

KONINKLIJK NEDERLANDS
METEOROLOGISCH INSTITUUT

D e B i l t

WETENSCHAPPELIJK RAPPORT

W.R. 72-11

S.J. Bijlsma en B.van Rietschote

Over het optimaal routeren van schepen
III. Een vergelijkend onderzoek (vervolg)

De Bilt, 1972

Publikatienummer: K.N.M.I. W.R. 72-11 (I)

Deel I zie W.R. 72-1

Deel II zie W.R. 72-2

U.D.C.: 551.509.58

I N H O U D

	pag.
Summary	2
Inleiding	3
1. De oplossingsmethode	4
2. Resultaten van het vergelijkend onderzoek	20
Literatuur	32
Bijlage 1: Het stroomdiagram	35
Bijlage 2: Het ALGOL-programma	41

Summary

The problem of minimal-time ship routing can be treated in two different ways: one way whereby one makes use of networks to compute the least time track and another way where the calculus of variations or optimal control theory is involved. In this report some methods resulting from the second way are discussed and some disadvantages are pointed out. Because of these a new method is introduced.

Moreover results of this method are compared with results of the manual and numerical time front methods. As in the previous reports the practical aspects of the problem will be emphasized more than the theoretical ones. Finally the computerprogram and a flow diagram are given.

Inleiding

Er zijn in de loop der jaren verschillende methoden ontwikkeld om met behulp van een rekenmachine de kortste-vaartijd route te bepalen voor een schip tussen twee bepaalde plaatsen op aarde, gegeven de golfsituatie en gegeven het gedrag van het schip in een golfveld. Deze methoden kan men ruwweg onderscheiden in discrete methoden (methoden, die gebruik maken van netwerken, die het gebied tussen begin en eindpunt geheel of gedeeltelijk bedekken) en continue methoden (methoden, die gebruik maken van de variatie rekening of de theorie van optimaal geregelde processen, zie Bijlsma en Van Rietschote (1972 I), waaraan we als methode die gebruik maakt van een netwerk nog toe kunnen voegen Zoppoli (1972)). We beperken ons hier tot de laatste groep. Alvorens nader in te gaan op de methoden uit deze groep eerst nog enkele algemene opmerkingen.

We veronderstellen voor de eenvoudigheid dat het schip een onbeperkte manoeuvreerbaarheid heeft. Dit heeft tot gevolg dat we ons, bij de afleiding van de uitgangsvergelijkingen, kunnen beperken tot de variatierekening, terwijl we anders gebruik hadden moeten maken van het algemenere Maximum Principe van Pontryagin (1962). Beperkte manoeuvreerbaarheid kan het gevolg zijn van o.a.:

- a. Verboden vaarrichtingen in verband met te grote schommelingen bij bepaalde golfhoogten. Hierover is weinig bekend.
- b. Oceaanstromingen ten gevolge waarvan het schip per tijdseenheid grotere verplaatsingen ondervindt dan ten gevolge van zijn eigen snelheid.

Met geen van beide gevallen houden we dus rekening. Keren we nu terug naar de methoden van de tweede groep dan kan men deze verdelen in methoden, waarbij men de kortste-vaartijd route door iteratie wil bepalen en tijdfront methoden d.w.z. methoden, waarbij men, uitgaande van het startpunt, de verzameling punten bepaalt, die in een bepaald tijdsinterval hoogstens bereikbaar zijn.

In dit rapport wordt een methode gepresenteerd, nauw verwant met beide voorgaande oplossingsmethoden die aanleiding is geweest tot dit vergelijkend onderzoek. De methode is in verschillende gevallen getoetst en de uitkomsten hiervan zijn vergeleken met de resultaten van zowel de numerieke als de manuele tijdfronten methode (Bijlsma en Van Rietschote, 1972 II). De methode wordt in het eerste hoofdstuk besproken, waar bovendien een overzicht gegeven wordt van de andere oplossingsmethoden. De resultaten komen in hoofdstuk 2 aan de orde en het ALGOL-programma wordt als bijlage toegevoegd.

1. De oplossingsmethode

We beginnen met een korte herhaling van de afleiding van de uitgangsvergelijkingen. Hierbij gaan we uit van een rechthoekig coördinatenstelsel met coördinaten x en y , waarin de snelheid van het schip gegeven wordt door $V = V(x,y,p,t)$. Hierin geeft $p = p(t)$ de hoek aan, die de snelheidsvector maakt met de positieve x -as. Het probleem is nu de stuurfunctie $p = p(t)$ zodanig te bepalen, dat de vaartijd T van een vast startpunt op tijdstip $t = 0$ naar een vast eindpunt op tijdstip $t = T$ minimaal is. Daartoe gaan we uit van de integraal

$$I = \int_0^T (1 + \lambda \phi_1 + \mu \phi_2) dt \quad (1)$$

waar

$$\phi_1 = \dot{x} - V \cos p = 0 \quad (2)$$

$$\phi_2 = \dot{y} - V \sin p = 0 \quad (3)$$

de bewegingsvergelijkingen van het schip zijn (de punt in (2) en (3) geeft differentiatie naar de tijd aan). De parameters $\lambda(t)$ en $\mu(t)$ zijn Lagrange multiplicatoren. Worden de variaties van $x = x(t)$, $y = y(t)$ en $p = p(t)$ gegeven door respectievelijk $\delta x(t)$, $\delta y(t)$ en $\delta p(t)$ dan vinden we, indien de aankomsttijd in het vaste eindpunt met δT wordt gevarieerd, als eerste variatie van (1)

$$\delta I = \delta T + [\lambda \delta x + \mu \delta y]_0^T - \int_0^T (\phi_3 \delta x + \phi_4 \delta y + \phi_5 \delta p) dt \quad (4)$$

Noodzakelijke voorwaarden voor een extremaal worden gevonden uit $\delta I = 0$. De coëfficiënten van δx , δy en δp geven de Euler vergelijkingen

$$\phi_3 = \dot{\lambda} + (\lambda \cos p + \mu \sin p) V_x = 0 \quad (5)$$

$$\phi_4 = \dot{\mu} + (\lambda \cos p + \mu \sin p) V_y = 0 \quad (6)$$

$$\phi_5 = \vec{\lambda} \cdot \vec{V}_p = 0 \quad (7)$$

waar

$$\vec{\lambda} = (\lambda, \mu)$$

$$\vec{V}_p = (V_p \cos p - V \sin p, V_p \sin p + V \cos p)$$

De index in (5), (6) en (7) geeft de differentiatie naar de desbetreffende variabele aan. Het verband tussen de variaties van de vaartijd en de plaatscoördinaten in het vaste eindpunt wordt gegeven door

$$x(T + \delta T) + \delta x(T + \delta T) = x(T)$$

$$y(T + \delta T) + \delta y(T + \delta T) = y(T)$$

Hieruit volgt voor kleine waarden van δT

$$\dot{x}(T) \delta T + \delta x(T) = 0 \quad (8)$$

$$\dot{y}(T) \delta T + \delta y(T) = 0 \quad (9)$$

Substitutie van (8) en (9) in (4) geeft na het nulstellen van de coëfficiënt van δT , daar $\delta x(0) = \delta y(0) = 0$,

$$(\vec{\lambda} \cdot \vec{V})_T = 1 > 0 \quad (10)$$

waarin $\vec{V} = (V \cos p, V \sin p)$. Gezien de homogeniteit van (5), (6) en (7) is van (10) alleen het teken belangrijk, waaruit blijkt dat de hoek tussen $\vec{\lambda}$ en \vec{V} scherp is. Het probleem van het optimaal routeren, althans in dit geval, bestaat uit het oplossen van het stelsel vergelijkingen (2), (3), (5), (6), (7) en (10). Hiervoor zijn verschillende methoden ontwikkeld. We zullen een kort overzicht geven.

a. Als eerste zullen we de methoden van Bleick en Faulkner (1965) bespreken. Hierin worden in het vertrekpunt startwaarden voor $\alpha = \arctan \left(\frac{\mu(0)}{\lambda(0)} \right)$ en de vaartijd T gekozen. Met behulp van deze startwaarden wordt het stelsel vergelijkingen (2), (3), (5), (6), (7) en (10) geïntegreerd. Een op deze wijze bepaalde kromme zal in het algemeen niet door het eindpunt (x_E, y_E) gaan, maar door bv. een punt $(x(T), y(T))$. We leiden nu een verband af tussen variaties in α en T en de hiermee gepaard gaande coördinaatverschillen $(\Delta x(T) = x_E - x(T), \Delta y(T) = y_E - y(T))$ tussen de bij α en T behorende coördinaten van het eindpunt $(x(T), y(T))$ en het vaste eindpunt (x_E, y_E) . We veronderstellen hiertoe dat λ_1, μ_1 en λ_2, μ_2 twee lineair onafhankelijke oplossingen zijn van de geadjungeerde vergelijkingen (5) en (6) en beschouwen nu $\lambda = \lambda_1 \cos \alpha + \mu_1 \sin \alpha$ en $\mu = \lambda_2 \cos \alpha + \mu_2 \sin \alpha$ met $\lambda(0) = \cos \alpha$ en $\mu(0) = \sin \alpha$. Om een verband tussen variaties in p en α te vinden herschrijven we (7) als

$$p = \arctan \left(\frac{\mu}{\lambda} \right) + \arctan \left(\frac{V_P}{V} \right) \quad (11)$$

Differentiatie van (11) geeft na substitutie van λ en μ

$$\delta p = \frac{R^2 (\lambda_1 \mu_2 - \mu_1 \lambda_2)}{\Lambda^2 (V^2 + 2V_P^2 - V V_{PP})} \delta \alpha \quad (12)$$

waarin $\Lambda = (\lambda^2 + \mu^2)^{\frac{1}{2}}$ en $R = (V^2 + V_p^2)^{\frac{1}{2}}$.

We willen nu een relatie vinden tussen de variaties $\delta x(T)$ en $\delta y(T)$ van de coördinaten $x(T)$ en $y(T)$ en de variatie van de richting α . Daartoe variëren we

$$\int_0^T (\lambda_i \phi_1 + \mu_i \phi_2) dt = 0 \quad (i = 1, 2)$$

waarmee we analoog aan (4) vinden, daar λ_1, μ_1 en λ_2, μ_2 voldoen aan (5) en (6), indien we T constant houden

$$\lambda_i(T) \delta x(T) + \mu_i \delta y(T) = \int_0^T \lambda_i (V_p \cos p - V \sin p) + \mu_i (V_p \sin p + V \cos p) \delta p dt \quad (i = 1, 2)$$

Hieruit volgt met behulp van (12)

$$\begin{aligned} \delta x(T) &= -J \mu(T) \delta \alpha \\ \delta y(T) &= J \lambda(T) \delta \alpha \end{aligned} \quad (13)$$

waarin

$$J = \frac{1}{\lambda_1(T)\mu_2(T) - \lambda_2(T)\mu_1(T)} \int_0^T \frac{R^3 (\lambda_1 \mu_2 - \mu_1 \lambda_2)^2}{\Lambda^3 (V^2 + 2V_p^2 - VV_{pp})} dt$$

Variatie van de vaartijd geeft bovendien

$$\begin{aligned} \delta x(T) &= \Delta x(T) - \dot{x}(T) \delta T \\ \delta y(T) &= \Delta y(T) - \dot{y}(T) \delta T \end{aligned} \quad (14)$$

zodat we na substitutie van (14) in (13) vinden

$$\begin{aligned} \dot{x}(T) \delta T - J \mu(T) \delta \alpha &= \Delta x(T) \\ \dot{y}(T) \delta T + J \lambda(T) \delta \alpha &= \Delta y(T) \end{aligned} \quad (15)$$

Met behulp van (15) worden uit de coördinaatverschillen $\Delta x(T)$ en $\Delta y(T)$ variaties in α en T en daarmee nieuwe startwaarden bepaald, die tot doel hebben deze coördinaatverschillen te corrigeren. De iteratie naar opeenvolgende trajectoriën wordt voortgezet, tot dat deze fouten acceptabel zijn.

- b. Een andere oplossingsmethode (Marks, 1968) vinden we door uit vergelijkingen (5), (6) en (7) de Lagrange multiplicatoren λ en μ te elimineren en zo een enkele vergelijking voor de stuurfunctie $p = p(t)$ af te leiden.

Deze vergelijking wordt de stuurvergelijking genoemd. Het berekenen van de route gebeurt op de volgende manier. In het vertrekpunt kiezen we een startwaarde $p(0)$ voor $p(t)$ en integreren vervolgens voorwaarts in de tijd. We gebruiken nu bijvoorbeeld de x-coördinaat van het eindpunt om het integreren te beëindigen en de y-coördinaat om de deviatie van het gewenste eindpunt vast te stellen. De informatie hierover kan weer gebruikt worden om de startwaarde $p(0)$ te verbeteren.

- c. We komen nu toe aan meer algemene methoden voor het oplossen van optimaliseringsproblemen. Voor een literatuuroverzicht wordt men verwezen naar Lee en Markus (1967). We zullen hier de methode van Bryson en Denham (1962) in een gewijzigde vorm toepassen. Hierbij gaan we weer uit van (4) en veronderstellen dat we een kromme hebben gevonden, die begin- en eindpunt verbindt, waarlangs $\lambda(t)$ en $\mu(t)$ voldoen aan (5) en (6). Uit $\delta I = 0$ (we veronderstellen $T = \text{constant}$) vinden we een verband tussen de variaties van de eindwaarden van x en y en de variatie in de stuurfunctie

$$\lambda(T)\delta x(T) + \mu(T)\delta y(T) = \int_0^T (\vec{\lambda} \cdot \vec{V}_p) \delta p \, dt \quad (16)$$

Een verband tussen de variaties van deze eindwaarden van x en y en de variatie van de vaartijd wordt gegeven door (zie (8) en (9))

$$\delta x(T) + V \cos p \, \Delta T = 0 \quad (17)$$

$$\delta y(T) + V \sin p \, \Delta T = 0$$

We noteren de variatie van de vaartijd hier i.v.m. (23) als ΔT . Combinatie van (16) en (17) geeft

$$\Delta T = - \frac{\int_0^T (\vec{\lambda} \cdot \vec{V}_p) \delta p \, dt}{(\vec{\lambda} \cdot \vec{V})_T} \quad (18)$$

Het is duidelijk dat indien we de optimale vaarroute door iteratie willen verkrijgen, we een bepaalde beperking aan de grootte van $\delta p(t)$ moeten stellen. We nemen aan dat

$$\Delta p^2 = \int_0^T \delta p^2(t) \, dt \quad (19)$$

klein genoeg gekozen is voor dit doel zodat we eventueel $\delta p(t)$ nog kleiner kunnen kiezen in gevoelige gebieden en iets groter in minder gevoelige gebieden. Opdat bij een bepaalde keuze van $\delta p(t)$ de afname ΔT

maximaal is beschouwen we weer een Lagrange variatieprobleem door m.b.v. (18) en (19) te schrijven

$$\Delta T = \int_0^T \left[- \frac{(\vec{\lambda} \cdot \vec{v}_p)}{(\vec{\lambda} \cdot \vec{v})_T} \delta p + v \delta p^2 \right] dt - v \Delta p^2$$

Hierin is v een Lagrange multiplicator. De variatie van ΔT wordt, indien we Δp^2 constant veronderstellen, gegeven door

$$\delta(\Delta T) = \int_0^T \left[- \frac{(\vec{\lambda} \cdot \vec{v}_p)}{(\vec{\lambda} \cdot \vec{v})_T} + 2v \delta p \right] \delta^2 p dt \quad (20)$$

Uit het nulstellen van de coëfficiënt van de tweede variatie $\delta^2 p(t)$ in (20) volgt

$$\delta p(t) = \frac{(\vec{\lambda} \cdot \vec{v}_p)}{2v(\vec{\lambda} \cdot \vec{v})_T} \quad (21)$$

Door substitutie van (21) in (19) vinden we

$$v = \pm \frac{1}{2\Delta p} \left(\int_0^T \frac{(\vec{\lambda} \cdot \vec{v}_p)^2}{(\vec{\lambda} \cdot \vec{v})_T^2} dt \right)^{\frac{1}{2}} \quad (22)$$

Het plusteken moet gebruikt worden indien we T willen minimaliseren, het minteken indien we een maximum zoeken. Substitutie van (22) in (21) en vervolgens van (21) in (18) geeft

$$\frac{\Delta T}{\Delta p} = - \frac{1}{(\vec{\lambda} \cdot \vec{v})_T} \left(\int_0^T (\vec{\lambda} \cdot \vec{v}_p)^2 dt \right)^{\frac{1}{2}} \quad (23)$$

Een nieuwe stuurfunctie wordt nu geconstrueerd door

$$\begin{array}{l} p(t) \\ \text{nieuw} \end{array} = \begin{array}{l} p(t) \\ \text{oud} \end{array} + \delta p(t)$$

waar $\delta p(t)$ gegeven wordt door (21). Dit gehele proces wordt herhaald totdat de "gradiënt" $\frac{\Delta T}{\Delta p}$ voldoende klein geworden is.

Er zij nog opgemerkt dat een analogon van bovenstaande procedure met succes toegepast kan worden in optimaliseringsproblemen betreffende het bepalen van maxima of minima van niet lineaire functies van meerdere variabelen. Literatuur over dit gebied van de optimalisering vindt men in Wilde en Brightler (1967) en Powell (1970).

Tegen de zo juist besproken methoden zijn, in verband met het doel waarvoor ze hier zouden moeten dienen (nl. het operationeel routeren van schepen) wel enige bezwaren aan te voeren o.a.

1. Het minimum dat men vindt behoeft geen absoluut minimum te zijn. Men zou derhalve over een voorprogramma moeten beschikken, dat een eerste indicatie geeft over de ligging van de route (een en ander gebeurt bij Marks (1968)).
2. Er kunnen convergentieproblemen ontstaan, die in individuele gevallen eenvoudig opgelost zouden kunnen worden, voor de oplossing waarvan moeilijk algemene voorschriften gegeven kunnen worden.

Het ligt dus voor de hand om naar methoden te zoeken die deze nadelen niet hebben. Hierbij komen we gemakkelijk op methoden die gebruik maken van tijdfrenten. Op de praktische aspecten hiervan komen we in het volgende punt terug.

Rest nog te vermelden dat Faulkner (1964) een vergelijkend onderzoek heeft gewijd aan de eerder besproken methoden van Bleick en Faulkner (1965) en Bryson en Denham (1962) waarvan de laatste in feite berust op het fundamentele lemma dat $\vec{\lambda} \cdot \vec{V}_p = 0$ langs een extremaal.

- d. We zullen hier in het kort ingaan op de meetkundige aspecten van het voorgaande en daarmee een nieuwe oplossingsmethode introduceren. We beschouwen weer vgl. (4). Onder voorwaarde dat aan vgl. (5) en (6) voldaan was hebben we afgeleid dat het verband tussen de variaties in de eindwaarden van $x(t)$ en $y(t)$ en de variatie in de stuurfunctie bij constante vaartijd voor een kromme tussen begin- en eindpunt gegeven wordt door

$$\lambda(T)\delta x(T) + \mu(T)\delta y(T) = \int_0^T (\vec{\lambda} \cdot \vec{V}_p) \delta p(t) dt$$

Het is duidelijk dat indien deze kromme een extremaal is dit overgaat in

$$\lambda(T)\delta x(T) + \mu(T)\delta y(T) = 0 \quad (24)$$

Veronderstellen we dat we hier te maken hebben met een veld van extremalen, allemaal afkomstig vanuit het vertrekpunt, dan wordt de richting van de raaklijn aan de kromme S, die punten met dezelfde vaartijd op de verschillende extremalen E verbindt (zie fig. 1), gevonden door $\lambda(T)$ en $\mu(T)$ te elimineren uit de lineaire, homogene vgl. (24) en (7). Men noemt deze richting de transversaalrichting en de kromme S transversaal of tijdfront.

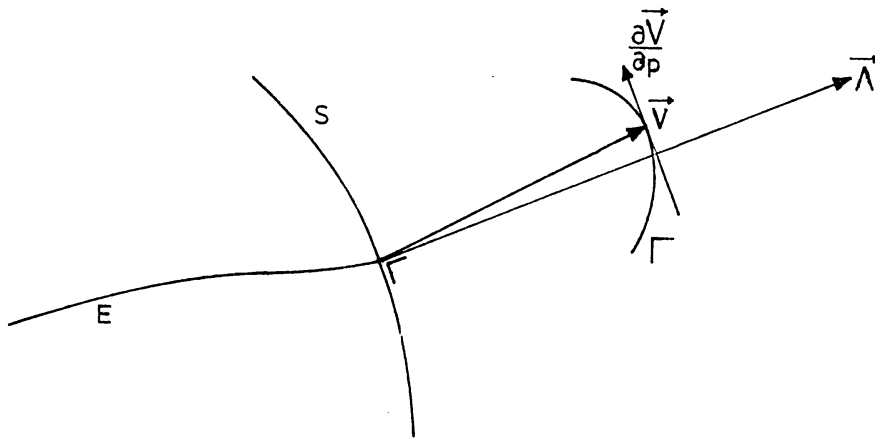


fig. 1. Constructie van de optimale vaarrichting \vec{V} .

De vector $\vec{\lambda} = (\lambda, \mu)$ staat dus loodrecht op het tijdfront. Men kan gebruikmakend van deze tijdfronten op een zeer eenvoudige wijze m.b.v. een rekenmachine extremalen bepalen door de constructie van fig. 1 toe te passen, waar de relatie tussen $\vec{\lambda}$, \vec{V} en $\frac{d\vec{V}}{dp}$ gegeven is (de kromme Γ geeft de snelheid als functie van de vaarrichting). Daartoe kiezen we een aantal startrichtingen $\vec{\lambda}(0) = (\lambda(0), \mu(0))$ en bepalen de punten die in de hierbij behorende optimale vaarrichtingen $\vec{V}(0)$ liggen en na een tijdsinterval Δt bereikt kunnen worden. Deze punten vormen een eerste tijdfront. Een tweede tijdfront wordt geconstrueerd door in de punten van het eerste tijdfront normaal richtingen te bepalen en weer de hierbij behorende optimale vaarrichtingen. Indien punten op een tijdfront te ver uit elkaar komen te liggen, kan men punten tussenvoegen. De constructie van tijdfronten wordt voortgezet totdat men een tijdfront heeft gevonden wat het eindpunt gepasseerd is. Door op het één na laatste tijdfront het punt te zoeken dat het dichtst bij het eindpunt ligt en vervolgens op het tijdfront daarvoor het punt van waaruit het afkomstig is (indien het een tussengevoegd punt is heeft men de keuze uit twee punten, waaraan men het kan toevoegen) kan men, indien men dit tot op

het eerste tijdfront voortzet, de optimale vaarroute bepalen. Deze meetkundige constructie is bijzonder eenvoudig toe te passen, zolang de extremalen elkaar niet snijden, zoals hier is verondersteld. Hebben we te maken met elkaar snijdende extremalen, dan krijgen we de gevallen zoals geschetst in fig. 2.

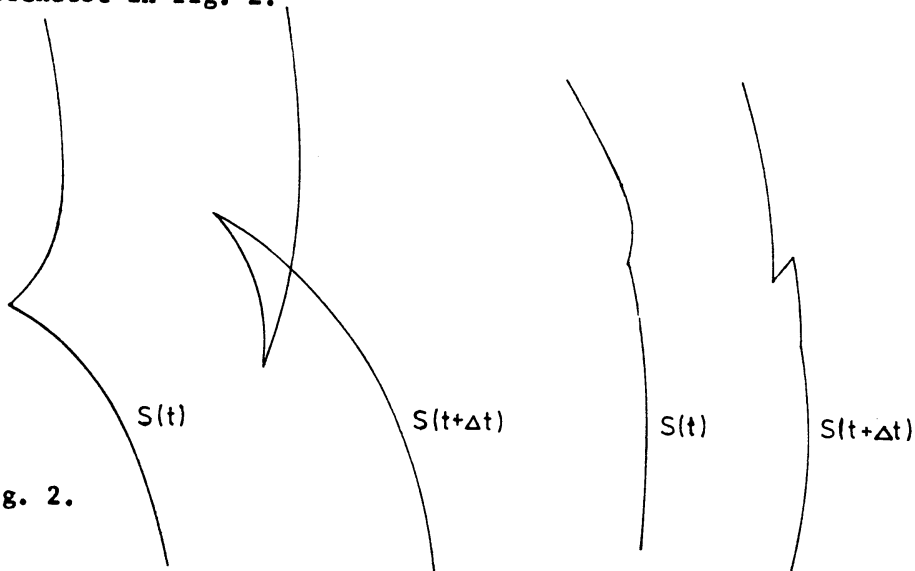


fig. 2.

Vindt dit snijden enkele keren achter elkaar plaats, dan kan men gevallen construeren, waarin men er met deze meetkundige constructie niet meer uitkomt. Een van de dingen die men nu kan doen om deze meetkundige constructie toch te kunnen blijven toepassen is achterblijvende gedeelten van tijdfronten weg te laten (zie hierover Bijlsma en Van Rietschote 1972 II). De vraag is nu wel of men op deze wijze geen oplossingen weglaat. Om deze vraag te kunnen beantwoorden zullen we overgaan op enkele (weliswaar intuïtieve) algemene beschouwingen over praktische aspecten van extremalen.

We beperken ons eerst tot het geval, waarin de golfsituatie niet met de tijd verandert.

Stationair geval

Veronderstel dat E een (absolute) extremaal is tussen de punten A en B (zie fig. 3) en dat P een punt is op deze extremaal.

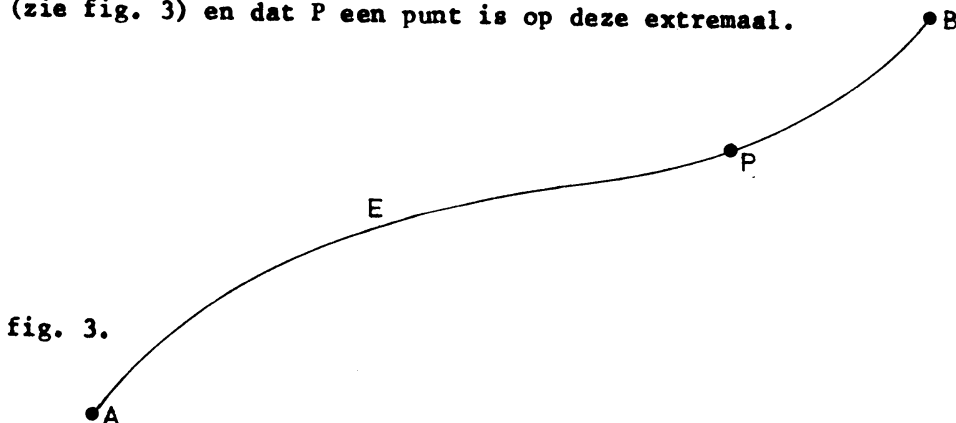


fig. 3.

Het is in dit geval duidelijk dat PB een absolute extremaal is. Was dit namelijk niet het geval, dan zou E geen absolute extremaal zijn. Hieruit volgt dus dat ook AP een absolute extremaal is. Derhalve kan in dit geval P nooit op een achterblijvend gedeelte van een tijdfront liggen en mag men deze gedeelten ongestraft weglaten. We zullen nu het niet-stationaire geval beschouwen.

Niet-stationaire geval

We nemen nu aan dat de golfsituatie met de tijd verandert en veronderstellen weer dat P ligt op de absolute extremaal van A naar B. Spreken we in dit geval over een absolute extremaal dan dienen we de vertrektijd expliciet te vermelden m.a.w. indien we nu zeggen dat PB een absolute extremaal is dan bedoelen we daarmee: een absolute extremaal bij b.v. een vertrektijd t_1 . De mogelijkheid is hiermee niet uitgesloten om op een tijdstip $t_2 > t_1$ uit P te vertrekken en toch op dezelfde tijd in B te arriveren. Er zij wel op gewezen, dat we hier de praktische situatie voor ogen hebben, waarin een schip in het algemeen een koers voor 12 uur uitzet. Bovendien betekent "gelijk aankomen" natuurlijk wel dat de aankomsttijden binnen zekere toegestane grenzen mogen verschillen.

Zouden we dergelijke toleranties in methoden a, b en c niet ingevoerd hebben dan zou het eindpunt waarschijnlijk bij geen van de iteratieve procedures bereikt worden. Keren we nu weer terug naar ons uitgangspunt dan blijkt dat AP geen absolute extremaal hoeft te zijn opdat E er een is. Het zou in dit geval dus onverstandig zijn achterblijvende gedeelten van tijdfronten weg te laten.

Gezien de argumenten die tegen bovenstaande methoden zijn geopperd ligt het voor de hand dat hier getracht is een methode te ontwikkelen die geen van bovengenoemde nadelen heeft. Deze methode zal in het volgende punt behandeld worden.

- e. Omdat het meetkundig construeren van tijdfronten, zoals we gezien hebben, in het algemeen geen geschikte methode is voor het bepalen van een optimale route, wordt het hier vervangen door het construeren van een waaier van extremalen, waaraan, indien de afstand tussen opeenvolgende extremalen te groot wordt na een bepaalde tijdstap, nieuwe extremalen worden toegevoegd. Dit gebeurt op de volgende manier. Analoog aan de vorige methode gaan we weer uit van een aantal startrichtingen $\vec{\lambda}_i(0) = (\lambda_i(0), \mu_i(0))$, $i = 1, \dots, N$. Vervolgens wordt het stelsel vergelijkingen

(2), (3), (5), (6), (7) met (10) simultaan geïntegreerd, waarbij als tijdstap 12 uur wordt aangehouden terwijl het polair snelheidsdiagram de vorm heeft van een ellips (zie Bijlsma en Van Rietschote 1972 II). Na elke tijdstap van 12 uur wordt nagegaan of de dan geconstrueerde punten te ver uit elkaar liggen. Indien dit het geval is wordt een punt tussengevoegd van waaruit een nieuwe extremaal gestart wordt. Als startwaarde voor λ in dit tussengevoegde punt nemen we het gemiddelde van de waarden van λ in de punten, waartussen het is toegevoegd. We zullen dit aan de hand van fig. 4 toelichten, waar we als voorbeeld uitgaan van de meest algemene situatie van fig. 2. Voor toepassingen zie fig. 5 t/m 8. De krommen C waarbinnen het snijden van de extremalen zich afspeelt, zijn bekend uit de optica (zie b.v. Born en Wolf, 1959) en worden brandkrommen genoemd.

Hoewel tijdfront S_1 gedefformeerd wordt in tijdfront S_2 is het duidelijk dat opeenvolgende punten op S_1 bij een bepaalde nummering, opeenvolgende punten op S_2 blijven.

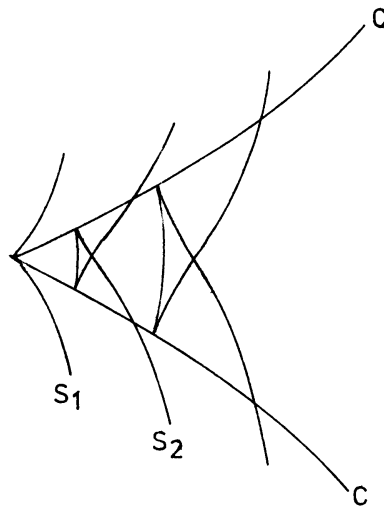


fig. 4.

Door nu, indien een punt is toegevoegd, de index van de daarop volgende punten met één te verhogen blijft de volgorde van punten op deze wijze behouden. Door vervolgens te onthouden hoeveel punten er vóór een bepaald punt zijn bijgekomen, kan men de optimale route terugvinden, op dezelfde manier als onder d. is aangegeven. Punten die op land terecht komen worden verwijderd. Hierdoor kan het voorkomen dat er tussen opeenvolgende

punten niet meer verfijnd mag worden. Een enkele opmerking nog over het route verloop na New Foundland (zie ook fig. 5 t/m 8). Daar het bij een noordelijke route in het algemeen optimaal zal zijn om de reis vanaf Cape Race (dit is de zuidelijkste punt van New Foundland) zo dicht mogelijk onder de kust te vervolgen, worden vanuit Cape Race op dezelfde wijze als vanuit het vertrekpunt een aantal startrichtingen geconstrueerd. Zo gauw Cape Race langs een extremaal binnen een tijdstap van 12 uur bereikt kan worden, worden m.b.v. deze toegevoegde startrichtingen voor een overeenkomstig gedeelte van de tijdstap van 12 uur een aantal punten gecreëerd, die aanleiding geven tot een stel toegevoegde extremen, die het gebied langs de kust bestrijken. De berekening wordt uiteraard beëindigd zo gauw een punt van een extremaal gevonden kan worden van waaruit het eindpunt binnen 12 uur bereikt kan worden. Voor enig inzicht in de gang van zaken in het rekenprogramma wordt men verwezen naar het stroomdiagram, dat als bijlage is toegevoegd. In het volgende hoofdstuk zal aandacht worden besteed aan enige resultaten van het rekenprogramma.

FIGURE

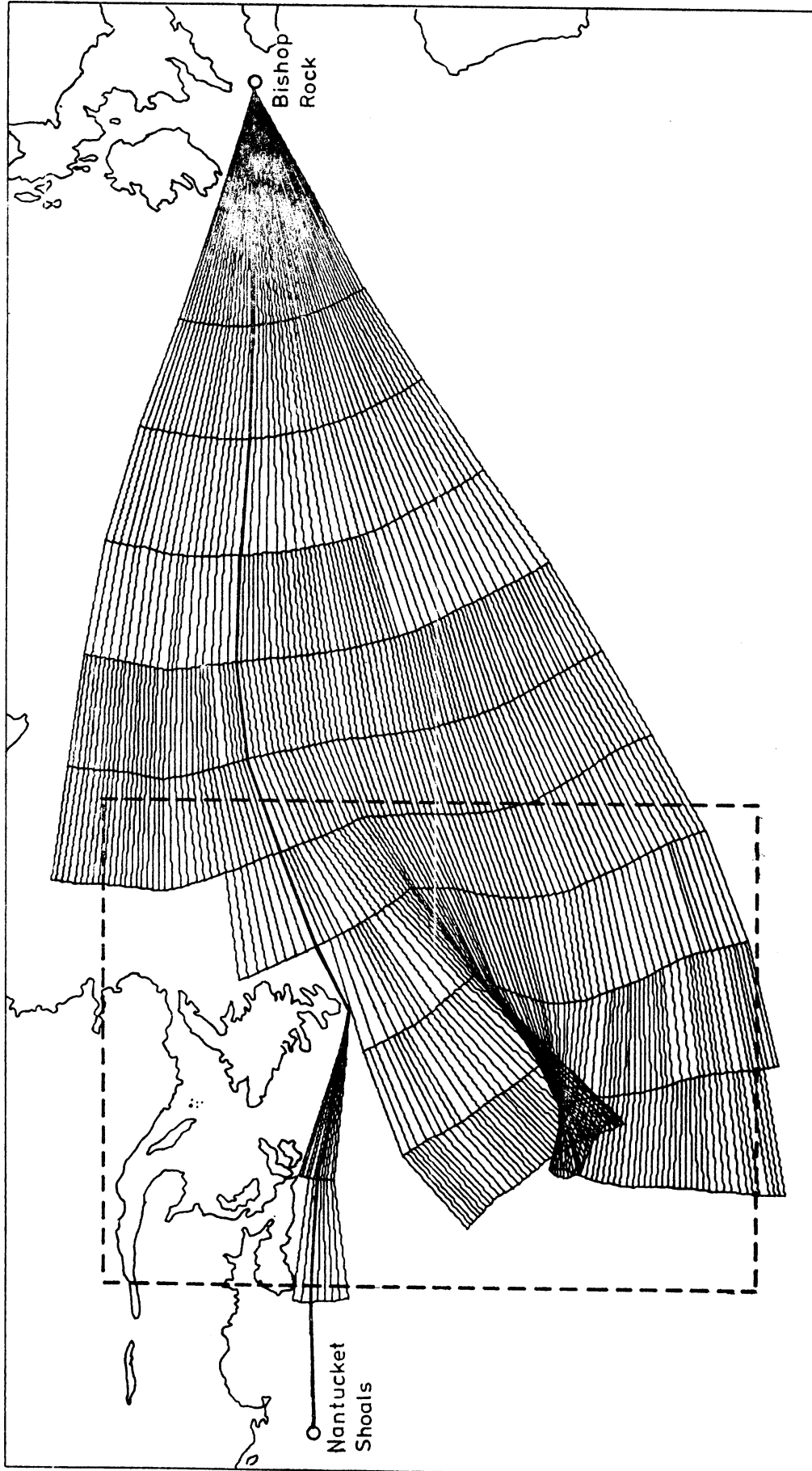


Fig. 5

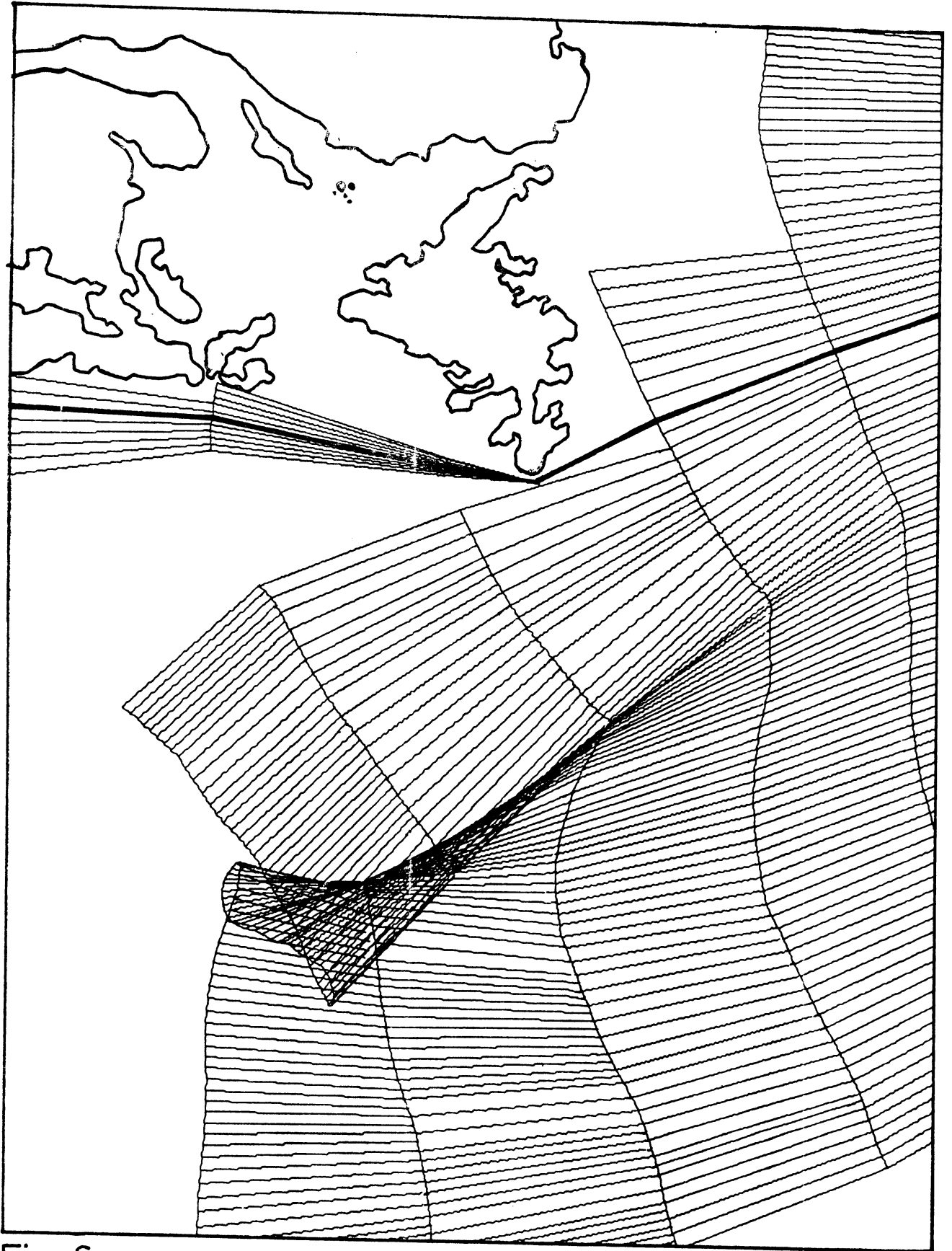


Fig. 6

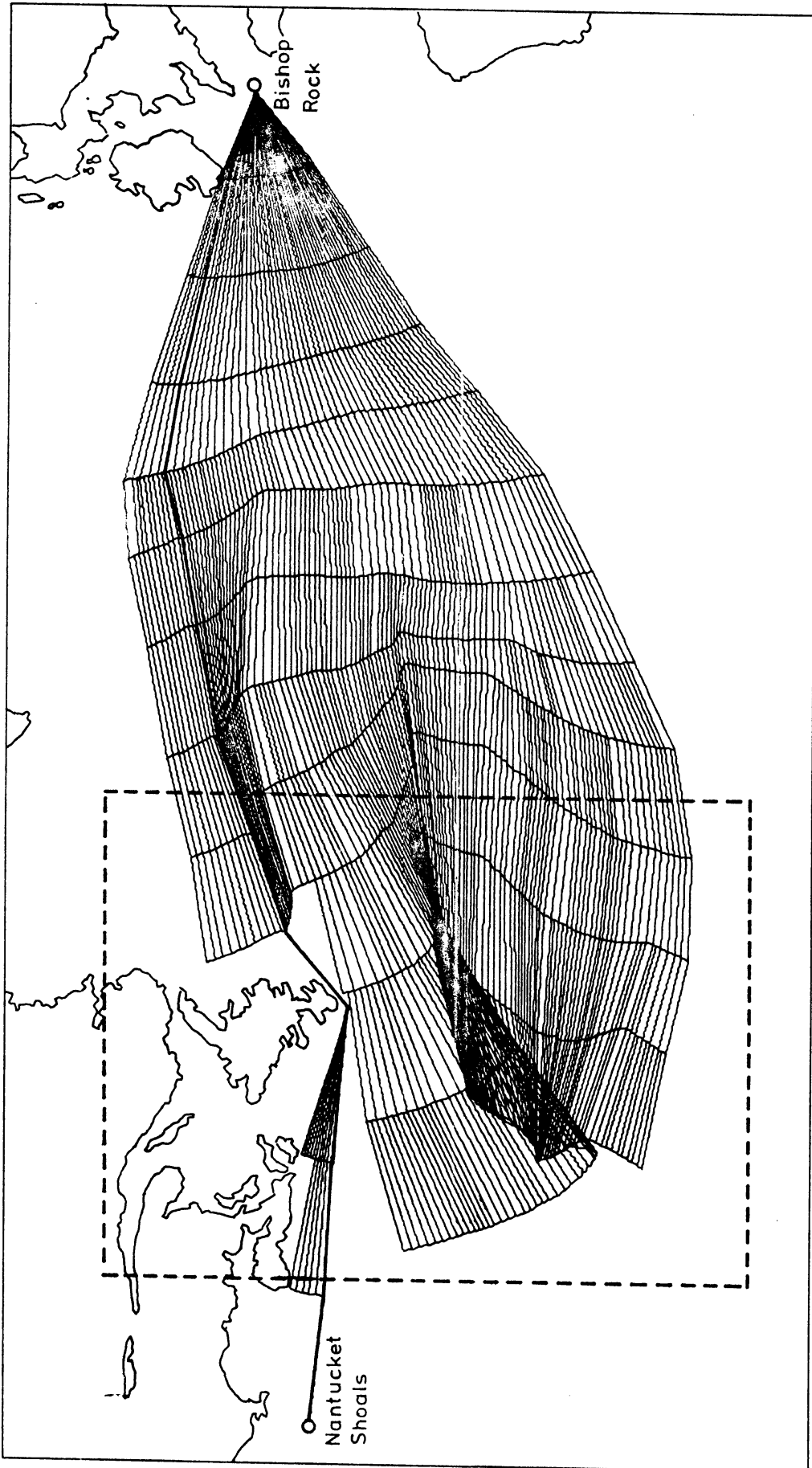


Fig. 7

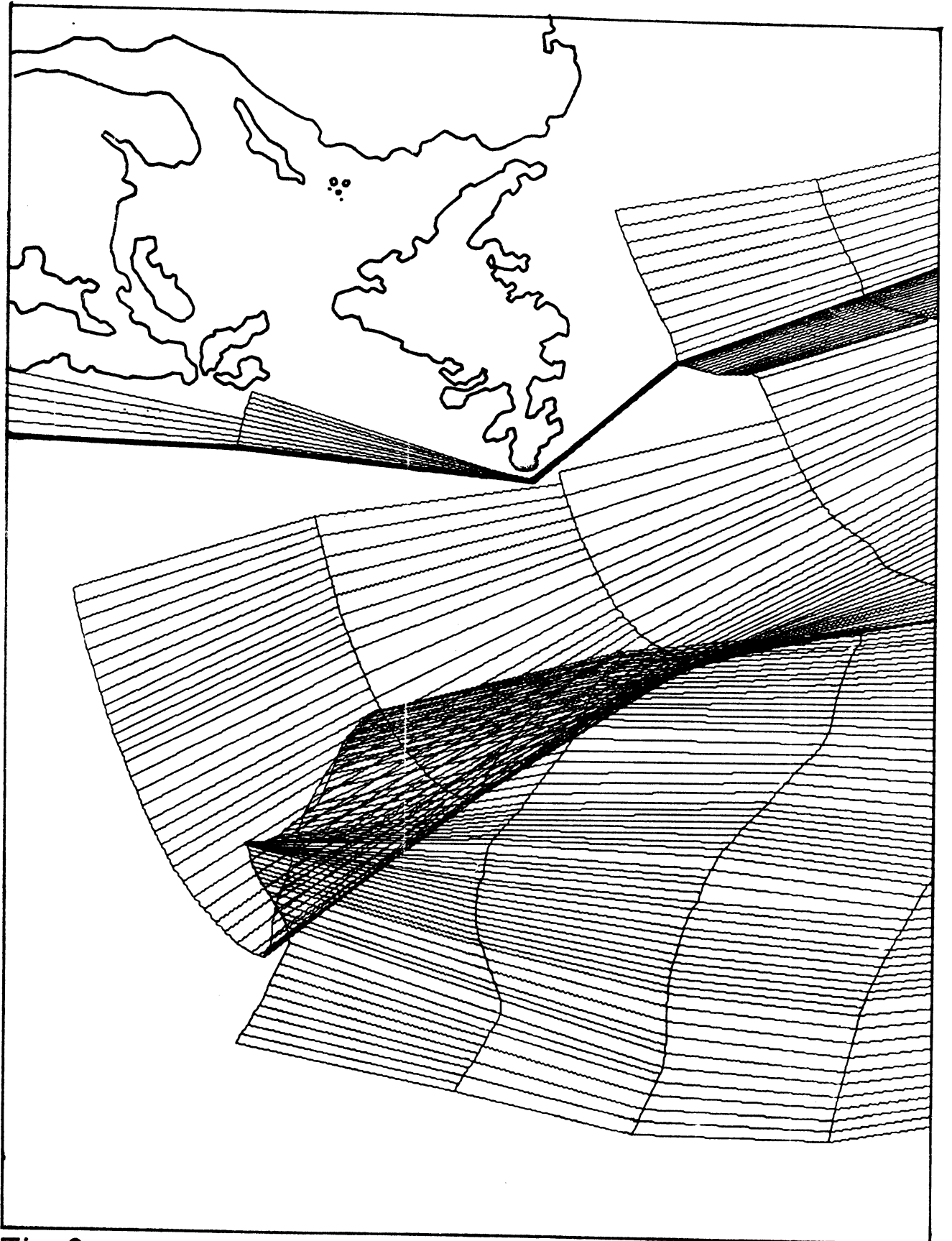


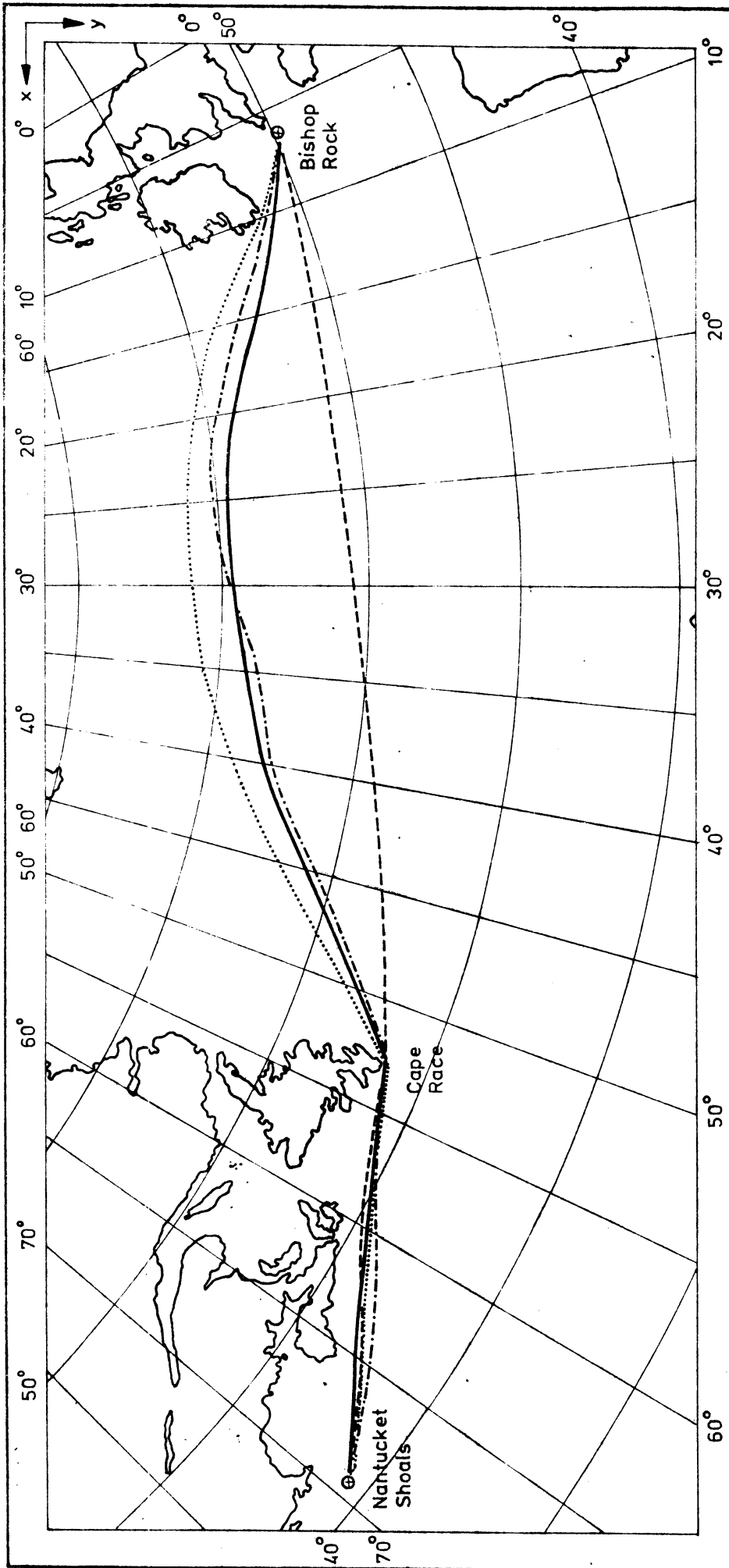
Fig. 8

2. Resultaten van het vergelijkend onderzoek

Wat betreft de resultaten van dit vergelijkend onderzoek, wordt men, voor een goed begrip van de hier getoonde resultaten verwezen naar ons eerder uitgevoerde vergelijkend onderzoek (Bijlsma en Van Rietschote, 1972 II), waar men gegevens vindt over het coördinatenstelsel en de gebruikte vaartafvalgrafieken en golfkaarten en waar men bovendien informatie vindt over de hier getoonde routes, die m.b.v. de manuele en numerieke tijdfront methoden geconstrueerd zijn. Voor de overzichtelijkheid is hier uit de veelheid van routes een keuze gemaakt, waarbij er voor gezorgd is dat elk karakteristiek geval vertegenwoordigd is (zie ook fig. 5 t/m 8). Het vergelijken van de routes, die verkregen zijn m.b.v. het hier gepresenteerde rekenprogramma, met die, welke verkregen zijn m.b.v. de numerieke tijdfront methode, is bijzonder nuttig geweest in verband met het benaderen van bepaalde grootheden zoals bv. ruimtelijke afgeleiden in het hier gebruikte rooster. Uit de hier geschetste voorbeelden blijkt duidelijk dat deze routes elkaar weinig ontlopen.

Bij de numerieke tijdfront methode worden extremalen toegevoegd aan de bovenzijde van een tijdfront om de route hoger te kunnen laten lopen. Dit kan in sommige gevallen aanleiding geven tot geringe verschillen in het verloop van de routes. Tenslotte is het duidelijk dat alle routes hier uitgesproken bovenlangs gaan, zodat het bij voorbaat uitgesloten was om alternatieve routes met gelijke vaartijd te vinden.

RESULTATEN



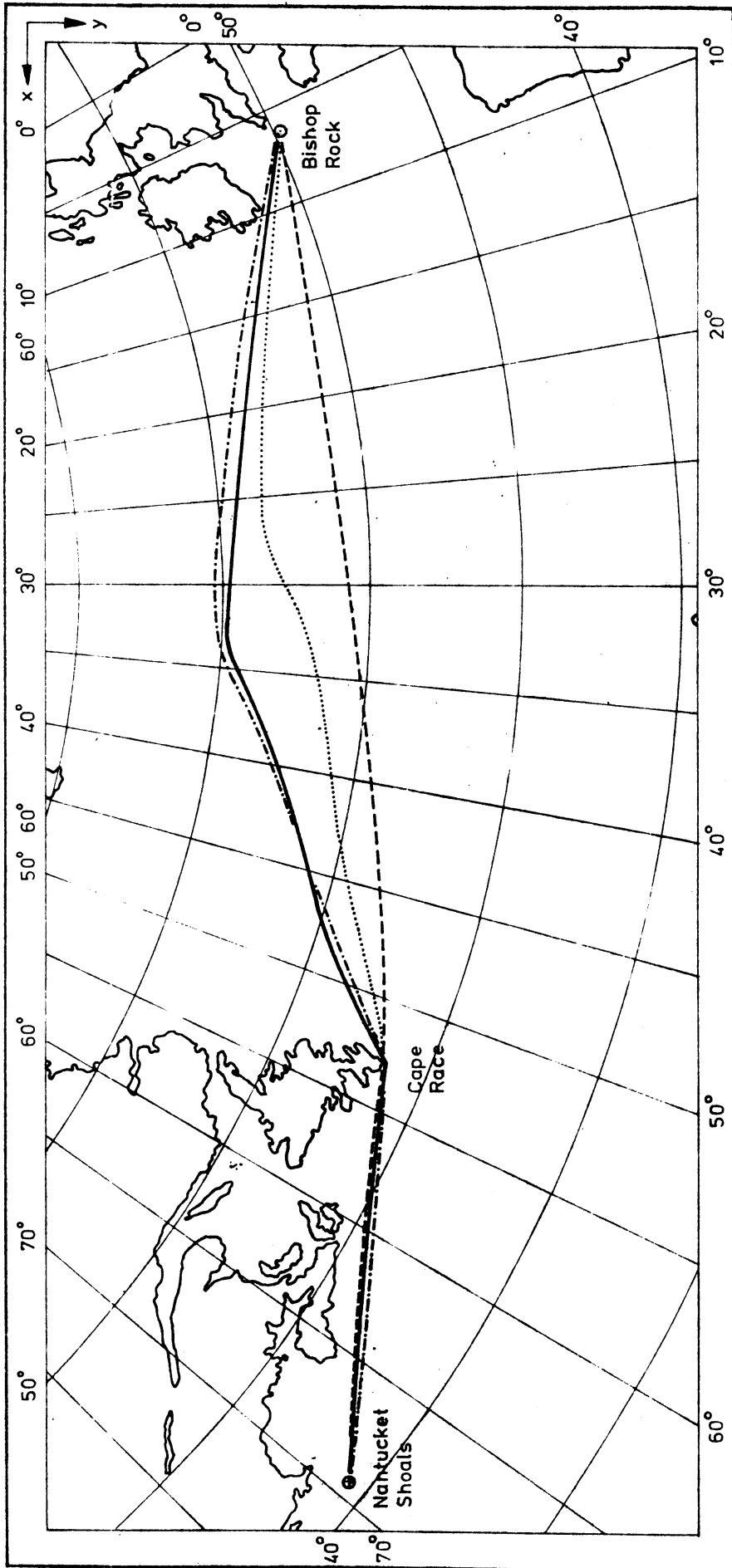
- de in dit rapport gevolgde methode
- de manuele tijdfrontmethode
- · - · de numerieke tijdfrontmethode
- de grootcirkel

PUNTEN OPTIMALE ROUTE:	X	Y	L	B
	18.08	4.82	68.44	41.11
	16.46	4.84	63.90	43.52
	14.85	5.02	58.52	45.29
	13.30	5.18	52.99	46.68
	11.91	4.64	48.57	49.37
	10.74	4.07	44.45	51.76
	9.58	3.65	39.72	53.53
	8.26	3.43	33.63	54.57
	6.99	3.36	27.52	54.81
	5.82	3.40	21.96	54.40
	4.65	3.63	16.77	53.18
	3.41	3.86	11.67	51.64

REISDUUR:	DAGEN	UREN
OPTIMALE ROUTE	6	3.0
GROOTCIRKEL	6	17.2
GC-LT	0	14.2

OMWEG:	MYLEN
OPTIMALE ROUTE	101

TIJDSTIP VAN VERTREK:	200170	00.00 Z
-----------------------	--------	---------



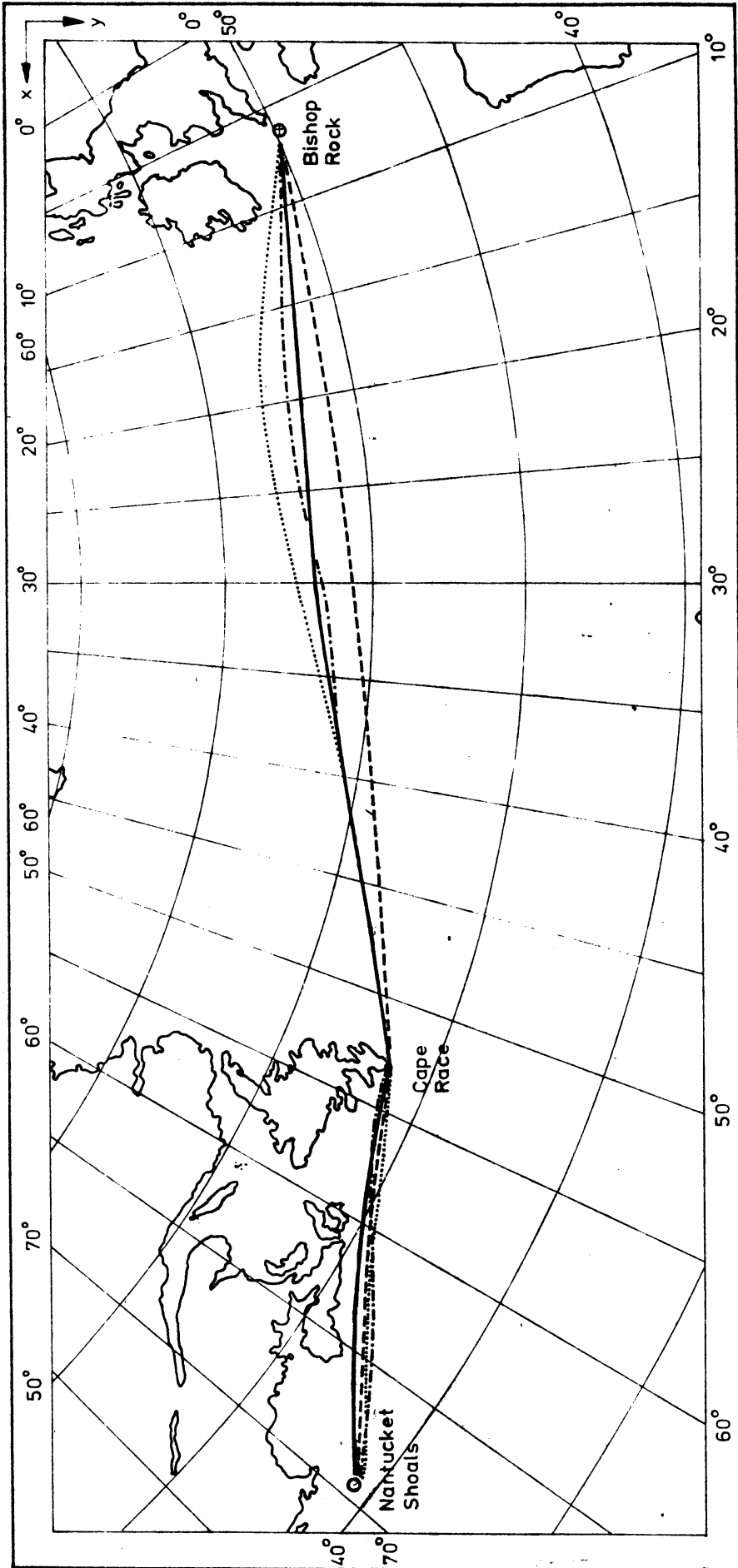
- de in dit rapport gevolgde methode
- de manuele tijdfrontenmethode
- · - · - de numerieke tijdfrontenmethode
- de grootcirkel

PUNTEN OPTIMALE ROUTE:	X	Y	L	B
	17.33	4.85	66.37	42.22
	15.72	4.97	61.40	44.27
	14.15	5.16	55.95	45.80
	12.66	4.98	50.95	47.83
	11.41	4.38	46.88	50.44
	10.11	4.01	41.78	52.31
	9.10	3.78	37.40	53.38
	8.26	3.40	33.65	54.65
	7.13	3.42	28.23	54.66
	5.80	3.63	22.02	53.75
	4.52	3.74	16.32	52.80
	3.30	3.85	11.21	51.56

REISDUUR:	DAGEN	UREN
OPTIMALE ROUTE	6	8.2
GROOTCIRKEL	6	20.8
GC-LT	0	12.6

OMWEG:	MYLEN
OPTIMALE ROUTE	89

TIJDSTIP VAN VERTREK:	220170	00.00 Z
-----------------------	--------	---------



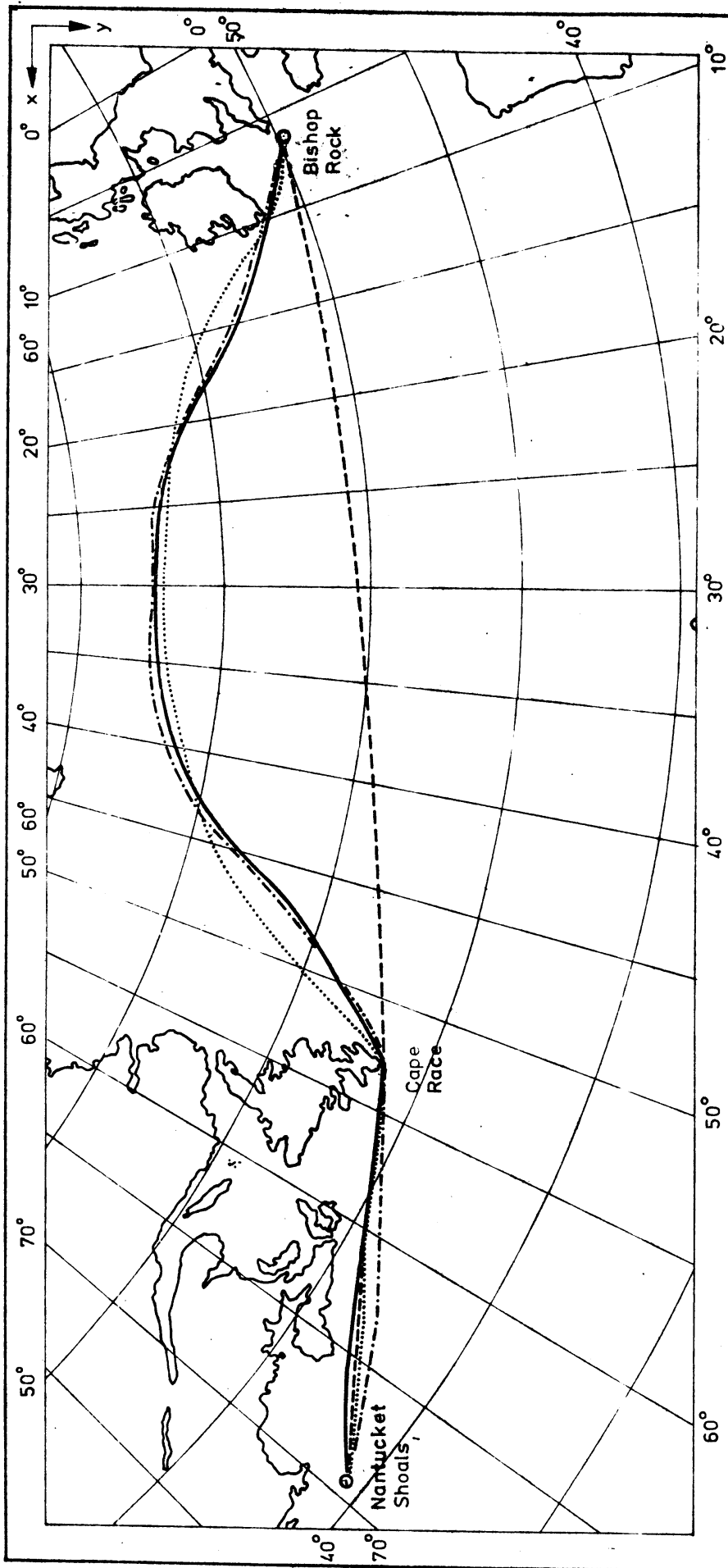
- de in dit rapport gevolgde methode
- de manuele tijdfrontenmethode
- - - - de numerieke tijdfrontenmethode
- de grootcirkel

PUNTEN OPTIMALE ROUTE:	X	Y	L	B
	18.04	4.82	68.37	41.19
	16.47	4.81	63.97	43.58
	14.88	4.95	58.76	45.41
	13.38	5.23	53.20	46.48
	11.90	5.09	47.95	48.21
	10.43	4.85	42.39	49.88
	9.00	4.69	36.49	50.94
	7.58	4.47	30.36	51.78
	6.18	4.34	24.13	51.95
	4.83	4.25	18.15	51.61
	3.45	4.16	12.26	50.90

REISDUUR:	DAGEN	UREN
OPTIMALE ROUTE	5	15.2
GROOTCIRKEL	5	15.3
GC-LT	0	.1

OMWEG:	MYLEN
OPTIMALE ROUTE	22

TIJDSTIP VAN VERTREK:	270170	00.00 Z
-----------------------	--------	---------



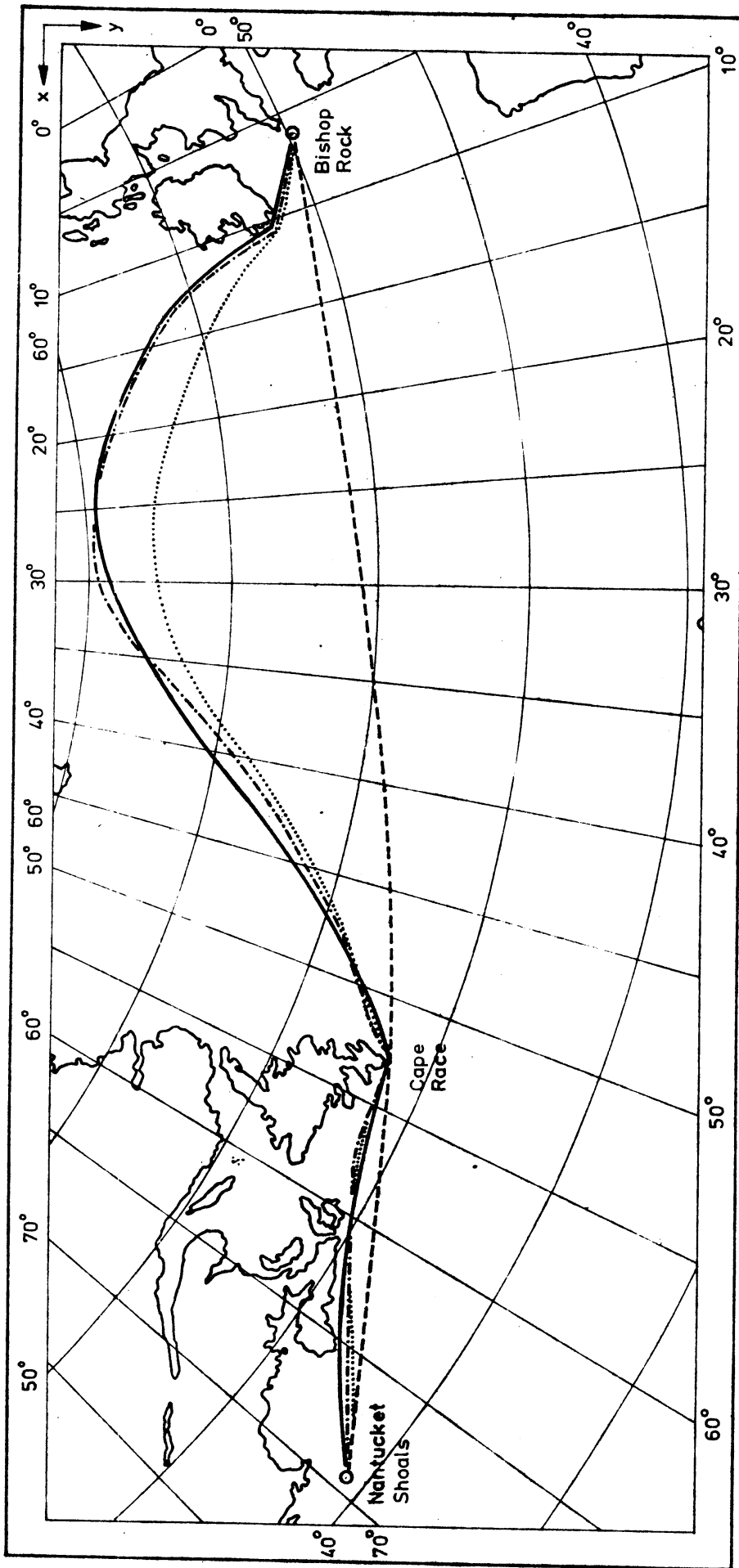
- de in dit rapport gevolgde methode
- de manuele tijdfrontenmethode
- · - · de numerieke tijdfrontenmethode
- de grootcirkel

PUNTEN OPTIMALE ROUTE:	X	Y	L	B
	18.35	4.73	69.36	40.86
	17.33	4.68	66.73	42.57
	16.32	4.78	63.59	43.86
	15.33	4.91	60.26	44.93
	14.34	5.09	56.73	45.75
	13.38	5.25	53.16	46.45
	12.59	4.82	50.91	48.30
	11.83	4.40	48.57	50.04
	11.14	3.91	46.36	51.86
	10.64	3.53	44.64	53.25
	10.25	3.20	43.22	54.38
	9.66	2.91	40.74	55.52
	8.86	2.74	36.90	56.35
	8.10	2.58	33.11	56.96
	7.43	2.52	29.64	57.19
	6.70	2.48	25.86	57.21
	5.94	2.67	22.06	56.46
	5.32	2.99	19.25	55.28
	4.80	3.27	17.06	54.23
	4.34	3.51	15.25	53.27
	3.69	3.69	12.65	52.32
	2.85	3.85	9.38	51.17

REISDUUR:	DAGEN	UREN
OPTIMALE ROUTE	11	1.3
GROOTCIRKEL	14	8.2
GC-LT	3	6.9

OMWEG:	MYLEN
OPTIMALE ROUTE	215

TIJDSTIP VAN VERTREK:		
	180170	00.00 Z



- de in dit rapport gevolgde methode
- de manuele tijdfrontenmethode
- · - · de numerieke tijdfrontenmethode
- - - - de grootcirkel

PUNTEN OPTIMALE ROUTE:

	X	Y	L	B
	18.39	4.70	69.52	40.87
	17.42	4.57	67.18	42.66
	16.41	4.61	64.19	44.09
	15.44	4.73	60.97	45.20
	14.63	4.92	57.98	45.79
	13.82	5.17	54.82	46.15
	12.89	5.11	51.60	47.27
	12.05	4.68	49.04	49.15
	11.30	4.15	46.74	51.12
	10.61	3.68	44.35	52.87
	10.04	3.24	42.23	54.42
	9.57	2.78	40.41	55.93
	9.16	2.60	38.48	56.61
	8.41	2.28	34.83	57.77
	7.92	1.92	32.28	58.85
	7.44	1.71	29.65	59.48
	6.67	1.68	25.33	59.47
	5.90	1.82	21.21	58.82
	5.13	2.04	17.33	57.80
	4.36	2.39	13.93	56.32
	3.65	2.79	11.19	54.64
	3.13	3.18	9.48	53.14
	2.66	3.60	8.19	51.60

REISDUUR:

DAGEN

UREN

OPTIMALE ROUTE
GROOTCIRKEL
GC-LT

11
13
2

13.2
14.1
.9

OMWEG:

MYLEN

OPTIMALE ROUTE

350

TIJDSTIP VAN VERTREK:

210170

00.00 Z

Literatuur

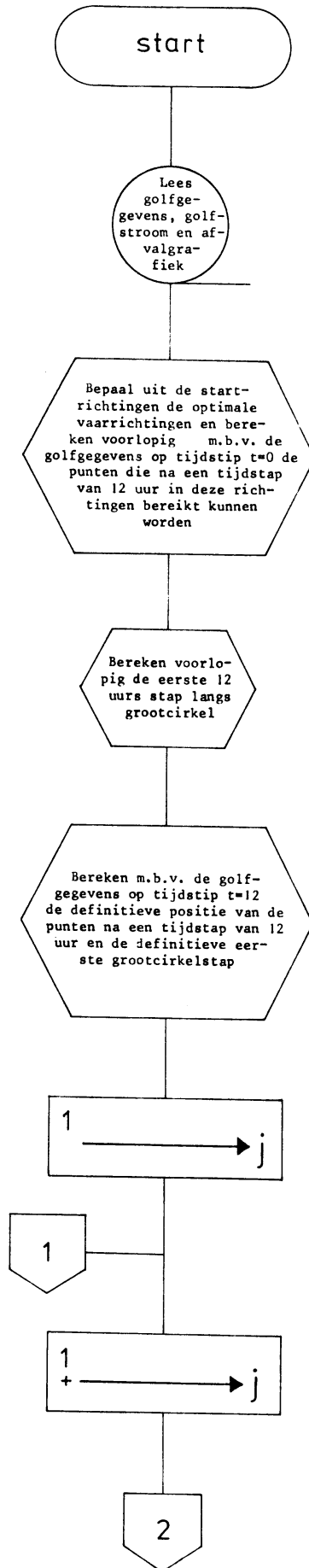
- Bleick, W.E. en F.D. Faulkner (1965) Minimal-time ship routing. Journ. appl. meteor. Vol. 4, no. 2, pp. 217-221.
- Born, M. en E. Wolf (1959) Principles of optics, Pergamon Press, London.
- Bryson, A.E. en W.F. Denham (1962) A steepest-ascent method for solving optimum programming problems. J. Appl. Mech. 29, 247-257.
- Bijlsma, S.J. en B. van Riet-schote (1972, I en II) Over het optimaal routeren van schepen,
I. een literatuuronderzoek W.R. 72-1
II. een vergelijkend onderzoek W.R. 72-2
K.N.M.I., De Bilt.
- Faulkner, F.D. (1964) A comparison between some methods for computing optimum paths in the problem of Bolza in:
Computing methods in optimization problem, Balakrishnan, A.V. and L.W. Neustads (eds), Academic Press, Inc. New York.
- Lee, E.E. en L. Markus (1967) Foundations of optimal control theory, John Wiley and sons, New York.
- Marks, W. et al. (1968) An automated system for optimum ship routing
Trans. Soc. Naval Architects and Marine Engineers. Vol. 76, pp. 22-55.
- Pontryagin, L.S. et al. (1962) The mathematical theory of optimal processes, Interscience Publishers, Inc., New York.
- Powell, M.J.D. (1970) Recent advances in unconstrained optimization, Math. Branch, A.E.R.E., Harwell, Berks.

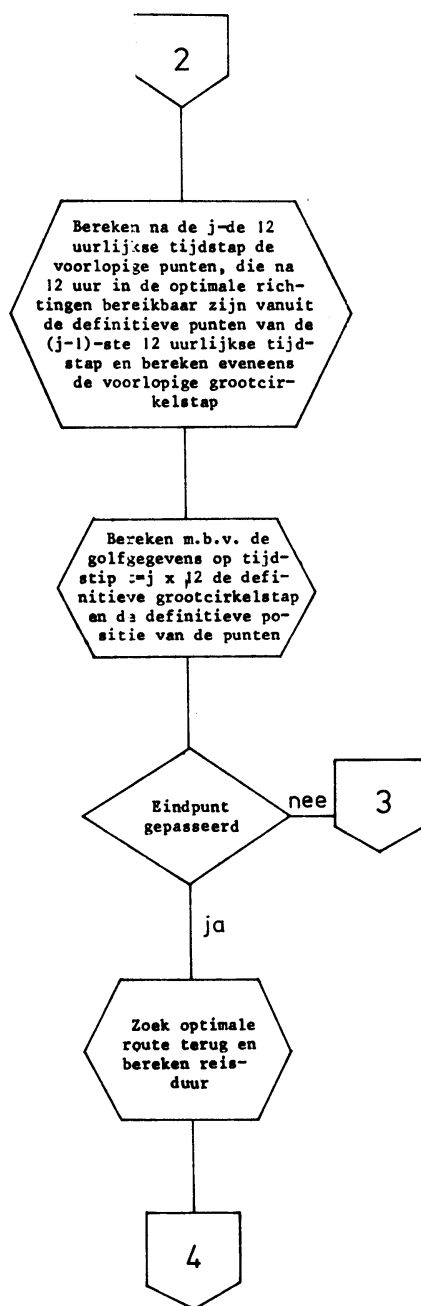
Wilde, D.J. en C.S. Breightler, Foundations of optimization, Prentice-Hall, (1967) Inc.

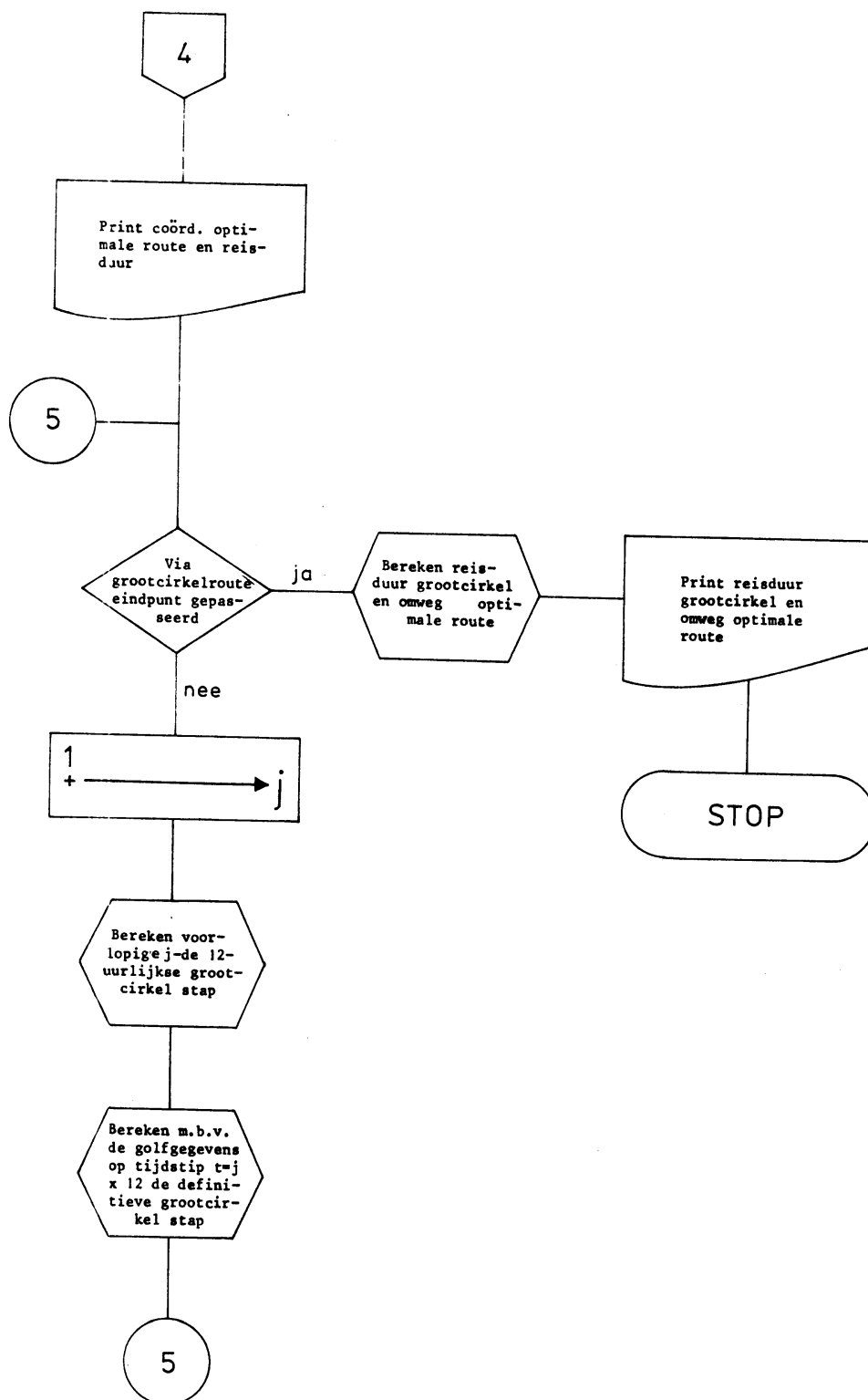
Zoppoli, R. (1972) Minimum-Time Routing as an N-stage Decision Process. Journ. appl. meteor. vol. 11, pp. 429-435.

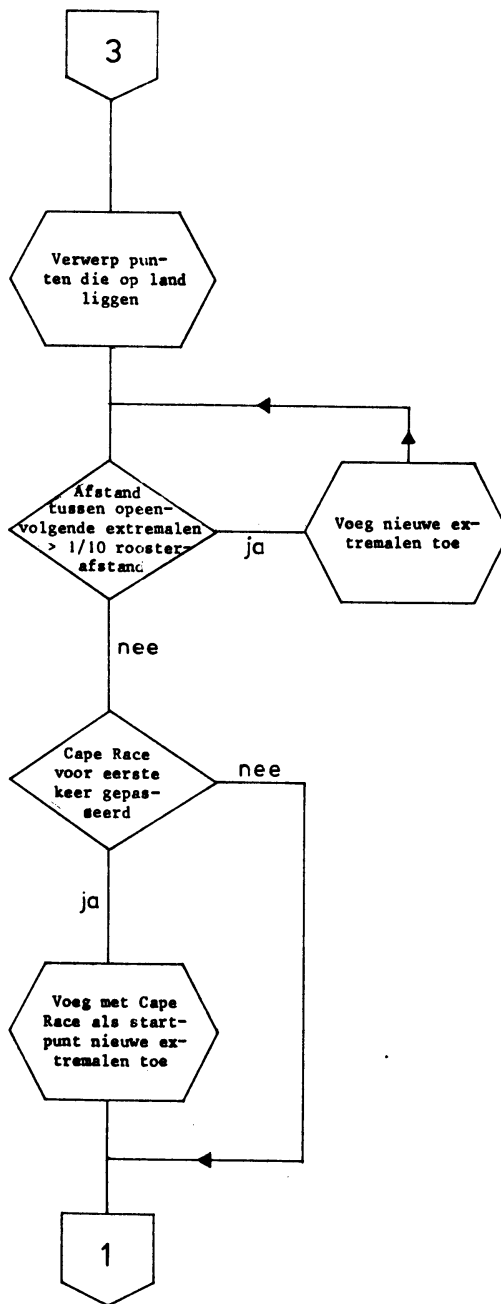
BIJLAGE 1

HET STROOMDIAGRAM









BIJLAGE 2

H E T A L G O L - P R O G R A M M A

```
begin comment KNMI-050972-BIJLS-RIEB.  
evaluatie van de kortste-vaartijd route en grootcirkel route;  
integer ant,at,at1,at11,at3,f,h,h1,hh,i,i0,i1,i11,i2,i00,i001,  
i01,i02,in,in1,ip,j,j0,j00,j2,jj,jj1,k,k1,pla,srt,tr,x,y;  
real a,aa,a0,afst1,afst2,amm,b,bb,bbb,c,cc,copgr,copi,d,dx,dy,  
e1,e2,e3,fr,ft,ftr,g,grd,l,lll,lim,lim1,olt,p,pp,pi,pi2,pi4,  
pi9,pim2,pr,q,r,s,sipgr,sipi,t,vh,vtu,xx,xxx,xe,xg,xk,xs,xv,  
yy,yyy,ye,yg,yk,ys,ysn,yv;  
boolean joke;  
integer array AA[1:3,0:30],I,PIA[1:30],  
GH,GR,XGS,YGS[0:27,0:17],PA,PB[1:150],XX,YY[1:12];  
array A,LA,MU,P,XT,YT[0:300],AH[1:3],AFX,AFY[1:4],L[1:12],  
X,Y[0:300,0:2],XGR,YGR[0:50];  
  
real procedure ARCTAN(x,v); value x,y; real x,y;  
ARCTAN:=if x>0^y<0 then -arctan(abs(y/x)) else  
if x<0^y>0 then pi-arctan(abs(y/x)) else  
if x<0^y<0 then -pi+arctan(abs(y/x)) else arctan(y/x);  
  
real procedure LM(a,b); value a,b; real a,b;  
LM:=sqrt(cc)*(arctan(sqrt((2*cc/(2*cc-  
(xs-a)^2-(ys-b)^2))^2-1)))/(AA[1,0]/170);  
  
procedure LT(a,b); value a,b; real a,b;  
begin real prdx,prdy,la,s;  
prdx:=prdy:=0;  
COBY: la:=pi2-  
2*arctan(sqrt((a+prdx/2-7.5)^2+(b+prdy/2+8.5)^2)*a0);  
s:=1852/((1+sin(la))*a0*6378178);  
if prdx#0 then goto KLM;  
prdx:=s*dx; prdy:=s*dy;  
goto COBY;  
KLM: dx:=s*dx; dy:=s*dy  
end LT;  
  
procedure AM(a,b); value a,b; real a,b;  
begin amm:=1000;  
tr:=0;  
for i:=11,i+1 while i<in do  
begin dx:=X[i,0]-a; dy:=Y[i,0]-b;  
d:=sqrt(dx*dx+dy*dy);  
if d<amm then begin amm:=d; tr:=i end  
end  
end AM;
```

```
procedure MAXAF(x,y,a); value x,y,a; real x,y,a;
begin   real coagr,cog,coggr,cogr,dx1,dy1,
        gh,gr,gr0,gr1,gr00,gr01,gr10,gr11,
        l1,x1,siagr,sig,siggr,sigr;
        integer h,h1,j1,k,k1,spr;
        spr:=0; j1:=j-1;
MGRT:   h:=entier(x); h1:=h+1;
        k:=if y>0 then entier(y) else 0; k1:=k+1;
        p:=x-h; q:=1-p;
        if y<0 then r:=0 else r:=y-k;
        s:=1-r;
        dx:=(s*(q*XGS[h,k]+p*XGS[h1,k])+
        r*(q*XGS[h,k1]+p*XGS[h1,k1]))/10;
        dy:=(s*(q*YGS[h,k]+p*YGS[h1,k])+
        r*(q*YGS[h,k1]+p*YGS[h1,k1]))/10;
        gh:=(s*(q*GH[h,k]+p*GH[h1,k])+r*(q*GH[h,k1]+
        p*GH[h1,k1]))*2;
        if gh>299 then gh:=299;
        gr00:=GR[h,k]; gr01:=GR[h,k1];
        gr10:=GR[h1,k]; gr11:=GR[h1,k1];
        if gr00-gr10>180 then gr10:=gr10+360;
        if gr10-gr00>180 then gr00:=gr00+360;
        if gr01-gr11>180 then gr11:=gr11+360;
        if gr11-gr01>180 then gr01:=gr01+360;
        gr0:=q*gr00+p*gr10; gr1:=q*gr01+p*gr11;
        if gr0-gr1>180 then gr1:=gr1+360;
        if gr1-gr0>180 then gr0:=gr0+360;
        gr:=(s*gr0+r*gr1)*grd;
        cogr:=cos(gr); sigr:=sin(gr);
        h:=entier(gh/10); h1:=h+1;
        p:=gh/10-h; q:=1-p;
        for k:=1,2,3 do AH[k]:=AA[k,h]*q+AA[k,h1]*p;
        e1:=(AH[1]+AH[3])/2;
        e2:=AH[1]-e1;
        e3:=e1*AH[2]/sqrt(AH[1]*AH[3]);
        if a=10 then
        begin   g:=arctan(-(xg-aa)/(yg-bb));
                cog:=cos(g); sig:=sin(g);
                pr:=dx*cog+dy*sig;
                coggr:=cos(g-gr); siggr:=sin(g-gr);
```

```
p:=e3Xe3Xcoggr; q:=e1Xe1Xsiggr;
r:=e2Xsiggr; r:=rXr;
s:=pXcoggr; t:=qXsiggr;
l:=(-pXe2+e1Xe3Xsqrt(t-r+s))/(t+s);
dx:=(1+pr)Xcog; dy:=(1+pr)Xsig
end else
begin coagr:=cos(a-gr); siagr:=sin(a-gr);
p:=e1Xcoagr; q:=e3Xsiagr;
r:=sqrt(pXp+qXq);
s:=e1Xp/r-e2; t:=e3Xq/r;
dx:=dx+sXcogr-tXsigr;
dy:=dy+sXsigr+tXcogr
end;
LT(x,y);
if a=10 then
begin if spr=0 then
begin dx1:=dx; dy1:=dy end else
begin dx:=(dx+dx1)/2; dy:=(dy+dy1)/2 end;
l1:=sqrt(dxXdx+dyXdy);
x1:=(l1Xl1X(aa-XGR[j1 ])+
l1X(YGR[j1 ]-bb)Xsqrt(4Xcc-l1Xl1 ))/(2Xcc);
x:=XGR[j1 ]+x1;
y:=YGR[j1 ]-(l1Xl1+2X(XGR[j1 ]-aa)Xx1)/
(2X(YGR[j1 ]-bb));
if spr#0 then
begin XGR[j]:=xg:=x; YGR[j]:=yg:=y;
goto MAC
end;
inarray(drum,20000+jX600,GH);
inarray(drum,50000+jX600,GR); hold(GR);
spr:=1;
goto MGRT;
MAC: end
end MAXAF;

ant:=HAND(1);
jji:=pia:=PIA[1 ]:=0;
srt:=ant-ant:100X100;
if 7 comparefile(tape(2),krieb rtn) then goto AF;
for j:=srt,j+1 while j<37 do
```

```
begin   inarray(tape(2),jx2-1,GH);
        inarray(tape(2),jx2,GR); hold(GR);
        outarray(drum,jj1x600+20000,GH);
        outarray(drum,jj1x600+50000,GR); hold(GR);
        jj1:=jj1+1
end;
inarray(tape(2),75,XGS);
inarray(tape(2),76,YGS);
inarray(tape(2),76+entier(ant/100),AA); hold(AA);
for j:=jj1,j+1 while j<jj1+6 do
begin   outarray(drum,jx600+20000,GH);
        outarray(drum,jx600+50000,GR)
end;
xs:=2.03; ys:=4.03;
xe:=18.46; ye:=4.8;
xk:=13.316; yk:=5.248;
afst1:=0; afst2:=2644;
aa:=13.00; bb:=-54.42; cc:=3536.54;
pi:=4xarctan(1); pim2:=pix2; grd:=pi/180;
pi2:=pi/2; pi4:=pi/4; pi9:=pi/9;
ft:=1+cos(pi/6);
a0:=15x2.12/(637.8178xft);
lim:=LM(xe,ye); lim1:=LM(xk,yk);
i1:=0; in:=40;
at:=14;
j:=1;
inarray(drum,20000,GH);
inarray(drum,50000,GR); hold(GR);
joke:=true;
for i:=i1,i+1 while i<in do
begin   A[i]:=(10-i)xgrd;
        IA[i]:=cos(A[i]); MU[i]:=sin(A[i]);
        MAXAF(xs,ys,A[i]);
        X[i,1]:=xs+dx;
        Y[i,1]:=ys+dy
end;
XGR[0]:=xg:=xs; YGR[0]:=yg:=ys;
MAXAF(xg,yg,10);
for i:=i1,i+1 while i<in do
```

```
begin MAXAF(X[i,1],Y[i,1],A[i]);  
X[i,2]:=(X[i,1]+xs+dx)/2;  
Y[i,2]:=(Y[i,1]+ys+dy)/2;  
P[i]:=ARCTAN(X[i,2]-xs,Y[i,2]-ys)  
end;  
for i:=i1,i+1 while i<in do  
begin X[i,0]:=XT[i]:=X[i,2]; Y[i,0]:=YT[i]:=Y[i,2] end;  
outarray(drum,0,XT);  
outarray(drum,602,YT); hold(YT);  
for h:=1,h+1 while h<151 do PB[h]:=PA[h]:=1000;  
outarray(drum,80001,PB);  
outarray(drum,80151,PA); hold(PA);
```

NEXTR: j:=j+1;

```
for i:=i1,i+1 while i<in do  
begin if Y[i,0]=-99 then goto L1;  
sipi:=sin(P[i]); cop1:=cos(P[i]);  
h:=entier(X[i,0]);  
k:=if Y[i,0]<1 then 1 else entier(Y[i,0]);  
j2:=1;  
for jj:=h-1,h-1,h,h,h,h,h+1,h+1,h+1,h+1,h+2,h+2 do  
begin XX[j2]:=jj; j2:=j2+1 end;  
j2:=1;  
for jj:=k,k+1,k-1,k,k+1,k+2,k-1,k,k+1,k+2,k,k+1 do  
begin YY[j2]:=jj; j2:=j2+1 end;  
for jj:=1,jj+1 while jj<13 do  
begin x:=XX[jj]; y:=YY[jj];  
pr:=XGS[x,y]xcop1+YGS[x,y]xsi1;  
hh:=entier(GH[x,y]/10);  
if hh>30 then hh:=30;  
a:=AA[1,hh]; b:=AA[2,hh]; c:=AA[3,hh];  
e1:=(a+c)/2; e2:=e1xb/sqrt(abc); e3:=a-e1;  
copgr:=cos(P[i]-GR[x,y]xgrd);  
sipgr:=sin(P[i]-GR[x,y]xgrd);  
p:=e2xe2xcopgr; q:=e1xe1xsipgr;  
r:=e3xsipgr; r:=rXr;  
s:=pxcopgr; t:=qxsipgr;  
l:=(-pxe3+e1xe2xsqrt(t-r+s))/(t+s);  
dx:=(1+pr)xcop1; dy:=(1+pr)xsip1;  
LT(x,y);  
L[jj]:=sqrt(dxXdx+dyXdy)  
end;
```



```
AFX[1]:=L[8]-L[1]; AFY[1]:=L[5]-L[3];
AFX[2]:=L[9]-L[2]; AFY[2]:=L[6]-L[4];
AFX[3]:=L[11]-L[4]; AFY[3]:=L[9]-L[7];
AFX[4]:=L[12]-L[5]; AFY[4]:=L[10]-L[8];
p:=X[i,0]-h; q:=1-p; r:=Y[i,0]-k; s:=1-r;
fr:=LA[i]×copi+MU[i]×sipi;
IA[i]:=LA[i]-fr×((p×(r×AFX[4]+s×AFX[3])+
q×(r×AFX[2]+s×AFX[1]))/2);
MU[i]:=MU[i]-fr×((p×(r×AFY[4]+s×AFY[3])+
q×(r×AFY[2]+s×AFY[1]))/2);
A[i]:=ARCTAN(IA[i],MU[i]);
MAXAF(X[i,0],Y[i,0],A[i]);
X[i,1]:=X[i,0]+dx;
Y[i,1]:=Y[i,0]+dy;
L1:  end;
      MAXAF(xg,yg,10);
      for i:=11,i+1 while i<=n do
      begin  if Y[i,0]=-99 then begin Y[i,2]=-99; goto L2 end;
            MAXAF(X[i,1],Y[i,1],A[i]);
            XT[i]:=X[i,2]:=(X[i,1]+X[i,0]+dx)/2;
            YT[i]:=Y[i,2]:=(Y[i,1]+Y[i,0]+dy)/2;
            P[i]:=ARCTAN(X[i,2]-X[i,0],Y[i,2]-Y[i,0]);
L2:  end;
      if j>entier(lim) then
      begin  AM(xe,ye);
            i0:=tr;
            if amm>sqrt((X[i0,2]-X[i0,0])2+(Y[i0,2]-Y[i0,0])2)
            then goto NEG else
            begin  pp:=sqrt((xe-X[i0,0])2+(ye-Y[i0,0])2)/
                  sqrt((X[i0,2]-X[i0,0])2+(Y[i0,2]-Y[i0,0])2);
                  goto EIND
            end;
      end;
NEG:  end;
      if j>entier(lim)∧joke then
      begin  AM(xk,yk);
            i00:=tr;
            vh:=1-amm/
            sqrt((X[i00,2]-X[i00,0])2+(Y[i00,2]-Y[i00,0])2);
            if vh<0 then goto NEG1
```

```
else begin joke:=false; j00:=j; i01:=i00 end;
NEG1:
end;
for i:=11,i+1 while i<=n do
begin if Y[i,0]=-99 then goto L1;
      if Y[i,2]>8.65^Y[i,2]<1.35 then goto L3;
      if X[i,2]>13^Y[i,2]<4.5 then goto L3;
      if X[i,2]>13^X[i,2]<13.58^Y[i,2]<5.18 then goto L3;
      if X[i,2]>13.58^X[i,2]<13.906^Y[i,2]<4.86 then goto L3;
      if X[i,0]<Xk^X[i,2]>Xk then
begin ysn:=(Y[i,2]-Y[i,0])/
      (X[i,2]-X[i,0])*(Xk-X[i,2])+Y[i,2];
      if ysn<Yk then goto L3
end;
      goto L4;
L3: Y[i,2]:=-99;
L4:
end;
L5: if Y[i1,2]=-99 then begin i1:=i1+1; goto L5 end;
L6: if Y[in,2]=-99 then begin in:=in-1; goto L6 end;
FIA[j]:=pia:=i1;
for i:=11,i+1 while i<=n do
begin X[i,1]:=X[i,0]; Y[i,1]:=Y[i,0];
      ip:=i-pia;
      if Y[i,0]=-99 then goto L7;
      X[i,0]:=X[i,2]; Y[i,0]:=Y[i,2];
L7: X[ip,0]:=X[i,0]; Y[ip,0]:=Y[i,0];
      X[ip,2]:=X[i,2]; Y[ip,2]:=Y[i,2];
      IA[ip]:=IA[i]; MU[ip]:=MU[i];
      P[ip]:=P[i];
      X[ip,1]:=X[i,1]; Y[ip,1]:=Y[i,1]
end;
i1:=i:=0;
in1:=in:=in-pia;
i00:=i00-pia;
L8: if Y[i,0]=-99^Y[i+1,0]=-99 then goto L9;
      xv:=X[i,0]-X[i+1,0]; yv:=Y[i,0]-Y[i+1,0];
      if xv*xv+yv*yv>0.01 then
begin for h:=in,h-1 while h>1 do
```

```
begin h1:=h+1;
      X[h1,0]:=X[h,0]; Y[h1,0]:=Y[h,0];
      IA[h1]:=IA[h]; MU[h1]:=MU[h];
      P[h1]:=P[h]
end;
i11:=i+1; i2:=i+2;
if j=j00^i<100 then i00:=i00+1;
X[i11,0]:=X[i,0]-xv/2; Y[i11,0]:=Y[i,0]-yv/2;
IA[i11]:=(IA[i]+IA[i2])/2; MU[i11]:=(MU[i]+MU[i2])/2;
if P[i]-P[i2]>pi then P[i2]:=P[i2]+pim2;
if P[i2]-P[i]>pi then P[i]:=P[i]+pim2;
P[i11]:=(P[i]+P[i2])/2;
in:=in+1;
goto L8
end else
L9: begin i:=i+1; if i<in then goto L8 end;
     if 7joke^j=j00 then
begin i02:=i01;
     if i01>in1+pia then
begin i00:=i00-i01+in1+pia; i01:=in1+pia end;
for h:=in,h-1 while h>i00 do
begin h1:=h+at;
      X[h1,0]:=X[h,0]; Y[h1,0]:=Y[h,0];
      IA[h1]:=IA[h]; MU[h1]:=MU[h];
      P[h1]:=P[h]
end;
inarray(drum,20000+(j-1)*600,GH);
inarray(drum,50000+(j-1)*600,GR); hold(GR);
at3:=at-3;
for i:=0,i+1 while i<at3 do
begin h:=i00+2+i;
      A[h]:=-(5+i)*grd;
      IA[h]:=cos(A[h]); MU[h]:=sin(A[h]);
      MAXAF(xk,yk,A[h]);
      X[h,1]:=vh*dx+xk; Y[h,1]:=vh*dy+yk
end;
inarray(drum,20000+j*600,GH);
inarray(drum,50000+j*600,GR); hold(GR);
for i:=0,i+1 while i<at3 do
```

```
begin h:=100+2+i;
      MAXAF(X[h,1],Y[h,1],A[h]);
      X[h,0]:=(X[h,1]+xk+vh×dx)/2;
      Y[h,0]:=(Y[h,1]+yk+vh×dy)/2;
      P[h]:=ARCTAN(X[h,0]-xk,Y[h,0]-yk)
end;
i001:=i00+at;
Y[i00+1,0]:=Y[i001-1,0]:=-99;
X[i001,0]:=X[i00,0]; Y[i001,0]:=Y[i00,0];
LA[i001]:=LA[i00]; MU[i001]:=MU[i00];
P[i001]:=P[i00];
for i:=in1+pie,i-1 while i>i01 do
begin at1:=at+i;
      XT[at1]:=XT[i]; YT[at1]:=YT[i];
      Y[pia+at1,2]:=Y[pia+i,2]
end;
for i:=1,i+1 while i<at do
begin XT[i01+i]:=X[i00+i,0];
      YT[i01+i]:=Y[i01-pia+i,2]:=Y[i00+i,0]
end;
in:=in+at;
at1:=at
end;
for i:=0,i+1 while i<in do Y[i,1]:=Y[i,0];
i:=0;
L10: if Y[i,0]=-99∧Y[i+1,0]=-99 then
begin for k:=i,k+1 while k<in do
begin k1:=k+1;
      X[k,0]:=X[k1,0]; Y[k,0]:=Y[k1,0];
      LA[k]:=LA[k1]; MU[k]:=MU[k1];
      P[k]:=P[k1]
end;
in:=in-1;
goto L10
end;
i:=i+1;
if i+1<in then goto L10;
outarray(drum,4×(j-1)×301,XT);
outarray(drum,2×(2×j-1)×301,YT); hold(YT);
k:=i:=f:=0;
```

```
L11:  if Y[i,2]=Y[k,1] then
      begin  k:=k+1; i:=i+1;
            if j $\neq$ j00 then at1:=0;
            if i<in1+at1 then goto L11 else goto L12
      end else
      begin f:=f+1; PB[f]:=k; k:=k+1; goto L11 end;
L12:  for h:=f+1,h+1 while h<151 do PB[h]:=1000;
      k:=i:=f:=0;
L13:  if Y[k,1]=Y[i,0] then
      begin  k:=k+1; i:=i+1;
            if i<in then goto L13 else goto L14
      end else
      begin  f:=f+1; PA[f]:=i; k:=k+1; goto L13 end;
L14:  for h:=f+1,h+1 while h<151 do PA[h]:=1000;
      outarray(drum,80001+2*(j-1)*150,PB);
      outarray(drum,80001+(2*j-1)*150,PA); hold(PA);
      goto NEXTR;
EIND: CARRIAGE(5); PRINTTEXT(⊥punten optimale route:⊥);
      PRINTTEXT(⊥      x      y      l      b⊥);
      CARRIAGE(2);
      j0:=j:=j-1;
      I[j]:=10;
L15:  inarray(drum,80001+2*(j-1)*150,PB);
      inarray(drum,80001+(2*j-1)*150,PA); hold(PA);
      f:=150;
L16:  if I[j]>PA[f] then I[j]:=I[j]+f else
      begin f:=f-1; if f>0 then goto L16 end;
      f:=150;
L17:  if I[j]>PB[f] then
      begin if I[j]=PB[f] then I[j]:=I[j]+1; I[j]:=I[j]-f end
      else begin f:=f-1; if f>0 then goto L17 end;
      if j=j0 then
      begin xxx:=xe; yyy:=ye end else
      begin xxx:=xx; yyy:=yy end;
      inarray(drum,4*(j-1)*301,XT);
      inarray(drum,2*(2*j-1)*301,YT); hold(YT);
      I[j]:=I[j]+PIA[j];
      xx:=XT[I[j]]; yy:=YT[I[j]];
```

```
l11:=arctan((xx-7.5)/(yy+8.5))/grd+30;
bbb:=90-2*arctan(a0*sqrt((xx-7.5)^2+(yy+8.5)^2))/grd;
ftr:=(636.5*cos(pi4-bbb*grd/2)^2)/ft;
if j=j00-1^(((yyy-yy)/(xxx-xx))*(xk-xxx)+yyy)<yk then
begin  afst1:=afst1+(sqrt((xx-xk)^2+(yy-yk)^2)+
      sqrt((xxx-xk)^2+(yyy-yk)^2))*ftr
end
else afst1:=afst1+sqrt((xxx-xx)^2+(yyy-yy)^2)*ftr;
if j=1 then
afst1:=afst1+sqrt((xx-xs)^2+(yy-ys)^2)*(636.5*cos(pi9)^2)/ft;
NLCR; SPACE(26);
ABSFIXT(2,2,xx); ABSFIXT(4,2,yy);
ABSFIXT(4,2,l11); ABSFIXT(4,2,bbb);
j:=j-1;
if j>0 then
begin  if j=j00-1^|I[j00]|>101^|I[j00]|<101+14
      then I[j]:=i02 else I[j]:=I[j+1];
      goto L15
end;
k:=entier(j0/2);
d:=(j0-2*k+pp)*12;
CARRIAGE(4); PRINTTEXT({reisduur:}); SPACE(23);
PRINTTEXT({dagen          uren}); CARRIAGE(3);
PRINTTEXT({optimale route}); SPACE(19);
ABSFIXT(2,0,k); ABSFIXT(14,1,d);
j:=j0+1;
if xg<xe then
begin
L18:      j:=j+1;
          MAXAF(xg,yg,10);
          if xg>xe then goto L20;
          goto L18
end else
begin
L19:      if XGR[j-1]>xe then
          begin j:=j-1; goto L19 end
end;
```

```
L20:  p:=(j-1+(xe-XGR[j-1])/(XGR[j]-XGR[j-1]))/2;
      k:=entier(p);
      q:=(p-k)*24;
      NLCR; PRINTTEXT(⟨grootcirkel⟩); SPACE(22);
      ABSFIXT(2,0,k); ABSFIXT(14,1,q);
      olt:=afst1/185200-afst2;
      vtu:=p*24-(j0+pp)*12;
      k:=entier(vtu/24);
      if k<0 then k:=0;
      q:=vtu-k*24;
      NLCR; PRINTTEXT(⟨gc-1t⟩); SPACE(28);
      ABSFIXT(2,0,k); ABSFIXT(14,1,q);
      CARRIAGE(4); PRINTTEXT(⟨omweg:⟩);
      SPACE(26); PRINTTEXT(⟨mylen⟩);
      CARRIAGE(3); PRINTTEXT(⟨optimale route⟩); SPACE(18);
      ABSFIXT(3,0,olt);
```

AF:

end