Knmi publicatie met nummer : 102.94

On minimal-time ship routing

S.J. Bijlsma

1975

Summary

Automation of the manual system for the meteorological navigation of ships in the North Atlantic Ocean requires that the following tasks be computerized:

(1) the construction of wind charts from pressure charts of the northern part of the Atlantic Ocean;
(2) the composition of wave charts with the aid of this wind information;
(3) the determination of the least-time track with the assistance of these wave charts.

In this paper the third point is dealt with. The problem of computing the least-time track is a special case of the control problem of Bolza. The problem of Bolza is treated in Section 3. In Section 4 it is shown that, under certain conditions, the solutions of the Euler-Lagrange equations depend continuously on the initial values of the Lagrange multipliers. This property, which holds both in the case of restricted and unrestricted manoeuvrability, forms the basis of a method for the computation of the least-time track. This method is considered in Section 6. An example is shown in Section 7. In Section 5 attention is paid to the problem of minimizing fuel consumption, and in Section 2 we give a description of the coordinate system used in practical computations. Finally an example of a computer program, written in ALGOL-60, is presented in Section 8.

Samenvatting

Automatisering van het manuele systeem voor de meteorologische navigatie van schepen op het noordelijk gedeelte van de Atlantisch Oceaan betekent dat de volgende werkzaamheden verricht zullen moeten worden met behulp van een rekenmachine:

(1) de constructie van windkaarten uit luchtdrukkaarten van het noordelijk gedeelte van de Atlantische Oceaan;
(2) de samenstelling van golfkaarten met behulp van deze windinformatie;
(3) de bepaling van de kortste-vaartijd route met gebruikmaking van deze golfkaarten.

In deze publicatie wordt het derde punt behandeld. Het probleem betreffende het berekenen van de kortste-vaartijd route is een speciaal geval van het probleem van Bolza uit de theorie van optimaal geregelde processen. Het probleem van Bolza wordt besproken in hoofdstuk 3. In hoofdstuk 4 wordt aangetoond dat de oplossingen van de Euler-Lagrange vergelijkingen, onder bepaalde voorwaarden continu afhangen van de beginwaarden van de Lagrange multiplicatoren. Deze eigenschap, die geldt zowel in het geval van beperkte als onbeperkte manoeuvreerbaarheid vormt de basis van een methode voor het berekenen van de korste-vaartijd route. Deze methode wordt beschouwd in hoofdstuk 6. Een voorbeeld wordt getoond in hoofdstuk 7. In hoofdstuk 5 besteden we aandacht aan het probleem van het minimaliseren van het brandstofgebruik en in hoofdstuk 2 geven we een beschrijving van het coördinatensysteem, dat gebruikt wordt bij de werkelijke berekeningen. Een voorbeeld van een rekenprogramma, geschreven in ALGOL-60, wordt tenslotte gepresenteerd in hoofdstuk 8.

Contents

1 Introduction

Optimal ship routing is a term meaning that we seek to find a route for a ship during its transit over the ocean between two given places so that a certain criterion (e.g. transit time, damage to cargo, fuel consumption or comfort of passengers) is optimized. In the case dealt with here the transit time is minimized, and in this connection we speak of minimal-time ship routing. In order to compute the least-time path, all the disturbing forces which may influence the ship on its way must be known. One of the most important forces which a ship encounters is formed by the disturbed ocean surface. As a consequence we assume here that the state of the ocean surface is fully known beforehand.

In practice this is not the case for the time being. Today we work with a weather forecast of 72 hours. In order to recommend the best possible route, this weather forecast is assumed to be valid during this time, leading to a route which may be changed every 12 hours when the next forecast is available. Nevertheless, it is of great importance now, when computing the best possible route step by step, to be compared at a later stage with the optimal route, and in the future, when longer-term forecasts will be available, to have a method at one's disposal for numerical computa-

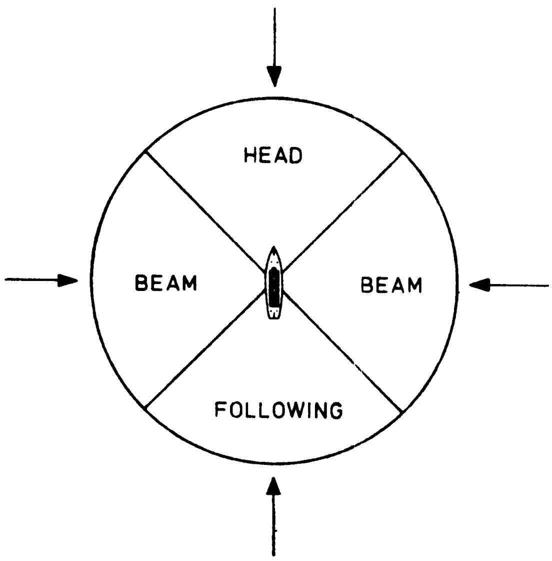


Fig. 1 Definition of bearing of ship to waves (arrow indicates wave motion).

tion of the least-time track on an operational scale. It was the aim of developing such a method which led to this paper.

Let us first consider the history of ship routing. This history goes back to the middle of the nineteenth century, when Maury (1855) recommended least-time tracks for sailing vessels based on statistical wind and current data compiled from ships' logs. In those days the wind direction was of course extremely important.

The conversion from sail to other means of propulsion made Maury's work obsolete. Although developments in aviation led to several investigations on this subject, it was not until more than a century later that James (1957) introduced a manual method for the calculation of least-time tracks for ships. His method can be described as follows. Depending on the relative wave direction resulting in head, beam and following waves (see Fig. 1) he constructed, in an empirical manner, graphs showing the connection between the ship's speed and the significant wave height as illustrated in Fig. 2.

By analysis of these data (including ocean current) the least-time track is constructed by introducing time fronts analogous to wave fronts in geometrical optics (see also Section 4).

The introduction of the computer led to the development of several methods for the computation of the least-time track. These methods can be roughly divided into two groups: one group using networks partly or wholly covering the navigation area (such as developed by Braddock (1968) and Zoppoli (1972)), and a second group

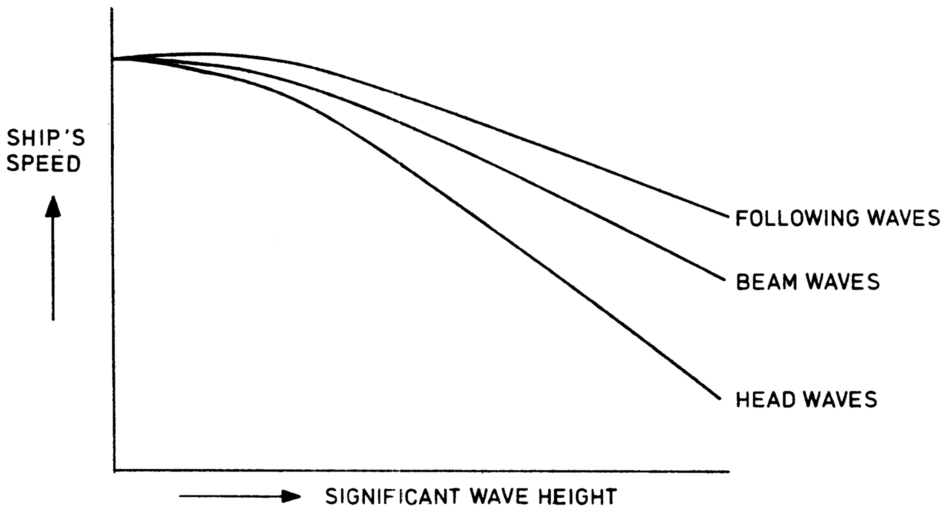


Fig. 2

containing applications of the calculus of variations or optimal control theory (e.g. Bleick and Faulkner (1965) or Marks et al. (1968)). In most methods the ship's velocity is approximated by a polar velocity diagram of elliptic form constructed by means of empirical data as presented in Fig. 2. We shall return to this subject in Section 6. For a theoretical treatment of the response of a ship to waves, the reader is referred to publications such as those of the Society of Naval Architects and Marine Engineers. In our opinion, application of the optimal control theory is preferable for practical reasons (see Section 6) when computing the least-time track. As a consequence the method presented here is an application of results of this theory.

Let us now consider some practical aspects of ship routing at the Royal Netherlands Meteorological Institute (KNMI). The Ship Routing Office was established at the KNMI in 1961. The activities of this Office can be generally outlined as follows.

(a) To supply information it is first necessary to collect a variety of data, such as the synoptic surface pressure analysis of the North Atlantic (delivered twice daily by the Weather Service of the KNMI), 500-mb charts for 72 hours ahead (transmitted by the National Oceanographic and Atmospheric Administration at Suitland), ship's reports containing wind and wave data, usually every 6 hours (from selected ships), information on ocean currents (from the US Hydrographic Office), the occurrence of pack ice, etc. The North Atlantic analyzed surface pressure charts are converted into corresponding wind velocity charts which are corrected by means of reported winds in the ship's reports.

(b) These wind charts are converted into wave charts, containing the significant wave height and direction of sea and swell, by manual analysis of the fetch and duration of the various wave generating areas. The wave charts are in turn corrected by ship's reports on waves.

(c) Given a specified ship and a well-defined operational policy, it is possible to obtain an average speed that the ship can maintain during 6 or 12 hours in different directions subject to the disturbing forces of the ocean surface.

With the aid of the above information the best possible route is recommended. Afterwards, following the manual method of James (1957), a least-time track is additionally constructed using the analyzed wave charts. Finally, sailing times along fixed routes, such as great circle and rhumb line, are calculated.

A survey of the points of departure and arrival of ships handled by the Routing Office is given in Fig. 3. The number of routed ships increased from 38 in 1961 to 711 in 1973. Recently an initial attempt was made at the KNMI to automate the manual routing method by successive numerical computation of wind charts, wave charts and least-time track. Results are found in Bijlsma en Van Rietschote (1972), Bijlsma en

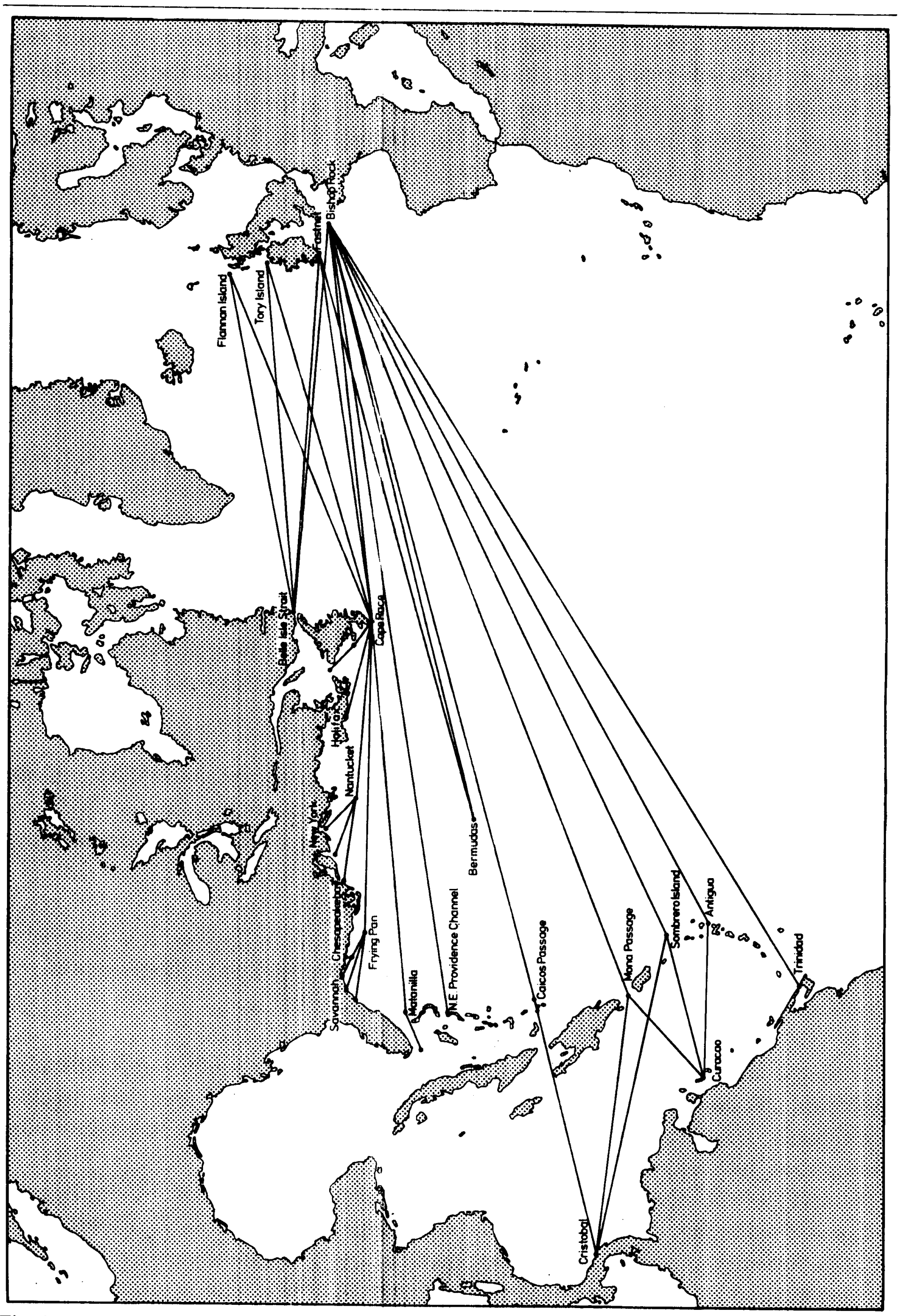

Flannan Island
Tory Island
Fastnet
Bishop Rock
Belle Isle Strait
Cape Race
Halifax
Nantucket
New York
Chesapeake
Frying Pan
Savannah
Matanilla
N.E. Providence Channel
Bermudas
Caicos Passage
Mona Passage
Sombrero Island
Antigua
Trinidad
Curacao
Cristobal

Fig. 3

Folkers (1973) and Bijlsma, Van Rietschote en Folkers (1973). As mentioned above only the numerical method for the computation of the least-time track is dealt with here. Theoretical aspects of this method are considered in Section 4, practical aspects in Section 6. Section 2 contains a description of the coordinate system and Section 3 a summary of relevant results of optimal control theory. An example is presented in Section 7. In Section 5, attention is paid to the problem of minimum-fuel routing.

2 The coordinate system

In order to simplify the computations, the navigation area is mapped conformally onto a plane by means of stereographic projection, which is commonly used in meteorology. We shall give a short description of this projection. In stereographic projection, the earth's surface is mapped onto a plane V, which is parallel or equal to the tangent plane at a point S of the earth's surface, by assigning to each point P of the spherical earth a point P′ which is the point of intersection of V and the straight line through P and S′, the point diametrically opposite S. In the case under consideration, the point S coincides with the north pole and the plane V passes through the circle of 60N latitude (see Fig. 4). We speak of a polar stereographic projection in this case.

Instead of geographical coordinates, we shall use here spherical coordinates θ and ϕ (see Fig. 5).

In the plane V we first introduce polar coordinates r and θ with the mapped north pole as origin and a scale factor s so that the following relation holds (see Fig. 4)

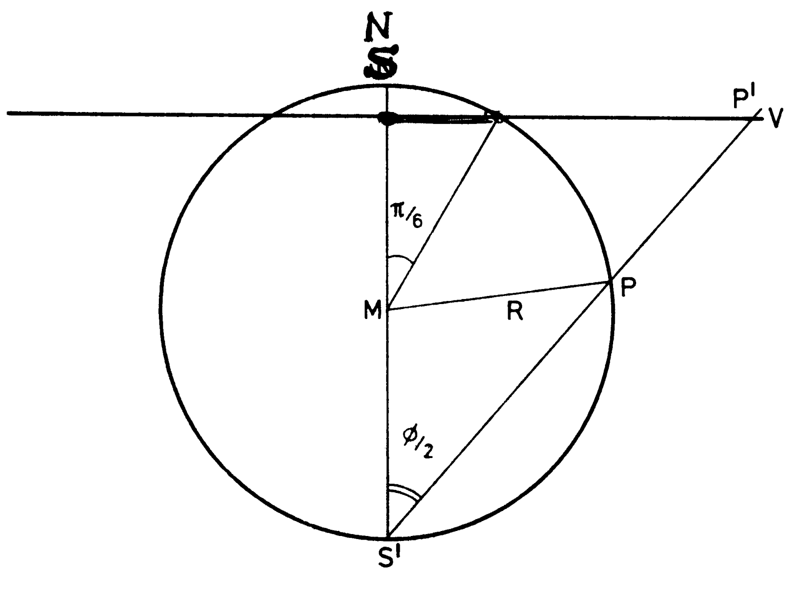


Fig. 4

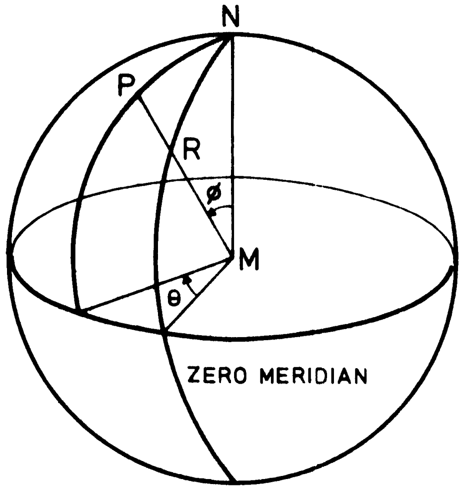


Fig. 5

$$r = \frac{R}{s}(1 + \cos\pi/6)\tan\phi/2.$$

Next we consider an orthogonal coordinate system with coordinates x_1 and x_2 where the x_2 axis is parallel to the projection of the meridian of 30W longitude (see Fig. 6).

The relation between polar and rectangular coordinates is given by

$$\begin{aligned} x_1 &= x_{1n} + \frac{r}{d_0}\sin(\theta - \pi/6), \\ x_2 &= x_{2n} + \frac{r}{d_0}\cos(\theta - \pi/6), \end{aligned} \qquad (2.1)$$

where x_{1n} and x_{2n} are the coordinates of the projected north pole and d_0 the mesh distance corresponding to 3180 km at 60N where the projection is true. From (2.1) we find the inverse transformation

$$\begin{aligned} \theta &= \arctan\{(x_1 - x_{1n}) / (x_2 - x_{2n})\} + \pi/6, \\ \phi &= 2\arctan\{a_0\sqrt{(x_1 - x_{1n})^2 + (x_2 - x_{2n})^2}\} \end{aligned} \qquad (2.2)$$

with

$$a_0 = \frac{s d_0}{R(1 + \cos\pi/6)}\ .$$

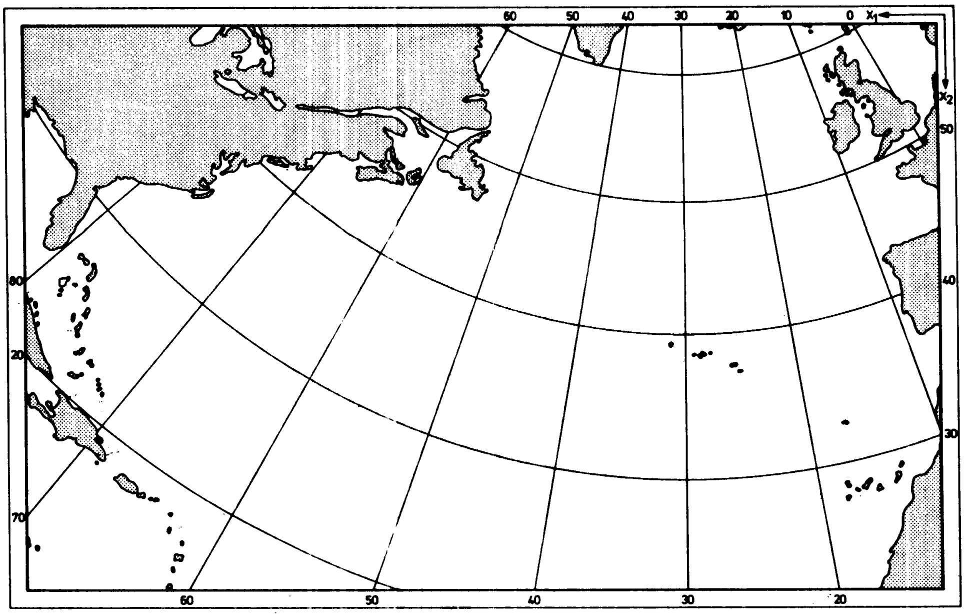


Fig. 6

By differentiating (2.1) we find the local transformation

$$dx_1 = m(\phi)\, R\, \{\sin(\theta - \pi/6)d\phi + \cos(\theta - \pi/6)\sin\phi\, d\theta\},$$

$$dx_2 = m(\phi)\, R\, \{\cos(\theta - \pi/6)d\phi - \sin(\theta - \pi/6)\sin\phi\, d\theta\}$$

with

$$m(\phi) = \frac{1 + \cos\pi/6}{sd_0(1 + \cos\phi)}\ .$$

Hence it follows that differential distances on the earth's surface must be multiplied by the map factor $m(\phi)$ in order to convert them into differential distances in the (x_1, x_2) plane. In the practice of ship routing we are also concerned with sailing along the great circle or rhumb line as we saw in Section 1. We shall therefore give a short derivation of their equations in the (x_1, x_2) or (r, θ) plane.

(a) great circle

Let the pole P of a great circle have the spherical coordinates θ_0 and ϕ_0. Then in the spherical triangle NPQ (see Fig. 7) the relation

$$\cos(\theta - \theta_0) = -(\tan\phi\tan\phi_0)^{-1} \qquad (2.3)$$

holds. Combination of (2.2) and (2.3) gives the equation of the projection of the great circle in the (x_1, x_2) plane, which is a circle again, as was to be expected. As a result we find

$$(x_1 - x_{1m})^2 + (x_2 - x_{2m})^2 = s_0^2$$

with

$$x_{1m} = x_{1n} + s_0 \sin \phi_0 \sin (\theta_0 - \pi/6),$$

$$x_{2m} = x_{2n} + s_0 \sin \phi_0 \cos (\theta_0 - \pi/6) \text{ and } s_0 = (a_0 \cos \phi_0)^{-1}.$$

The spherical coordinates θ_0 and ϕ_0 are simply found by substituting the coordinates of two points of the great circle in (2.3).

(b) rhumb line

Owing to the fact that we are applying conformal mapping, it is clear that the rhumb line is mapped as a logarithmic spiral, with the projection of the north pole as its centre, satisfying the equation

$$r = r_1 \exp \{\alpha(\theta - \theta_1)\}$$

with

$$\alpha = \ln (r_2/r_1) \,/\, (\theta_2 - \theta_1) \text{ and}$$

$$r_i = \frac{R}{s} (1 + \cos \pi/6) \tan \phi_i/2 \qquad (i = 1, 2),$$

where (ϕ_1, θ_1) and (ϕ_2, θ_2) are the coordinates of two points of the rhumb line.

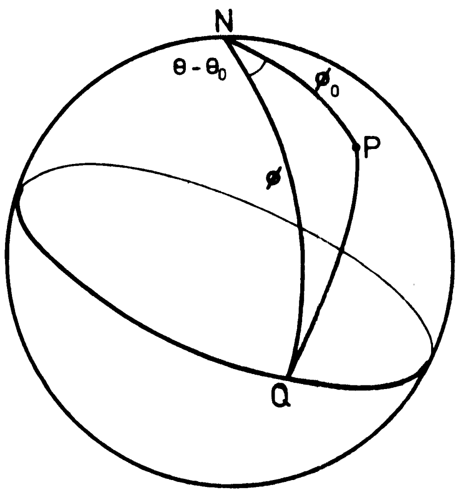


Fig. 7

3 Necessary conditions for a minimum

In this section we shall summarize the conditions necessary for a minimum for the problem of Bolza. As an introduction to this, let us first consider our basic problem of minimizing the transit time for a ship of known performance. The equations of motion of a ship in a Cartesian coordinate system with coordinates x_1 and x_2 are given by

$$\dot{x}_1 = V(t, x_1, x_2, p) \cos p + S_1(t, x_1, x_2),$$
$$\dot{x}_2 = V(t, x_1, x_2, p) \sin p + S_2(t, x_1, x_2).$$

Here the dot denotes differentiation to the time t. Further $V(t, x_1, x_2, p)$ gives the (maximum attainable) speed of the ship relative to the water as a function of the position coordinates x_1 and x_2, the time t and the course p. The functions $S_1(t, x_1, x_2)$ and $S_2(t, x_1, x_2)$ denote the velocity components of the ocean current. In practice these components are nearly time-independent during a winter or summer season.

The path of the ship is completely determined by the initial values of x_1 and x_2 and by the value of $p(t)$ over the voyage, so that the problem is to determine the function

$$p(t) \qquad 0 \leq t \leq t_1$$

which will minimize the transit time t_1 among all paths with prescribed conditions on the initial and final values $x_1(0)$, $x_2(0)$, $x_1(t_1)$ and $x_2(t_1)$. Instead of the problem described above, we consider the following, more general problem in the calculus of variations.

Consider a class of functions

$$u_j(t) \qquad (j = 1, \ldots, p;\ t_0 \leq t \leq t_1)$$

and a class of arcs

$$x_i(t) \qquad (i = 1, \ldots, n),$$

connected by the differential equations

$$\dot{x}_i = f_i(t, x, u) \qquad (u = u_1, \ldots, u_p;\ x = x_1, \ldots, x_n) \tag{3.1}$$

and end conditions

$$x_i(t_0) = x_{i0},\ x_i(t_1) = x_{i1}. \tag{3.2}$$

We now seek to find a set of functions $u_j(t)$ and $x_i(t)$ which minimizes an integral of the form

$$I = \int_{t_0}^{t_1} f_0(t, x, u)dt. \tag{3.3}$$

This problem is called the control problem of Bolza. The functions

$$u_j(t)$$

are called control functions. The problem described initially is clearly a special case of the control problem of Bolza. We now distinguish two cases. First we consider the unrestricted case where no conditions are imposed on the control functions u_j or the coordinates x_i. In the second case, however, these variables are restricted to a closed set.

In practical ship routing this means that the navigation area for the ship is bounded or that certain courses are forbidden.

CASE 1: The unrestricted case

We now return to the control problem of Bolza as formulated above. We restrict ourselves in this case to elements t, $x = (x_1, \ldots, x_n)$ and $u = (u_1, \ldots, u_p)$ which lie in open sets T, X and U. These elements are called admissible. The cases where U or X is a closed set are extensions of the classical calculus of variations and are dealt with in textbooks on optimal control theory (see for instance Pontryagin (1962) or Hestenes (1966)). These cases are considered in the second part of this section.
We consider arcs

$$C: \quad x_i(t), u_j(t) \quad (t_0 \leq t \leq t_1; i = 1, \ldots, n; j = 1, \ldots, p)$$

such that

(a) the functions $u_j(t)$ are continuous on the interval $t_0 \leq t \leq t_1$. In addition the functions $f_i(t, x, u)$ $(i = 0, \ldots, n)$ are assumed to be continuously differentiable of the second order with respect to x and u and continuously differentiable with respect to t.

(b) its elements (t, x, u) are admissible.

Suppose now that an arc

$$C^0: \quad x^0(t), u^0(t)$$

which is normal in the sense described below in this section and satisfies conditions (a) and (b) minimizes (3.3) subject to equations (3.1) and (3.2). Then there exist continuously differentiable multipliers $\lambda(t) = \{\lambda_0(t), \ldots, \lambda_n(t)\}$, $\lambda_0(t) = \text{constant} \leq 0$ not vanishing simultaneously and a function

$$H(t, x, u, \lambda) = \sum_{i=0}^{n} \lambda_i f_i,$$

so that conditions (1), (2), (3) and (4) described below hold on C^0. For a proof of the results of this section one is referred to Hestenes (1966).

(1) The first necessary condition. On C^0 the Euler-Lagrange equations

$$\dot{x}_i = H_{\lambda_i}, \dot{\lambda}_i = -H_{x_i}, H_{u_j} = 0 \qquad (i = 1, \ldots, n; j = 1, \ldots, p)$$

hold and also the equation

$$\frac{dH}{dt} = H_t.$$

Variables as subscripts denote partial differentiation. In the case of movable ends the differentials of x_{is} $(s = 0, 1)$ are connected by the transversality condition

$$\sum_{i=1}^{n} \lambda_i(t_s)\, dx_{is} = 0 \qquad (s = 0, 1).$$

(2) The necessary condition of Weierstrass. Along C^0 the inequality

$$H(t, x^0, u, \lambda) \leq H(t, x^0, u^0, \lambda)$$

must hold for every admissible element (t, x^0, u). In addition

$$H\{t_s, x^0(t_s), u^0(t_s), \lambda(t_s)\} = 0.$$

The generalization of the conditions (1) and (2) in the case where U is a closed set is known as Pontryagin's maximum principle and is considered in the second part of this section. Solutions of the Euler-Lagrange equations with continuous control functions are called extremals. As an immediate consequence of the condition of Weierstrass we have

(3) The necessary condition of Legendre. At each element (t, x^0, u^0, λ) of C^0 the inequality

$$\sum_{j,k=1}^{p} H_{u_j u_k} \xi_j \xi_k \leq 0$$

holds for any non-zero $\xi = (\xi_1, \ldots, \xi_p)$. If the determinant

$$| H_{u_j u_k} | \qquad (j, k = 1, \ldots, p)$$

is different from zero the arc C^0 will be called non-singular. This concept of non-singularity will be dealt with in more detail in Section 4. There still remains one necessary condition. In order to describe this condition we introduce the concept of variation.

Let

$$C^{\varepsilon}: \qquad x_i(t, \varepsilon), u_j(t, \varepsilon) \qquad (i = 1, \ldots, n; j = 1, \ldots, p)$$

be a one-parameter family of arcs satisfying equations (3.1) and (3.2) and containing C^0 for $\varepsilon = 0$. Since equations (3.1) and (3.2) are identities in ε we find, taking derivatives with respect to ε at $\varepsilon = 0$,

$$\dot{y}_i = \sum_{h=1}^{n} A_{ih} y_h + \sum_{j=1}^{p} B_{ij} v_j \qquad (i = 1, \ldots, n), \tag{3.4}$$

$$y_i(t_s) = 0 \qquad (s = 0, 1), \tag{3.5}$$

where

$$A_{ih} = f_{ix_h}, \; B_{ij} = f_{iu_j}, \; y_i(t) = x_{i\varepsilon}(t, 0) \text{ and } v_j(t) = u_{j\varepsilon}(t, 0).$$

The arc

$$\gamma: \qquad y_i(t), v_j(t) \qquad (i = 1, \ldots, n; j = 1, \ldots, p; \; t_0 \leq t \leq t_1)$$

is called the variation of the family C^{ε} along x^0 if the functions $v_j(t)$ are continuous and the functions $y_i(t)$ have continuous derivatives for $t_0 \leq t \leq t_1$. We are now in a position to define normality. The arc C^0 will be called normal if there exist $2n$ variations

$$\gamma^{\sigma}: \qquad y_i^{\sigma}(t), v_j^{\sigma}(t) \qquad (\sigma = 1, \ldots, 2n)$$

satisfying equation (3.4) such that no linear combination of these arcs satisfies equation (3.5). This is the case if and only if the determinant

$$\begin{vmatrix} y_i^{\sigma}(t_0) \\ y_i^{\sigma}(t_1) \end{vmatrix} \qquad (i = 1, \ldots, n; \sigma = 1, \ldots, 2n)$$

is different from zero. Normality ensures the existence of the one-parameter family C^ε. The second variation $I_2(\gamma)$ of I along C^0 is obtained by evaluating I along the family C^ε and then differentiating twice to ε at $\varepsilon = 0$. This second variation is given by

$$I_2(\gamma) = -\int_{t_0}^{t_1} 2\omega\{t, y(t), v(t)\}\,dt,$$

where

$$2\omega(t, y, v) = \sum_{i,h=1}^{n} H_{x_i x_h} y_i y_h + 2\sum_{i=1}^{n}\sum_{k=1}^{p} H_{x_i u_k} y_i v_k + \sum_{j,k=1}^{p} H_{u_j u_k} v_j v_k.$$

The fourth necessary condition can now be formulated as follows.

(4) The second-order condition. The second variation $I_2(\gamma)$ of I along C^0 is such that

$$I_2(\gamma) \geq 0$$

for every variation γ satisfying equations (3.4) and (3.5).

As a consequence of normality, the equality sign for the multiplier λ_0 is excluded, so that by choosing $\lambda_0 = -1$ the multipliers $\lambda_i(t)$ $(i = 1, \ldots, n)$ are unique. In the practical problems discussed in this paper, the condition of normality is assumed to be fulfilled.

CASE 2: The restricted case

The results of the foregoing part of this section can be generalized in the case where the arcs

$$C: \quad x_i(t),\ u_j(t)$$

not only satisfy equations (3.1) and (3.2) but also a set of additional constraints. Since we are primarily interested in the governing equations of our problem contained in the necessary conditions (1) and (2), we shall only consider generalizations of these conditions. First we assume that U is a closed set.

U IS A CLOSED SET.

We consider inequality constraints of the form

$$\phi_l(t, x, u) \leq 0 \qquad (l = 1, \ldots, q) \tag{3.6}$$

which involve the control variable explicitly. The functions $\phi_l(t, x, u)$ are assumed to be continuously differentiable. An element (t, x, u) is called admissible if the sets T and X are open, while u is restricted to a closed set U. Let

$$C^0: \qquad x^0(t), u^0(t)$$

be an arc which satisfies conditions (a) and (b) and minimizes (3.3) subject to equations (3.1), (3.2) and condition (3.6). Let moreover the matrix

$$(\phi_{lu_k}) \qquad (l = l_1, \ldots, l_r; \; k = 1, \ldots, p)$$

have rank r at each point (t, x, u) satisfying $\phi_l = 0$ $(l = l_1, \ldots, l_r)$. Then there exist continuously differentiable and continuous multipliers $\lambda(t) = \{\lambda_0(t), \ldots, \lambda_n(t)\}$, $\lambda_0 =$ constant ≤ 0 and $\mu(t) = \{\mu_1(t), \ldots, \mu_q(t)\}$, and a function

$$H(t, x, u, \lambda, \mu) = \sum_{i=0}^{n} \lambda_i f_i - \sum_{l=1}^{q} \mu_l \phi_l,$$

so that conditions (1) and (2) described below hold on C^0.

(1) The first necessary condition. On C^0 the equations

$$\dot{x}_i = H_{\lambda_i}, \; \dot{\lambda}_i = -H_{x_i}, \; H_{u_j} = 0, \; \phi_l \leq 0 \qquad (i = 1, \ldots, n; \; j = 1, \ldots, p; \; l = 1, \ldots, q)$$

hold and also the equation

$$\frac{dH}{dt} = H_t.$$

Moreover $\mu_l(t) \geq 0$ $(l = 1, \ldots, q)$ where $\mu_l(t) = 0$ at each point of C^0 at which $\phi_l < 0$.

(2) The second necessary condition. Along C^0 the inequality

$$H(t, x^0, u, \lambda, 0) \leq H(t, x^0, u^0, \lambda, 0)$$

must hold for every admissible element (t, x^0, u) for which $\phi_l(t, x^0, u) \leq 0$.

So far we have considered the maximum principle contained in conditions (1) and (2) on an open set X. We now continue with the case where the trajectory x lies partly or wholly on the boundary of a closed set X, i.e. we consider constraints of the form

$$\psi_m(x) \leq 0 \qquad (m = 1, \ldots, r). \tag{3.7}$$

The functions $\psi_m(x)$ are assumed to be continuously differentiable of the second order.

The set U may be open or closed. Analogous to the foregoing, the concept of admissibility is defined.

X IS A CLOSED SET.

As before, we consider an arc

$$C^0: \qquad x^0(t), u^0(t)$$

which satisfies equations (3.1), (3.2) and condition (3.7), and minimizes (3.3).

Let the matrix

$$\left(\sum_{i=1}^{n} \psi_{mx_i} f_{iu_k}\right) \qquad (m = 1, \ldots, r;\ k = 1, \ldots, p)$$

have rank r. Then there exist continuously differentiable and continuous multipliers $\lambda(t) = \{\lambda_0(t), \ldots, \lambda_n(t)\}$, $\lambda_0 = \text{constant} \le 0$ and $v(t) = \{v_1(t), \ldots, v_r(t)\}$, and a function

$$H(t, x, u, \lambda, v) = \sum_{i=0}^{n} \lambda_i f_i - \sum_{m=1}^{r} \sum_{i=1}^{n} v_m \psi_{mx_i} f_i ,$$

so that the following equations hold on C^0.

(1) The first necessary condition. On C^0 the equations

$$\dot{x}_i = H_{\lambda_i}, \dot{\lambda}_i = -H_{x_i}, H_{u_j} = 0 \qquad (i = 1, \ldots, n; j = 1, \ldots, p)$$

hold and also

$$\frac{dH}{dt} = H_t.$$

Moreover the multipliers $v_m(t)$ $(m = 1, \ldots, r)$ are non-increasing on $t_0 \le t \le t_1$ and are constant on every interval on which

$$\psi_m\{x^0(t)\} < 0.$$

(2) The second necessary condition. Along C^0 the inequality

$$H(t, x^0, u, \lambda, v) \le H(t, x^0, u^0, \lambda, v)$$

must hold for every admissible element (t, x^0, u) for which $\psi_m(x^0) \leq 0$.

Moreover, if U is a closed set, u must also satisfy conditions of the form (3.6). This concludes the results of the general theory. In the next section we shall apply these results to the problem of ship routing in order to derive the necessary equations for this problem.

4 Time fronts and extremals

In this section we return to the problem of minimizing the transit time. We shall derive some properties of extremals which form the basis of a method for the computation of the least-time track. The problem under consideration was to determine a function

$$p(t) \qquad 0 \leq t \leq t_1,$$

so that it minimizes the transit time t_1 subject to the equations

$$\dot{x}_1 = V(t, x_1, x_2, p) \cos p + S_1(t, x_1, x_2), \tag{4.1}$$

$$\dot{x}_2 = V(t, x_1, x_2, p) \sin p + S_2(t, x_1, x_2) \tag{4.2}$$

with

$$x_i(0) = x_{io}, \; x_i(t_1) = x_{i1} \qquad (i = 1, 2).$$

Let us first consider the case of unlimited manoeuvrability.

CASE 1: Unlimited manoeuvrability

Application of the results of Section 3 with $n = 2$, $u_1 = p$, $t_0 = \text{fixed} = 0$, $f_0 = 1$, $f_1 = V \cos p + S_1$ and $f_2 = V \sin p + S_2$ yields the equations

$$\dot{\lambda}_1 = -\sum_{i=1}^{2} \lambda_i (V_{ix_1} + S_{ix_1}), \tag{4.3}$$

$$\dot{\lambda}_2 = -\sum_{i=1}^{2} \lambda_i (V_{ix_2} + S_{ix_2}), \tag{4.4}$$

$$\sum_{i=1}^{2} \lambda_i V_{ip} = 0, \tag{4.5}$$

where $V_1 = V \cos p$ and $V_2 = V \sin p$.

First of all we would point out that the H-function vanishes along an extremal, so that

$$\sum_{i=1}^{2} \lambda_i (V_i + S_i) = -\lambda_0 > 0. \tag{4.6}$$

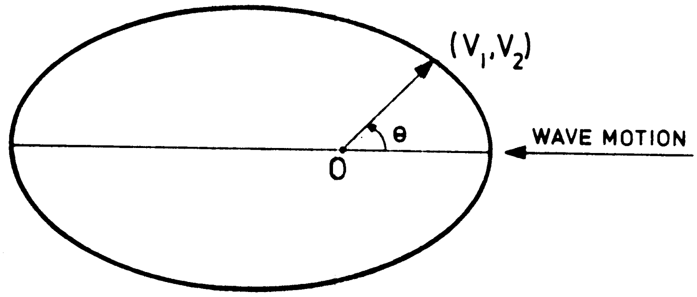


Fig. 8 The ship is situated at O. θ is the angle between the ship's heading and the wave direction.

A second remark concerns the case of a movable end point. In the case of an optimal transit time t_1 differential changes in the coordinates of the end point are connected by the relation

$$\sum_{i=1}^{2} \lambda_i(t_1)dx_{i1} = 0. \tag{4.7}$$

Equations (4.5), (4.6) and (4.7) have a simple geometrical meaning, as we shall see below. At this point it is convenient to introduce a polar velocity diagram as shown in Fig. 8, giving the ship's velocity as a function of the angle between the ship's heading and the wave direction for fixed values of x_1, x_2 and t (see Section 6). This velocity diagram will be called the indicatrix after Caratheodory (1904), (see also Bolza (1909)), although strictly speaking the ocean current should also have been taken into account.

Next we consider a theorem on systems of differential equations

$$\dot{y}_k = f_k(t, y_1, \ldots, y_n, s) \qquad (k = 1, \ldots, n)$$

depending on a parameter s. For a proof one is referred to Coddington and Levinson (1955, p. 58).

THEOREM 1.

For $k = 1, \ldots, n$ the following assumptions are made concerning the real functions $f_k(t, y_1, \ldots, y_n, s)$.

(a) $f_k(t, y_1, \ldots, y_n, s)$ is a continuous and bounded function of the variables $y_1, \ldots, y_n$, s and t for $a < t < b$.

(b) for $s = s_0$ and $\alpha_k(s_0) = \alpha_k^0$ there exists a unique system of functions $y_k(t, s_0) = y_k^0(t)$ satisfying the equations

$$y_k(t, s) = \alpha_k(s) + \int_{t_0}^{t} f_k\{t, y_1(t, s), \ldots, y_n(t, s), s\}\, dt \qquad (4.8)$$

with $a < t_0 < b$.

Let moreover

$$\lim_{s \to s_0} \alpha_k(s) = \alpha_k^0,$$

then every solution $y_k(t, s)$ of (4.8) also satisfies the relation

$$\lim_{s \to s_0} y_k(t, s) = y_k^0(t).$$

In conclusion we give a theorem on implicit functions. For a proof one is referred to Hestenes (1966, p. 22).

THEOREM 2.

Consider a system of n simultaneous equations

$$f_i(z_1, \ldots, z_m, y_1, \ldots, y_n) = 0 \quad (i = 1, \ldots, n).$$

We assume that

(a) the functions $f_i(z_1, \ldots, z_m, y_1, \ldots, y_n)$ are real continuous functions on an open set S.

(b) the partial derivatives

$$f_{iy_j} \qquad (i, j = 1, \ldots, n)$$

are continuous on S. Let the functional determinant be given by

$$D = |f_{iy_j}|.$$

Suppose that the relations

$$f_i(z_1^0, \ldots, z_m^0, y_1^0, \ldots, y_n^0) = 0, D(z_1^0, \ldots, z_m^0, y_1^0, \ldots, y_n^0) \neq 0$$

hold at a point $(z_1^0, \ldots, z_m^0, y_1^0, \ldots, y_n^0)$ in S. Then there exist continuous functions

$$y_i(z) = y_i(z_1, \ldots, z_m)$$

in a neighbourhood of z^0 such that

$$y_i(z^0) = y_i^0, f_i\{z, y_1(z), \ldots, y_n(z)\} = 0 \quad (i = 1, \ldots, n)$$

hold only in case

$$y_i = y_i(z).$$

We now return to equation (4.5). The geometrical meaning of this equation is illustrated in Fig. 9. Instead of considering p as a function of the time t, we now regard p as a function of the variables x_1, x_2, λ_1, λ_2 and t given by (4.5).

In fact p is a function of arctan (λ_2/λ_1) as easily follows by writing equation (4.5) in the form

$$p = \arctan(\lambda_2/\lambda_1) + \arctan(V_p/V).$$

In the following we are interested in continuous functions $p(t, x_1, x_2, \lambda_1, \lambda_2)$. Application of Theorem 2 to equation (4.5) yields the non-singularity condition

$$\sum_{i=1}^{2} \lambda_i V_{ipp} \neq 0, \tag{4.9}$$

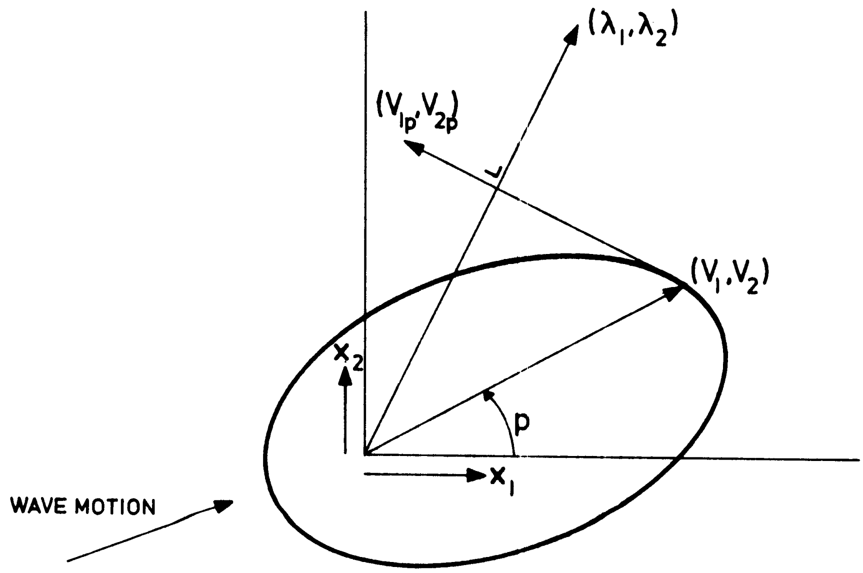


Fig. 9 In view of the Weierstrass or Legendre condition there is a unique choice for p. The course p is measured as indicated.

so that in view of the Legendre condition

$$\sum_{i=1}^{2} \lambda_i V_{ipp} < 0. \tag{4.10}$$

In addition V, V_p and V_{pp} must be continuous functions with respect to x_1, x_2, p and t.

Combination of (4.5) and (4.9) gives

$$V^2 + 2V_p^2 - VV_{pp} \neq 0. \tag{4.11}$$

The geometrical interpretation of this result is elementary, as illustrated in Fig. 10 (see for instance Bolza (1909, p. 369)).

Writing $\lambda_1(0) = \cos a$ and $\lambda_2(0) = \sin a$, Theorem 1 can be applied to equations (4.1), (4.2), (4.3), (4.4) with (4.5) if we set $t_0 = 0$, $s = a$, $y_1 = x_1$, $y_2 = x_2$, $y_3 = \lambda_1$, $y_4 = \lambda_2$, $\alpha_1(a) = x_{10}$, $\alpha_2(a) = x_{20}$, $\alpha_3(a) = \cos a$, $\alpha_4(a) = \sin a$, $f_1 = V \cos p + S_1$, $f_2 = V \sin p + S_2$, $f_3 = -\sum_{i=1}^{2} \lambda_i(V_{ix_1} + S_{ix_1})$ and $f_4 = -\sum_{i=1}^{2} \lambda_i(V_{ix_2} + S_{ix_2})$ and if we assume that the right-hand sides of (4.1), (4.2), (4.3) and (4.4) with (4.5) are continuous

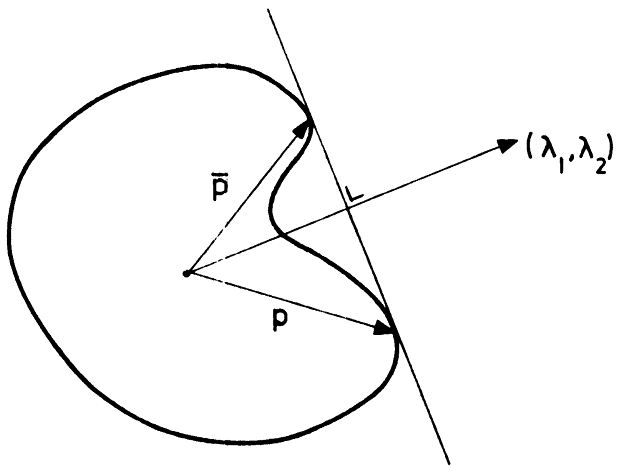


Fig. 10 Condition (4.11) implies that the indicatrix must be convex, so that the tangents for p and $\bar{p}$ perpendicular to the direction (λ_1, λ_2) cannot coincide, which would lead to a corner in the corresponding solution.

and bounded, and satisfy the Lipschitz condition so that condition (b) of Theorem 1 is fulfilled. In view of the homogeneity of equations (4.3), (4.4) and (4.5) it is clear that all extremals starting from (x_{10}, x_{20}) are found by varying parameter a. Summarizing the foregoing, we have the following result.

RESULT 1.

Let the functions V, V_p, V_{pp}, V_{x_1}, V_{x_2}, S_i, S_{ix_1} and S_{ix_2} $(i = 1, 2)$ be continuous for $0 \leq t \leq t_1$ and let relation (4.10) be valid. Futher let the right-hand sides of equations (4.1), (4.2), (4.3) and (4.4) with (4.5) be bounded and satisfy a Lipschitz condition with respect to x_1, x_2, λ_1 and λ_2. Then $x_i(t, a)$ and $\lambda_i(t, a)$ $(i = 1, 2;\ 0 \leq t \leq t_1)$ as solutions of equations (4.1), (4.2), (4.3) and (4.4) with $x_i(0) = x_{i0}$ $(i = 1, 2)$, where p is given by (4.5), are continuously differentiable with respect to t and continuous in their dependence on the parameter a defined by $\lambda_1(0) = \cos a$ and $\lambda_2(0) = \sin a$.

Result 1 enables us to introduce a numerical method for the computation of the least-time track which appears to be very suitable in practical cases. The practical aspects of this method are discussed in Section 6. Let us proceed with a geometrical interpretation of the foregoing. Integrating the system of equations and varying parameter a, we find after a time τ a set of points $S(\tau)$ which can be reached ultimately. For the sake of simplicity we first restrict ourselves to a field of extremals, which means that the extremals starting from the point O at time $t = 0$ (see Fig. 11) do not intersect. The set of points $S(\tau)$ is called a time front. Let $\{x_1(\tau, a), x_2(\tau, a)\}$ be a point of $S(\tau)$. We further assume that the tangent $\{x_{1a}(\tau, a), x_{2a}(\tau, a)\}$ in this point exists. Let us now consider $\{x_1(\tau, a), x_2(\tau, a)\}$ as a movable end point. Since $x_i(t, a)$ $(i = 1, 2;\ 0 \leq t \leq \tau)$ is a solution of our optimal problem with movable ends, the differentials of the coordinates $\{x_1(\tau, a), x_2(\tau, a)\}$ with respect to a must satisfy relation (4.7), yielding

$$\sum_{i=1}^{2} \lambda_i(\tau, a)\, x_{ia}(\tau, a) = 0.$$

In view of this relation one can construct time fronts in a geometrical manner by determining the direction of the normal to $S(\tau)$ and applying the construction of Fig. 11. According to equation (4.6), the angle between the normal to $S(\tau)$ and $(V_1 + S_1, V_2 + S_2)$ must be acute. The first time front which contains the fixed end point obviously yields the least-time track.

Let us now consider the more realistic situation that usually occurs in practice when a part of the one-parameter family of extremals has an envelope C, as sketched for instance in Fig. 12.

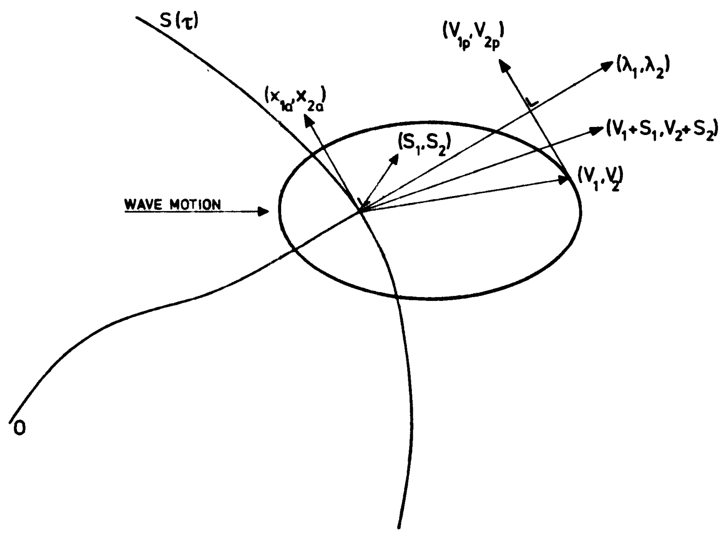


Fig. 11

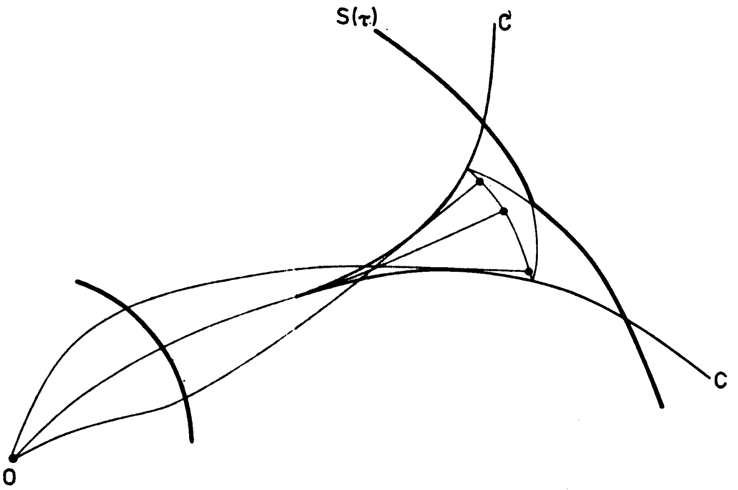


Fig. 12 $S(\tau)$, the set of ultimately attainable points at time $t = \tau$ for a ship starting at time $t = 0$ at O, is indicated by a heavy line.

The curves C are called caustic curves. In particular they can degenerate into one point. The tangent point of an extremal with an envelope is called a conjugate point to O. As a consequence of the fourth necessary condition a non-singular extremal between two points cannot be a minimal curve if it contains a conjugate point between those points. This condition is called the necessary condition of Jacobi. The necessary conditions of Euler-Lagrange (I), Weierstrass (II), Legendre (III) and Jacobi (IV) from the classical calculus of variations were in fact discovered in the sequence I, III, IV, II. An arc satisfying these four conditions furnishes a relative minimum for our problem. In order to find an absolute minimum we have to consider the time front $S(\tau)$ (Fig. 12), which is in fact the boundary of the set of points that could possibly be reached at the time $t = \tau$. An extremal $x_i(t)$ $(i = 1, 2)$ furnishes an absolute minimum for our problem if $x_i(\tau)$ $(i = 1, 2)$ belongs to the boundary $S(\tau)$ for all τ with $0 \leq \tau \leq t_1$. This property has been proved by Halkin (1964, see p. 9).

CASE 2: Modifications in the case of prohibited courses

In the first part of this section we ignored the fact that some courses could be forbidden. We shall consider this case here. The prohibition of a course is due to heavy rolling which occurs if the course is at a certain angle, depending on the ship's velocity and wave period, with respect to the wave direction. The polar velocity diagram is changed for instance as indicated in Fig. 13.

The course p is here restricted to sector I or II determined by the angles $M_l(t, x_1, x_2)$ $(l = 1, 2, 3, 4)$. For courses p which are interior courses of these sectors, the above equations remain valid. If p would exceed one of the boundary courses, say M_j, it is equalized to this value. According to the theory of Section 3, equations (4.3) and (4.4) change into

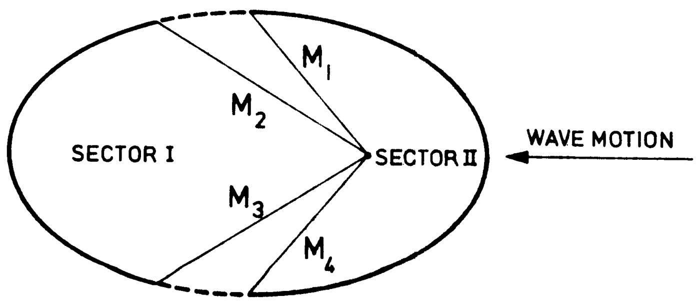


Fig. 13 The non-coinciding boundary courses $M_l(t, x_1, x_2)$ $(l = 1, 2, 3, 4)$ are measured in the same way as the course p.

$$\dot{\lambda}_1 = -\sum_{i=1}^{2} \lambda_i (V_{ix_1} + S_{ix_1}) - \mu_j M_{jx_1}, \tag{4.12}$$

$$\dot{\lambda}_2 = -\sum_{i=1}^{2} \lambda_i (V_{ix_2} + S_{ix_2}) - \mu_j M_{jx_2}, \tag{4.13}$$

where μ_j is a Lagrange multiplier given by (4.15).

Let us now consider the variables p and μ_j in their dependence on x_1, x_2, λ_1, λ_2 and t. We start with the course p. If p is an interior course of sector I or II (we first assume x_1, x_2 and t to be fixed) then $p(t, x_1, x_2, \lambda_1, \lambda_2)$ is a continuous function of $\arctan(\lambda_2/\lambda_1)$. As soon as this angle equals or exceeds a boundary value, say $\arctan\{\lambda_2(M_j)/\lambda_1(M_j)\}$, p obtains the value $M_j(t, x_1, x_2)$. So at the boundary the following relation holds

$$p = \begin{cases} M_j(t, x_1, x_2) & \text{if } \arctan(\lambda_2/\lambda_1) \geq \arctan\{\lambda_2(M_j)/\lambda_1(M_j)\}. \\ p(t, x_1, x_2, \lambda_1, \lambda_2) & \text{otherwise.} \end{cases} \tag{4.14}$$

Of course the inequality sign may be changed in the opposite sense, depending on the situation.

If we require $M_l(t, x_1, x_2)$ $(l = 1, 2, 3, 4)$ to be continuous functions of x_1, x_2 and t and moreover if p is restricted to lie in one of sectors I or II, then p is continuous in its dependence on x_1, x_2, λ_1, λ_2 and t. With respect to μ_j we have

$$\mu_j = \begin{cases} \sum_{i=1}^{2} \lambda_i V_{ip} \text{ with } p = M_j & \text{if } \arctan(\lambda_2/\lambda_1) \geq \arctan\{\lambda_2(M_j)/\lambda_1(M_j)\}. \\ 0 & \text{otherwise.} \end{cases} \tag{4.15}$$

We now return to equations (4.1), (4.2), (4.12) and (4.13). Analogous to the previous case we have the following result.

RESULT 2.

Let $M_j(t, x_1, x_2)$ and M_{jx_i} $(j = 1, 2, 3, 4;\ i = 1, 2)$ be continuous with respect to x_1, x_2 and t. Further let the conditions of Result 1 be fulfilled for courses in sectors I and II, including the boundary courses for $0 \leq t \leq t_1$. Then $x_i(t, a)$ and $\lambda_i(t, a)$ $(i = 1, 2;\ 0 \leq t \leq t_1)$ as solutions of equations (4.1), (4.2), (4.12) and (4.13) with $x_i(0) = x_{i0}$ $(i = 1, 2)$, where p and μ_j $(j = 1, 2, 3, 4)$ are given by (4.14) and (4.15), are continuously differentiable with respect to t and continuous in their dependence on the parameter a defined by $\lambda_1(0) = \cos a$ and $\lambda_2(0) = \sin a$.

Regions of limited and unlimited manoeuvrability are assumed to be separated from each other by closed curves in the (x_1, x_2) plane changing continuously with time and containing the limiting points where M_1 and M_2 as well as M_3 and M_4 coincide. So that extremals along which the conditions of Result 1 and Result 2 are satisfied still depend continuously on their initial values, we have to make additional assumptions with respect to λ_1 and λ_2 along the boundary curves. The range of values of λ_1 and λ_2 along these curves must be such that the corresponding courses p in the regions of limited and unlimited manoeuvrability pass into each other continuously. When computing the least-time track afterwards we shall not use this concept of limited manoeuvrability, because the least-time track will never go through such a region of higher waves. To conclude this section we shall pay some attention to the behaviour of extremals near a fixed boundary. In practical ship routing this boundary may be formed by land or ice. Assuming that such a boundary can be described locally by functions which are linear in the variables x_1 and x_2, say $c_1x_1 + c_2x_2 \leq 0$, the following equations hold

$$\dot{\lambda}_1 = -\sum_{i=1}^{2} (\lambda_i - c_i v_j)(V_{ix_1} + S_{ix_1}),$$

$$\dot{\lambda}_2 = -\sum_{i=1}^{2} (\lambda_i - c_i v_j)(V_{ix_2} + S_{ix_2}),$$

$(j = 1, \ldots, n$; where n is the number of inequalities needed to describe the whole boundary) where v_j is again a Lagrange multiplier analogous to (4.15). In connection with the position of beginning and end points in practical cases it is assumed here that extremals either touch these boundaries, so that $v_j = 0$, or will be cancelled.

5 Minimum fuel consumption

For the sake of completeness, we shall now consider the problem of minimizing costs during transit, apart from the costs determined by the time of arrival, which in general are minimized by minimizing the transit time. It is supposed here that the costs of a ship during an ocean crossing are mainly due to fuel consumption. As a result, it is the minimization of this quantity which will be discussed. We therefore first consider the speed V as a new control variable. Moreover, we assume that the rate of decrease of fuel can be described by the equation

$$\dot{x}_0 = f_0(t, x_1, x_2, V, p), \tag{5.1}$$

where $x_0(t)$ denotes the fuel as a function of the time t. According to the theory of Section 3, we now seek to find functions $p(t)$ and $V(t)$ satisfying the equations

$$\dot{x}_1 = V \cos p + S_1(t, x_1, x_2), \tag{5.2}$$

$$\dot{x}_2 = V \sin p + S_2(t, x_1, x_2) \tag{5.3}$$

with

$$x_i(0) = x_{i0},\ x_i(t_1) = x_{i1} \qquad (i = 1, 2)$$

which minimize

$$\int_0^{t_1} f_0(t, x_1, x_2, V, p)\, dt. \tag{5.4}$$

Of course the speed V is restricted to a range of values given by

$$V_{\min}(t, x_1, x_2, p) \leq V \leq V_{\max}(t, x_1, x_2, p),$$

where $V_{\max}(t, x_1, x_2, p)$ denotes the maximum attainable speed depending on wave height and wave direction and $V_{\min}(t, x_1, x_2, p)$ an acceptable minimum. Since we assume that V will not take these boundary values, the corresponding Lagrange multipliers vanish. Moreover, we choose $\lambda_0 = -1$. For the sake of simplicity we neglect ocean current. Application of the theory of Section 3 yields

$$\dot{\lambda}_1 = f_{0x_1}, \tag{5.5}$$

$$\dot{\lambda}_2 = f_{0x_2}, \tag{5.6}$$

$$-f_{0V} + \lambda_1 \cos p + \lambda_2 \sin p = 0, \tag{5.7}$$

$$-f_{0p} - \lambda_1 V \sin p + \lambda_2 V \cos p = 0. \tag{5.8}$$

Since the H-function vanishes along an extremal we also have

$$-f_0 + \lambda_1 V \cos p + \lambda_2 V \sin p = 0. \tag{5.9}$$

From (5.7) and (5.9) it follows that the speed V along an extremal must be so chosen that it maximizes the quotient

$$\frac{V}{f_0(t, x_1, x_2, V, p)}. \tag{5.10}$$

As a result we could prescribe in every point of the (x_1, x_2) plane an optimal speed $V(t, x_1, x_2, p)$ satisfying (5.10), leading to a more direct approach to our problem analogous to the time-optimal case. As before we can derive conditions which must be satisfied so that the solutions of equations (5.2), (5.3), (5.5), (5.6), (5,7), (5.8) and (5.9) depend continuously on their initial values.

Let the initial values of the multipliers λ_1 and λ_2 be denoted by a and b. Then because of the inhomogeneity of equations (5.5) through (5.9) we must consider $x_i(t, a, b)$ and $\lambda_i(t, a, b)$ $(i = 1, 2)$ in the (a, b) plane. Although it is possible to proceed in this way, it seems more sensible in view of the accuracy of the fuel functions $f_0(t, x_1, x_2, V, p)$, which are derived from empirical data, to apply suitable approximations in practical cases. For instance, if we assume that $f_0\{t, x_1, x_2, V(t, x_1, x_2, p), p\}$ does not depend on x_1, x_2 and p, we can apply the results of the preceding section.

6 The computation of the least-time track

In this section we shall apply the results of Section 4 to the computation of the least-time track. In this connection some remarks will be made on practical aspects of the problem.

(1) When solving equations (4.1), (4.2), (4.3), (4.4) with (4.5) numerically, the time step is determined by the fact that wave charts are available every 12 hours. Depending on the ship's speed one can introduce 6-hour or 12-hour time steps interpolating between two successive wave charts, so that the distance which can be covered by the ship is of the order of magnitude of the mesh distance. As a consequence of practical experience, it is assumed that the wave height and direction between grid points can be obtained by bilinear interpolation.

(2) A second point concerns the approximation of the derivatives with respect to the coordinates x_1 and x_2 in the grid. If we denote the x_1 and x_2 coordinates of a grid point by (i, j) and for instance the ship's speed at this point by $V(i, j)$ then derivatives are approximated by

$$V_{x_1}(i,j) = \tfrac{1}{2}\{V(i+1,j) - V(i-1,j)\},$$

$$V_{x_2}(i,j) = \tfrac{1}{2}\{V(i,j+1) - V(i,j-1)\},$$

where distances are measured in mesh units. Moreover, it is assumed that derivatives between grid points can be obtained by bilinear interpolation. The validity of this approximation has been tested by means of a numerical construction of time fronts in the geometrical way as described in Section 4. The actual geographical velocity diagram showing the relation between ship's velocity and wave direction (and wave height) is assumed to be of elliptic form. It is constructed with the aid of the actual geographical values of the ship's speed in the case of following, beam and head waves (see Fig. 14). These values, which must be multiplied by the map factor $m(\phi)$ when applied in the (x_1, x_2) plane, are obtained from empirical data. It is supposed here that the speed exceeds the numerical value of the ocean current for all waveheights.

(3) We shall now pay some attention to the occurrence of forbidden courses. Consider a ship with zero velocity in a wave field composed of waves of a single period. If the direction of the waves lies in a suitable sector and if their period corresponds to the resonance period of the ship, then the ship will undergo violent movements.

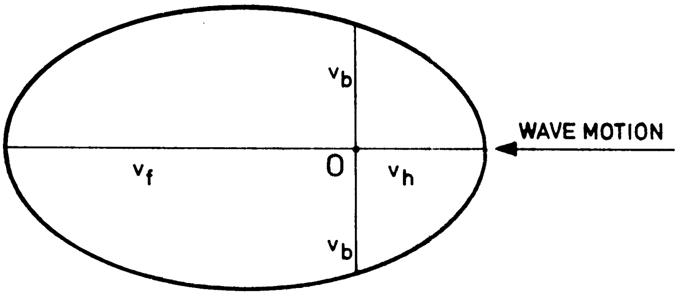


Fig. 14 The velocity diagram. The ship is situated at O. v_f, v_b and v_h correspond with the ship's speed in the case of following, beam and head waves.

As a consequence, one has to change the position of the ship with respect to the wave direction in order to avoid heavy rolling. If we now consider the more realistic situation of a moving ship in a seaway, some modifications will appear due to the fact that

(a) the wave period which causes resonance and also the corresponding sector of relative wave direction are dependent on the velocity of the ship, and

(b) instead of waves of a single period, a spectrum of wave periods is present. As a representative period, one could take in this case the period which is related to the maximum energy in the spectrum.

As we saw in Section 4, the velocity diagram must be changed, as shown in Fig. 15, where the sectors of forbidden courses are indicated. For simplicity it is assumed here that these sectors depend merely on the significant wave height (which in fact determines the maximum ship's speed). Owing to the danger of heavy rolling, a ship will in general not follow a strategy as mentioned by De Wit (1968), (see also Zoppoli (1972)), where points in the forbidden sectors can be reached by steering a combination of boundary courses.

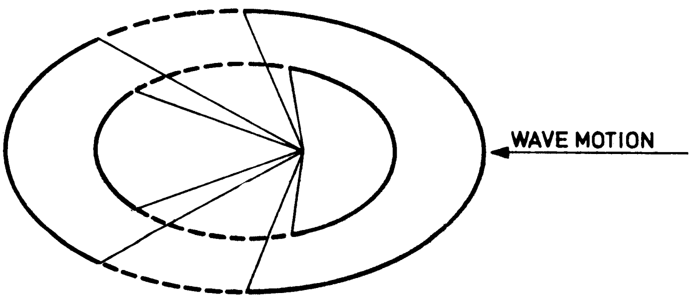


Fig. 15 The forbidden sectors increasing with the significant wave height.

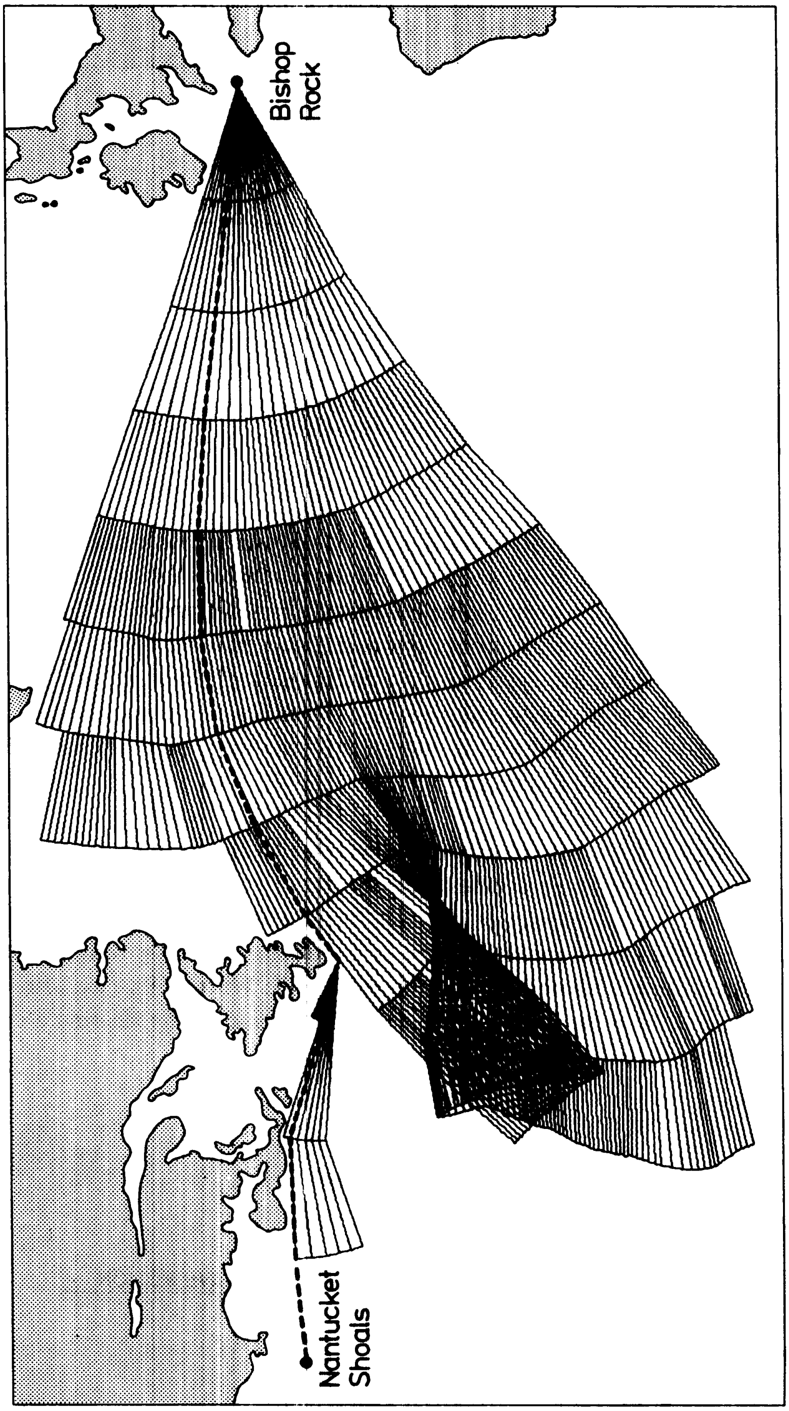


Fig. 16 Computer produced by means of an incremental plotter using wave information over the period 17 January-23 January 1970, fictitious ship's data and a 12-hour time step. The least-time track is indicated by the dashed line.

(4) Let us now consider the continuous dependence on the initial values (Results 1 and 2 of Section 4) from a practical point of view. Consider two extremals $\{x_i(t), \lambda_i(t)\}$ and $\{x_i(t) + \Delta x_i(t), \lambda_i(t) + \Delta\lambda_i(t)\}$ $(i = 1, 2)$ satisfying equations (4.1), (4.2), (4.3) and (4.4), starting from the point (x_{10}, x_{20}) with parameter values a and $a + \Delta a$ (see again Results 1 and 2 of Section 4). Since these solutions depend continuously on the initial values, it follows that $\Delta x_i(t), \Delta\lambda_i(t) \to 0$ $(i = 1, 2; 0 \leq t \leq t_1)$ if $\Delta a \to 0$.

As a consequence, we can start an extremal at time $t = \tau$ $(0 \leq \tau \leq t_1)$ with initial values which can be obtained by linear interpolation between the corresponding values of its neighbours if the distance $\sqrt{\Delta x_1(\tau)^2 + \Delta x_2(\tau)^2}$ is small enough (see Fig. 16). The above procedure can be applied in the case of forbidden courses, if the courses p and $p + \Delta p$ connected with the two extremals both belong to sector I or sector II (Result 2, Section 4), otherwise a gap occurs between the two extremals.

(5) Further we may remark on the behaviour of extremals near a boundary. In some cases there are parts of the ocean region which cannot be reached along extremals. To avoid this situation, points are chosen in a suitable way along which extremals can penetrate these areas. These points can be considered as new points of departure or arrival (see Fig. 16). The computation of an extremal is stopped if it intersects the boundary. As a result near a boundary only those extremals will be considered which coincide with it partly or wholly. We shall conclude with a remark on the exclusion of extremals.

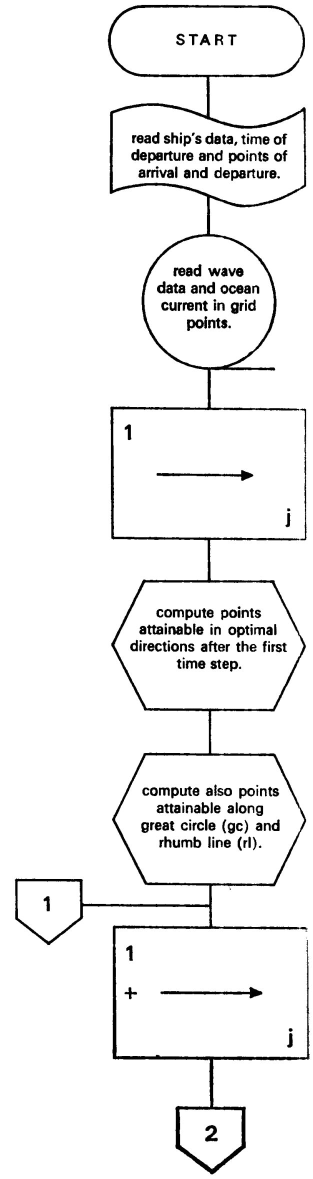
As we saw in Section 4, extremals can be excluded because of

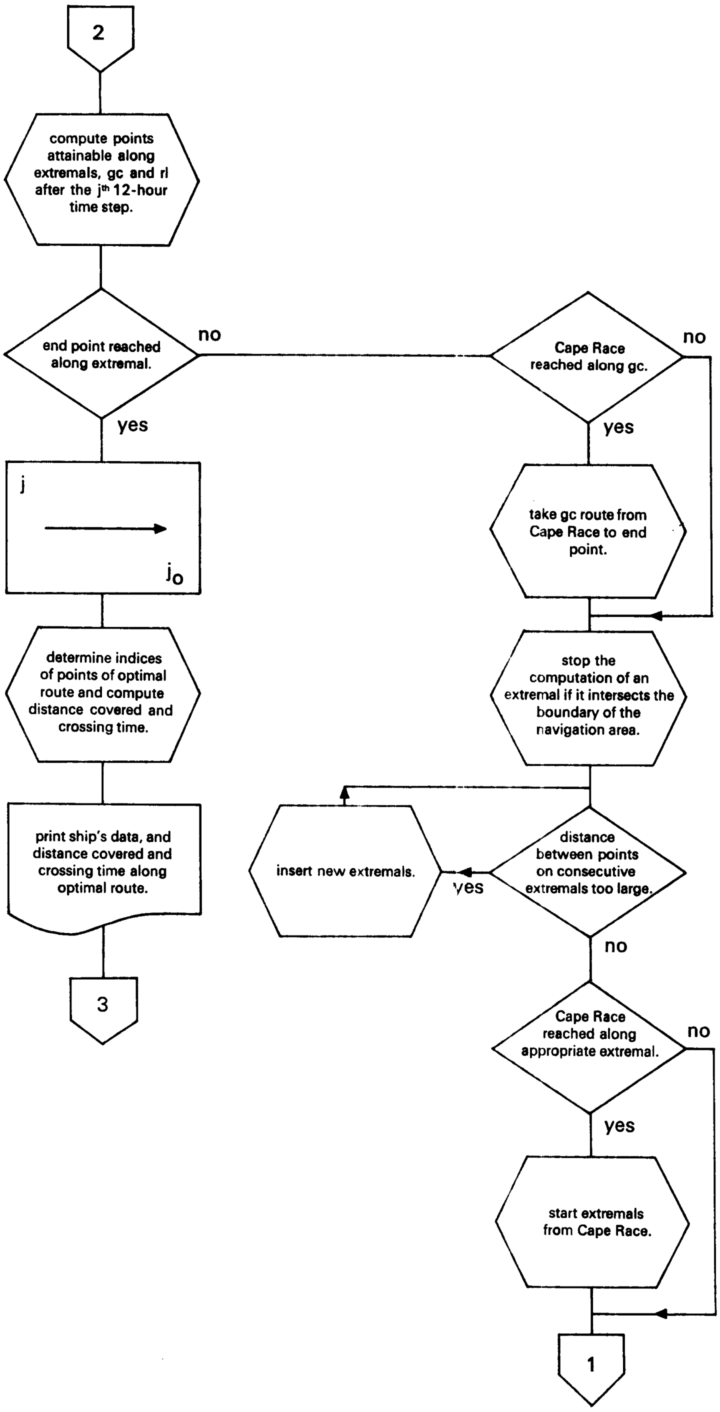
(a) the presence of a conjugate point between beginning and end point (no relative minimum), and

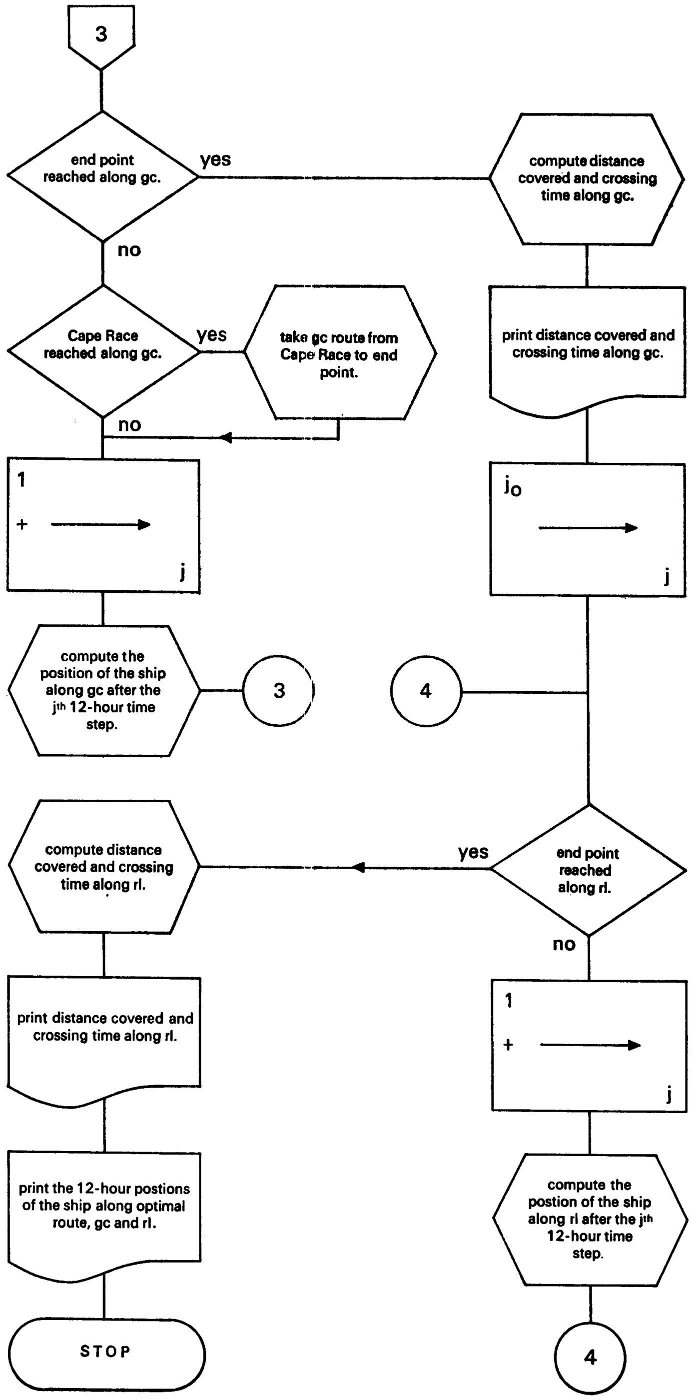
(b) the fact that not every point of this extremal belongs to the boundary of the corresponding set of reachable points (no absolute minimum).

In the practical situations treated here (with 6-hour or 12-hour time steps) the second condition will not be satisfied in all cases. Although in general we consider all extremals when computing the least-time track, there are situations where the exclusion of extremals on account of condition (a) saves a lot of computing time. These situations can be handled very simply. After these introductory remarks, we shall give a survey of the method for the computation of the least-time track in the form of a flow diagram. This flow diagram gives a description of the computation of the route which is presented in Section 7. The corresponding computer program is found in Section 8.

Flow diagram




2
compute points attainable along extremals, gc and rl after the jth 12-hour time step.
end point reached along extremal.
no
yes
j
jo
determine indices of points of optimal route and compute distance covered and crossing time.
print ship's data, and distance covered and crossing time along optimal route.
3
Cape Race reached along gc.
no
yes
take gc route from Cape Race to end point.
stop the computation of an extremal if it intersects the boundary of the navigation area.
distance between points on consecutive extremals too large.
yes
insert new extremals.
no
Cape Race reached along appropriate extremal.
no
yes
start extremals from Cape Race.
1


3
end point reached along gc.
yes
compute distance covered and crossing time along gc.
no
Cape Race reached along gc.
yes
take gc route from Cape Race to end point.
print distance covered and crossing time along gc.
no
1
+
j
j₀
j
compute the position of the ship along gc after the jth 12-hour time step.
3
4
compute distance covered and crossing time along rl.
yes
end point reached along rl.
no
print distance covered and crossing time along rl.
1
+
j
print the 12-hour postions of the ship along optimal route, gc and rl.
compute the postion of the ship along rl after the jth 12-hour time step.
STOP
4

At this point we shall touch lightly on other methods applied in ship routing for solving equations (4.1), (4.2), (4.3), (4.4) with (4.5). We can distinguish two types of iterative methods: a steepest-ascent method or method of gradients, Kelley (1962), Bryson and Denham (1962), and a method which uses variational relations between initial values and end conditions, Marks (1968), Bleick and Faulkner (1965), see also Faulkner (1963, 1964). Of course a third method can be found by using the geometrical relations between time fronts and extremals as mentioned in Section 4 (see De Wit (1968)). The disadvantages with respect to the application of the iterative computing methods are first of all, that the solution depends on the choice of the initial course, which is unacceptable for operational use, and secondly, that there may be convergence problems, especially in view of the use of empirical data. The geometrical time front method is more suited for manual use than for numerical application because of the introduction of random errors when computing the normal direction, which in fact is assumed to exist. In conclusion we consider the possibilities of a practical application of the numerical method. At the Routing Office of the KNMI, transit times along least-time track and standard routes are provided afterwards, using analyzed wave charts in order to check the value of the recommended route.

Since the number of ships had increased to 711 by 1973, as we saw in Section 1, computerization of only this evaluation would be time-saving.

7 Results

In this section an example is presented of an ocean crossing between Greenock (Tory Island) and Charleston (Frying Pan) (see also Fig. 3). The service speed of the chosen general cargo-ship was 21 knots. This ship departed from Tory Island on the 25th October 1973 at 03.00 GMT. The recommended route manually constructed with the aid of the 72-hour weather forecasts is indicated in Fig. 17. The transit time was 8 days and 2.5 hours. Before being used for computations with respect to the least-time track, great circle and rhumb line the ship's performance data were corrected by recomputing this route with analyzed wave charts. The crossing times along the least-time route, great circle and rhumb line (see Fig. 17) were respectively 7 d 20.7 h, 8 d 19 h and 8 d 7.3 h.

The detours of the least-time route and recommended route with respect to the great circle were 296 and 253 miles.

This section concludes with a series of wave charts showing the 12-hour ship's positions along the least-time track and great circle. These wave charts contain significant wave height contours in meters and wave motion arrows. Sea is indicated by solid lines, swell by dashed lines.

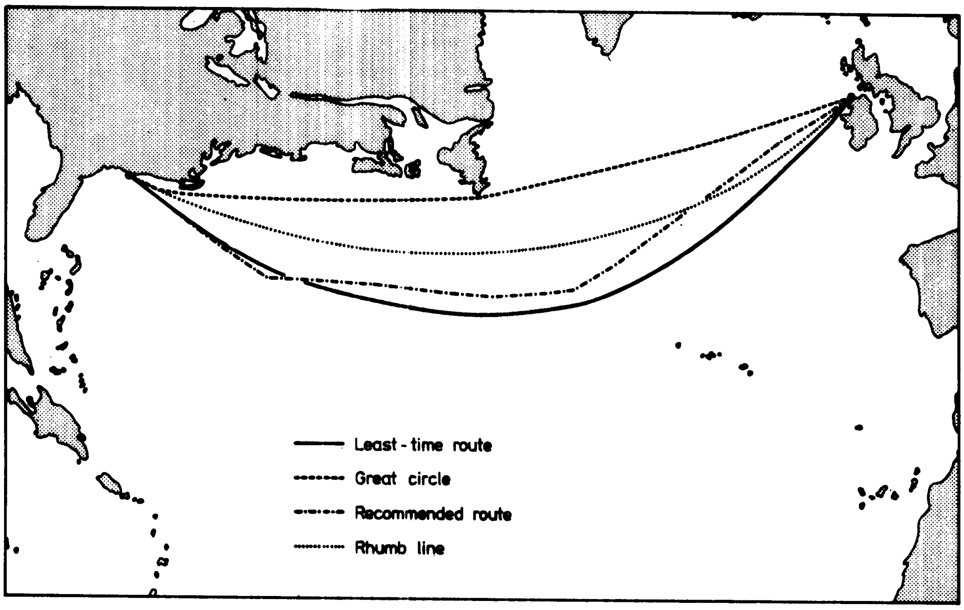
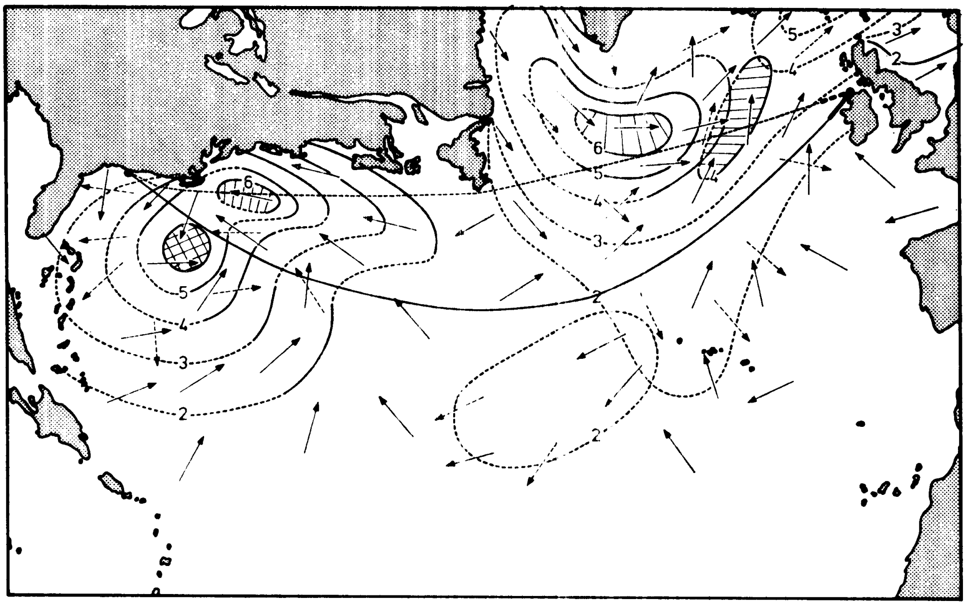
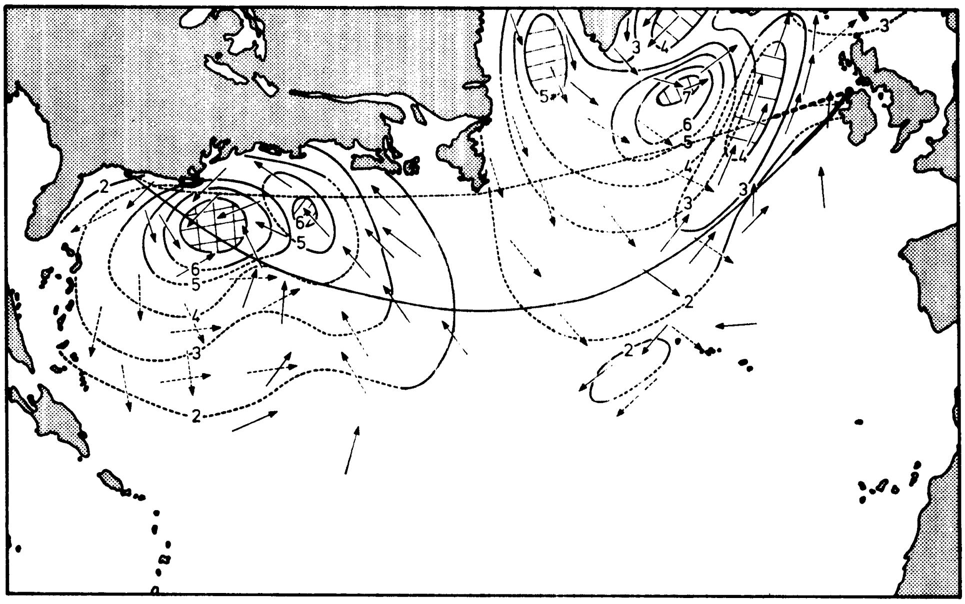


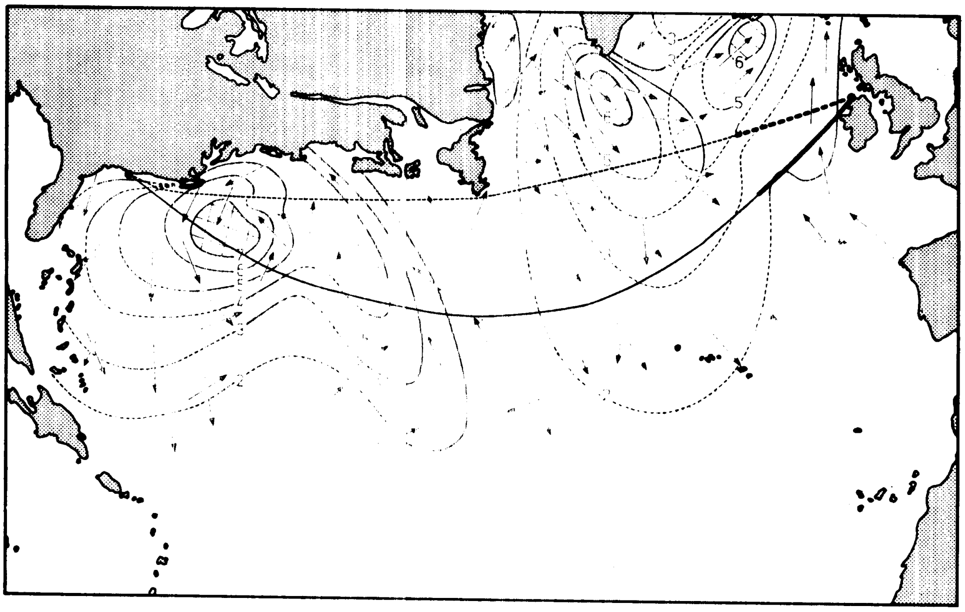
Fig. 17



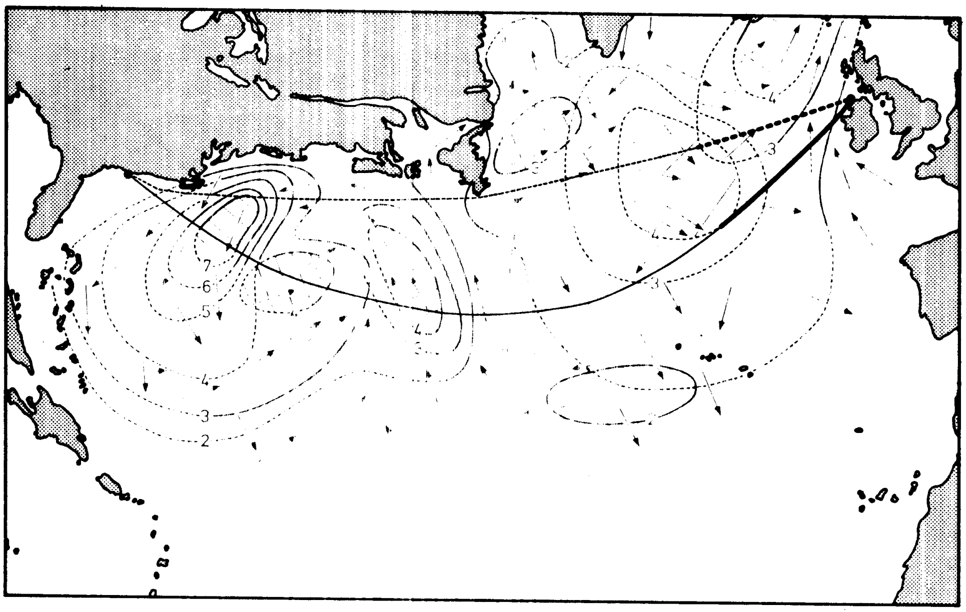
Wave chart 25 October 1973 12.00 GMT



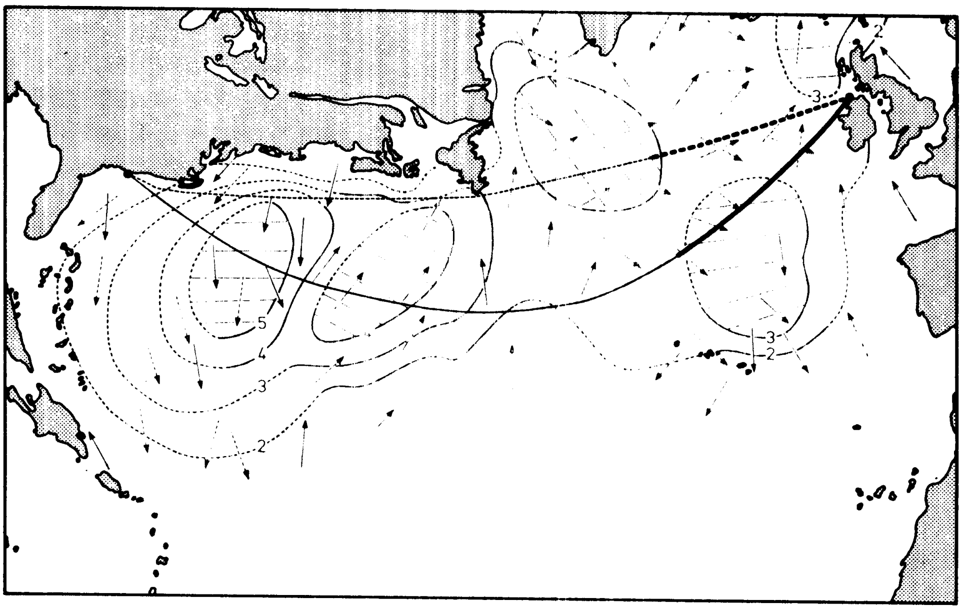
Wave chart 26 October 1973 00.00 GMT



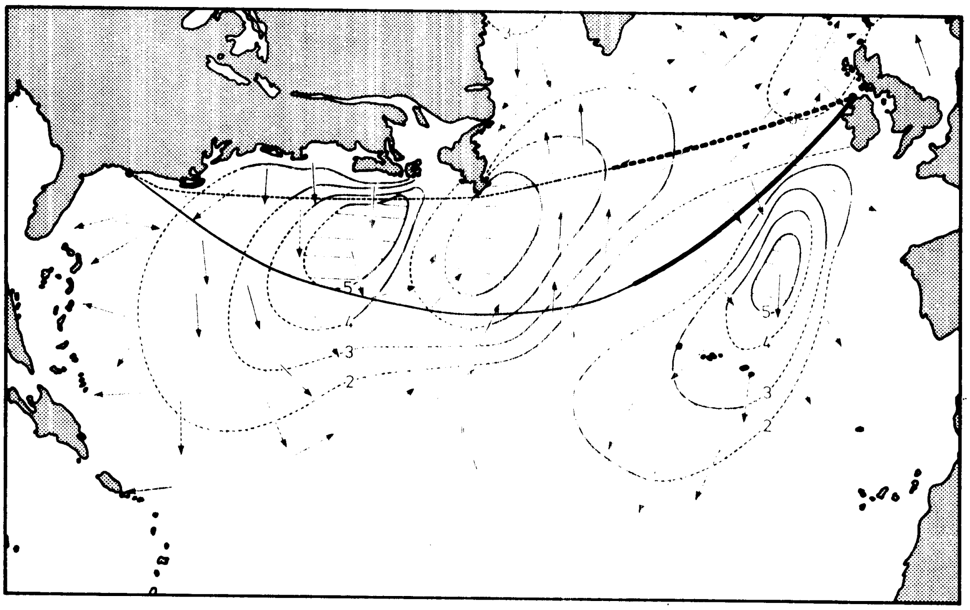
Wave chart 26 October 1973 12.00 GMT



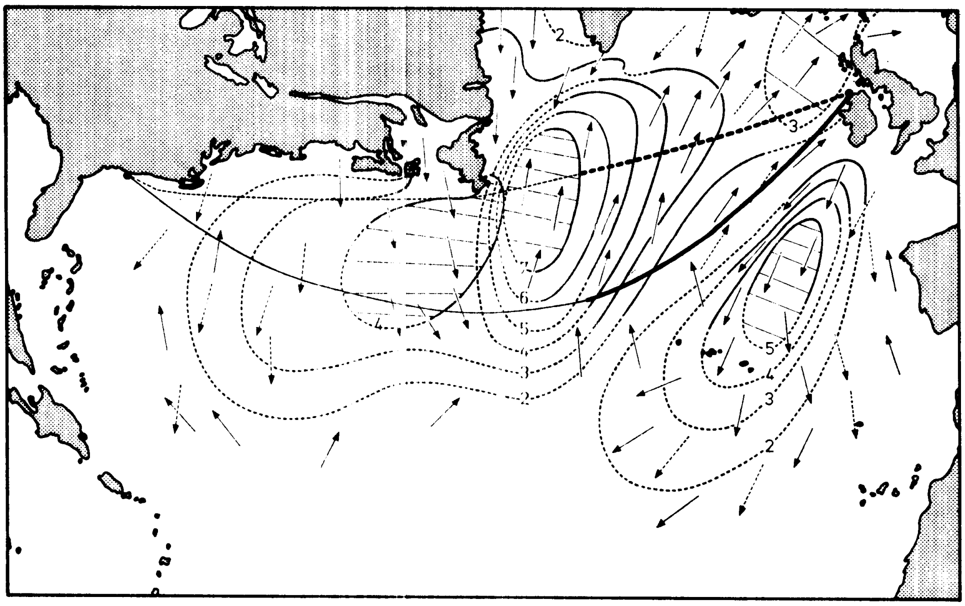
Wave chart 27 October 1973 00.00 GMT



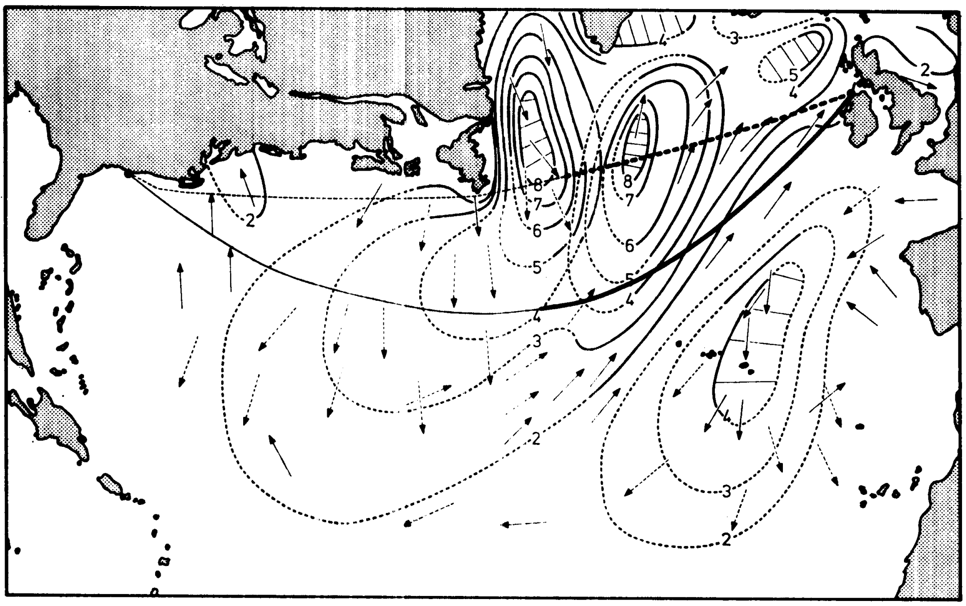
Wave chart 27 October 1973 12.00 GMT



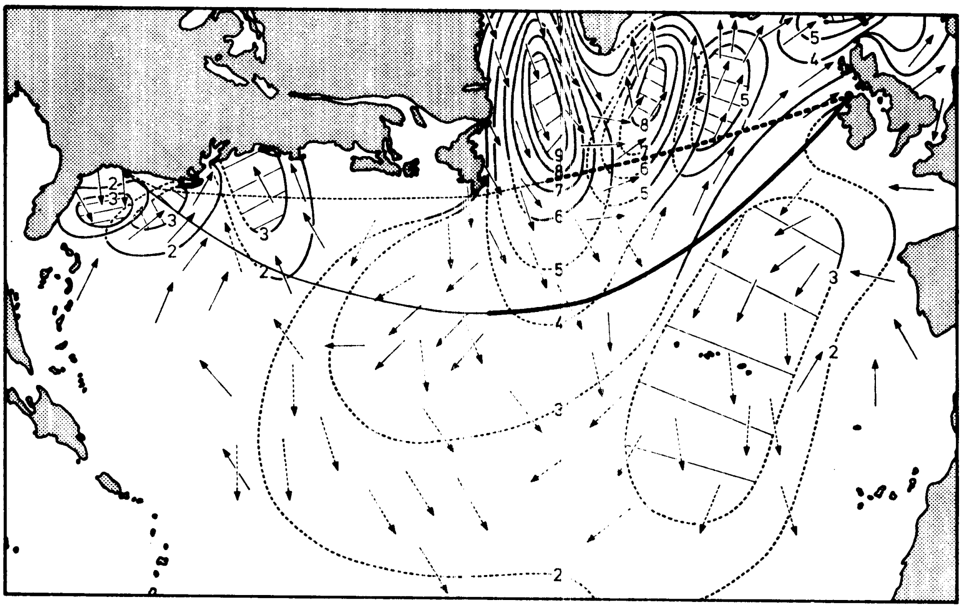
Wave chart 28 October 1973 00.00 GMT



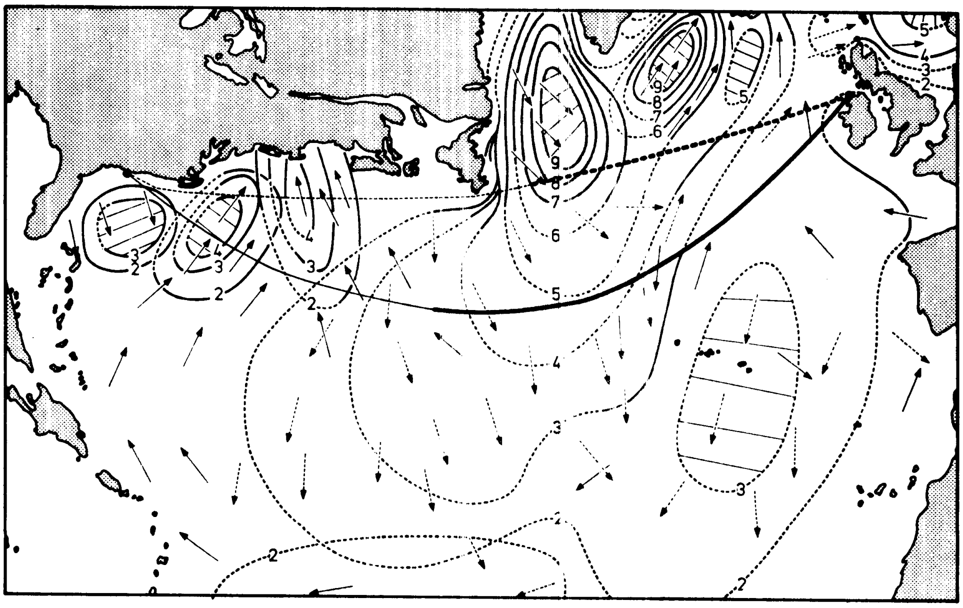
Wave chart 28 October 1973 12.00 GMT



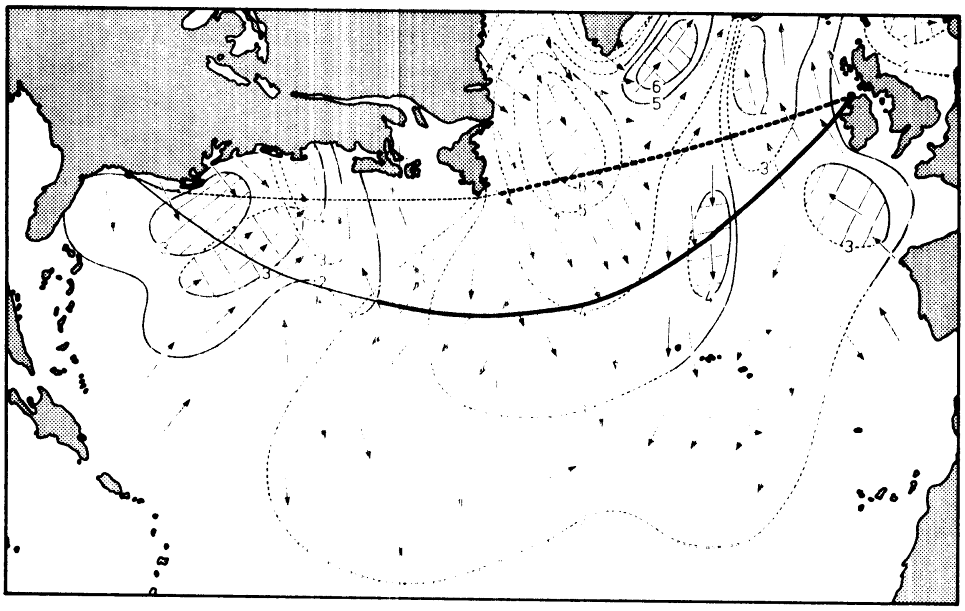
Wave chart 29 October 1973 00.00 GMT



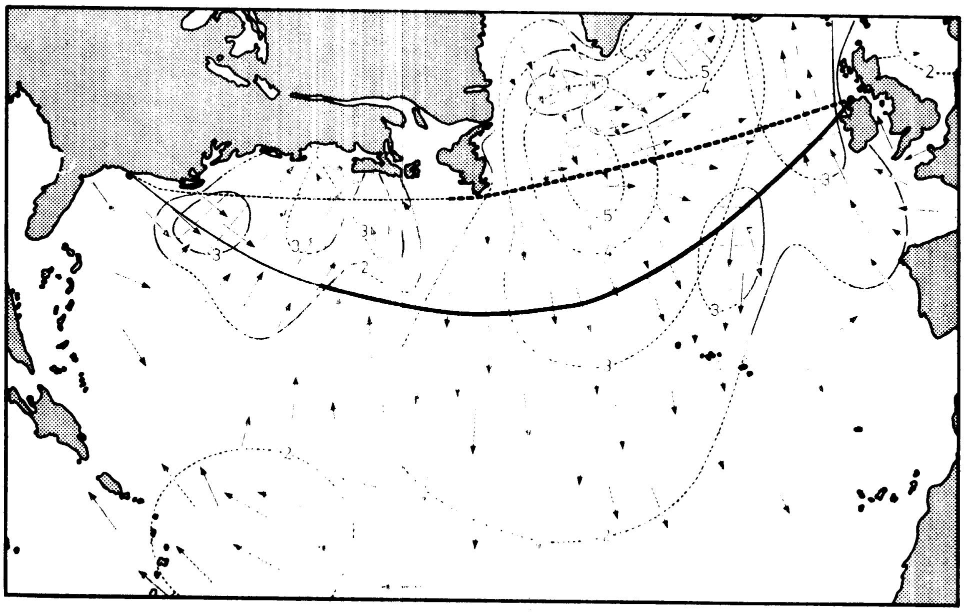
Wave chart 29 October 1973 12.00 GMT



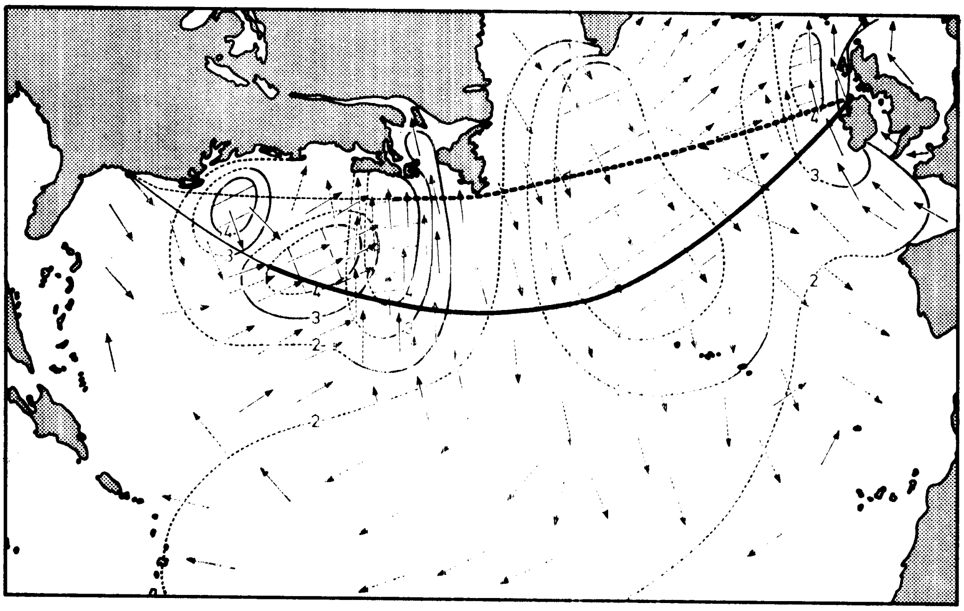
Wave chart 30 October 1973 00.00 GMT



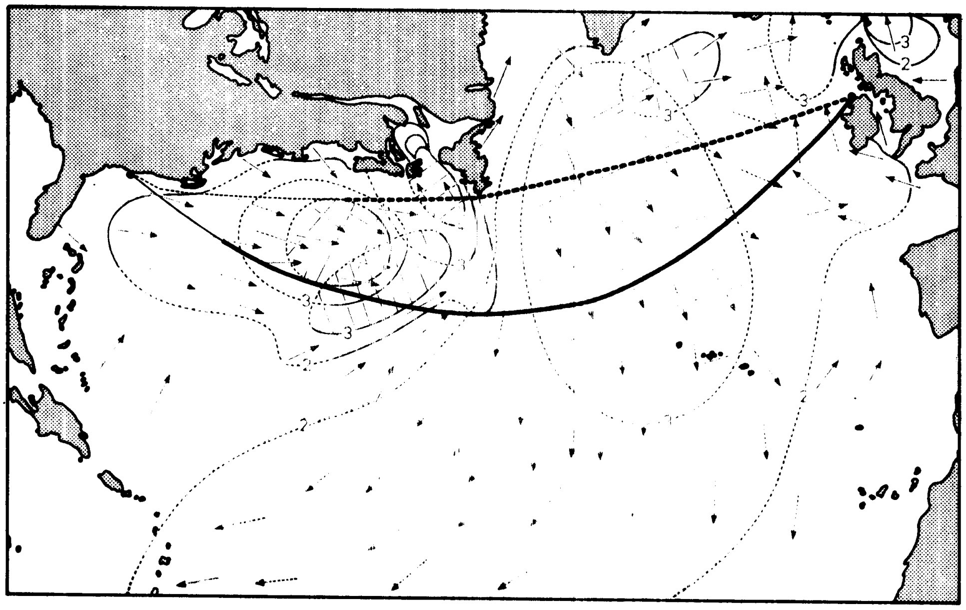
Wave chart 30 October 1973 12.00 GMT



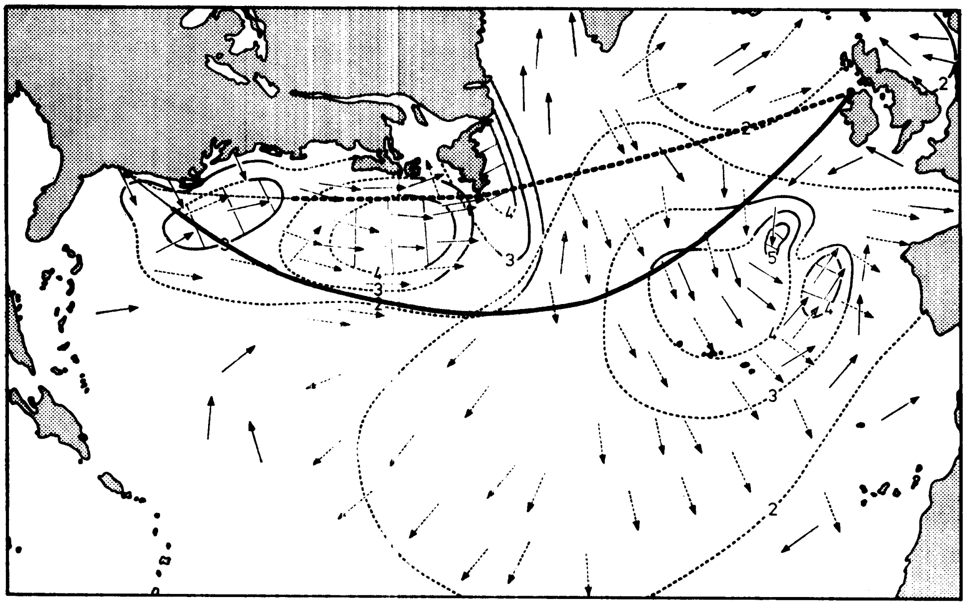
Wave chart 31 October 1973 00.00 GMT



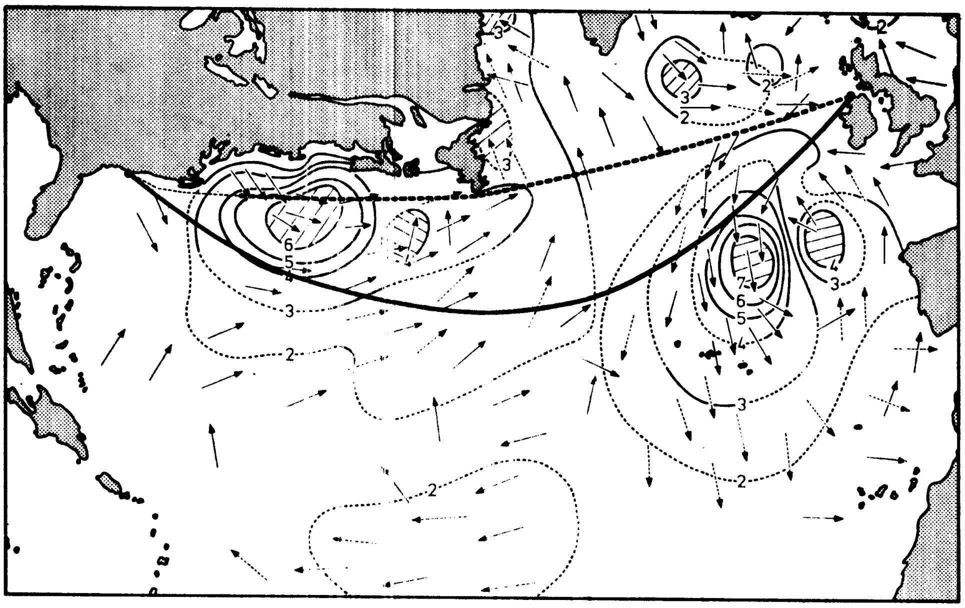
Wave chart 31 October 1973 12.00 GMT



Wave chart 1 November 1973 00.00 GMT



Wave chart 1 November 1973 12.00 GMT



Wave chart 2 November 1973 00.00 GMT

8 An example of a computer program, written in ALGOL-60

```
begin  comment This computer program has been used for the computation of the
       least-time track from Greenock to Charleston , a route which is presented in
       Section 7. It is a simplified version of the program applied at the KNMI.
       The input/output procedures are determined by the use of an EL-X8 computer;
       integer ant,at,at1,at3,ctr,d1,d2,f,h,h1,hh,hk,hk1,hk2,i,i0,i1,i2,i00,i01,i02,
       i11,i001,i0001,ii,in,in1,ip,j,j0,j2,j00,j001,jj,jj1,jg,jl,jr,k,k1,l1,mnd,pia,
       pt,srt,t1,tot,tk,tr,x,y;
       real a0,aa,aa1,afst1,afst2,afst3,alpha,amn,bb,bb1,cc,cc1,copgr,copi,d,dx,dx1,
       dy,dy1,e1,e2,e3,ft,g,grd,gur,l,l2,l3,l4,l01,l02,l11,l111,lim,lim1,p,p2,p3,p4,
       p01,p02,pi,pi2,pi4,pi6,pi9,pim2,pp,pr,pr1,pu1,pu2,pu3,q,r,r0,r1,r2,s,s1,sipgr
       ,sipi,t,th,tp2,tp3,tp4,tt,tt1,tt2,ur,vh,xe,xg,xk,xl,xs,xv,xx,xx1,xxx,ye,yg,yk
       ,yl,ys,ysn,yv,yy,yyy;
       integer array AF[1:3],XX,YY[1:12],I,PIA[1:50],RS[1:100],PA,PB[1:150],
       GH,GR,XGS,YGS[0:27,0:19];
       array AH[1:3],AFX,AFY[1:4],L[1:12],XGC,YGC,XLX,YLX,XOR,YOR[0:50],
       A,LA,MU,P,PX,PY[0:300],AA[1:3,0:15],X,Y[0:300,0:2];
       boolean joke,joke1;

       real procedure BR(x,y); value x,y; real x,y;
       BR:=-arctan(sqrt((x-7.5)↑2+(y+8.5)↑2)×a0)×2+pi2;

       real procedure LE(x,y); value x,y; real x,y;
       LE:=arctan((x-7.5)/(y+8.5))+pi6;

       procedure P1(u); value u; real u;
       begin   integer d; real ru;
               d:=entier(u/24); ru:=u-d×24;
               SPACE(19); ABSFIXT(2,0,d); ABSFIXT(14,1,ru); NLCR; NLCR
       end P1;

       procedure P2(A,B,g,tkt); value g; integer g; array A,B; string tkt;
       begin   integer i;
               NEWPAGE; CARRIAGE(5);
               PRINTTEXT(⌈points ⌉); PRINTTEXT(tkt);
               PRINTTEXT(⌈ route:        x1        x2        la        lo⌉);
               CARRIAGE(2);
               for i:=0,i+1 while i≤g do
               begin   NLCR; SPACE(29);
                       ABSFIXT(2,2,A[i]); ABSFIXT(4,2,B[i]);
                       ABSFIXT(4,2,BR(A[i],B[i])/grd);
                       ABSFIXT(4,2,LE(A[i],B[i])/grd)
               end
       end P2;
```

```
procedure LS;
begin   integer j1;
        j1:=j-1;
        vh:=1-AFST(XGC[j1],YGC[j1],xk,yk)/
        AFST(XGC[j1],YGC[j1],XGC[j],YGC[j]);
        if vh<0 then goto RIK else
        begin   joke1:=false;
                aa:=aa1; bb:=bb1; cc:=cc1;
                j001:=j;
                l11:=vh×l11; l111:=l11×l11;
                xx1:=((yk-bb)×sqrt(cc×4-l11)×tk+aa-xk)×l11/(cc×2);
                XGC[j]:=xg:=xk+xx1;
                YGC[j]:=yg:=yk-((xk-aa)×xx1×2+l111)/((yk-bb)×2)
        end;
RIK:
end LS;

procedure CB(x,y); value x,y; real x,y;
begin   real prdx,prdy,s;
        prdx:=prdy:=0;
COBY:   s:=(((prdx/2+x-7.5)↑2+(prdy/2+y+8.5)↑2)×a0×a0+1)×s1;
        if prdx≠0 ∨ prdy≠0 then goto KLM;
        prdx:=s×dx; prdy:=s×dy;
        goto COBY;
KLM:    dx:=s×dx; dy:=s×dy
end CB;

real procedure AFST(x1,y1,x2,y2); value x1,y1,x2,y2; real x1,y1,x2,y2;
AFST:=sqrt((x1-x2)↑2+(y1-y2)↑2)×636⏨5/
((((x1+x2)/2-7.5)↑2+((y1+y2)/2+8.5)↑2)×a0×a0+1)/ft;

procedure AM(x,y); value x,y; real x,y;
begin   integer i;
        amn:=2⏨15; tr:=0;
        for i:=i1,i+1 while i≤in do
        begin   d:=AFST(X[i,0],Y[i,0],x,y);
                if d<amn then begin amn:=d; tr:=i end
        end
end AM;

real procedure ARCTAN(x,y); value x,y; real x,y;
ARCTAN:= if x=0 then sign(y)×pi2 else
if x<0 then arctan(y/x)+sign(y)×pi+(if y=0 then pi else 0) else arctan(y/x);

real procedure LM(x,y); value x,y; real x,y;
begin   real cd,csc,sc;
        cd:=arctan(cos(LE(x,y)-l2)×tp2);
        csc:=cos(p2)×cos(pi2-BR(x,y)-cd)/cos(cd); sc:=sqrt(1-csc×csc);
        LM:=arctan(sc/csc)×6378178/1852/AA[1,2]
end LM;
```

```
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,l11,siagr,sig,siggr,sigr,x1;
        integer h,h1,j1,k,k1,spr;

        real procedure IP(A); integer array A;
        IP:=(s×(q×A[h,k]+p×A[h1,k])+r×(q×A[h,k1]+p×A[h1,k1]))/10;

        spr:=0; j1:=j-1;
        if a=20 ∧ j<j0+2 then
        begin   inarray(drum,j1×600,GH);
                inarray(drum,j1×600+30000,GR); hold(GR)
        end;
MGRT:   h:=entier(x); h1:=h+1; k:=entier(y); k1:=k+1;
        p:=x-h; q:=1-p; r:=y-k; s:=1-r;
        dx:=IP(XGS); dy:=IP(YGS);
        gh:=IP(GH);
        if j=1 then begin dx:=dx×gur; dy:=dy×gur end;
        if gh>14.9 then gh:=14.9;
        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;
        if gr<0 then gr:=gr+pim2;
        if gr>pim2 then gr:=gr-pim2;
        cogr:=cos(gr); sigr:=sin(gr);
        h:=entier(gh); h1:=h+1;
        p:=gh-h; q:=1-p;
        for k:=1,2,3 do AH[k]:=AA[k,h]×q+AA[k,h1]×p;
        if j=1 then for k:=1,2,3 do AH[k]:=AH[k]×gur;
        e1:=(AH[1]+AH[3])/2; e2:=AH[1]-e1; e3:=e1×AH[2]/sqrt(AH[1]×AH[3]);
        if a=10 ∨ a=20 ∨ a=30 then
        begin   if a=30 then goto PL;
                if a=10 then g:=ARCTAN(tk,(aa-x)/(y-bb)×tk) else
                g:=ARCTAN(tk,((y+8.5)/alpha-x+7.5)/((x-7.5)/alpha+y+8.5)×tk);
        PL:     cog:=cos(g); sig:=sin(g);
                pr:=dx×cog+dy×sig;
                coggr:=cos(g-gr); siggr:=sin(g-gr);
                p:=e3×e3×coggr; q:=e1×e1×siggr;
                r:=e2×siggr; r:=r×r;
                s:=p×coggr+q×siggr;
                l:=(sqrt(s-r)×e1×e3-e2×p)/s;
                pr1:=l+pr;
                if a=30 then goto MAC;
                dx:=pr1×cog; dy:=pr1×sig
        end else
```

```
            begin  coagr:=cos(a-gr); siagr:=sin(a-gr);
                   p:=e1×coagr; q:=e3×siagr; r:=sqrt(p×p+q×q);
                   s:=e1×p/r-e2; t:=e3×q/r;
                   tt1:=s×cogr-t×sigr; tt2:=s×sigr+t×cogr;
                   dx:=dx+tt1; dy:=dy+tt2
            end;
            CB(x,y);
            if a=10 ∨ a=20 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(dx×dx+dy×dy); l11:=l1×l1;
                   if a=10 then
                   begin  x1:=(l11×(aa-XGC[j1])+
                          tk×l1×(YGC[j1]-bb)×sqrt(4×cc-l11))/(2×cc);
                          x:=XGC[j1]+x1;
                          y:=YGC[j1]-((XGC[j1]-aa)×x1×2+l11)/((YGC[j1]-bb)×2);
                          if spr≠0 then
                          begin XGC[j]:=xg:=x; YGC[j]:=yg:=y; goto MAC end
                   end else
                   begin  l1:=l1/sqrt(1/(alpha×alpha)+1);
                          tt:=ln(tk×l1/(r0×alpha)+
                          exp((LE(XLX[j1],YLX[j1])-th)/alpha))×alpha+th;
                          r2:=exp((tt-th)/alpha)×r0;
                          tt:=tt-pi6;
                          x:=sin(tt)×r2+7.5; y:=cos(tt)×r2-8.5;
                          if spr≠0 then
                          begin XLX[j]:=xl:=x; YLX[j]:=yl:=y; goto MAC end
                   end;
                   inarray(drum,j×600,GH);
                   inarray(drum,30000+j×600,GR); hold(GR);
                   spr:=1;
                   goto MGRT
            end;
      MAC:
      end MAXAF;

      if ¬ comparefile(tape(2),⟨krieb-ortn⟩) then goto AFL;
      tot:=jj1:=pia:=PIA[1]:=0;
      xk:=13.316; yk:=5.248;
      afst1:=afst2:=afst3:=0;
      pi:=arctan(1)×4; pim2:=pi×2; pi2:=pi/2; pi4:=pi/4; pi6:=pi/6; pi9:=pi/9;
      grd:=pi/180; ft:=cos(pi6)+1;
      at:=53; j00:=45; j0:=100; t1:=1;
      a0:=31.8/(637.1229×ft); s1:=1852/(a0×12742458);
      joke:=joke1:=true;
L1:   k:=RESYM; if k≠120 then goto L1;
L2:   k:=RESYM; if k=120 then goto L3;
      RS[t1]:=k; t1:=t1+1; goto L2;
L3:   for i:=0,i+1 while i<16 do for j:=1,2,3 do AA[j,i]:=READ×1.2;
      jr:=READ; mnd:=READ; d1:=READ; ur:=READ;
```

```
xs:=XOR[0]:=READ; ys:=YOR[0]:=READ;
xe:=READ; ye:=READ;
tt:=th:=l2:=LE(xs,ys);
l3:=LE(xe,ye); l4:=LE(xk,yk);
if xs>xe then begin tk:=-1; hk1:=195; hk2:=240; in:=75; at:=90 end else
begin tk:=1; hk1:=35; hk2:=10; in:=75; at:=33 end;
p2:=-BR(xs,ys)+pi2; p3:=-BR(xe,ye)+pi2; p4:=-BR(xk,yk)+pi2;
tp2:=sin(p2)/cos(p2); tp3:=sin(p3)/cos(p3); tp4:=sin(p4)/cos(p4);
r0:=sin(p2/2)/cos(p2/2)/a0; r1:=sin(p3/2)/cos(p3/2)/a0;
alpha:=(l3-l2)/ln(r1/r0);
l01:=arctan((cos(l2)/tp4-cos(l4)/tp2)/(sin(l4)/tp2-sin(l2)/tp4));
l02:=arctan((cos(l4)/tp3-cos(l3)/tp4)/(sin(l3)/tp4-sin(l4)/tp3));
p01:=arctan(-cos(p2)/(sin(p2)×cos(l2-l01)));
p02:=arctan(-cos(p4)/(sin(p4)×cos(l4-l02)));
cc:=1/a0/cos(p01); cc1:=1/a0/cos(p02);
aa:=sin(p01)×sin(l01-pi6)×cc+7.5; aa1:=sin(p02)×sin(l02-pi6)×cc1+7.5;
bb:=sin(p01)×cos(l01-pi6)×cc-8.5; bb1:=sin(p02)×cos(l02-pi6)×cc1-8.5;
cc:=cc×cc; cc1:=cc1×cc1;
for i:=jr-1 step -1 until 73 do
if i:4×4=i then tot:=tot+366 else tot:=tot+365;
for i:=1 step 1 until mnd-1 do
begin   if i=4 ∨ i=6 ∨ i=9 ∨ i=11 then l1:=30 else l1:=31;
        if i=2 then begin if jr:4×4=jr then l1:=29 else l1:=28 end;
        tot:=tot+l1
end;
if ur≥12 then begin d2:=3; ur:=ur-12 end else d2:=2;
gur:=1-ur/12;
srt:=(tot+d1-274)×2+d2;
inarray(tape(2),1,XGS); inarray(tape(2),2,YGS); hold(YGS);
ant:=srt+30;
for j:=srt,j+1 while j<ant do
begin   inarray(tape(2),j×2-1,GH); inarray(tape(2),j×2,GR); hold(GR);
        outarray(drum,jj1×600,GH); outarray(drum,jj1×600+30000,GR); hold(GR);
        jj1:=jj1+1
end;
closefile(tape(2));
i1:=0; j:=1;
lim:=LM(xe,ye); lim1:=LM(xk,yk);
inarray(drum,0,GH); inarray(drum,30000,GR); hold(GR);
for i:=0,i+1 while i≤in do
begin   A[i]:=(hk1-i)×grd;
        LA[i]:=cos(A[i]); MU[i]:=sin(A[i]);
        MAXAF(xs,ys,A[i]);
        X[i,1]:=xs+dx; Y[i,1]:=ys+dy;
        dx:=tt1; dy:=tt2;
        CB(xs,ys);
        PX[i]:=dx; PY[i]:=dy
end;
XGC[0]:=XLX[0]:=xg:=xl:=xs; YGC[0]:=YLX[0]:=yg:=yl:=ys;
MAXAF(xg,yg,10); MAXAF(xl,yl,20);
```

```
        for i:=0,i+1 while i≤in do
        begin  MAXAF(X[i,1],Y[i,1],A[i]);
               X[i,0]:=X[i,2]:=(X[i,1]+xs+dx)/2;
               Y[i,0]:=Y[i,2]:=(Y[i,1]+ys+dy)/2;
               dx:=tt1; dy:=tt2;
               CB(X[i,1],Y[i,1]);
               P[i]:=ARCTAN(PX[i]+dx,PY[i]+dy)
        end;
        outarray(drum,100000,X); outarray(drum,100902,Y); hold(Y);
        for h:=1,h+1 while h<151 do PB[h]:=PA[h]:=1000;
        outarray(drum,60000,PB); outarray(drum,60250,PA); hold(PA);
NEXTR:  j:=j+1;
        for i:=0,i+1 while i≤in do
        begin  if Y[i,0]=-99 then goto L4;
               sipi:=sin(P[i]); copi:=cos(P[i]);
               h:=entier(X[i,0]); k:=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];
                       hh:=entier(GH[x,y]/10);
                       p:=GH[x,y]/10-hh; q:=1-p;
                       for r:=1,2,3 do AH[r]:=AA[r,hh]×q+AA[r,hh+1]×p;
                       e1:=(AH[1]+AH[3])/2;
                       e2:=e1×AH[2]/sqrt(AH[1]×AH[3]); e3:=AH[1]-e1;
                       copgr:=cos(P[i]-GR[x,y]×grd); sipgr:=sin(P[i]-GR[x,y]×grd);
                       p:=e2×e2×copgr; q:=e1×e1×sipgr;
                       r:=e3×sipgr; r:=r×r;
                       s:=p×copgr+q×sipgr;
                       l:=(-p×e3+e1×e2×sqrt(s-r))/s×(LA[i]×copi+MU[i]×sipi)+
                       (LA[i]×XGS[x,y]+MU[i]×YGS[x,y])/10;
                       dx:=l×copi; dy:=l×sipi;
                       CB(x,y);
                       L[jj]:=sqrt(dx×dx+dy×dy)
               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;
               LA[i]:=LA[i]-(p×(r×AFX[4]+s×AFX[3])+q×(r×AFX[2]+s×AFX[1]))/2;
               MU[i]:=MU[i]-(p×(r×AFY[4]+s×AFY[3])+q×(r×AFY[2]+s×AFY[1]))/2;
               A[i]:=ARCTAN(LA[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;
```

```
                dx:=tt1; dy:=tt2;
                CB(X[i,0],Y[i,0]);
                PX[i]:=dx; PY[i]:=dy;
L4:     end;
        MAXAF(xg,yg,10);
        l11:=sqrt(dx×dx+dy×dy);
        MAXAF(xl,yl,20);
        for i:=0,i+1 while i≤in do
        begin   if Y[i,0]=-99 then begin Y[i,2]:=-99; goto L5 end;
                MAXAF(X[i,1],Y[i,1],A[i]);
                X[i,2]:=(X[i,1]+X[i,0]+dx)/2; Y[i,2]:=(Y[i,1]+Y[i,0]+dy)/2;
                dx:=tt1; dy:=tt2;
                CB(X[i,1],Y[i,1]);
                P[i]:=ARCTAN(PX[i]+dx,PY[i]+dy);
L5:     end;
        if j≥entier(lim) then
        begin   AM(xe,ye);
                i0:=tr;
                if amn>AFST(X[i0,2],Y[i0,2],X[i0,0],Y[i0,0]) then goto NEG else
                begin   pp:=AFST(xe,ye,X[i0,0],Y[i0,0])/
                        AFST(X[i0,2],Y[i0,2],X[i0,0],Y[i0,0])-ur/12;
                        goto EIND
                end;
NEG:
        end;
        if j≥entier(lim1) ∧ joke then
        begin   AM(xk,yk);
                i00:=tr;
                vh:=1-amn/AFST(X[i00,2],Y[i00,2],X[i00,0],Y[i00,0]);
                if vh<0 then goto NEG1 else begin joke:=false; j00:=j; i01:=i00 end;
NEG1:
        end;
        if j≥entier(lim1) ∧ joke1 then LS;
        for i:=0,i+1 while i≤in do
        begin   if Y[i,0]=-99 then goto L7;
                if Y[i,2]>10 ∨ Y[i,2]<1.35 ∨ X[i,2]<1.5 ∨ (X[i,2]≥13 ∧ Y[i,2]<4.5)
                then goto L6;
                if X[i,2]≥13 ∧ X[i,2]<13.58 ∧ Y[i,2]<5.18 then goto L6;
                if X[i,2]≥13.58 ∧ X[i,2]<13.906 ∧ Y[i,2]<4.86 then goto L6;
                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 L6
                end;
                goto L7;
L6:             Y[i,2]:=-99;
L7:
        end;
L8:     if Y[i1,2]=-99 then begin i1:=i1+1; goto L8 end;
L9:     if Y[in,2]=-99 then begin in:=in-1; goto L9 end;
        PIA[j]:=pia:=i1;
```

```
        for i:=i1,i+1 while i≤in do
        begin   X[i,1]:=X[i,0]; Y[i,1]:=Y[i,0];
                ip:=i-pia;
                if Y[i,0]=-99 then goto L10;
                X[i,0]:=X[i,2]; Y[i,0]:=Y[i,2];
L10:            X[ip,0]:=X[i,0]; Y[ip,0]:=Y[i,0];
                X[ip,1]:=X[i,1]; Y[ip,1]:=Y[i,1];
                X[ip,2]:=X[i,2]; Y[ip,2]:=Y[i,2];
                LA[ip]:=LA[i]; MU[ip]:=MU[i]; P[ip]:=P[i]
        end;
        i1:=i:=0; in1:=in:=in-pia; i00:=i00-pia;
L11:    ip:=i+pia;
        if Y[i,0]=-99 ∨ Y[i+1,0]=-99 then goto L12;
        xv:=X[i,0]-X[i+1,0]; yv:=Y[i,0]-Y[i+1,0];
        if xv×xv+yv×yv>0.02 then
        begin   for h:=in,h-1 while h>i do
                begin   h1:=h+1;
                        X[h1,0]:=X[h,0]; Y[h1,0]:=Y[h,0];
                        LA[h1]:=LA[h]; MU[h1]:=MU[h]; P[h1]:=P[h]
                end;
                i11:=i+1; i2:=i+2;
                if j=j00 ∧ i<i00 then i00:=i00+1;
                X[i11,0]:=X[i,0]-xv/2; Y[i11,0]:=Y[i,0]-yv/2;
                LA[i11]:=(LA[i]+LA[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 L11
        end else
L12:    begin i:=i+1; if i<in then goto L11 end;
        if ¬ joke ∧ j=j00 then
        begin   i02:=i01;
                if i01>in1+pia then begin i00:=i00-i01+in1+pia; i01:=in1+pia end;
                i0001:=i00;
                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];
                        LA[h1]:=LA[h]; MU[h1]:=MU[h]; P[h1]:=P[h]
                end;
                inarray(drum,(j-1)×600,GH);
                inarray(drum,(j-1)×600+30000,GR); hold(GR);
                at3:=at-3;
                for i:=0,i+1 while i<at3 do
                begin   h:=i00+2+i;
                        A[h]:=(hk2-i)×grd;
                        LA[h]:=cos(A[h]); MU[h]:=sin(A[h]);
                        MAXAF(xk,yk,A[h]);
                        X[h,1]:=xk+vh×dx; Y[h,1]:=yk+vh×dy
                end;
```

```
            inarray(drum,j×600,GH); inarray(drum,j×600+30000,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]:=(vh×dx+X[h,1]+xk)/2; Y[h,0]:=(vh×dy+Y[h,1]+yk)/2;
                    P[h]:=ARCTAN(X[h,0]-xk,Y[h,0]-yk)
            end;
            1001:=100+at;
            Y[100+1,0]:=Y[1001-1,0]:=-99;
            X[1001,0]:=X[100,0]; Y[1001,0]:=Y[100,0];
            LA[1001]:=LA[100]; MU[1001]:=MU[100];
            P[1001]:=P[100];
            for i:=in1+pia step -1 until 101+1 do Y[-pia+at+i,2]:=Y[-pia+i,2];
            for i:=1,i+1 while i≤at do Y[101-pia+i,2]:=Y[100+i,0];
            in:=in+at; at1:=at
      end;
      for i:=0,i+1 while i≤in do Y[i,1]:=Y[i,0];
      i:=0;
L13:  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 L13
      end;
      i:=i+1;
      if i+1<in then goto L13;
      outarray(drum,4×(j-1)×451+100000,X);
      outarray(drum,2×(2×j-1)×451+100000,Y); hold(Y);
      k:=i:=f:=0;
L14:  if Y[i,2]=Y[k,1] then
      begin   k:=k+1; i:=i+1;
              if j≠j00 then at1:=0;
              if i≤in1+at1 then goto L14
      end
      else begin f:=f+1; PB[f]:=k; k:=k+1; goto L14 end;
L15:  for h:=f+1,h+1 while h<151 do PB[h]:=1000;
      k:=i:=f:=0;
L16:  if Y[k,1]=Y[i,0] then
      begin   k:=k+1; i:=i+1;
              if i≤in then goto L16
      end
      else begin f:=f+1; PA[f]:=i; k:=k+1; goto L16 end;
L17:  for h:=f+1,h+1 while h<151 do PA[h]:=1000;
      outarray(drum,2×(j-1)×250+60000,PB);
      outarray(drum,(2×j-1)×250+60000,PA); hold(PA);
      goto NEXTR;
```

```
EIND:  j0:=j:=j-1; PIA[j+1]:=0; I[j]:=10; goto L21;
L18:   inarray(drum,2×j×250+60000,PB);
       inarray(drum,(2×j+1)×250+60000,PA); hold(PA);
       f:=150;
L19:   if I[j]≥PA[f] then I[j]:=I[j]+f else
       begin f:=f-1; if f>0 then goto L19 end;
       f:=150;
L20:   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 L20 end;
L21:   if j=j0 then begin xxx:=XOR[j+1]:=xe; yyy:=YOR[j+1]:=ye end
       else begin xxx:=xx; yyy:=yy end;
       inarray(drum,4×(j-1)×451+100000,X);
       inarray(drum,2×(2×j-1)×451+100000,Y); hold(Y);
       I[j]:=I[j]+PIA[j+1];
       xx:=X[I[j],0]; yy:=Y[I[j],0];
       if j=j00-1 ∧ (((yyy-yy)/(xxx-xx))×(xk-xxx)+yyy)<yk
       then afst1:=afst1+AFST(xx,yy,xk,yk)+AFST(xxx,yyy,xk,yk)
       else afst1:=afst1+AFST(xx,yy,xxx,yyy);
       if j=1 then afst1:=(afst1+AFST(xx,yy,xs,ys))/185200;
       XOR[j]:=xx; YOR[j]:=yy;
       j:=j-1;
       if j>0 then
       begin  if j=j00-1 ∧ I[j00]>10001 ∧ I[j00]<10001+at-1 then
              begin  PIA[j+1]:=0; I[j]:=102; goto L21 end else
              begin  if j=j00-1 ∧ I[j00]>10001+at-2 then I[j]:=I[j+1]-at
                     else I[j]:=I[j+1]
              end;
              goto L18
       end;
       pu1:=(j0+pp)×12;
       CARRIAGE(5);
       for i:=1,i+1 while i<t1 do PRSYM(RS[i]);
       CARRIAGE(8); PRINTTEXT(⊰crossing time:⊱); SPACE(18);
       PRINTTEXT(⊰days              hours⊱); CARRIAGE(3);
       PRINTTEXT(⊰optimal  route⊱);
       P1(pu1);
       j:=j0+1;
       if (xg<xe ∧ tk>0) ∨ (xg>xe ∧ tk<0) then
       begin
L22:          if joke1 then LS;
              j:=j+1;
              MAXAF(xg,yg,10);
              ll1:=sqrt(dx×dx+dy×dy);
              if (xg≥xe ∧ tk>0) ∨ (xg≤xe ∧ tk<0) then goto L24;
              goto L22
       end else
       begin
L23:          if (XGC[j-1]>xe ∧ tk>0) ∨ (XGC[j-1]<xe ∧ tk<0)
              then  begin j:=j-1; goto L23 end
       end;
```

```
L24:  pu2:=(j-1+(xe-XGC[j-1])/(XGC[j]-XGC[j-1]))×12-ur;
      for i:=j,i-1 while i>j001 do begin XGC[i]:=XGC[i-1]; YGC[i]:=YGC[i-1] end;
      XGC[j001]:=xk; YGC[j001]:=yk;
      jg:=j:=j+1;
      for i:=1,i+1 while i<j do afst2:=afst2+AFST(XGC[i],YGC[i],XGC[i-1],YGC[i-1]);
      afst2:=(afst2+AFST(xe,ye,XGC[j-1],YGC[j-1]))/185200;
      XGC[j]:=xe; YGC[j]:=ye;
      PRINTTEXT(⊀great circle  ⊁);
      P1(pu2);
      j:=j0+1;
      if (xl<xe ∧ tk>0) ∨ (xl>xe ∧ tk<0) then
      begin
L25:         j:=j+1;
             MAXAF(xl,yl,20);
             if (xl≥xe ∧ tk>0) ∨ (xl≤xe ∧ tk<0) then goto L27;
             goto L25
      end else
      begin
L26:         if (XLX[j-1]>xe ∧ tk>0) ∨ (XLX[j-1]<xe ∧ tk<0)
             then begin j:=j-1; goto L26 end
      end;
L27:  jl:=j;
      pu3:=(j-1+(xe-XLX[j-1])/(XLX[j]-XLX[j-1]))×12-ur;
      for i:=1,i+1 while i<j do afst3:=afst3+AFST(XLX[i],YLX[i],XLX[i-1],YLX[i-1]);
      afst3:=(afst3+AFST(xe,ye,XLX[j-1],YLX[j-1]))/185200;
      XLX[j]:=xe; YLX[j]:=ye;
      PRINTTEXT(⊀rhumb line    ⊁);
      P1(pu3);
      CARRIAGE(3); PRINTTEXT(⊀distance covered:              miles⊁);
      CARRIAGE(3); PRINTTEXT(⊀optimal route⊁); SPACE(17); ABSFIXT(5,0,afst1);
      NLCR; NLCR; PRINTTEXT(⊀great circle⊁); SPACE(18); ABSFIXT(5,0,afst2);
      NLCR; NLCR; PRINTTEXT(⊀rhumb line⊁); SPACE(20); ABSFIXT(5,0,afst3);
      P2(XOR,YOR,j0+1,⊀     optimal⊁);
      P2(XGC,YGC,jg,⊀great circle⊁); P2(XLX,YLX,jl,⊀  rhumb line⊁);
AFL:
end
```

References

BLEICK, W. E. and F. D. FAULKNER ,1965. Minimal-time ship routing. Journ. appl. meteor., vol. 4-no. 2, pp. 217-221.

BOLZA, O., 1909. Vorlesungen über variationsrechnung. B. G. Teubner, Leipzig.

BRADDOCK, R. D., 1968. Optimal problems in physical oceanography, research papers 19 and 21. Horace Lamb Centre, Flinders University of South Australia.

BRYSON, A. E. and W. F. DENHAM, 1962. A steepest-ascent method for solving optimum programming problems. Journ. appl. mech., vol. 29, pp. 247-257.

BIJLSMA, S. J. en B. VAN RIETSCHOTE, 1972. Over het optimaal routeren van schepen, I Een literatuuronderzoek. II Een vergelijkend onderzoek. III Een vergelijkend onderzoek (vervolg). Wetensch. rapporten Kon. Ned. Meteor. Inst., De Bilt, WR 72-1, WR 72-2, WR 72-11.

BIJLSMA, S. J. en G. D. G. FOLKERS, 1973. Over het optimaal routeren van schepen, IV Het berekenen van golfvelden uit windvelden op de Atlantische Oceaan. Wetensch. rapporten Kon. Ned. Meteor. Inst., De Bilt, WR 73-2.

BIJLSMA, S. J., B. VAN RIETSCHOTE en G. D. G. FOLKERS, 1973. Over het optimaal routeren van schepen, V Een geautomatiseerd systeem. Wetensch. rapporten Kon. Ned. Meteor. Inst. De Bilt, WR 73-5.

CARATHEODORY, C., 1904. Über die diskontinuierlichen lösungen der variationsrechnung. Dissertation. Göttingen.

CODDINGTON, A. E. and N. LEVINSON, 1955. Theory of ordinary differential equations. McGraw-Hill Book Co., Inc., New York.

FAULKNER, F. D., 1963. Numerical methods for determining optimum ship routes. Journ. Inst. of Navigation, vol. 10-no. 4, pp. 351-367.

FAULKNER, F. D., 1964. A comparison between some methods for computing optimum paths in the problem of Bolza. In: Computing methods in optimization problems. A. V. Balakrishnan and L. W. Neustadt (eds), Acad. Press Inc., New York.

HALKIN, H., 1964. On the necessary conditions for optimal control of nonlinear systems. Journ. d'analyse math., Vol. XII, Jerusalem.

HESTENES, M. R., 1966. Calculus of variations and optimal control theory. J. Wiley, Inc., New York.

JAMES, R. W., 1957. Application of wave forecasts to marine navigation. Navy Hydrogr. Office, SP-1. Washington D.C.

KELLEY, H. J., 1962. Method of gradients. In: Optimization techniques. George Leitmann (ed.), Acad. Press Inc., New York.

MARKS, W. ET AL., 1968. An automated system for optimum ship routing. Trans. Soc. of Naval Architects and Marine Engineers, vol. 76, pp. 22-55.

MAURY, M. F., 1855. Physical geography of the sea. Harper Bros, New York.

PONTRYAGIN, L. S. ET AL., 1962. The mathematical theory of optimal processes. Interscience publishers, Inc., New York.

WIT, C. DE, 1968. Mathematical treatment of optimal ocean ship routeing. Dissertation. Techn. Hogeschool, Delft.

ZOPPOLI, R., 1972. Minimum-time routing as an N-stage decision process. Journ. appl. meteor., vol. 11-no. 3, pp. 429-435.