(uvw+ucomp))+sqr(RYY*(ter#vuu))
+sqr(RXX*ucomp)+sqr(hhfuv#wber);
deltaxkwadr:=sqr(TXX)+sqr(TXY)+sqr(TXZ)+sqr(RYZ*(vbw+vcomp))
+sqr(RYX*vcomp)+sqr(RZY*(uuv+uvw+ucomp))+
sqr(ucomp#RXZ)+sqr(RYY*(ter#vuu))+
sqr(RZZ*(uvw+ucomp));
END; -

```


PROGRAM CONFIGURATIE;

```
($I A:typedef.sys)           (Include deze files van de Turbo-diskette)
($I A:graphix.sys)
($I A:kernel.sys)
($I A:windows.sys)
```

Type letters=string[20];

Type machine= record

```
    uber,vber,wber:real;
    startpunt,twegel,np,uas,vas,was:char;
    draaiing:real;
    uuv,uvm,vuv,vvm,muv,mvm:real;
    ucomp,vcomp,wcomp:real;
    verb1,verb2,verb3:char;
    verb4,verb5,verb6:char;
    merk,tiepe:letters;
end;
```

```
Var uber,vber,wber:real;           (dit blok variabelen staat hier)
Var startpunt,twegel,np,uas,vas,was:char; (om de volgorde hetzelfde te houden)
Var draaiing:real;
Var uuv,uvm,vuv,vvm,muv,mvm:real;
Var ucomp,vcomp,wcomp:real;
Var verb1,verb2,verb3:char;
Var verb4,verb5,verb6:char;
Var merk,tiepe:letters;
```

```
Var datafile:file of machine; (file waarin de configuratie van de machine zit)
Var config:machine;           (variabele bestaand uit alle machinekenmerken)
Var filename:string[20];      (naam van de op te roepen file)
Var variabele:file;           (hulpvariabele voor fileoproep)
Var tekening,bedoeldemach:char; (ja/nee variabelen voor keuzes)
```

```
Function Exist(filename:letters):boolean; (kijkt of de file bestaat)
Var fil:file;
BEGIN
    Assign(fil,'B:'+filename); (geef de ingegeven filenaam aan fil)
    {$I-} Reset(fil); {$I+} (reset de file, wanneer de file al bestaat)
    Exist:=(IOresult=0); (dan geeft de IOresultfunctie nul aan)
END;
```

```
Procedure Leesdiskfile; (leest de file van de diskette)
BEGIN
    assign(datafile,'B:'+filename); (geef de filenaam aan datafile)
    reset(datafile); (zet de startwaarde vooraan de file)
    read(datafile,config); (lees uit de datafile de variabele config)
    close(datafile); (sluit de datafile weer af)
END;
```

```

BEGIN
bedoeldemach:='N';           {geef de variabele een initiële waarde}
repeat                       {blijf in de volgende lus tot bedoeldemach de}
  BEGIN                       {waarde 'Y' krijgt toegewezen}
  clrscr;
  gotoxy(12,8);
  write('Toets de filenaam in :');
  read(filename);
  filename:=filename+'.FIG'; {voeg aan de gewenste filenaam '.FIG' toe}
  if exist(filename)=true then {als de file bestaat doe dan het volgende}
  BEGIN
    LEESDISKFILE;           {lees de variabele config uit de file}
    clrscr;
    gotoxy(8,6);
    write('Merk : ');
    writeln(config.merk);   {zet het merk v/d machine op het scherm}
    gotoxy(8,8);
    write('Type : ');
    writeln(config.tiepe); {zet het type v/d machine op het scherm}
    gotoxy(8,10);
    write('Bereik eerste geleiding : ');
    writeln(config.uber:8:2); {zet het bereik v/d U-as op het scherm}
    gotoxy(8,12);
    write('Bereik tweede geleiding : ');
    writeln(config.vber:8:2); {evenzo voor de V-as}
    gotoxy(8,14);
    write('Bereik derde geleiding : ');
    writeln(config.wber:8:2); {en voor de W-as}
    gotoxy(6,20);
    write('Is dit de machine die U bedoeld ? (Y/N) ');
    read(kbd,bedoeldemach);
    bedoeldemach:=upcase(bedoeldemach);
  END
else                           {als er geen machine onder de ingetikte}
  BEGIN                           {naam op disk staat}
  clrscr;
  gotoxy(8,10);
  write('Er is geen file aanwezig met de door U opgegeven naam. ');
  gotoxy(8,12);
  write('Toets N om opnieuw te proberen of T om uw machine ');
  gotoxy(8,14);
  write('zelf te gaan tekenen. ');
  read(kbd,bedoeldemach);
  bedoeldemach:=upcase(bedoeldemach);
  if bedoeldemach='T' then {nu gaan we zelf een machine tekenen}
  BEGIN
    clrscr;
    gotoxy(8,8);
    write('Wat is het merk van de machine : ');
    read(merk);
    gotoxy(8,10);
    write('Welk type is de machine : ');
    read(tiepe);
  END
  END

```

```

        assign(variabel,'kinekett.chn');      {koppel het tekenblok}
        chain(variabel);
    END;
END;
until bedoeldemach='Y';

uber:=config.uber;      {wanneer we de bedoelde machine hebben}
vber:=config.vber;     {maken we alle variabelen uit de verzamel-}
wber:=config.wber;     {variabele config gelijk aan de namen die}
startpunt:=config.startpunt; {gebruikt worden in de andere blokken}
twegel:=config.twegel;
np:=config.np;
uas:=config.uas;
vas:=config.vas;
was:=config.was;
draaiing:=config.draaiing;
uuv:=config.uuv;
uvw:=config.uvw;
vuv:=config.vuv;
vvw:=config.vvw;
wuv:=config.wuv;
wvw:=config.wvw;
ucomp:=config.ucomp;
vcomp:=config.vcomp;
wcomp:=config.wcomp;
verb1:=config.verb1;
verb2:=config.verb2;
verb3:=config.verb3;
verb4:=config.verb4;
verb5:=config.verb5;
verb6:=config.verb6;
merk:=config.merk;
tiepe:=config.tiepe;
clrscr;
gotoxy(5,12);
write('Wilt U een modeltekening of');
write(' rechtstreeks de foutentabel zien ? (T/F) ');
read(tekening);
tekening:=upcase(tekening);
if tekening='T' then assign(variabel,'autokett.chn')      {koppel of het}
                    else assign(variabel,'variabel.chn'); {automatisch teken-}
chain(variabel);      {pakket of het blok dat de foutentabel uitgeeft}
END.

```

```
PROGRAM AUTOKINEKETT;
```

```
{DIT PROGRAMMA STEMT VRIJWEL OVEREEN MET HET PROGRAMMA KINEKETT MET DAT }  
{VERSCHIL DAT ALLE INPUTVRAGEN HIER VERVANGEN ZIJN DOOR REEDS BEKENDE }  
{WAARDEN. VOOR VERDERE UITLEG OMTRENT DE WERKING WORDT HIER DAN OOK }  
{VERWEZEN NAAR HET BLOK KINEKETT.PAS }
```

```
($I a:typedef.sys) (include deze files van het pakket Turbo-Graphics)  
($I a:graphix.sys)  
($I a:kernel.sys)  
($I a:windows.sys)
```

```
VAR uber,vber,wber:real;  
VAR startpunt,twegel,np,uas,vas,was:char; (np=nulpunt,uas=x/y/z,twegel=y/n)  
VAR draaiing:real;  
VAR uuv,uvm,vuv,vvm,wuv,wvw:real; (de componenten v/d verbindingvectoren)  
VAR ucomp,vcomp,wcomp:real; (dit zijn de tastercomponenten)  
VAR verb1,verb2,verb3:char;  
Var verb4,verb5,verb6:char;  
Var vb:char;  
VAR ht:^integer;  
VAR xb,yb,xe,ye,keus:real; (begin en eindpunten in de TEKEN procedure)  
VAR richting:char; (richting=h/s/v)  
VAR hs,vs,ss:real; (deze geven het bereik en de schermfactoren)  
VAR unul,vnul,wnul:real; (de nul vectorcomponenten)  
VAR ve,s,d,vhulp,shulp,dhulp:real;(d=hor.richting=1 of-1,s=schuin,ve=verticaal)  
VAR kleiner:char;  
VAR k,i,t,q,l:integer;(k=schaalfactor voor verkleinen,i en t zitten in for-to )  
VAR a,b,c,g,uverpl,vverpl,wverpl:real; (zie proc.meettafel2)  
VAR x1,y1,x2,y2,x3,y3,x4,y4,x5,y5,x6,y6,x7,y7:real;  
VAR pyl,cr:char; (pyl geeft richting hoedje aan op de vectoren)  
VAR nr,f:integer; (zie proc. verbind en proc.fout)  
VAR vector,vraag:boolean; (vector:zie proc.teken,vraag:zie proc.verbind)  
VAR htast,vtast,wtast:real; (zie proc.tastvector)  
VAR zeker:char;  
VAR variabel:file;
```

```
PROCEDURE INITIAL; (hier gaan we in de graphische mode)  
BEGIN  
initgraphic;  
drawborder;  
definewindow(1,0,10,79,199); (geef als eerste window het scherm op)  
defineworld(1,0,1400,1400,0); (definieer de rekengrootte 1400 bij 1400)  
selectworld(1);  
setaspect(1);  
END;
```

```

PROCEDURE MEETTAfel;    (teken de eerste meettafel waarop het meetvolume staat)
BEGIN
  drawline(100,1275,750,1275);
  drawline(750,1275,750,1375);
  drawline(750,1375,100,1375);
  drawline(100,1375,100,1275);
  drawline(100,1275,700,675);
  drawline(700,675,1350,675);
  drawline(1350,675,750,1275);
  drawline(1350,675,1350,775);
  drawline(1350,775,750,1375);
END;

```

```

PROCEDURE MEETVOLUME2;  (teken het meetvolume op de tafel als stippellijnen)
BEGIN
  setlinestyle(2);
  drawline(100,1275,100,625);
  drawline(100,625,750,625);
  drawline(750,625,750,1275);
  drawline(100,625,700,25);
  drawline(700,25,1350,25);
  drawline(1350,25,750,625);
  drawline(1350,25,1350,675);
  drawline(700,25,700,675);
  drawline(100,1275,750,1275);
  drawline(750,1275,1350,675);
  drawline(1350,675,700,675);
  drawline(100,1275,700,675);
  setlinestyle(0);
END;

```

```

PROCEDURE TEKEN (keus:real;vector:boolean;richting:char);
BEGIN
  if vector=true then          {true=gel.vector,false=verbindingsvector}
  BEGIN
    if richting='H' then      {teken de hor.geleidingsvector}
    BEGIN
      xe:=xb+(d#325);        {bepaal de eindpunten tov de beginpunten}
      ye:=yb;
      drawline(xb,yb-8,xb,yb+8);
      drawline(xb,yb-8,xe-(d#12),ye-8);          {teken vector}
      drawline(xb,yb+8,xe-(d#12),ye+8);
      drawline(xe,ye,xe-(d#20),ye-30);          {pijl aan de vector}
      drawline(xe,ye,xe-(d#20),ye+30);
    END;
  END;

```



```

if richting='V' then          {teken de verticale geleidingsvector}
  BEGIN
    xe:=xb;
    ye:=yb-(ve#400);
    drawline(xb-5,yb,xb+5,yb);
    drawline(xb-5,yb,xb-5,ye);
    drawline(xb+5,yb,xb+5,ye);
    drawline(xe,ye,xe-20,ye+(ve#20));
    drawline(xe,ye,xe+20,ye+(ve#20));
  END;
if richting='S' then        {teken de 'schuine' geleidingsvector}
  BEGIN
    xe:=xb+(s#200);
    ye:=yb-(s#200);
    drawline(xb-10,yb,xb+10,yb);
    if s=1 then BEGIN drawline(xb-10,yb,xe-10,ye);
                      drawline(xb+10,yb,xe,ye+10); END
    else BEGIN drawline(xb-10,yb,xe-3,ye-10);
               drawline(xb+10,yb,xe+10,ye); END;
    drawline(xe,ye,xe-(s#35),ye);
    drawline(xe,ye,xe+(s#8),ye+(s#30));
  END;
END
else
  BEGIN
    if richting = 'H' then          {teken de verbindingsvectoren}
      BEGIN
        xe:= xb+(d#keus#hs);
        ye:=yb;
      END;
    if richting = 'V' then
      BEGIN
        ye:= yb-(ve#keus#vs);
        xe:=xb;
      END;
    if richting = 'S' then
      BEGIN
        xe:= xb+(s#keus#ss#0.70710678);
        ye:= yb-(s#keus#ss#0.70710678);
      END;
    drawline(xb,yb,xe,ye);
  END;
  xb:=xe;
  yb:=ye;
END;

PROCEDURE HOED(PYL:CHAR);          {tekent een hoedje aan de vector}
  BEGIN
    if pyl='H' then
      BEGIN
        drawline(xe,ye,xe-(d#((k/2)+1)#15),ye-(d#((k/2)+1)#15));
        drawline(xe,ye,xe-(d#((k/2)+1)#15),ye+(d#((k/2)+1)#15));
      END;
    END;
  END;

```

```

if pyl='S' then
  BEGIN
    drawline(xe,ye,xe+(s*((k/2)+1)*3),ye+(s*((k/2)+1)*15));
    drawline(xe,ye,xe-(s*((k/2)+1)*22),ye);
  END;
if pyl='V' then
  BEGIN
    drawline(xe,ye,xe-(ve*((k/2)+1)*15),ye+(ve*((k/2)+1)*15));
    drawline(xe,ye,xe+(ve*((k/2)+1)*15),ye+(ve*((k/2)+1)*15));
  END;
END;

PROCEDURE SCHRIJF;           {maakt een regel onder op het scherm leeg}
BEGIN                       {en gaat alvast met de cursor onder staan}
  gotoxy(8,25);
  write(' ');
  gotoxy(9,25);
END;

PROCEDURE UVMKRUIS;        {tekent het gekozen UVW model}
BEGIN
  xb:=1225;yb:=1250;TEKEN((65*((k/2)+1)/hs),false,'H');HOED('H');
  drawtextw(xe+(d*20),ye,1,'U');
  xb:=1225;yb:=1250;TEKEN((68*((k/2)+1)/ss),false,'S');HOED('S');
  if twegeel='N' then drawtextw(xe+(s*30),ye-(s*30),1,'V')
    else drawtextw(xe+(s*30),ye-(s*30),1,'W');
  xb:=1225;yb:=1250;TEKEN((78*((k/2)+1)/vs),false,'V');HOED('V');
  if twegeel='N' then BEGIN
    if ve=1 then drawtextw(xe-30,ye-20,1,'W')else drawtextw(xe-30,ye+10,1,'W');
    END
    else BEGIN
    if ve=1 then drawtextw(xe-30,ye-20,1,'V')else drawtextw(xe-30,ye+10,1,'V');
    END;
END;

PROCEDURE START;
BEGIN
  if (startpunt='A') or (startpunt='B') or (startpunt='a')
    or (startpunt='b') then xb:=100 else xb:=750;
  if (startpunt='A') or (startpunt='C') or (startpunt='a')
    or (startpunt='c') then yb:=625 else yb:=1275;
  setlinestyle(2);
  if (startpunt='A') or (startpunt='a') then drawline(100,625,100,1275);
  if (startpunt='C') or (startpunt='c') then drawline(750,625,750,1275);
  setlinestyle(0);
END;

```

```

PROCEDURE TASTTEKEN;
BEGIN
  htast:=ucomp#hs;
  if twege1='N' then
    BEGIN
      vtast:=vcomp#ss;
      wtast:=wcomp#vs;
    END
  else
    BEGIN
      vtast:=wcomp#ss;
      wtast:=vcomp#vs;
    END;
  xe:=xb+(d#htast)+(s#0.7071#vtast);
  ye:=yb-(ve#wtast)-(s#0.7071#vtast);
  drawline(xb,yb,xe,ye);
  drawcircle(xe,ye,0.05);
END;

PROCEDURE VERBIND (vb:char;nr:integer;vraag:boolean);
BEGIN
  if (vb='u') or (vb='U') then
    BEGIN
      if nr=1 then TEKEN(uuv,false,'H') else TEKEN(uvw,false,'H');
    END;
  if (vb='v') or (vb='V') then
    BEGIN
      if twege1='N' then
        BEGIN
          if nr=1 then TEKEN(vuv,false,'S') else TEKEN(vvw,false,'S');
        END
      else
        BEGIN
          if nr=1 then TEKEN(vuv,false,'V') else TEKEN(vvw,false,'V');
        END;
    END;
  if (vb='w') or (vb='W') then
    BEGIN
      if twege1='N' then
        BEGIN
          if nr=1 then TEKEN(wuv,false,'V') else TEKEN(wvw,false,'V');
        END
      else
        BEGIN
          if nr=1 then TEKEN(wuv,false,'S') else TEKEN(wvw,false,'S');
        END;
    END;
END;

```

```

PROCEDURE NUL;                {teken de vector Aou van het nulpunt naar }
BEGIN                          {het begin v/d U-geleiding}
  START;
  TEKEN(unul,false,'H');
  if twegel='N' then TEKEN(vnul,false,'S') else TEKEN(vnul,false,'V');
  if twegel='N' then TEKEN(wnul,false,'V') else TEKEN(wnul,false,'S');
END;

PROCEDURE KETTING;           {tekent de ingevoerde vectorenketting}
BEGIN
  TEKEN(ve,true,'H');
  VERBIND(verb1,1,false);
  VERBIND(verb2,1,false);
  VERBIND(verb3,1,false);
  if twegel='N' then TEKEN(wuv,true,'S') else TEKEN(wuv,true,'V');
  VERBIND(verb4,2,false);
  VERBIND(verb5,2,false);
  VERBIND(verb6,2,false);
  if twegel='N' then TEKEN(wvw,true,'V') else TEKEN(wvw,true,'S');
  TASTTEKEN;
END;

PROCEDURE MEETTAFEL2;        {tekent een grote meettafel onder het meetvolume}
BEGIN                          {om duidelijk te laten zien waar de geleidingen liggen}
  unul:=-uuv-uvw;vnul:=-vuv-vvw;wnul:=-wuv-wvw;
  if (ve=-1) and (twegel='N') then wnul:=wnul-wber;
  if (ve=-1) and (twegel='Y') then vnul:=vnul-wber;
  if ((startpunt='A') or (startpunt='C') or (startpunt='a')
    or (startpunt='c')) and (ve=-1) then
    BEGIN
      if (ve=-1) and (twegel='N') then wnul:=wnul+(650/vs);
      if (ve=-1) and (twegel='Y') then vnul:=vnul+(650/vs);
    END;
  if twegel='N' then
    BEGIN
      if wnul<0 then BEGIN a:=0;b:=1;c:=0;g:=wnul; END;
      if wnul>0 then BEGIN a:=0;b:=1;c:=1;g:=0; END
      else BEGIN a:=0;b:=2;c:=1;g:=0; END;
    END
  else
    BEGIN
      if vnul<0 then BEGIN a:=1;b:=0;c:=0;g:=vnul; END;
      if vnul>0 then BEGIN a:=1;b:=0;c:=1;g:=0; END
      else BEGIN a:=2;b:=0;c:=1;g:=0; END;
    END;
  END;
  uverpl:=abs(unul#hs);
  vverpl:=0.7071#ss#vnul;
  wverpl:=0.7071#ss#wnul;
  x1:=(100-uverpl+(a#wverpl)+(b#vverpl));
  y1:=(1275-(a#wverpl)-(b#vverpl));
  x2:=750+uverpl+(a#wverpl)+(b#vverpl);
  y2:=y1;
  x3:=x2;
  y3:=y2+(c#100)-(g#2#vs);
  x4:=x1;

```

```

y4:=y3;
x5:=700-uverpl-(a#wverpl)-(b#vverpl);
y5:=675+(a#wverpl)+(b#vverpl);
x6:=1350+uverpl-(a#wverpl)-(b#vverpl);
y6:=y5;
x7:=x6;
y7:=y6+(c#100)-(g#2#vs);
drawline(x1,y1,x2,y2);
drawline(x2,y2,x3,y3);
drawline(x3,y3,x4,y4);
drawline(x4,y4,x1,y1);
drawline(x1,y1,x5,y5);
drawline(x5,y5,x6,y6);
drawline(x6,y6,x2,y2);
drawline(x6,y6,x7,y7);
drawline(x7,y7,x3,y3);
END;

```

```

PROCEDURE XYZASSEN;           {bepaal het startpunt voor het xyz assenkruis}

```

```

BEGIN
  if (np='A') or (np='a') then BEGIN xb:=100;yb:=625; END;
  if (np='B') or (np='b') then BEGIN xb:=100;yb:=1275; END;
  if (np='C') or (np='c') then BEGIN xb:=750;yb:=625; END;
  if (np='D') or (np='d') then BEGIN xb:=750;yb:=1275; END;
  if (np='E') or (np='e') then BEGIN xb:=700;yb:=25; END;
  if (np='F') or (np='f') then BEGIN xb:=700;yb:=675; END;
  if (np='G') or (np='g') then BEGIN xb:=1350;yb:=25; END;
  if (np='K') or (np='k') then BEGIN xb:=1350;yb:=675; END;
  if (np='A') or (np='a') or (np='B') or (np='b') or (np='E')
    or (np='e') or (np='F') or (np='f') then d:=1 else d:=-1;
  if (np='A') or (np='a') or (np='B') or (np='b') or (np='C')
    or (np='c') or (np='D') or (np='d') then s:=1 else s:=-1;
  if (np='B') or (np='b') or (np='D') or (np='d') or (np='F')
    or (np='f') or (np='K') or (np='k') then ve:=1 else ve:=-1;
  draaiing:=(d#ve#s);
END;

```

```

PROCEDURE VERKLEIN;          {verkleint het window en daarmee de tekening}

```

```

BEGIN
  SCHRIF;
  write('Verkleinen of vergroten (V/G) of verdergaan (CR) ?');
  {$I-} read(kbd,kleiner);{$I+}
  if (kleiner='v') or (kleiner='V') or (kleiner='g') or (kleiner='G') then
    BEGIN
      if k>=1 then setclippingoff;           {teken ook buiten het window}
      if (kleiner='v') or (kleiner='V') then k:=k+1 else k:=k-1;
    END
  ;

```

```

    if k<0 then BEGIN SCHRIF;
                    write(' ');
                    write(' ');
                    SCHRIF;
                    write('U kunt niet groter tekenen !');
                    {$I-} read(kbd,kleiner);{$I+}
                    k:=k+2;
                END;
    if k>6 then BEGIN SCHRIF;
                    write(' ');
                    write(' ');
                    SCHRIF;
                    write('U kunt niet kleiner tekenen !');
                    {$I-} read(kbd,kleiner);{$I+}
                    k:=k-2;
                END;
    clearscreen;
    redefinewindow(1,6*k,15*k,79-(6*k),199-(15*k));
    selectworld(1);
    selectwindow(1);
    MEETVOLUME2;
    MEETTAFEL2;START;NUL;KETING;UVMKRUIS;
    VERKLEIN;
END;

```

PROCEDURE TOTAAL1;

```

BEGIN
    (HIER BEGINT HET EIGENLIJKE PROGRAMMA)
    if (startpunt='C') or (startpunt='D') or (startpunt='c')
        or (startpunt='d') then d:=-1;
    if (startpunt='A') or (startpunt='a') or (startpunt='B')
        or (startpunt='b') then d:=1;
END;

```

PROCEDURE TUSSEN23;

```

BEGIN
    if (vas='A') or (vas='a') then BEGIN twegel:='N'; s:=1; END;
    if (vas='B') or (vas='b') then BEGIN twegel:='N'; s:=-1; END;
    if (vas='C') or (vas='c') then BEGIN twegel:='Y'; ve:=1; END;
    if (vas='D') or (vas='d') then BEGIN twegel:='Y'; ve:=-1; END;
END;

```

PROCEDURE TUSSEN34;

```

BEGIN
    if (twegel='N') and ((was='A') or (was='a')) then ve:=1;
    if (twegel='N') and ((was='B') or (was='b')) then ve:=-1;
    if (twegel='Y') and ((was='A') or (was='a')) then s:=1;
    if (twegel='Y') and ((was='B') or (was='b')) then s:=-1;
END;

```

```

PROCEDURE TUSSEN45;
BEGIN
  hs:=650/uber;           {bepaal de schermfactor in hor. richting}
  if twegel='N' then ss:=850/vber else vs:=650/vber; {richting afh.v/d keuze}
  if twegel='M' then vs:=650/wber else ss:=850/wber;
  UVWKRUIS;
END;

```

```

PROCEDURE TUSSEN67;
BEGIN
  MEETVOLUME2;
  MEETAFEL2;START;NUL;KETTING; {teken het meetvolume en de vectorenketting}
  VERKLEIN;                    {verklein tot de juiste grootte}
  SCHRIJF;
  write('                      ');
  MEETAFEL2;
END;

```

```

PROCEDURE TUSSEN78;
BEGIN
  dhulp:=d;                 {bewaar de d,s en v waarden van de vectorenketting}
  shulp:=s;                 {in dhulp,shulp en vhulp}
  vhulp:=ve;
  XYZASSEN;
  TEKEN((65*((k/2)+1)/hs),false,'H');HOED('H'); {teken het XYZassenkruis}
  XYZASSEN;                {in het juiste hoekpunt en}
  TEKEN((68*((k/2)+1)/ss),false,'S');HOED('S'); {gebruik daarbij de }
  XYZASSEN;                {schaalfactor k}
  TEKEN((78*((k/2)+1)/vs),false,'V');HOED('V');
  d:=dhulp;                {geef d,s en v de oude waarden terug}
  s:=shulp;
  ve:=vhulp;
END;

```

```

PROCEDURE TOTAALB;
BEGIN
  if (uas='x') or (uas='X') then {en leidt hieruit af waar de V en W}
  BEGIN {geleiding voor staan omdat het een}
    XYZASSEN; {rechtsdraaiend assenkruis moet zijn}
    TEKEN((65*((k/2)+1)/hs),false,'H');
    drawtextw(xe+(d*20),ye+30,1,'X');
    XYZASSEN;
    TEKEN((68*((k/2)+1)/ss),false,'S');
    if (d*ve*s)=1 then drawtextw(xe-15,ye-(s*50),1,'Y')
      else drawtextw(xe-15,ye-(s*50),1,'Z');
    XYZASSEN;
    TEKEN((78*((k/2)+1)/vs),false,'V');
    if (d*ve*s)=1 then drawtextw(xe-35,ye-(ve*30),1,'Z')
      else drawtextw(xe-35,ye-(ve*30),1,'Y');
  END;
END;

```

```

if (uas='y') or (uas='Y') then
  BEGIN
    XYZASSEN;
    TEKEN((65*((k/2)+1)/hs),false,'H');
    drawtextw(xe+(d*20),ye+30,1,'Y');
    XYZASSEN;
    TEKEN((68*((k/2)+1)/ss),false,'S');
    if (d#ve#s)=1 then drawtextw(xe-15,ye-(s*50),1,'Z')
      else drawtextw(xe-15,ye-(s*50),1,'X');
    XYZASSEN;
    TEKEN((78*((k/2)+1)/vs),false,'V');
    if (d#ve#s)=1 then drawtextw(xe-35,ye-(ve*30),1,'X')
      else drawtextw(xe-35,ye-(ve*30),1,'Z');
  END;

```

```

if (uas='z') or (uas='Z') then
  BEGIN
    XYZASSEN;
    TEKEN((65*((k/2)+1)/hs),false,'H');
    drawtextw(xe+(d*20),ye+30,1,'Z');
    XYZASSEN;
    TEKEN((68*((k/2)+1)/ss),false,'S');
    if (d#ve#s)=1 then drawtextw(xe-15,ye-(s*50),1,'X')
      else drawtextw(xe-15,ye-(s*50),1,'Y');
    XYZASSEN;
    TEKEN((78*((k/2)+1)/vs),false,'V');
    if (d#ve#s)=1 then drawtextw(xe-35,ye-(ve*30),1,'Y')
      else drawtextw(xe-35,ye-(ve*30),1,'X');
  END;

```

```
END;
```

```
PROCEDURE TUSSEN89;
```

```

  BEGIN
    gotoxy(60,24);
    write(' ');
    SCHRIJF;
    write('Druk Shft PrtSc');
    delay(1500);
    SCHRIJF;
    delay(3000);
    SCHRIJF;
    write('Toets <CR> voor foutentabel. ');
    read;
    leavegraphic;
  END;

```

{wacht op een return en
{verlaat de grafische mode}

BEGIN

k:=0; (zet k op 0 dus gebruik het hele scherm als window)

INITIAL; (ga in de graphische mode)

selectwindow(1); (kies het scherm als window)

TOTAAL1;

TUSSEN23;

TUSSEN34;

TUSSEN45;

TUSSEN67;

TUSSEN78;

TOTAAL8;

TUSSEN89;

assign(variabel, 'A:variabel.chn');

chain(variabel);

END.