

21 JUNI 1971

KONINKLIJK NEDERLANDS  
METEOROLOGISCH INSTITUUT

De Bilt

Verslagen

V - 232

S.J. Bijlsma

en

B. van Rietschote

Het bepalen van de kortste-vaartijdroute

De Bilt, 1971

Kon. Ned. Meteor. Inst.  
De Bilt



I N H O U D

	pag.
Inleiding	2
1. De stereografische projectie	3
2. De vaartkarakteristiek	8
3. Bepaling van de kortste-vaartijdroute door de invoering van tijdfrenten en toepassing van Pontryagin's maximum principe	11
4. Het ALGOL-programma	16
5. Discussie	21
Literatuur	21

## Inleiding

In de volgende bladzijden wordt ten behoeve van eventuele gebruikers een beschrijving gegeven van een rekenprogramma dat de kortste-vaartijdroute bepaalt voor een gegeven schip tussen twee bepaalde plaatsen op aarde, in ons geval Engels Kanaal en New York. De stationaire Golfstroom wordt bekend verondersteld, evenals de met de tijd veranderende golfhoogten en-richtingen in het te bevaren gebied. Bovendien wordt aangenomen dat de invloed, die deze golven hebben op het gedrag van het schip bekend is. Het te bevaren gebied wordt ter vereenvoudiging van de berekeningen door middel van de stereografische projectie afgebeeld in een plat vlak. Het construeren van de kortste-vaartijdroute geschiedt door het invoeren van tijdfuncties en het toepassen van Pontryagin's maximum principe. Het rekenprogramma is ontwikkeld door De Wit (1968). Aangezien hierin o.a. een andere kaartprojectie wordt gebruikt, moesten er enkele veranderingen worden aangebracht. Het doel dat we bij het schrijven van dit verhaal voor ogen hadden was allereerst een beschrijving te geven van het rekenprogramma, zodat van de mathematische achtergrond slechts de voor een goed begrip strikt noodzakelijke resultaten vermeld zullen worden. Alvorens over te gaan op de bespreking van het rekenprogramma zal eerst enige aandacht worden besteed aan datgene wat aan het bepalen van de kortste-vaartijdroute ten grondslag ligt. Voor nadere bijzonderheden en verdere literatuur wordt men verwezen naar De Wit (1968).

## 1. De stereografische projectie

De stereografische projectie beeldt het oppervlak van de aarde af in een plat vlak, rakend aan het aardoppervlak in een punt S, door aan elk punt P van het aardoppervlak als projectie toe te voegen het snijpunt van de lijn door P en S', het tegenpunt van S, met het projectievlak. Als projectievlak kan men in principe ieder vlak evenwijdig aan het raakvlak kiezen. Al naar gelang S' samenvalt met een pool of op de equator ligt, spreekt men van een polair of equatoriaal stereografische projectie. In ons geval is er sprake van een polair stereografische projectie (met de zuidpool als projectiecentrum) terwijl als projectievlak, het vlak door de breedtecirkel van 60 graden N.B. is genomen (zie fig. 1). Zijn l (lengte) en b (breedte) de geografische coördinaten van een punt op aarde, dan worden de coördinaten van de projectie als poolcoördinaten ten opzichte van de geprojecteerde noordpool (als oorsprong) gegeven. De modulus bedraagt

$$\frac{R}{s} (1 + \cos 30) \operatorname{tg} (45 - b/2)$$

waarin s een schaalfactor is (in ons geval  $1.5 \cdot 10^7$ ) en R de straal van de aarde.

De fasehoek wordt gegeven door de lengte l. We transformeren nu het punt met bovenstaande poolcoördinaten naar een rechthoekig coördinatenstelsel met coördinaten x en y, waarin de y-as evenwijdig is aan de lijn  $l=30$ . Kiezen we de oriëntatie zodanig dat de positieve y-as loopt in de richting van de afnemende breedte en de positieve x-as in de richting van toenemende lengte, dan vinden we, indien de coördinaten van de noordpool  $x_n$  en  $y_n$  zijn, dat de transformatie gegeven wordt door

$$x = x_n + \frac{R}{s d_0} (1 + \cos 30) \operatorname{tg} (45-b/2) \sin (l-30) \quad (1)$$

$$y = y_n + \frac{R}{s d_0} (1 + \cos 30) \operatorname{tg} (45-b/2) \cos (l-30)$$

Hierin is  $d_0$  gelijk aan de roosterafstand (in ons geval 2.12). Door differentiatie volgt uit (1) dat de transformatie lokaal geschreven kan worden als

$$\begin{aligned} dx &= -A(b)R(db \sin (l-30) - dl \cos (l-30) \cos b) \\ dy &= -A(b)R(db \cos (l-30) + dl \sin (l-30) \cos b) \end{aligned} \quad (2)$$

waarin  $A(b) = \frac{1 + \cos 30}{2 s d_0 \cos^2 (45-b/2)}$

Uit (2) volgt dat de transformatie conform is, terwijl bovendien de draaiing van de lokaal orthogonale coördinatenstelsels ten opzichte van elkaar gegeven wordt. De omrekeningsfactor, die gebruikt moet worden om kleine afstanden op aarde om te zetten in afstanden in het (x, y)-vlak is gelijk aan  $d_0 A(b)$ , immers

$$ds^2_{(x,y)\text{-vlak}} = d_0^2 (dx^2 + dy^2) = d_0^2 A^2(b) R^2 (db^2 + dl^2 \cos^2 b) = d_0^2 A^2(b) ds^2_{\text{aarde}}$$

Tenslotte volgt uit (1) voor de inverse transformatie

$$l = \arctg \left( \frac{x-x_n}{y-y_n} \right) + 30$$

$$b = 90 - 2 \arctg \left( a_0 \sqrt{(x-x_n)^2 + (y-y_n)^2} \right)$$

$$\text{waarin } a_0 = \frac{s d_0}{R (1 + \cos 30)}$$

We zullen nu met behulp van bovenstaande aantonen dat de projectie van een grootcirkel wederom een cirkel is. Weliswaar geldt deze eigenschap veel algemener, nl. voor alle cirkels op de bol, maar we zullen ons hier beperken tot de voor ons meest noodzakelijke eigenschappen. Bovendien heeft de hier gevolgde methode het voordeel dat we de vergelijking van de geprojecteerde grootcirkel expliciet vinden. Veronderstel dat de pool van een grootcirkel gekarakteriseerd wordt door de parameters  $p_0$  en  $l_0$ , waarvan  $l_0$  de lengte en  $b_0 = 90 - p_0$  de breedte van de pool is (zie fig. 2). Een punt op de grootcirkel wordt nu gekarakteriseerd door  $l - l_0$  en  $p$ , waarvan  $l$  de lengte is van het punt en  $b = 90 - p$  de breedte. In feite moeten  $l$  en  $p$  beschouwd worden als bolcoördinaten. In de boldriehoek BNP geldt nu

$$\cos (l - l_0) = -\text{ctg } p \text{ ctg } p_0 \quad (3)$$

waarmee de vergelijking van de grootcirkel gevonden is. De parameters  $p_0$  en  $l_0$  worden eenvoudig bepaald door de lengten en breedten van twee punten van de grootcirkel (in ons geval begin en eindpunt van de reis, waartussen we de vaartijd langs de grootcirkelroute willen bepalen) in (3) te substitueren. Combinatie van (1) en (3) geeft de vergelijking van de geprojecteerde grootcirkel. Uit (1) volgt door kwadrateren

$$\text{tg } p/2 = a_0 \sqrt{(x-x_n)^2 + (y-y_n)^2}$$

Stellen we  $l-l_0 = l'$  dan volgt eveneens uit (1)

$$\cos l' = \frac{(x-x_n) \sin (l_0-30) + (y-y_n) \cos (l_0-30)}{\sqrt{(x-x_n)^2 + (y-y_n)^2}}$$

Substitutie in (3) geeft, daar  $\operatorname{tg} p = \frac{2 \operatorname{tg} p/2}{1-(\operatorname{tg} p/2)^2}$ , als vergelijking voor de geprojecteerde grootcirkel

$$2 a_0 \left[ (x-x_n) \sin (l_0-30) + (y-y_n) \cos (l_0-30) \right] = \operatorname{ctg} p_0 \left[ a_0^2 ((x-x_n)^2 + (y-y_n)^2) - 1 \right] \quad (4)$$

We kunnen (4) herschrijven als

$$(x-x_m)^2 + (y-y_m)^2 = r^2$$

met  $x_m = x_n + r \sin p_0 \sin (l_0-30)$ ,  $y_m = y_n + r \sin p_0 \cos (l_0-30)$  en

$$r = (a_0 \cos p_0)^{-1}$$

waarmee de coördinaten van het middelpunt  $(x_m, y_m)$ , alsook de straal van de geprojecteerde grootcirkel gevonden zijn, uiteraard in roostereenheden.

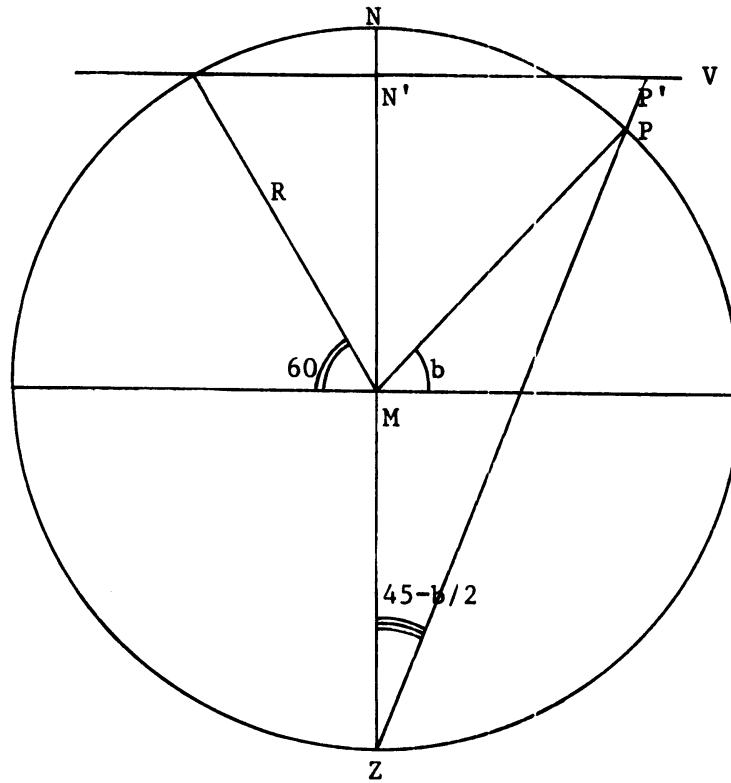


fig. 1. Indien de coördinaten van  $P'$  als poolcoördinaten worden gegeven t.o.v.  $N'$ , dan is de modulus  $N'P' = R (1 + \cos 30) \operatorname{tg} (45 - b/2)$ . In het projectievlak  $V$ , loodrecht op het vlak van tekening wordt nu een rechthoekig coördinatensysteem aangebracht met de  $y$ -as langs de lijn  $1 = 30$ .

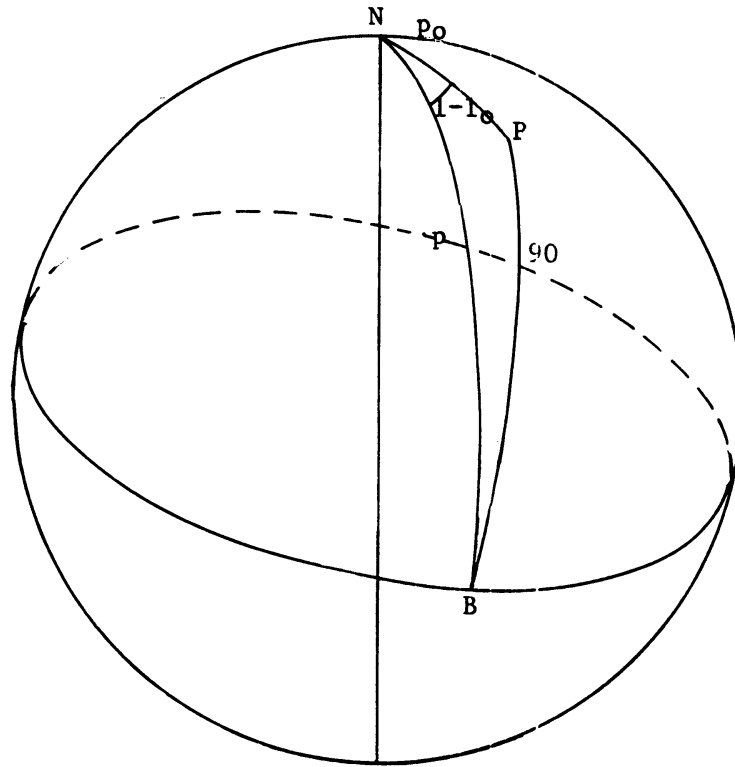


fig. 2.  $P$  is de pool van de grootcirkel met lengte  $l_0$  en breedte  $b_0 = 90 - p_0$ .  
De lengte en breedte van punt  $B$  op de grootcirkel zijn respectievelijk  
 $l$  en  $b = 90 - p$ .



## 2. De vaartkarakteristiek

Onder de vaartkarakteristiek van een schip verstaan we een beschrijving van zijn maximale snelheid als functie van de vaarrichting, bij gegeven (significante) golfhoogte en golfrichting. Daar er theoretisch nog weinig bekend is over het gedrag van een schip in een golfveld, worden dergelijke karakteristieken meestal op experimentele gronden geconstrueerd. Door het golfveld gedurende een bepaalde tijdsinterval stationair te veronderstellen kunnen we aangeven welke afstand, afhankelijk van de richting, in dit tijdsinterval kan worden afgelegd. Nu wordt om de 12 uur een golfkaart verstrekt, waarop bovenstaande golfgegevens zijn vastgelegd, wat tot gevolg heeft dat de tijdstapjes, waarin we de golfsituatie als stationair beschouwen in ons geval 12 uur bedragen.

Geven we nu de positie van het schip aan door een punt S en veronderstellen we golfhoogte en -richting vast, dan kunnen we door een gesloten kromme om S te construeren de in 12 uur maximaal haalbare afstanden weergeven. Voor de eenvoudigheid wordt deze kromme, in navolging van De Wit (1968) benaderd door een ellips met de lange as in de golfrichting, terwijl punt S op de lange as ligt en ten opzichte van het centrum van de ellips verschoven is tegen de golfrichting in, zoals geschetst in fig. 3. Bovendien voert men om zwaar slingeren te voorkomen een verboden vaarsector in d.w.z. een gebied van verboden vaarrichtingen. Wordt deze vaarsector bepaald door de hoeken  $g_1$  en  $g_2$  dan zal men om een punt binnen deze vaarsector te bereiken een bepaalde combinatie van de koersen  $g_1$  en  $g_2$  moeten varen.

Van belang is ook nog het volgende. Veronderstel dat een rechte gegeven is door het punt S die een hoek  $g_0$  maakt met de positieve x-as (zie fig. 3). Gevraagd wordt nu een koers te bepalen zodanig dat de afgelegde afstand geprojecteerd op de gegeven rechte maximaal is. Het is duidelijk dat dit gebeurt door een lijn te construeren, die loodrecht staat op de gegeven rechte en tevens raakt aan de ellips. Het raakpunt bepaalt de gevraagde koers. Indien het raakpunt zich in de verboden sector zou bevinden wordt een methode gevolgd zoals beschreven in fig. 4.

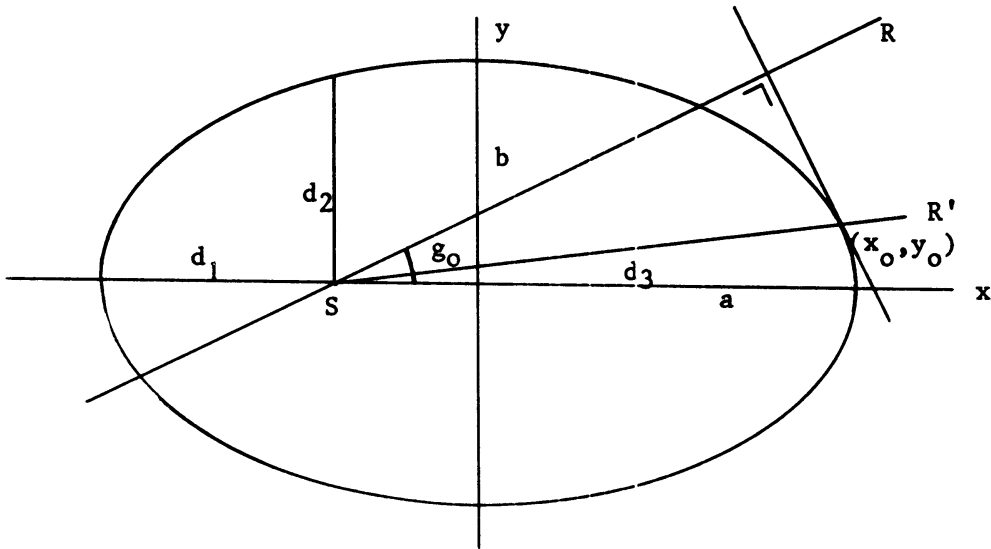


fig. 3. In het rekenprogramma worden van de ellips de lijnstukken  $d_1$ ,  $d_2$  en  $d_3$  ingelezen, zodat de lange as van de ellips gegeven wordt door  $a = \frac{d_1 + d_3}{2}$  en de korte as door  $b = \frac{ad_2}{\sqrt{d_1 d_3}}$

Maakt de gegeven rechte  $R$  een hoek  $g_0$  met de positieve  $x$ -as (we veronderstellen de golfrichting in de richting van de positieve  $x$ -as) dan snijden we de ellips  $\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1$  met de lijn  $y = -x \operatorname{ctg} g_0 + z$  en bepalen de parameter  $z$  zodanig dat de twee snijpunten samenvallen.

We vinden dan als coördinaten van het raakpunt

$$x_0 = \frac{a^2 \cos g_0}{\sqrt{a^2 \cos^2 g_0 + b^2 \sin^2 g_0}} \quad \text{en} \quad y_0 = \frac{b^2 \sin g_0}{\sqrt{a^2 \cos^2 g_0 + b^2 \sin^2 g_0}}$$

zodat de koers bepaald wordt door de lijn  $R'$ .

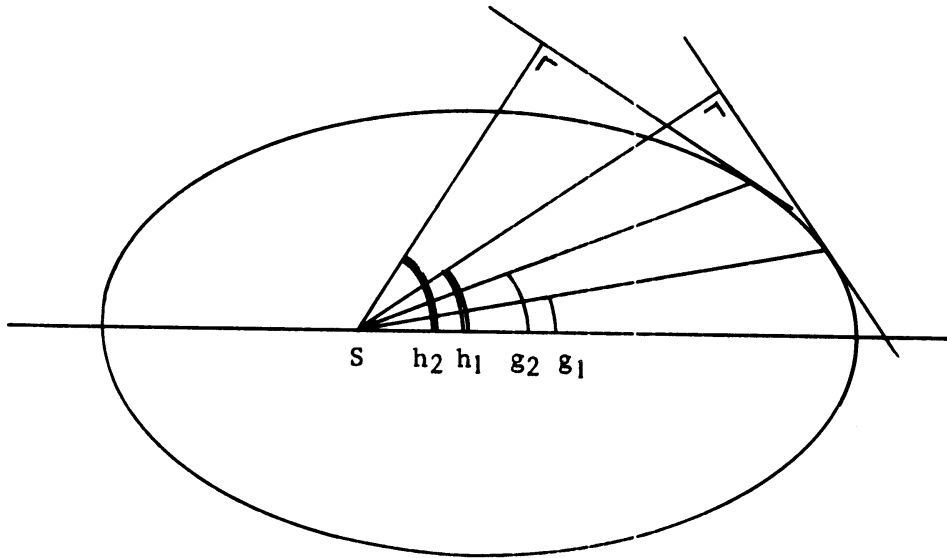


fig. 4. Indien de hoeken  $g_1$  en  $g_2$  de verboden zone karakteriseren, moeten, als begrenzing voor de gegeven richting, de hoeken  $h_1$  en  $h_2$  gegeven worden. Indien de richting  $g_0$  zich nu bevindt tussen  $h_1$  en  $h_2$ , berekenen we de koers, die gevaren moet worden opdat de projectie van de afgelegde afstand in de richting  $g_0$  maximaal is, door een combinatie van de koersen  $g_1$  en  $g_2$  te nemen. De factor, die deze combinatie bepaalt bedraagt

$$\frac{\cos h_2 - \cos g_0}{\cos h_2 - \cos h_1}$$

### 3. Bepaling van de kortste-vaartijdroute door de invoering van tijdfrenten en toepassing van Pontryagin's maximum principe

In dit hoofdstuk zal volstaan worden met het aanroeren van enkele, voor ons belangrijke resultaten uit de theorie van de optimaal geregelde processen. Voor meer informatie wordt men verwezen naar de desbetreffende literatuur (zie Inleiding). We keren nu terug naar het in het eerste hoofdstuk ingevoerde coördinatenstelsel en veronderstellen dat de coördinaten van begin en eindpunt van de reis respectievelijk gegeven worden door  $(x_0, y_0)$  en  $(x_B, y_B)$ . Bovendien nemen we in eerste instantie aan dat in alle roosterpunten van het te bevaren gebied de golfhoogte en-richting om de 12 uur voor de duur van de reis bekend zijn. De waarden van de Golfstroomcomponenten veranderen weinig, zodat per jaar volstaan kan worden met twee Golfstroomkaarten. In de tussenliggende punten van het rooster worden de waarden van Golfstroom en golfveld verkregen door bilineaire interpolatie. Vanuit het punt  $(x_0, y_0)$  worden nu een aantal richtingen uitgezet min of meer symmetrisch ten opzichte van de grootcirkel door  $(x_0, y_0)$  en  $(x_B, y_B)$ . We komen nu toe aan de bespreking van tijdfrenten en extremalen.

Onder een tijdfrent op een tijdstip  $t$  verstaan we in ons geval de verzameling punten, die een schip uitgaande van een bepaald startpunt, door het volgen van alle mogelijke vaarrichtingen, gedurende een tijdsinterval  $t$  maximaal kan bereiken. De route die het schip heeft moeten volgen om vanuit het startpunt een punt op het tijdfrent te bereiken, heet een extremaal. Er bestaat nl. geen route, waarlangs het schip het desbetreffende punt op het tijdfrent eerder zou bereiken. Er is nu een stelling, die zegt, dat als we een extremaal in stukjes verdelen, elk stukje opnieuw een extremaal is. Deze stelling maakt het ons mogelijk tijdfrenten en extremalen op eenvoudige wijze te construeren. Veronderstel nl. dat we twee tijdfrenten hebben  $S(t)$  en  $S(t+\Delta t)$ , respectievelijk op de tijdstippen  $t$  en  $t+\Delta t$  (zie fig. 5). Gesteld nu dat  $s$  een extremaal is, dan zegt de stelling dat het lijnstuk  $PQ$  dat door de tijdfrenten van de extremaal wordt afgesneden, eveneens een extremaal is, m.a.w.  $PQ$  zal lokaal gezien (d.w.z. voor voldoende kleine  $\Delta t$ ) een zodanige richting hebben dat de afstand tussen de tijdfrenten op een zo vlug mogelijke manier wordt afgelegd. Dit zal het geval zijn, indien de snelheid in  $P$ , in de richting  $PQ$ , geprojecteerd op de normaal op  $S(t)$  in  $P$  maximaal is. Deze constructie kan beschouwd worden als een toepassing van Pontryagin's maximum principe, in dit speciale geval. We keren nu weer terug naar ons uitgangspunt nl. het construeren van de kortste-vaartijdroute voor een schip tussen

de punten  $(x_0, y_0)$  en  $(x_B, y_B)$ , gegeven golfhoogte en -richting en Golfstroomcomponenten in de roosterpunten en gegeven een vaartkarakteristiek.

Uitgaande van de gegeven richtingen, die vanuit  $(x_0, y_0)$  zijn uitgezet, kunnen we, na een tijdstapje van 12 uur een eerste tijdfront construeren. Het bepalen van die koers, zodanig dat de projectie van de afgelegde weg op een gegeven richting maximaal is, zoals beschreven in het tweede hoofdstuk, gebeurt in de procedure OPTISTEP. Bij de op deze wijze gevonden punten moeten we, omdat we ons a.h.w. op een bewegend wateroppervlak bevinden door de invloed van de stationaire Golfstroom, uiteraard nog de 12-uurlijkse verplaatsing ten gevolge van deze stroming optellen. Dit gebeurt eveneens in de procedure OPTISTEP. Vervolgens bepalen we in de punten van het nu gevonden eerste tijdfront de normaal richtingen. Dit gebeurt in de procedure NORMAAL. Ten opzichte van deze normaal richtingen worden weer de optimale vaarrichtingen bepaald en een volgend tijdfront wordt gevonden. Deze gang van zaken wordt herhaald tot we een laatste tijdfront hebben gevonden dat het punt  $(x_B, y_B)$  is gepasseerd. Het constateren of op een gegeven ogenblik een tijdfront het punt  $(x_B, y_B)$  is gepasseerd gebeurt in procedure FUIK. Beschouwen we nu het op één na laatste tijdfront en zoeken we hierop het punt Q dat het dichtst bij  $(x_B, y_B)$  ligt, dan kunnen we door terug te zoeken van welk punt P, op het tijdfront daarvoor, het afkomstig is (zoals in fig. 5) en deze handeling te herhalen tot we op het eerste tijdfront zijn teruggekeerd, de extremaal d.w.z. de kortste-vaartijdroute bepalen.

Er kunnen onderweg evenwel nog enkele complicaties optreden. Ten eerste kan de afstand tussen de punten op een tijdfront te groot worden. Er worden dan punten aan toegevoegd. Dit gebeurt in procedure VERFIJN. Bovendien kunnen er punten buiten de grenzen van het te bevaren gebied terecht komen en zullen deze punten geëlimineerd moeten worden. Dit gebeurt in procedure RAND (zie fig. 8). Een andere moeilijkheid die kan ontstaan is de volgende. Veronderstel dat een tijdfout  $S(t)$  op een gegeven moment de vorm heeft zoals geschetst in fig. 6. Het tijdfront van 12 uur later kan nu het verloop van  $S(t+12)$  hebben. Om deze meerwaardigheid te vermijden is de procedure ONTLUSSEN ingevoerd, die de punten van de lus verwijdert. Een andere moeilijkheid, die kan optreden wordt weergegeven in fig. 7. De procedure, die de achterblijvende punten hier verwijdert heet SCHOONMAAK. Het is duidelijk dat door toepassing van deze procedures, punten met een gelijke index op twee opeenvolgende tijdfronten niet noodzakelijk met elkaar corresponderen. Door nu bij de index van een punt te onthouden hoeveel punten ervóór zijn verwijderd of zijn toegevoegd, is het toch mogelijk de kortste-vaartijdroute te construeren. Hierover meer in de programmabeschrijving.

Tenslotte berekent de procedure STAP de vaartijd langs de grootcirkelroute. Hiertoe bepalen we, beginnende in punt  $(x_0, y_0)$ , de raaklijn aan de grootcirkel en snijden deze met de ellips van de vaartkarakteristiek. Het nu gevonden lijnstuk dat de in 12 uur afgelegde afstand voorstelt in de richting van de raaklijn, wordt omgecirkeld tot op de grootcirkel door de procedure FIT. Indien de raaklijn in het verboden vaargebied valt, moeten we weer een combinatie van koersen varen, zoals in het tweede hoofdstuk is vermeld. Rest nog iets te zeggen over het invoeren van subextremalen. Om een koers om tamelijk scherpe landsgrenzen heen te kunnen volgen (zie fig. 8), voegt men aan de normaalrichtingen van een tijdfront nog een aantal richtingen toe, getrokken vanuit het punt R van het tijdfront dat het dichtst bij deze scherpe landsgrens ligt en berekent men met een eventueel kleiner tijdstapje het volgende tijdfront. De extremalen die m.b.v. deze toegevoegde richtingen geconstrueerd worden, noemt men subextremalen. Omdat de vaartkarakteristiek (evenals de Golfstroomkaart) de werkelijke afstand geeft die in 12 uur wordt afgelegd is het duidelijk dat deze afstand m.b.v. de in het eerste hoofdstuk berekende factor  $A(b)$  gereduceerd moet worden, om er mee te kunnen werken in het door ons ingevoerde coördinatenstelsel. Dit hoofdstuk was bedoeld om een globaal idee te krijgen, hoe men te werk gaat bij het bepalen van een kortste-vaartijdroute. Een meer kwantitatieve beschrijving zal in het volgende hoofdstuk gegeven worden, waar aan de hand van een regelbespreking het gebruik van het programma verduidelijkt zal worden.

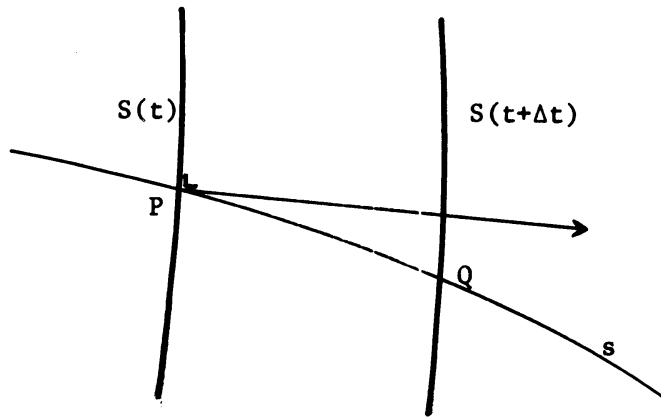


fig. 5.

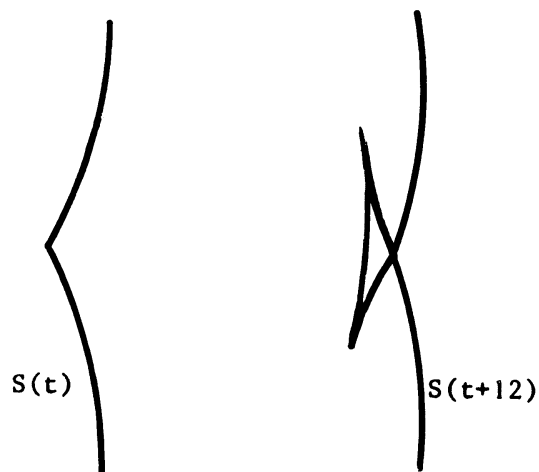


fig. 6.

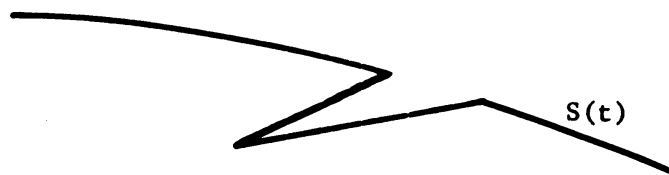


fig. 7.





#### 4. Het ALGOL-programma

Om het aanpassen aan andere situaties te vereenvoudigen leek het ons raadzaam om, in plaats van een stroomdiagram een regelbespreking van het programma te geven. We zullen hier uitgaan van de situatie, zoals geschetst in fig. 8. In het door ons gebruikte coördinatenstelsel had de noordpool de coördinaten (7.5, -8.5). We beginnen onze bespreking daar waar het programma gaat rekenen. De procedures zullen dan, voorzover dit al niet gebeurd is, vanzelf ter sprake komen.

Regelnr.

- 388 ik en im zijn de indices van begin en eindpunt van een tijdfront.
- 389 j telt het aantal tijdfronten. De parameter kk komt voor in procedure LEES(rooster, s) (zie regel 404).
- 390  $x_0$  en  $y_0$  zijn de coördinaten van het beginpunt van de reis.
- 391  $x_B$  en  $y_B$ , de coördinaten van het eindpunt.
- 392 aa, bb en cc zijn respectievelijk de x-en y-coördinaat van het middelpunt van de geprojecteerde grootcirkel en het kwadraat van zijn straal.
- 393 De vaartkarakteristiek wordt uitgelezen (zie fig. 3).
- 394 De cosinussen van de hoeken  $h_1$  en  $h_2$  (zie fig. 4) worden uitgelezen.
- 396/403 De x-en y-componenten van de Golfstroom in de roosterpunten worden in 3 blokken van 10 x 18 stuks ingelezen. Dit betekent dat er door deze constructie, daar het rooster 29 x 18 punten telt, 18 waarden meer worden gevraagd. Hiervoor kunnen gefingeerde getallen genomen worden. De componenten worden vervolgens in de juiste volgorde op de trommel gezet.
- 404 De golfhoogten en -richtingen in de roosterpunten worden ingelezen d.m.v. de procedure LEES(rooster, s) (regel 26), waarin s een vermenigvuldigingsfactor is, zodanig dat  $s = 1$  voor de hoogten en  $s = gr$  voor de richtingen, waardoor de richtingen in radialen worden gegeven t.o.v. de richting van de positieve x-as tegen de wijzers van de klok in. In procedure LEES wordt per keer slechts een blok van 10 x 18 waarden ingelezen t.o.v. een y-as, die bepaald wordt door het punt van het tijdfront met de kleinste x-waarde. Dit punt

wordt bepaald in de procedure ARRMIN (regel 14). Opdat dit punt niet verandert indien LEES voor de tweede keer wordt aangeroepen, is in de procedure LEES de parameter kk ingevoerd. Er is uiteraard vanuit gegaan dat de maximale spreiding in de x-richting van de punten van een tijdfrent kleiner is dan 10 roosterafstanden.

- 405 Zie regel 474/489.
- 406/407 De 10 x 18 x- respectievelijk y-componenten van de Golfstroom worden van de trommel gehaald. Wegens het realkarakter van de arrays beslaat elk element twee geheugenplaatsen.
- 408 UGR[j] en VGR[j] zijn de x- en y-coördinaat van een punt van de grootcirkelroute, bereikt na de j-de tijdstap. In de berekeningsfase worden deze coördinaten weergegeven door xgk en ygk.
- 409/410 gk geeft de helling van de raaklijn aan de grootcirkel in het punt met coördinaten xgk en ygk.
- 411/413 In procedure STAP wordt de afgelegde afstand in de richting van deze raaklijn uitgerekend. In procedure FIT wordt deze afstand omgecirkeld tot op de grootcirkel en de coördinaten van het snijpunt uitgerekend.
- 414/419 Hierin worden, uitgaande van 40 gegeven richtingen, door procedure OPTISTEP de punten van het eerste tijdfrent in eerste instantie uitgerekend.
- 420/426 Door de berekening van regel 411/413 in het aldaar verkregen punt met een nieuwe golfkaart te herhalen en de afgelegde afstanden te middelen wordt een verbeterde grootcirkel stap berekend.
- 427/432 Het in regel 414/419 besprokene wordt in de aldaar verkregen punten herhaald met een nieuwe golfkaart en definitieve punten van het eerste tijdfrent worden berekend.
- 433/438 De x- en y-componenten van het punt van het tijdfrent met index 1 worden opgeslagen in de arrays U[1] en V[1] en op de trommel gezet. De array-elementen met een index vallend buiten de range (ik, im) worden gelijk gemaakt aan -1.
- 439/440 In de buurt van New Foundland (zie fig. 8) moet men overgaan van 12-op 3-uurs tijdstapjes. In verband hiermee is de identifier "grens" ingevoerd. Hierin is d10 de afstand die een schip maximaal in 12 uur kan varen. Een roosterafstand komt overeen met 170 mijl.

- 441 De parameter  $j$  houdt het aantal 3-uurs tijdstappen bij.
- 443/444 Zie regel 530/547.
- 445/459 Indien de  $x$ -coördinaat van het "bovenste" punt van het tijdfront de waarde van "grens" overschreden heeft, gaan we over op 3-uurs tijdstappen en moet de vaartkarakteristiek door 4 gedeeld worden. Door  $\gamma$  gebeurt dit slechts eenmaal. Is aan de voorwaarde van regel 447 niet meer voldaan dan zorgt de boolean  $\epsilon$  ervoor dat de vaartkarakteristiek weer met 4 wordt vermenigvuldigd.  $\Delta$  zorgt er weer voor dat dit slechts eenmaal gebeurt.
- 460/466 Het volgende punt van de grootcirkelstap wordt weer in eerste instantie berekend. De grootcirkel door begin en eindpunt Bishop Rock en Nantucket Shoals gaat over land zodat we de grootcirkel door Bishop Rock en Cape Race ( $x = 13.33437$ ) nemen en vanaf Cape Race rechtstreeks naar het eindpunt varen. In procedure STAP wordt derhalve de vaarrichting voor  $x < 13.33437$  bepaald door de raaklijn aan de grootcirkel en moet de afgelegde weg d.m.v. procedure FIT weer omgecirkeld worden. Voor  $x > 13.33437$  is de richting uiteraard bepaald door de helling van de lijn Cape Race - Nantucket Shoals.
- 467 Zie regel 511/512.
- 468/473 De punten van het volgende tijdfront worden weer in eerste instantie uitgerekend.
- 474/489  $IG[j]$  wordt gelijk gemaakt aan de index van het "bovenste" punt van het  $j$ -de tijdfront, indien vanuit dit punt subextremalen zijn ingevoerd. Dit is het geval indien we ons bij New Foundland bevinden, alsook indien de  $x$ -coördinaat van het "bovenste" punt van het tijdfront groter is dan 4 en de normaal in dit punt een hoek maakt met de positieve  $x$ -as (kloksgewijs) die kleiner is dan 20 graden.
- 490/494 Indien we niet of niet meer met 3-uurs tijdstapjes werken of indien we er 4 achter de rug hebben, worden nieuwe golfgegevens ingelezen.
- 495/502 De definitieve grootcirkelstap wordt uitgerekend en opgeslagen.
- 503/510 De punten van het nieuwe tijdfront worden definitief berekend (zie regel 427/432 en 433/438) en in regel 528/529 op de trommel gezet.
- 511/512 In procedure FUIK wordt nagegaan of het einddoel gepasseerd is. Is dit het geval dan krijgt de boolean  $\alpha$  de waarde true en de reis is afgelopen.

- 513/520 Door de procedures RAND, ONTLUSSEN en SCHOONMAAK kunnen er punten van het tijdfrent zijn verwijderd en een nieuwe indicering zijn aangebracht. Daar dit voor de betreffende punten gebeurt aan de hand van hun coördinaten  $X[i, 2]$  en  $Y[i, 2]$ , kunnen we door deze coördinaten aanvankelijk gelijk te maken aan respectievelijk  $X[i, 1]$  en  $Y[i, 1]$  en ze naderhand weer met elkaar te vergelijken, nagaan welke punten verwijderd zijn. (Zie regel 530/547.)
- 521/526 Hier gebeurt hetzelfde voor de punten die er door procedure VERFIJN bijgekomen zijn, door  $X[i, 0]$ ,  $Y[i, 0]$  en  $X[i, 2]$ ,  $Y[i, 2]$  te beschouwen. Bovendien wordt in regel 525 het punt van het tijdfrent met de kleinste  $x$ -waarde bepaald.
- 530/547 Hierin wordt nagegaan, hoeveel punten vòòr een bepaald punt van het tijdfrent zijn verwijderd of zijn toegevoegd, in verband met het terugzoeken van de kortste-vaartijdroute. Het array  $MU[h, j]$  geeft aan dat vòòr het punt met index  $MU$  van het  $j$ -de tijdfrent,  $h$  punten zijn toegevoegd. Het array  $NU$  geeft analoog aan, hoeveel er verwijderd zijn.
- 548 Indien het einddoel nog niet gepasseerd is, gaan we weer naar het volgende tijdfrent.
- 549/550 In regel 511/512 is geconstateerd dat het  $j$ -de tijdfrent het eindpunt gepasseerd is. In procedure FUIK wordt in het punt  $i_0$  bepaald, dat op het  $(j-1)$ ste tijdfrent ligt en zich het dichtst bij het eindpunt bevindt. Het array-element  $IB[j]$  dat de index van het punt van de kortste-vaartijdroute geeft op het  $(j-1)$ ste tijdfrent wordt hieraan gelijk gemaakt.
- 551 Indien  $IB[j]$  groter is dan de index van het punt, van waaruit sub-extremalen zijn ingevoerd, wordt hij gelijk gemaakt aan deze index.
- 552/554 Indien  $IB[j]$  groter is dan of gelijk is aan de index van het punt, vòòr hetwelk  $f$  punten zijn toegevoegd, dan moeten deze punten worden afgetrokken.
- 555/557 Hier geldt een analoge redenering voor punten, die verwijderd zijn.
- 558/559 De ongecorrigeerde coördinaten van de punten van het desbetreffende tijdfrent worden van de trommel gehaald (dit waren nl. de coördinaten voor dat de handelingen van regel 513/520 en 521/526 plaatsvonden).

- 560/562 De index van het punt van het  $j$ -de tijdfrent, waar de kortste-vaartijdroute door heen gaat wordt uitgeprint, evenals zijn coördinaten.
- 563/564 De hele gang van zaken, vanaf regel 551, wordt herhaald voor het voorgaande tijdfrent.
- 565/574 Hebben we alle tijdfrenten gehad, dan wordt de vaartijd langs de grootcirkelroute uitgeprint. Het aantal 3-uurs tijdstappen wordt bepaald door  $jj$  en het aantal 12-uurs tijdstappen door  $j_0-jj$ . De tijd nodig om de afstand van het laatst bereikte punt van de grootcirkelroute tot het eindpunt te overbruggen wordt bepaald door het in regel 565 aangegeven gedeelte van 12 uur te nemen. Het uitprinten van desbetreffende gegevens voor de kortste-vaartijdroute gebeurt in de procedure FUIK.

## 5. Discussie

In het voorgaande is verondersteld dat de golfhoogten en -richtingen in alle roosterpunten om de 12 uur voor de duur van de reis bekend waren. Dit nu zal voorlopig nog niet het geval zijn. Daarom zullen we moeten werken met de golfkaarten zoals die momenteel met de hand gemaakt worden. Hiermee ontstaan twee problemen. Allereerst geeft een golfkaart de stereografische projectie van het noordelijk gedeelte van de Atlantische Oceaan, waarop d.m.v. krommen (lijnen van gelijke golfhoogte) en pijltjes (golfrichtingen) golfsituaties zijn vastgelegd. Derhalve is het eerste probleem: hoe op een verantwoorde wijze de manueel verkregen golfgegevens om te zetten in golfhoogte en -richting gegevens in alle punten van ons rooster.

Het tweede probleem is minder eenvoudig op te lossen. Het betreft hier de kern van de zaak nl. het bepalen van een kortste-vaartijdroute. Indien de golfgegevens niet voor de duur van de reis bekend zijn is een routevoorspelling, zoals in het voorgaande is gesuggereerd, uitgesloten. Er blijven ons nu twee mogelijkheden open. Ten eerste kunnen we aan de hand van oude golfkaarten kortste-vaartijdroutes bepalen en een verband met de destijds bestaande weerkundige situaties zien te ontdekken.

De andere mogelijkheid is: aan de hand van dagelijkse golfkaarten op een nog nader te bepalen manier toch te proberen kortste-vaartijdroutes te berekenen en deze zowel te vergelijken met de routes, die tot nu toe met de hand worden bepaald als ook met de routes, die achteraf door het rekenprogramma worden berekend.

Mogelijk kan men op deze wijze, ondanks het gemis aan een voorspeld golfveld, toch een zinvolle methode ontwikkelen om tot een kortste-vaartijdroutebepaling te komen. Een en ander zal nader onderzocht worden.

### Literatuur:

de Wit, C., 1968.

Mathematical treatment of optimal ocean ship routeing (diss. Delft)

```

1  begin   comment KNMI-710126-BIJLS-RIEB-scheepsrouting-
2          evaluatie van kortste-vaartijd-route en grootcirkelroute
3          volgens methode de Wit;
4          integer i,j,h,k,ik,im,f,g,i0,i1,j0,j1,ik1,im1,kk,jj,ii,yas,skip;
5          real  a,b,pi,dx,dy,x0,y0,xb,yb,gr,l,m,n,p,q,r,s,t,xii,yii,xhi,yhi
6          ,xhh,yhh,det,min,dmin,pp,qq,rr,ss,xk,yk,unit,prdx,prdy,d10,grens
7          ,aa,bb,cc,gk,sgk,xgk,ygk,cosgk,singk,dx0,dy0;
8          boolean alfa,beta,gamma,delta,eps;
9          integer array NU,MU[1:40,1:25],IB,IG[1:25];
10         array SXR,SYR,HZR,TZR[0:9,0:17],X,Y[0:80,0:2],
11         U,V[0:80],D,COSPSIG[1:3,0:30],DE,COSG[1:3],DXGR,DYGR,SING[1:2],
12         UGR,VGR[0:25];
13
14         real procedure arrmin(A,m); array A; integer m;
15         begin   comment arrmin bepaalt x-minimum van de punten van een
16                 tijdfrent;
17                 real a;
18                 a:=A[ik,2];
19                 m:=0;
20         next:   m:=m+1;
21                 if m>im-ik then goto end;
22                 if A[ik+m,2]<a then a:=A[ik+m,2];
23                 goto next;
24         end:    arrmin:=a;
25         end arrmin;
26
27         procedure LEES(rooster,s); real s; array rooster;
28         begin   ii:=(-1)kk;
29                 if ii<0 then
30                 begin   yas:=if j=1 then entier(x0) else
31                         if j=2 then entier(arrmin(X,m)) else
32                         entier(min)
33                 end;
34                 for h:=0 step 1 until yas-1 do
35                 for k:=0 step 1 until 17 do
36                 skip:=read;
37                 for h:=yas step 1 until yas+9 do
38                 for k:=0 step 1 until 17 do
39                 rooster[h-yas,k]:=s*read;
40                 for h:=yas+10 step 1 until 28 do
41                 for k:=0 step 1 until 17 do
42                 skip:=read;
43                 kk:=kk+1;
44         end LEES;

```

```

45
46 procedure NORMAAL(X,Y,ii,jj,aa); array X,Y; integer ii,jj;
47 real aa;
48 begin real dh,eh,dt,et,bb;
49 if ii<im then
50 begin dh:=X[ii+1,jj]-X[ii,jj];
51 eh:=Y[ii,jj]-Y[ii+1,jj];
52 if abs(dh)<abs(eh) then aa:=arctan(dh/eh) else
53 aa:=pi*sign(dh)*sign(eh)/2-arctan(eh/dh)
54 end;
55 if ii>ik then
56 begin dt:=X[ii,jj]-X[ii-1,jj];
57 et:=Y[ii-1,jj]-Y[ii,jj];
58 if abs(dt)<abs(et) then bb:=arctan(dt/et) else
59 bb:=pi*sign(dt)*sign(et)/2-arctan(et/dt)
60 end;
61 if ii>ik^ii<im then aa:=(aa+bb)/2;
62 if ii=im then aa:=bb;
63 end NORMAAL;
64
65 procedure OPTISTEP(xs,ys,a); real xs,ys,a;
66 begin real fp,fpc,fq,fqc,tz,hz,tz0,tz1,tz00,tz01,tz10,tz11,
67 hz1,hz2,ai,bi,ei,copsi,sipsi,dx1,dy1, costz,sintz,ia;
68 h:=entier(xs);
69 k:=entier(ys);
70 fp:=xs-h; fpc:=1-fp;
71 fq:=ys-k; fqc:=1-fq;
72 h:=h-yas;
73 dx:=fqc*(fpc*SXR[h,k]+fp*SXR[h+1,k])+
74 fq*(fpc*SXR[h,k+1]+fp*SXR[h+1,k+1]);
75 dy:=fqc*(fpc*SYR[h,k]+fp*SYR[h+1,k])+
76 fq*(fpc*SYR[h,k+1]+fp*SYR[h+1,k+1]);
77 hz:=(fqc*(fpc*HZR[h,k]+fp*HZR[h+1,k])
78 +fq*(fpc*HZR[h,k+1]+
79 fp*HZR[h+1,k+1]))*2;
80 if hz>300 then goto DELFT;
81 tz00:=TZR[h,k]; tz01:=TZR[h,k+1];
82 tz10:=TZR[h+1,k]; tz11:=TZR[h+1,k+1];
83 A1: if tz00-tz10>pi then begin tz10:=tz10+2*pi; goto A1 end;
84 A2: if tz10-tz00>pi then begin tz00:=tz00+2*pi; goto A2 end;
85 A3: if tz01-tz11>pi then begin tz11:=tz11+2*pi; goto A3 end;
86 A4: if tz11-tz01>pi then begin tz01:=tz01+2*pi; goto A4 end;
87 tz0:=fpc*tz00+fp*tz10; tz1:=fpc*tz01+fp*tz11;

```



```

88      A5:  if tz0-tz1>pi then begin tz1:=tz1+2×pi; goto A5 end;
89      A6:  if tz1-tz0>pi then begin tz0:=tz0+2×pi; goto A6 end;
90      tz:=fq×tz0+fq×tz1;
91      costz:=cos(tz); sintz:=sin(tz);
92      copsi:=cos(a-tz); sipsi:=sin(a-tz);
93      h:=entier(hz/10);
94      fp:=hz/10-h; fpc:=1-fp;
95      for k:=1,2,3 do DE[k]:=D[k,h]×fpc+D[k,h+1]×fp;
96      for k:=1,2 do
97      COSG[k]:=fpc×COSPSIG[k,h]+fp×COSPSIG[k,h+1];
98      a1:=(DE[1]+DE[3])/2;
99      e1:=DE[1]-a1;
100     b1:=a1×DE[2]/sqrt(DE[1]×DE[3]);
101     if copsi>COSG[1]∨copsi<COSG[2] then
102     begin  p:=a1×copsi; q:=b1×sipsi;
103           r:=sqrt(p×p+q×q);
104           dx1:=a1×p/r-e1; dy1:=b1×q/r;
105           dx:=dx+dx1×costz-dy1×sintz;
106           dy:=dy+dx1×sintz+dy1×costz;
107           goto DELFT
108     end;
109     for k:=1,2 do
110     begin  p:=a1×COSG[k];
111           SING[k]:=sign(sipsi)×sqrt(1-COSG[k]×COSG[k]);
112           q:=b1×SING[k]; r:=sqrt(p×p+q×q);
113           DXGR[k]:=a1×p/r; DYGR[k]:=b1×q/r
114     end;
115     fp:=(COSG[1]-copsi)/(COSG[1]-COSG[2]);
116     fpc:=1-fp;
117     dx1:=fp×DXGR[2]+fpc×DXGR[1]-e1;
118     dy1:=fp×DYGR[2]+fpc×DYGR[1];
119     dx:=dx+dx1×costz-dy1×sintz;
120     dy:=dy+dx1×sintz+dy1×costz;
121     DELFT: begin  real procedure IA(w1,w2); real w1,w2;
122                IA:=pi/2-2×arctan(sqrt((w1-7.5)2+(w2+8.5)2)/
123                ((1+cos(pi/6))×637817800/(15n6×2.12)));;
124                comment IA berekent de breedte van de plaats;
125                la:=IA(xs,ys);
126                s:=(1+cos(pi/6))/
127                (2×cos(pi/4-la/2)×cos(pi/4-la/2));
128                unit:=1852/(2.12×15n4);

```

```

129      comment unit geeft aan, hoeveel rooster-eenheden
130      een zeemijl is op 60 graden NB;
131      prdx:=sXunitXdx; prdy:=sXunitXdy;
132      la:=LA(xs+prdx/2,ys+prdy/2);
133      s:=(1+cos(pi/6))/
134      (2Xcos(pi/4-la/2)Xcos(pi/4-la/2));
135      dx:=sXunitXdx; dy:=sXunitXdy;
136      end
137  end OPTISTEP;
138
139  procedure ONTLUSSEN;
140  begin   i:=ik;
141          h:=i+2;
142  LL1:   xii:=X[i+1,2]-X[i,2];
143          yii:=Y[i+1,2]-Y[i,2];
144  LL2:   xhi:=X[h+1,2]-X[i,2];
145          yhi:=Y[h+1,2]-Y[i,2];
146          xhh:=X[h+1,2]-X[h,2];
147          yhh:=Y[h+1,2]-Y[h,2];
148          det:=xiiXyhh-yiiXxhh;
149          l:=(xhiXyhh-yhiXxhh)/det;
150          m:=(xiiXyhi-yiiXxhi)/det;
151          if l>0^l<1^m>0^m<1 then
152          begin   for k:=i+1 step 1 until im-h+i do
153                  begin   X[k,0]:=X[k+h-1,0];
154                          Y[k,0]:=Y[k+h-1,0];
155                          X[k,2]:=X[k+h-1,2];
156                          Y[k,2]:=Y[k+h-1,2]
157                  end;
158                  im:=im-h+1;
159                  i:=i+1;
160                  h:=i+2;
161                  if h<im then goto LL1 else goto L0;
162          end;
163          h:=h+1;
164          if h<=i+15^h<im then goto LL2;
165          i:=i+1;
166          h:=i+2;
167          if h<im then goto LL1;
168  L0:
169  end ONTLUSSEN;
170

```

```

171      procedure SCHOONMAAK;
172      begin   for i:=ik step 1 until im do
173      L1:    begin   xii:=X[i,2]-X[i,0];
174              yii:=Y[i,2]-Y[i,0];
175              for h:=ik step 1 until im do
176              begin   if h=i-1\h=i then goto L2;
177                      xhi:=X[h,2]-X[i,0];
178                      yhi:=Y[h,2]-Y[i,0];
179                      xhh:=X[h,2]-X[h+1,2];
180                      yhh:=Y[h,2]-Y[h+1,2];
181                      det:=xii*yhh-yii*xhh;
182                      l:=(xhi*yhh-yhi*xhh)/det;
183                      m:=(xii*yhi-yii*xhi)/det;
184                      if l>1\m>0\m<1 then
185                      begin   for k:=i step 1 until im-1 do
186                              begin   X[k,0]:=X[k+1,0];
187                                      Y[k,0]:=Y[k+1,0];
188                                      X[k,2]:=X[k+1,2];
189                                      Y[k,2]:=Y[k+1,2]
190                              end;
191                              im:=im-1;
192                              goto L1
193                      end;
194      L2:    end;
195      end;
196      if ( im-ik)>20 then h:=20 else h:=im-ik;
197      for i:=ik+1 step 1 until ik+h do
198      begin
199      L3:    if Y[i,2]>Y[ik,2] then
200              begin   for k:=i step 1 until im-1 do
201                      begin   X[k,0]:=X[k+1,0];
202                              Y[k,0]:=Y[k+1,0];
203                              X[k,2]:=X[k+1,2];
204                              Y[k,2]:=Y[k+1,2]
205                      end;
206                      im:=im-1;
207                      goto L3
208              end;
209      end;
210      for i:=im-h step 1 until im-1 do
211      begin
212      L4:    if Y[i,2]<Y[im,2] then

```

```

213         begin   for k:=i step 1 until im-1 do
214             begin   X[k,0]:=X[k+1,0];
215                     Y[k,0]:=Y[k+1,0];
216                     X[k,2]:=X[k+1,2];
217                     Y[k,2]:=Y[k+1,2]
218             end;
219             im:=im-1;
220             goto L4
221         end
222     end;
223 end SCHOONMAAK;
224
225 procedure RAND;
226 begin
227 A7:   if Y[ik,2]>9 then begin ik:=ik+1; goto A7 end;
228       if X[im,2]<13 then
229       begin
230 A8:   if Y[im,2]<1 then
231       begin im:=im-1; goto A8 end;
232       end;
233       if X[im,2]≥13∧X[im,2]<14 then
234       begin
235 A9:   if Y[im,2]<5.2 then
236       begin im:=im-1; goto A9 end;
237       end;
238       if X[im,2]≥14∧X[im,2]<21 then
239       begin
240 A10:  if Y[im,2]<4.5 then
241       begin im:=im-1; goto A10 end;
242       end;
243       if X[im,2]≥21 then
244       begin
245 A11:  if Y[im,2]<5 then
246       begin im:=im-1; goto A11 end;
247       end
248 end RAND;
249
250 procedure STAP;
251 begin   real fp, fpc, fq, fqc, tz, hz, tz0, tz1, tz00, tz01, tz10, tz11, ac,
252         ec, bc, st, cofi, sifi, dx1, dy1, costz, sintz, la;
253         h:=entier(xgk);
254         k:=if ygk>0 then entier(ygk) else 0;

```

```

255      fp:=xgk-h; fpc:=1-fp;
256      fq:=ygk-k; fqc:=1-fq;
257      h:=h-yas;
258      st:=(fqc*(fpcXSXR[h,k]+fpXSXR[h+1,k])
259      +fq*(fpcXSXR[h,k+1]+fpXSXR[h+1,k+1]))*cosgk+
260      (fqc*(fpcXSYR[h,k]+fpXSYR[h+1,k])+fq*(fpcXSYR[h,k+1]+
261      fpXSYR[h+1,k+1]))*singk;
262      hz:=(fqc*(fpcXHZR[h,k]+fpXHZR[h+1,k])
263      +fq*(fpcXHZR[h,k+1]+
264      fpXHZR[h+1,k+1]))*2;
265      if hz>300then
266      begin l:=0; goto ZWET end;
267      tz00:=TZR[h,k]; tz01:=TZR[h,k+1];
268      tz10:=TZR[h+1,k]; tz11:=TZR[h+1,k+1];
269      AS1: if tz00-tz10>pi then
270      begin tz10:=tz10+2*pi; goto AS1 end;
271      AS2: if tz10-tz00>pi then
272      begin tz00:=tz00+2*pi; goto AS2 end;
273      AS3: if tz01-tz11>pi then
274      begin tz11:=tz11+2*pi; goto AS3 end;
275      AS4: if tz11-tz01>pi then
276      begin tz01:=tz01+2*pi; goto AS4 end;
277      tz0:=fpc*tz00+fp*tz10; tz1:=fpc*tz01+fp*tz11;
278      AS5: if tz0-tz1>pi then begin tz1:=tz1+2*pi; goto AS5 end;
279      AS6: if tz1-tz0>pi then begin tz0:=tz0+2*pi; goto AS6 end;
280      tz:=fqc*tz0+fq*tz1;
281      costz:=cos(tz); sintz:=sin(tz);
282      h:=entier(hz/10);
283      fp:=hz/10-h; fpc:=1-fp;
284      for k:=1,2,3 do
285      DE[k]:=fpc*D[k,h]+fp*D[k,h+1];
286      for k:=1,2 do
287      COSG[k]:=fpcxCOSPSIG[k,h]+fpxCOSPSIG[k,h+1];
288      ac:=(DE[1]+DE[3])/2; ec:=DE[1]-ac;
289      bc:=ac*DE[2]/sqrt(DE[1]*DE[3]);
290      cofi:=cos(gk-tz); sifi:=sin(gk-tz);
291      p:=bc*bc*cofi; q:=ac*ac*sifi;
292      r:=ec*sifi; r:=r*r;
293      s:=p*cofi; t:=q*sifi;
294      l:=(-p*ec+ac*bc*sqrt(t-r+s))/(t+s);
295      dx1:=l*cofi; dy1:=l*sifi;
296      for i:=1,2 do

```

```

297         begin   p:=acxacxcosg[i];
298               SING[i]:=sign(sifi)*sqrt(1-cosg[i]*cosg[i]);
299               q:=bcxbcxsing[i];
300               r:=sqrt(pxcosg[i]+qxsing[i]);
301               DXGR[i]:=p/r; DYGR[i]:=q/r
302         end;
303         if DXGR[2]<dx1+ec^dx1+ec<DXGR[1] then
304           l:=((DXGR[1]-ec)*DYGR[2]-(DXGR[2]-ec)*DYGR[1])/
305             ((DYGR[2]-DYGR[1])*cofi+(DXGR[1]-DXGR[2])*sifi);
306   ZWET:   dx:=(1+st)*cosgk; dy:=(1+st)*singk;
307         begin   real procedure LA(w1,w2); real w1,w2;
308               LA:=pi/2-2*arctan(sqrt((w1-7.5)^2+(w2+8.5)^2)/
309                 ((1+cos(pi/6))*637817800/(15*6*2.12)));
310               la:=LA(xgk,ygk);
311               s:=(1+cos(pi/6))/
312                 (2*cos(pi/4-la/2)*cos(pi/4-la/2));
313               unit:=1852/(2.12*15*4);
314               prdx:=s*xunit*dx; prdy:=s*xunit*dy;
315               la:=LA(xgk+prdx/2,ygk+prdy/2);
316               s:=(1+cos(pi/6))/
317                 (2*cos(pi/4-la/2)*cos(pi/4-la/2));
318               dx:=s*xunit*dx; dy:=s*xunit*dy;
319         end LA;
320   end STAP;
321
322   procedure FIT(a,b,c); real a,b,c;
323   begin   real l,x1;
324           l:=sqrt(dx*dx+dy*dy);
325           x1:=(l*(a-UGR[j-1])+
326             l*(VGR[j-1]-b)*sqrt(4*c-l*x1))/(2*c);
327           ygk:=VGR[j-1]-(l*(1+2*(UGR[j-1]-a)*x1)/
328             (2*(VGR[j-1]-b)));
329           xgk:=UGR[j-1]+x1;
330   end FIT;
331
332   procedure FUIK;
333   begin   real d;
334           dmin:=1000;
335           i0:=0;
336           for i:=ik step 1 until im do
337             begin   dx:=X[i,0]-xb; dy:=Y[i,0]-yb;
338                   d:=sqrt(dx*dx+dy*dy);

```

```

339         if d<dmin then begin dmin:=d; i0:=i end
340         end;
341         i1:=i0-1;
342         rr:=(xb-X[i0,0])*(Y[i0,2]-Y[i0,0])-
343         (yb-Y[i0,0])*(X[i0,2]-X[i0,0]);
344     CONTR: pp:=(xb-X[i0,0])*(Y[i1,0]-Y[i0,0])-
345         (yb-Y[i0,0])*(X[i1,0]-X[i0,0]);
346         qq:=(xb-X[i0,2])*(Y[i1,2]-Y[i0,2])-
347         (yb-Y[i0,2])*(X[i1,2]-X[i0,2]);
348         ss:=(xb-X[i1,0])*(Y[i1,2]-Y[i1,0])-
349         (yb-Y[i1,0])*(X[i1,2]-X[i1,0]);
350         if pp*qq<0^rr*ss<0 then goto HOERA;
351         if i1=i0-1 then begin i1:=i0+1; goto CONTR end;
352         goto NOGNIE;
353     HOERA: j1:=j-1;
354         alfa:=true;
355         f:=i0;
356         CARRIAGE(3); PRINTTEXT(⟨reisduur:⟩); SPACE(15);
357         pp:=pp/(pp-qq);
358         k:=entier((j1-jj)/2+jj/8);
359         d:=((j1-jj)+jj/4-2*k+pp)*12;
360         ABSFIXT(5,0,k); PRINTTEXT(⟨  dagen  ⟩);
361         ABSFIXT(5,1,d); PRINTTEXT(⟨  uren  ⟩);
362         CARRIAGE(4);
363         PRINTTEXT(⟨punten baanƿromme:⟩); CARRIAGE(2);
364         goto LBB;
365     NOGNIE: NLCR;
366     LBB:
367     end FUIK;
368
369     procedure VERFIJN;
370     begin i:=ik;
371     A20: xii:=X[i,0]-X[i+1,0];
372         yii:=Y[i,0]-Y[i+1,0];
373         if xii*xii+yii*yii>0.0625 then
374         begin for h:=im step -1 until i+1 do
375             begin X[h+1,0]:=X[h,0];
376                 Y[h+1,0]:=Y[h,0]
377             end;
378         X[i+1,0]:=X[i,0]-xii/2;
379         Y[i+1,0]:=Y[i,0]-yii/2;
380         im:=im+i;

```

```

381             i:=i+1;
382             if i<im then goto A20
383             end else
384             begin i:=i+1; if i<im then goto A20 end
385     end VERFIJN;
386
387     pi:=3.141592653590; gr:=pi/180;
388     ik:=0; im:=40;
389     j:=1; kk:=1;
390     x0:=read; y0:=read;
391     xb:=read; yb:=read;
392     aa:=read; bb:=read; cc:=read;
393     for h:=0 step 1 until 30 do for k:=1,2,3 do D[k,h]:=read;
394     for h:=0 step 1 until 30 do for k:=1,2 do COSPSIG[k,h]:=read;
395     for g:=1 step 1 until 3 do
396     begin   for h:=0 step 1 until 9 do
397           for k:=0 step 1 until 17 do
398           begin   SXR[h,k]:=read;
399                   SYR[h,k]:=read;
400           end;
401           outarray(drum,20000+(g-1)*360,SXR); hold(SXR);
402           outarray(drum,30000+(g-1)*360,SYR); hold(SYR)
403     end;
404     LEES(HZR,1); LEES(TZR,gr);
405     beta:=true;
406     inarray(drum,20000+yas*36,SXR); hold(SXR);
407     inarray(drum,30000+yas*36,SYR); hold(SYR);
408     UGR[0]:=xgk:=x0; VGR[0]:=ygk:=y0;
409     gk:=arctan(-(xgk-aa)/(ygk-bb));
410     cosgk:=cos(gk); singk:=sin(gk);
411     STAP;
412     dx0:=dx; dy0:=dy;
413     FIT(aa,bb,cc);
414     for i:=ik step 1 until im do
415     begin   a:=(40-2*i)*gr;
416           OPTISTEP(x0,y0,a);
417           X[i,1]:=x0+dx;
418           Y[i,1]:=y0+dy;
419     end;
420     LEES(HZR,1); LEES(TZR,gr);
421     gk:=arctan(-(xgk-aa)/(ygk-bb));
422     cosgk:=cos(gk); singk:=sin(gk);

```



```

423     STAP;
424     dx:=(dx+dx0)/2; dy:=(dy+dy0)/2;
425     FII(aa,bb,cc);
426     UGR[j]:=xgk; VGR[j]:=ygk;
427     for i:=ik step 1 until im do
428     begin   NORMAAL(X,Y,i,1,a);
429             OPTISTEP(X[i,1],Y[i,1],a);
430             X[i,2]:=(X[i,1]+x0+dx)/2;
431             Y[i,2]:=(Y[i,1]+y0+dy)/2;
432     end;
433     if im#80 then for i:=im+1 step 1 until 80 do U[i]:=V[i]:=-1;
434     for i:=ik step 1 until im do
435     begin   X[i,0]:=U[i]:=X[i,2]; Y[i,0]:=V[i]:=Y[i,2] end;
436     delta:=gamma:=eps:=true;
437     outarray(drum,0,U); hold(U);
438     outarray(drum,2x81,V); hold(V);
439     d10:=D[1,0];
440     grens:=14-1.1xd10/170;
441     jj:=0;
442     im1:=im;
443     for h:=1 step 1 until 40 do
444     MU[h,1]:=NU[h,1]:=100;
445     TIODFR: j:=j+1;
446     if X[im1,2]>grens then
447     begin   if X[im1,2]≤14v(X[im1,2]>14^(jj<4v(jj>4vjj<8)))
448             then jj:=jj+1 else eps:=false;
449             if gamma then
450             for h:=0 step 1 until 30 do
451             for k:=1,2,3 do D[k,h]:=D[k,h]/4;
452             gamma:=false
453     end;
454     if !eps then
455     begin   if delta then
456             for h:=0 step 1 until 30 do
457             for k:=1,2,3 do D[k,h]:=4xD[k,h];
458             delta:=false
459     end;
460     gk:=if xgk<13.33437 then arctan(-(xgk-aa)/(ygk-bb)) else
461     arctan((yb-5.2449134784)/(xb-13.3343696151));
462     cosgk:=cos(gk); singk:=sin(gk);
463     STAP;
464     dx0:=dx; dy0:=dy;

```

```

465      xgk:=xgk+dx; ygk:=ygk+dy;
466      if xgk<13.33437 then FIT(aa,bb,cc);
467      alfa:=false;
468      for i:=ik step 1 until im do
469      begin   NORMAAL(X,Y,i,0,a);
470            OPTISTEP(X[i,0],Y[i,0],a);
471            X[i,1]:=X[i,0]+dx;
472            Y[i,1]:=Y[i,0]+dy;
473      end;
474      IG[j-1]:=100;
475      if X[im,0]>13.6^beta then begin beta:=false; goto A16 end;
476      if a>-20*gr^X[im,0]>4.0 then
477  A16:  begin   f:=15;
478            if im>65 then f:=80-im;
479            IG[j-1]:=im;
480            for i:=im+1 step 1 until im+f do
481            begin   a:=a-4*gr;
482                    X[i,0]:=X[im,0];
483                    Y[i,0]:=Y[im,0];
484                    OPTISTEP(X[i,0],Y[i,0],a);
485                    X[i,1]:=X[i,0]+dx;
486                    Y[i,1]:=Y[i,0]+dy;
487            end;
488            im:=im+f;
489      end;
490      if X[im,2]<=grensvjj=4vjj=8 then
491      begin   LEES(HZR,1); LEES(TZR,gr);
492            inarray(drum,20000+yas*36,SXR); hold(SXR);
493            inarray(drum,30000+yas*36,SYR); hold(SYR);
494      end;
495      gk:=if xgk<13.33437 then arctan(-(xgk-aa)/(ygk-bb)) else
496      arctan((yb-5.2449134784)/(xb-13.3343696151));
497      cosgk:=cos(gk); singk:=sin(gk);
498      STAP;
499      dx:=(dx+dx0)/2; dy:=(dy+dy0)/2;
500      xgk:=UGR[j-1]+dx; ygk:=VGR[j-1]+dy;
501      if xgk<13.33437 then FIT(aa,bb,cc);
502      UGR[j]:=xgk; VGR[j]:=ygk;
503      if ik>0 then for i:=0 step 1 until ik-1 do U[i]:=V[i]:=-1;
504      for i:=ik step 1 until im do

```

```

505     begin   NORMAAL(X,Y,i,1,a);
506           OPTISTEP(X[i,1],Y[i,1],a);
507           U[i]:=X[i,2]:=(X[i,1]+X[i,0]+dx)/2;
508           V[i]:=Y[i,2]:=(Y[i,1]+Y[i,0]+dy)/2;
509     end;
510     if im#80 then for i:=im+1 step 1 until 80 do U[i]:=V[i]:=-1;
511     if j>13 then FUIK;
512     if alfa then goto AFGEL;
513     for i:=ik step 1 until im do
514     begin X[i,1]:=X[i,2]; Y[i,1]:=Y[i,2] end;
515     for i:=ik step 1 until im-1 do
516     begin if Y[i,2]<Y[i+1,2] then goto A14 end;
517     goto A15;
518 A14:   ONTLUSSEN;
519       SCHOONMAAK;
520 A15:   RAND;
521       ik1:=ik; im1:=im;
522       X[im1,2]:=X[im,2]; X[im1,1]:=X[im,1]; X[im1,0]:=X[im,0];
523       for i:=ik step 1 until im do
524       begin X[i,0]:=X[i,2]; Y[i,0]:=Y[i,2] end;
525       min:=arrmin(X,m);
526       VERFIJN;
527       h:=k:=i:=ik1; f:=g:=0;
528       outarray(drum,4*(j-1)*81,U); hold(U);
529       cutarray(drum,2*(2*j-1)*81,V); hold(V);
530 A22:   if X[h,1]=X[i,2]/X[i,2]=X[k,0] then
531     begin   h:=h+1; k:=k+1; i:=i+1;
532           if i<im1 then goto A22 else goto A23
533     end;
534     if X[h,1]#X[i,2]/X[i,2]=X[k,0] then
535     begin h:=h+1; g:=g+1; NU[g,j]:=i; goto A22 end;
536     if X[h,1]#X[i,2]/X[i,2]#X[k,0] then
537     begin   h:=h+1; g:=g+1;
538           NU[g,j]:=i;
539           f:=f+1;
540           MU[f,j]:=k;
541           k:=k+1;
542           goto A22
543     end;
544     if X[h,1]=X[i,2]/X[i,2]#X[k,0] then
545     begin f:=f+1; MU[f,j]:=k; k:=k+1; goto A22 end;

```

```

546 A23:   for h:=g+1 step 1 until 40 do NU[h,j]:=100;
547       for h:=f+1 step 1 until 40 do MU[h,j]:=100;
548       if j<25 then goto TIJDFR else goto EINDE;
549 AFGEL: j0:=j:=j-1;
550       IB[j]:=i0;
551 CCO:   if IB[j]>IG[j] then IB[j]:=IG[j];
552       f:=40;
553 CC1:   if IB[j]>MU[f,j] then begin IB[j]:=IB[j]-f; goto CC2 end else
554       begin f:=f-1; if f>0 then goto CC1 else goto CC2 end;
555 CC2:   g:=40;
556 CC3:   if IB[j]>NU[g,j] then begin IB[j]:=IB[j]+g; goto CC4 end else
557       begin g:=g-1; if g>0 then goto CC3 else goto CC4 end;
558 CC4:   inarray(drum,4*(j-1)*81,U); hold(U);
559       inarray(drum,2*(2*j-1)*81,V); hold(V);
560       NLCR; PRINTTEXT(⟨ib[⟨j⟩]; ABSFIXT(2,0,j); PRINTTEXT(⟨]=⟨j⟩);
561       ABSFIXT(2,0,IB[j]); SPACE(10);
562       ABSFIXT(2,3,U[IB[j]]); ABSFIXT(2,3,V[IB[j]]); NLCR;
563       j:=j-1;
564       if j>0 then begin IB[j]:=IB[j+1]; goto CCO end else
565       begin   p:=(j0-jj+jj/4+(xb-UGR[j0])/(UGR[j0+1]-UGR[j0]))/2;
566           k:=entier(p);
567           q:=(p-k)*24;
568           CARRIAGE(3);
569           PRINTTEXT(⟨duur grootcirkelroute: ⟨j⟩);
570           ABSFIXT(5,0,k); PRINTTEXT(⟨  dagen  ⟨j⟩);
571           ABSFIXT(5,1,q); PRINTTEXT(⟨  uren  ⟨j⟩
572       end;
573 EINDE:
574 end

```