1986-03

Some tactical algorithms for spherical geometry

Shudde, Rex H.

Monterey, California. Naval Postgraduate School

NPS55-86-008

NAVAL POSTGRADUATE SCHOOL

Monterey, California

TECHNICAL

SOME TACTICAL ALGORITHMS FOR
SPHERICAL GEOMETRY

REX H. SHUDDE

MARCH 1986

Prepared for:
Director of Tactical Readiness Division
Office of the Chief of Naval Operations
Department of the Navy
Washington, D. C. 20350

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral Robert H. Shumaker
Superintendent

David A. Schrady
Provost

This report was prepared by:

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION: UNCLASSIFIED

1b RESTRICTIVE MARKINGS:

2a SECURITY CLASSIFICATION AUTHORITY:

2b DECLASSIFICATION/DOWNGRADING SCHEDULE:

3 DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited

4 PERFORMING ORGANIZATION REPORT NUMBER(S): NPS55-86-008

5 MONITORING ORGANIZATION REPORT NUMBER(S):

6a NAME OF PERFORMING ORGANIZATION: Naval Postgraduate School

6b OFFICE SYMBOL (If applicable): Code 55

7a NAME OF MONITORING ORGANIZATION: Director of Tactical Readiness Division Office of Chief of Naval Operations

6c ADDRESS (City, State, and ZIP Code): Monterey, CA 93943-5000

7b ADDRESS (City, State, and ZIP Code): Washington, D.C. 20350

8a NAME OF FUNDING/SPONSORING ORGANIZATION: Naval Air Development Center

8b OFFICE SYMBOL (If applicable):

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER:

8c ADDRESS (City, State, and ZIP Code): Warminster, PA 18974

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification): SOME TACTICAL ALGORITHMS FOR SPHERICAL GEOMETRY

12 PERSONAL AUTHOR(S): SHUDDE, REX H.

13a TYPE OF REPORT: Technical

13b TIME COVERED: FROM ______ TO ______

14 DATE OF REPORT (Year, Month, Day): 1986, March

15 PAGE COUNT: 30

16 SUPPLEMENTARY NOTATION:

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Navigation, CPA, Intercept, Great Circle, Closest Point of Approach, Spherical Earth Vectors

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This report presents two great circle navigation algorithms, a closest point of approach (CPA) algorithm and intercept algorithm. It also presents an implementation program for the algorithms that is written in the BASIC language for an IBM PC. Instead of using classical spherical geometry or the general spherical triangle, these algorithms incorporate rectangular coordinates and vectors on the surface of the spherical . The intent of the report is to provide algorithms for spherical earth models that can be used for situations in which flat earth models are not appropiate, but that do not require the sophistication of rotating earth models.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL: Rex H. Shudde

22b TELEPHONE (Include Area Code): (408) 646-2303

22c OFFICE SYMBOL: Code 55 u

CONTENTS

I. INTRODUCTION.

This report presents two great circle navigation algorithms, a closest point of approach (CPA) algorithm and an intercept algorithm. It also presents an implementation program that is written in the BASIC language for an IBM PC. Instead of using classical spherical geometry or the general spherical triangle [Ref. 1], these algorithms incorporate rectangular coordinates and vectors on the surface of the sphere. Portions of the vector formalism were suggested by Reference 2.

The intent of the report is to provide algorithms for spherical earth models that can be used for situations in which flat earth models are not appropriate, but that do not require the sophistication of rotating earth models.

II. RECTANGULAR COORDINATES AND VECTORS ON A SPHERE

In a spherical earth model, a point P on the earth's surface can be located by a position vector $\mathbf{p} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ in a rectangular coordinate system with origin at the earth's center. In matrix form, and with x, y and z expressed in spherical coordinates,

$$\mathbf{p} = \begin{bmatrix} r\cos\phi\cos\lambda \\ r\cos\phi\sin\lambda \\ r\sin\phi \end{bmatrix}$$

where ϕ is the latitude and λ is the longitude at the point and r is the distance of the point from the earth's center. See Figure 1.

In terms of the unit vector $\mathbf{x}$ where

$$\mathbf{x} = \begin{bmatrix} x_\mathbf{i} \\ x_\mathbf{j} \\ x_\mathbf{k} \end{bmatrix} = \begin{bmatrix} \cos\phi\cos\lambda \\ \cos\phi\sin\lambda \\ \sin\phi \end{bmatrix}, \tag{1}$$

$\mathbf{p} = r\mathbf{x}$. We can think of $\mathbf{x}$ as the unit vector normal to the surface at the point. It is convenient to define two other unit vectors in the tangent plane to the earth's surface at the point. These are local north $\mathbf{n}$ and local east $\mathbf{e}$, defined by

$$\mathbf{n} = \frac{\mathbf{x}_\phi}{|\mathbf{x}_\phi|} = \begin{bmatrix} -\sin\phi\cos\lambda \\ -\sin\phi\sin\lambda \\ \cos\phi \end{bmatrix} \quad \text{and} \quad \mathbf{e} = \frac{\mathbf{x}_\lambda}{|\mathbf{x}_\lambda|} = \begin{bmatrix} -\sin\lambda \\ \cos\lambda \\ 0 \end{bmatrix}, \tag{2}$$

where

$$\mathbf{x}_\phi \equiv \frac{\partial \mathbf{x}}{\partial \phi} \quad \text{and} \quad \mathbf{x}_\lambda \equiv \frac{\partial \mathbf{x}}{\partial \lambda}.$$

The vectors $\mathbf{x}$, $\mathbf{n}$ and $\mathbf{e}$ provide the basis for a right-handed orthogonal coordinate system.

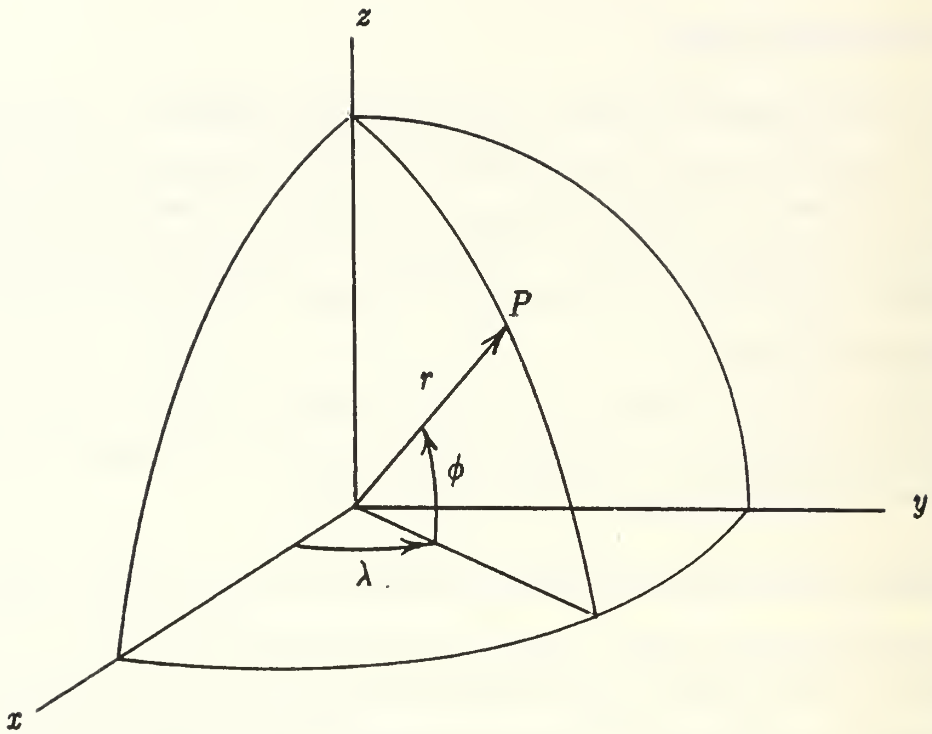

Figure 1. The Earth Coordinate System

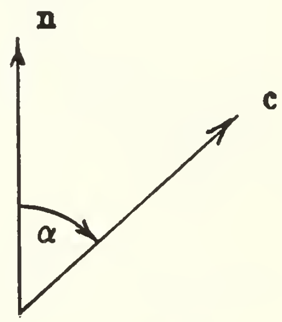

Figure 2. The Course Vector

Let a course α be designated at a point on the earth's surface. We wish to determine the unit course vector $\mathbf{c}$ in the direction α which lies in the tangent plane at the point in terms of ϕ, λ and α (see Fig. 2). An arbitrary vector $\mathbf{a}$, defined at the point P with coordinates (r, λ, ϕ) and lying in the tangent plane at the point, can be rotated clockwise (looking from P toward the origin) through an angle α by using the operation $\mathbf{R}_p(\alpha)\mathbf{a}$, where $\mathbf{R}_p(\alpha)$ is the rotation operator. $\mathbf{R}_p(\alpha)$ is a composite rotation in 3-space and can be decomposed into fundamental rotations in one of several ways. One way is to proceed as follows: First, move P (carrying $\mathbf{a}$ with P) along its parallel of latitude to the x-z plane (the Greenwich meridian); this is equivalent to a clockwise rotation about the z-axis through an angle λ and is denoted by $\mathbf{R}_z(\lambda)$. Next, move P along the Greenwich meridian to the equator: this is equivalent to a counterclockwise rotation about the y-axis through an angle ϕ and is denoted by $\mathbf{R}_y(-\phi)$. The point P now lies on the x-axis with coordinates $(r, 0, 0)$ and the vector $\mathbf{a}$ at P makes the same angle with the Greenwich meridian as it did with respect to the original meridian at (r, λ, ϕ). Next, leave P on the Greenwich meridian at the equator and rotate $\mathbf{a}$ through an angle α about the x-axis; this clockwise rotation is denoted by $\mathbf{R}_x(\alpha)$. The vector $\mathbf{a}$ has now been rotated through the desired angle α with respect to the Greenwich meridian. When P is returned to its original position by reversing the steps which got it to the equator on the Greenwich meridian, it will have been rotated through the angle α with respect to the original meridian of P, *i.e.* $\mathbf{R}_y(\phi)$ followed by $\mathbf{R}_z(-\lambda)$. The composite rotation operator $\mathbf{R}_p(\alpha)$ is

$$\mathbf{R}_p(\alpha) = \mathbf{R}_z(-\lambda)\mathbf{R}_y(\phi)\mathbf{R}_x(\alpha)\mathbf{R}_y(-\phi)\mathbf{R}_z(\lambda).$$

The course vector can then be written as $\mathbf{c} = \mathbf{R}_p(\alpha)\mathbf{n}$. The fundamental x-, y- and z-axis rotation operators are given by

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix},$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \quad \text{and}$$

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

These rotation operators are consistent with a right-handed coordinate system and positive signs of θ for a counterclockwise rotation of the coordinate system or a clockwise rotation of a point P about the x-, y- or z-axes, respectively, as viewed looking toward the origin from the positive end of the rotation axis [Ref. 3, pg. 43 and Ref. 4, pg. 100]. Some simplification in determining $\mathbf{R}_p(\alpha)\mathbf{n}$ can be obtained by noting that

$$\mathbf{R}_y(-\phi)\mathbf{R}_z(\lambda)\mathbf{n} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \equiv \mathbf{k}.$$

Thus $\mathbf{c}$ can be found from

$$\mathbf{c} = \mathbf{R}_z(-\lambda)\mathbf{R}_y(\phi)\mathbf{R}_x(\alpha)\mathbf{k}. \tag{3}$$

If an object's course vector $\mathbf{c}$ is known at some point with position vector $\mathbf{p}$ then it is easily shown (see Fig. 3) that

$$\mathbf{n} \cdot \mathbf{c} = \cos\alpha \quad \text{and}$$

$$\mathbf{e} \cdot \mathbf{c} = \sin\alpha.$$

From these relations, the course α is found to be

$$\alpha = \mathrm{qatn}(\mathbf{e} \cdot \mathbf{c}, \mathbf{n} \cdot \mathbf{c}) \tag{4}$$

where qatn is a quadrant determining arctangent function (see Appendix A and Ref. 5).

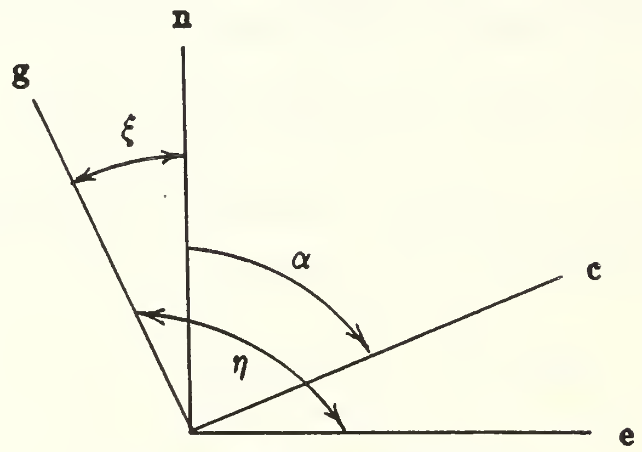

Figure 3. An Object's Course

The course can also be determined from the great circle vector $\mathbf{g}$ that is defined by

$$\mathbf{g} = \mathbf{x} \times \mathbf{c}, \tag{5}$$

where $\mathbf{p} = r\mathbf{x}$. From Figure 3 we see that the following relationships hold:

$$\mathbf{n} \cdot \mathbf{g} = \cos\xi = \sin\alpha \quad \text{and}$$

$$\mathbf{e} \cdot \mathbf{g} = \cos\eta = -\cos\alpha,$$

whence

$$\alpha = \mathrm{qatn}(\mathbf{n} \cdot \mathbf{g}, -\mathbf{e} \cdot \mathbf{g}) \tag{6}$$

If the object is traveling with speed v and is not maneuvering, it's course will be a great circle. Let $\mathbf{v}_0 = v\mathbf{c}_0$ denote the object's velocity vector, where $\mathbf{c}_0$ is its course and $\mathbf{p}_0$ is its position vector at time 0. At some subsequent time t, the object's position vector will be $\mathbf{p}(t)$ and

$$\mathbf{p}(t) = a\mathbf{p}_0 + b\mathbf{v}_0 t \tag{7}$$

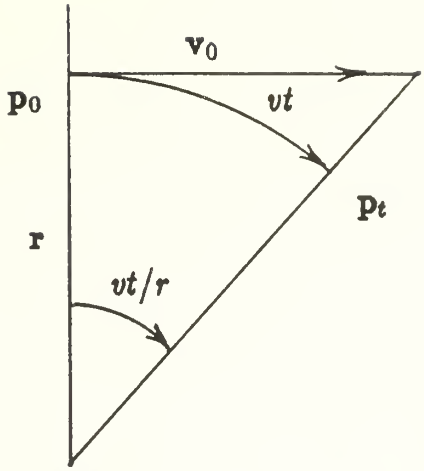

Figure 4. The Velocity Vector

where a and b are to be determined (see Fig. 4). Dotting Equ. 7 from the right by $\mathbf{p}_0$, we see that

$$\mathbf{p}(t) \cdot \mathbf{p}_0 = a\mathbf{p}_0 \cdot \mathbf{p}_0$$

or, with angles in radians,

$$a = \frac{\mathbf{p}(t) \cdot \mathbf{p}_0}{\mathbf{p}_0 \cdot \mathbf{p}_0} = \frac{1}{r^2}\mathbf{p}(t) \cdot \mathbf{p}_0 = \frac{1}{r^2}\left[r^2 \cos\left(\frac{vt}{r}\right)\right]$$

or

$$a = \cos\left(\frac{vt}{r}\right).$$

Similarly, dotting Equ. 7 from the right by $\mathbf{v}_0$ we find

$$\mathbf{p}(t) \cdot \mathbf{v}_0 = b\mathbf{v}_0 \cdot \mathbf{v}_0 t = bv^2 t$$

or

$$b = \frac{1}{v^2 t}\mathbf{p}(t) \cdot \mathbf{v}_0 = \frac{1}{v^2 t}\left[rv \cos\left(\frac{\pi}{2} - \frac{vt}{r}\right)\right],$$

so that

$$b = \frac{r}{vt} \sin\left(\frac{vt}{r}\right).$$

Thus,

$$\mathbf{p}(t) = \mathbf{p}_0 \cos\left(\frac{vt}{r}\right) + \mathbf{v}_0 \frac{r}{v} \sin\left(\frac{vt}{r}\right).$$

In terms of the unit vectors, this equation becomes

$$r\mathbf{x}(t) = r\mathbf{x}_0 \cos\left(\frac{vt}{r}\right) + v\mathbf{c}_0 \frac{r}{v} \sin\left(\frac{vt}{r}\right),$$

or

$$\mathbf{x}(t) = \mathbf{x}_0 \cos\left(\frac{vt}{r}\right) + \mathbf{c}_0 \sin\left(\frac{vt}{r}\right). \tag{8}$$

Applications of the these relations are made in the following sections of this report.

III. GREAT CIRCLE NAVIGATION.

The "Direct Solution Algorithm" computes the latitude and longitude of a position P_2 and the backward azimuth from P_2 to P_1 given the latitude and longitude of a position P_1, the forward azimuth from P_1 to P_2 and the distance from P_1 to P_2. The "Inverse Solution Algorithm" computes the distance from position P_1 to position P_2, the forward azimuth from P_1 to P_2, and the backward azimuth from P_2 to P_1 given the latitude and longitude of positions P_1 and P_2. Details of these algorithms using the concept of the general spherical triangle are presented in Reference 5. These models are redeveloped here using the concepts of Section II.

A. The Direct Solution Algorithm. Given $P_1(\phi_1, \lambda_1)$, forward azimuth (bearing) α_{12} and distance d, find ϕ_2 and λ_2 of P_2 and the backward azimuth α_{21}. Proceed as follows:

1. From ϕ_1 and λ_1, compute the components of $\mathbf{x}_1$ using Equ. 1.
2. Compute $\mathbf{c}_1$ from
$$\mathbf{c}_1 = \mathbf{R}_z(-\lambda_1)\mathbf{R}_y(\phi_1)\mathbf{R}_x(\alpha_{12})\mathbf{k}.$$
3. Compute $\mathbf{x}_2$ using
$$\mathbf{x}_2 = \mathbf{x}_1 \cos\left(\frac{d}{r}\right) + \mathbf{c}_1 \sin\left(\frac{d}{r}\right).$$
Note, with $d = vt$ in nautical miles and $r = 60(180°/\pi)$, Equ. 8 becomes
$$\mathbf{x}_2 = \mathbf{x}_1 \cos\left(\frac{d}{60}\right) + \mathbf{c}_1 \sin\left(\frac{d}{60}\right),$$
where the arguments of the cosine and sine are now in degrees.
4. From the components of $\mathbf{x}_2$ compute
$$\phi_2 = \sin^{-1}(x_{\mathbf{k}2}) \quad \text{and}$$
$$\lambda_2 = \text{qatn}(x_{\mathbf{j}2}, x_{\mathbf{i}2}).$$
5. Compute $\mathbf{g} = \mathbf{x}_1 \times \mathbf{c}_1$.
6. Compute $\mathbf{n}_2$ and $\mathbf{e}_2$ using Equ. 2.
7. Using Equ. 6 compute $\alpha_{21} = -\,\text{qatn}(\mathbf{n}_2 \cdot \mathbf{g}, -\mathbf{e}_2 \cdot \mathbf{g})$. Note that the sign change occurs because α_{21} is the *backward* azimuth.

B. The Inverse Solution Algorithm. Given $P_1(\phi_1, \lambda_1)$ and $P_2(\phi_2, \lambda_2)$, find the distance d from P_1 to P_2, the forward azimuth α_{12} from P_1 to P_2 and the backward azimuth α_{21} from P_2 to P_1. Proceed as follows:

1. From ϕ_i and λ_i compute $\mathbf{x}_i$, for $i = 1, 2$.
2. Compute $d = r\cos^{-1}(\mathbf{x}_1 \cdot \mathbf{x}_2)$, whence $d = 60(180/\pi)\cos^{-1}(\mathbf{x}_1 \cdot \mathbf{x}_2)$ is the distance in nautical miles.
3. Compute $\mathbf{n}_i$ and $\mathbf{e}_i$ for $i = 1, 2$ (see Equ. 2).
4. Compute
$$\mathbf{g} = \frac{\mathbf{x}_1 \times \mathbf{x}_2}{|\mathbf{x}_1 \times \mathbf{x}_2|}.$$
This equation arises because the great circle passes through both $\mathbf{x}_1$ and $\mathbf{x}_2$, hence $\mathbf{g}$ must be perpendicular to $\mathbf{x}_1$ and to $\mathbf{x}_2$.
5. Compute $\alpha_{12} = \operatorname{qatn}(\mathbf{n}_1 \cdot \mathbf{g}, -\mathbf{e}_1 \cdot \mathbf{g})$.
6. Compute $\alpha_{21} = -\operatorname{qatn}(\mathbf{n}_2 \cdot \mathbf{g}, -\mathbf{e}_2 \cdot \mathbf{g})$.

IV. CLOSEST POINT OF APPROACH (CPA) PROBLEM.

Consider two objects traveling on different great circle paths. From Equ. 8, their tracks will be characterized by the equations

$$\mathbf{x}_i(t) = \mathbf{x}_{i0}\cos\omega_i t + \mathbf{c}_{i0}\sin\omega_i t, \quad \text{for } i = 1, 2, \tag{9}$$

where $\omega_i \equiv v_i/r$. At any time t, their angular separation $s(t)$ in radians is determined by

$$\cos s(t) = \mathbf{x}_1(t) \cdot \mathbf{x}_2(t). \tag{10}$$

The time of minimum separation (CPA) is that time T when $\frac{d}{dt}[\cos s(t)] = 0$. That is, we must find T such that

$$\mathbf{x}_1(T) \cdot \frac{d\mathbf{x}_2(T)}{dt} + \frac{d\mathbf{x}_1(T)}{dt} \cdot \mathbf{x}_2(T) = 0.$$

Unfortunately, this equation cannot be solved explicitly. One approach is to use the Newton-Raphson iteration method [Ref. 6] to find the root T of $f(t)$, where

$$f(t) = \mathbf{x}_1(t) \cdot \frac{d\mathbf{x}_2(t)}{dt} + \frac{d\mathbf{x}_1(t)}{dt} \cdot \mathbf{x}_2(t).$$

Taking the required derivatives of Equ. 9 and performing the required dot products, we find that

$$f(t) = A\sin\omega_1 t\sin\omega_2 t + B\cos\omega_1 t\cos\omega_2 t + C\sin\omega_1 t\cos\omega_2 t + D\cos\omega_1 t\sin\omega_2 t,$$

where

$$A = -(\omega_1 \mathbf{x}_{10} \cdot \mathbf{c}_{20} + \omega_2 \mathbf{x}_{20} \cdot \mathbf{c}_{10}),$$
$$B = \omega_1 \mathbf{c}_{10} \cdot \mathbf{x}_{20} + \omega_2 \mathbf{c}_{20} \cdot \mathbf{x}_{10},$$
$$C = -(\omega_1 \mathbf{x}_{10} \cdot \mathbf{x}_{20} - \omega_2 \mathbf{c}_{20} \cdot \mathbf{c}_{10}) \quad \text{and}$$
$$D = \omega_1 \mathbf{c}_{10} \cdot \mathbf{c}_{20} - \omega_2 \mathbf{x}_{20} \cdot \mathbf{x}_{10}.$$

The use of the Newton-Raphson method requires that the derivative of $f(t)$ with respect to t, namely $f'(t)$, be computed or estimated. We find that

$$f'(t) = -(C\omega_2 + D\omega_1)\sin\omega_1 t \sin\omega_2 t + (D\omega_2 + C\omega_1)\cos\omega_1 t \cos\omega_2 t$$
$$+ (A\omega_2 - B\omega_1)\sin\omega_1 t \cos\omega_2 t - (B\omega_2 - A\omega_1)\cos\omega_1 t \sin\omega_2 t.$$

The Newton-Raphson method requires us to compute

$$t_{i+1} = t_i - \frac{f(t_i)}{f'(t_i)} \quad \text{for } i = 0, 1, \ldots \tag{11}$$

where t_0 is an initial estimate of T, until some value of i is found for which $f(t_i)/f'(t_i)$ is sufficiently close to zero.

The CPA option in the computer program (Appendix B) will print out the time to CPA (from time $t = 0$), the distance between objects at CPA and the bearing from object 1 from object 2 at CPA. Also printed is the number of iterations required for convergence of Equ. 11 to $|f(t_i)/f'(t_i)|$ less than 0.00001 hours. A negative time to CPA indicates that CPA has already occurred.

V. INTERCEPT PROBLEM.

The intercept problem is divided into two problems, each of which may require an answer. In both problems we are given the initial position, course and speed of a target as well as the position of an interceptor or launch platform. In the first problem we are given the time (or elapsed time) at which intercept is desired and are required to compute the vehicle speed needed for an intercept to take place. In the second problem we are given the speed of an intercept vehicle and wish to compute the time required for an intercept to occur provided an intercept can be made. Provision for both of these options is made in the program presented in Appendix B.

A. Speed Required to Intercept. Inputs are the initial latitude and longitude of an interceptor and a target, and the target course and speed. Also input is the time of the desired intercept. Outputs are the speed required of the interceptor, the course of the interceptor, the distance traveled to intercept, and the latitude and longitude of the intercept.

From the inputs, use Equ. 1 to compute $\mathbf{x}_{10}$ and $\mathbf{x}_{20}$, the position vectors of the interceptor and target, respectively, at time $t = 0$. Denote the time required to intercept by t. Compute the target course vector, $\mathbf{c}_{20}$, using Equ. 3 and then compute $\mathbf{x}_2(t)$, the position of the target at the time of intercept, using Equ. 8. The remainder of the problem is solved using the inverse solution algorithm discussed previously. The speed required for intercept is given by $v_1 = d/t$, where d is the distance from the initial interceptor position to the target position at the time of intercept.

B. Time Required to Intercept. Inputs are the initial latitude and longitude of an interceptor and a target, and the target course and speed. For a given interceptor speed, it may or may not be possible to make an intercept. We develop two sub-algorithms. The first algorithm is to compute the minimum interceptor speed required to achieve intercept and the time required to make such an intercept. The second algorithm queries the user to input an interceptor speed. If the speed is at least that required for intercept, then the time required to intercept is computed. (If the interceptor speed is greater than the minimum required, there are two possible solutions for the time to intercept—the shortest time to intercept is computed by the algorithm). Outputs are the minimum required interceptor speed, the time to intercept at minimum speed, the course of the interceptor, the distance traveled to intercept, and the latitude and longitude of the intercept.

The first problem is to determine the minimum speed, v_m, required to make an interception. This can be accomplished by finding the time of intercept t_m which makes $dv/dt = 0$. We can relate v to s, the angular separation in radians, between the two points $\mathbf{x}_{10}$ and $\mathbf{x}_2(t)$ by the relation $v(t) = rs(t)/t$. We find that $dv(t)/dt = 0$ implies

$$r\left[\frac{1}{t}\frac{ds}{dt} - \frac{s}{t^2}\right] = 0. \tag{12}$$

Anticipating that it will not be possible to find a closed form solution, we prepare to use the Newton-Raphson procedure (Equ. 11). Multiplying Equ. (12) by t^2 gives us the function

$$f(t) = t\frac{ds}{dt} - s$$

for which we wish to find t_m such that $f(t_m) = 0$. Also needed is

$$f'(t) = t\frac{d^2s}{dt^2}.$$

Using Equ. 10 to determine s(t) we find that

$$\frac{ds}{dt} = -\frac{1}{\sin s}\mathbf{x}_{10} \cdot \frac{dx_2(t)}{dt} \quad \text{and}$$

$$\frac{d^2s}{dt^2} = -\frac{1}{\sin s}\left[\frac{\cos s}{\sin^2 s}\left(\mathbf{x}_{10} \cdot \frac{d\mathbf{x}_2(t)}{dt}\right)^2 + \mathbf{x}_{10} \cdot \frac{d^2\mathbf{x}_2(t)}{dt^2}\right],$$

where

$$\mathbf{x}_{10} \cdot \frac{d\mathbf{x}_2(t)}{dt} = -\omega_2\left[\mathbf{x}_{10} \cdot \mathbf{x}_{20} \sin \omega_2 t - \mathbf{x}_{10} \cdot \mathbf{c}_{20} \cos \omega_2 t\right] \quad \text{and}$$

$$\mathbf{x}_{10} \cdot \frac{d^2\mathbf{x}_2(t)}{dt^2} = -\omega_2^2\left[\mathbf{x}_{10} \cdot \mathbf{x}_{20} \cos \omega_2 t + \mathbf{x}_{10} \cdot \mathbf{c}_{20} \sin \omega_2 t\right].$$

The Newton-Raphson procedure continues until t_m is found, then $v_m = sr/t_m$.

The second problem is to find the time of interception t_I when the interceptor's speed v_1 is given. Once more the Newton-Raphson procedure is used. As before, we can relate v_1 to $s(t)$, the angular separation in radians, between two points $\mathbf{x}_{10}$ and $\mathbf{x}_2(t)$ by the relation $v(t) = rs(t)/t$, which tells us that we must require $\omega_1 = s(t)/t$. That is, we wish to find t_I for which $s(t_I)/t_I$ equals the constant ω_1, or equivalently, we wish to find t_I such that $f(t_I) = 0$ where

$$f(t) = \frac{s}{t} - \omega_1.$$

Also needed is

$$f'(t) = \frac{1}{t}\left(\frac{ds}{dt} - \frac{s}{t}\right).$$

The equation for ds/dt is the same as that given in the previous paragraph. The remaining output is found using the inverse solution algorithm for the points $\mathbf{x}_{10}$ and $\mathbf{x}_2(t_I)$.

VI. SAMPLE PROBLEMS

Master Menu. The master menu for algorithm demonstration program is shown below.

```
                ALGORITHM DEMO

            1) DIRECT SOLUTION
            2) INVERSE SOLUTION
            3) FIND CPA
            4) SPEED NEEDED TO INTERCEPT
            5) TIME NEEDED TO INTERCEPT

            6) QUIT
```

Problem 1. Suppose you are at San Francisco (latitude 37°47′ north and longitude 122°25′ west), that your initial course is 260° and that you travel a distance of 4000 n. mi. What is your final position? Select Option 1 from the master menu:

```
              DIRECT SOLUTION

        1st LATITUDE   DD.MMSS (-S) ? 37.47
        1st LONGITUDE DDD.MMSS (-E) ? 122.25
        INITIAL COURSE  DDD.MMSS    ? 260
        DISTANCE (n. mi.)           ? 4000

      SPHERICAL EARTH DIRECT SOLUTION
          2nd LATITUDE       6°41.9′
          2nd LONGITUDE  -172°00.7′
          BACK AZIMUTH      51°35.9′

        PRESS ANY KEY TO CONTINUE
```

Problem 2. Suppose you are at San Francisco (latitude 37°47′ north and longitude 122°25′ west) and that your destination is Sydney, Australia (latitude 33°51′ south and longitude 151°13′ east). How far do you travel, what is your initial course, and what is the backward azimuth from Sydney to San Francisco? Select Option 2 from the master menu.

```
          INVERSE SOLUTION

     1st LATITUDE    DD.MMSS (-S) ? 37.47
     1st LONGITUDE DDD.MMSS (-E) ? 122.25
     2nd LATITUDE    DD.MMSS (-S) ? -33.51
     2nd LONGITUDE DDD.MMSS (-E) ? -151.13

   SPHERICAL INVERSE SOLUTION
       DISTANCE            6446.3 n.mi.
       FORWARD COURSE      240°18.9′
       BACK COURSE          55°45.9′

     PRESS ANY KEY TO CONTINUE
```

Problem 3. Suppose an observer is at 20° north, 60° west traveling on a course of 010° at a speed of 15 knots. A target is reported to be at 34° north, 50° west on a course of 220° at a speed of 300 knots. Assuming that neither observer or target changes course or speed, how much time will elapse until CPA and at CPA where will the target be with respect to the observer? Select Option 3 from the master menu.

```
                FIND CPA

1st LATITUDE   DD.MMSS (-S) ? 20
1st LONGITUDE DDD.MMSS (-E) ? 60
INITIAL COURSE  DDD.MMSS    ? 10
SPEED (knots)               ? 15
2nd LATITUDE   DD.MMSS (-S) ? 34
2nd LONGITUDE DDD.MMSS (-E) ? 50
INITIAL COURSE  DDD.MMSS    ? 220
SPEED (knots)               ? 300

TIME TO CPA     = 3h09m48s
DISTANCE AT CPA = 67.03 n.mi.
BEARING AT CPA  = 304°06.3'
NO. ITERATIONS  = 3

PRESS ANY KEY TO CONTINUE
```

As an additional output, a table of observer positions and target positions is given at CPA and six equally spaced times before and after CPA.

FIND CPA

TIME	LAT 1	LONG 1	DISTANCE	BEARING(1->2)
00s	20°00.0′	60°00.0′	994.34	30°18.8′
31m38s	20°07.8′	59°58.5′	829.45	29°31.6′
1h03m16s	20°15.6′	59°57.1′	664.78	28°21.6′
1h34m54s	20°23.4′	59°55.6′	500.56	26°26.3′
2h06m32s	20°31.2′	59°54.1′	337.43	22°39.8′
2h38m10s	20°38.9′	59°52.7′	178.42	12°02.7′
3h09m48s	20°46.7′	59°51.2′	67.03	304°06.3′
3h41m27s	20°54.5′	59°49.7′	178.43	236°10.0′
4h13m05s	21°02.3′	59°48.2′	337.43	225°33.2′
4h44m43s	21°10.1′	59°46.7′	500.58	221°47.3′
5h16m21s	21°17.9′	59°45.3′	664.81	219°52.7′
5h47m59s	21°25.7′	59°43.8′	829.5	218°43.7′
6h19m37s	21°33.4′	59°42.3′	994.41	217°57.6′

PRESS ANY KEY TO CONTINUE

Problem 4. Suppose an observer at 20° north, 60° west wishes to launch an interceptor at a target reported to be at 34° north, 50° west on a course of 220° at a speed of 600 knots. If interception is required to take place 45 minutes (2700 seconds) after launch, how fast must the interceptor travel, and where will the intercept take place? (Assume that the target does not change course or speed.) Select Option 4 from the master menu.

```
SPEED NEEDED TO INTERCEPT (1->2)

1st LATITUDE   DD.MMSS (-S) ? 20
1st LONGITUDE DDD.MMSS (-E) ? 60
2nd LATITUDE   DD.MMSS (-S) ? 34
2nd LONGITUDE DDD.MMSS (-E) ? 50
2nd COURSE    DDD.MMSS      ? 220
2nd SPEED (knots)           ? 600

TIME TO INTERCEPT (SECONDS) ? 2700

SPEED REQUIRED          =  730.1 knots
BEARING TO INTERCEPT    =  26°06.9'
RANGE TO INTERCEPT      =  547.5 n.mi.
LATITUDE OF INTERCEPT   =  28°08.0'
LONGITUDE OF INTERCEPT  =  55°27.6'

    1) CHANGE TIME OF INTERCEPT
    2) NEW PROBLEM
    3) MASTER MENU
```

Problem 5. As in the previous problem, suppose an observer at 20° north, 60° west wishes to launch an interceptor at a target reported to be at 34° north, 50° west on a course of 220° at a speed of 600 knots. If the maximum speed of the interceptor is 700 knots, how long will it take before interception can occur, what should be the interceptor's initial great circle heading, and where will the intercept take place? (Assume that the target does not change course or speed.) Select Option 5 from the master menu.

```
TIME NEEDED TO INTERCEPT (1->2)

1st LATITUDE   DD.MMSS (-S) ? 20
1st LONGITUDE DDD.MMSS (-E) ? 60
2nd LATITUDE   DD.MMSS (-S) ? 34
2nd LONGITUDE DDD.MMSS (-E) ? 50
2nd COURSE    DDD.MMSS      ? 220
2nd SPEED (knots)          ? 600

MINIMUM SPEED REQUIRED TO INTERCEPT    =  52.6 knots
TIME REQ'D TO INTERCEPT AT MIN SPEED   =  1h39m50s

INTERCEPTOR SPEED (knots)  ? 700

TIME REQUIRED           = 46m03s
BEARING TO INTERCEPT    = 25°56.1'
RANGE TO INTERCEPT      = 537.2 n.mi.
LATITUDE OF INTERCEPT   = 27°59.6'
LONGITUDE OF INTERCEPT  = 55°34.7'

    1) CHANGE INTERCEPTOR SPEED
    2) NEW PROBLEM
    3) MASTER MENU
```

V. REFERENCES.

1. Chauvenet, Wm., *A Treatise on Plane and Spherical Trigonometry*, 7th ed., J. B. Lippincott & Co., 1869.
2. "Computer Aided Stationing Tool (CAST)", Programmable Description Document, 11 May 1984, Prepared by MANTECH International Corporation
3. Mueller, I. I., *Spherical and Practical Astronomy as Applied to Geodesy*, Frederick Unger Publishing Co., 1969.
4. Goldstein, H., *Classical Mechanics*, Addison–Wesley Press, Inc., 1950.
5. Shudde, R. H., "Some Navigation and Almanac Algorithms", Naval Postgraduate School Technical Report NPS55–85–023, September 1985.
6. Hildebrand, F. B., *Introduction to Numerical Analysis*, McGraw-Hill Book Company, Inc., 1956.

APPENDIX A: The QATN Function

This routine is the standard arctangent function corrected for quadrant. The quadrant arctangent function is occasionally implemented as the ATAN2 function, the ANGLE function or the Rectangular-to-Polar function.

Entering variables are the x- and y-coordinates, X and Y. The exiting variable is the angle Θ, where $-\pi < \Theta \leq \pi$. Use of the quadrant arctangent function is denoted by $\Theta = \text{qatn}(Y, X)$.

1. If $X \neq 0$, go to step 4.
2. Set $\Theta = (\pi/2) * \text{sgn}(Y)$.
3. Go to step 8.
4. Set $\Theta = \arctan(Y/X)$.
5. If $X > 0$, go to step 8.
6. Set $\Theta = \Theta + \pi * \text{sgn}(Y)$.
7. If $Y = 0$, set $\Theta = \pi$.
8. Return.

Note:

If $Y > 0$ then $\text{sgn}(Y) = +1$.
If $Y = 0$ then $\text{sgn}(Y) = 0$.
If $Y < 0$ then $\text{sgn}(Y) = -1$.

Users of Microsoft BASIC can simplfy the qatn function significantly by using the code given below. To return an angle of Θ (designated by A in the code) in the range of $(-\pi, \pi)$, use:

```
PI = 4*ATN(1):  TP = PI + PI: EPS = 1E-33
A = ATN(Y/(X-EPS*(X=0))) - PI*(X<0)*(SGN(Y) - (Y=0))
```

To return a value of A in the range of $(0, 2\pi)$, use:

```
PI = 4*ATN(1):  TP = PI + PI: EPS = 1E-33
A = ATN(Y/(X-EPS*(X=0))) - PI*(X<0) + TP*(X >= 0)*(Y<0)
```

APPENDIX B: Program Listing

```
  10 ' "RECT COORD ALGORITHMS" R.H. SHUDDE, 03-03-86.  REV. 03-19-86 1000
  13 ' RECTALGR",A
 100 DEFDBL A-Z
 110 PI4=ATN(1#):PI2=PI4+PI4:PI=PI2+PI2:TP=PI+PI:RD=PI/180#:EPS=1D-33
 120 AE=60#*360#/TP
 130 DEF FNM(X)=X-MO*INT(X/MO):'  X MOD MO FUNCTION.
 140 DEF FNL(X)=X-TP*INT((X+PI)/TP):' LONGITUDE ADJUST (-PI,PI)
 150 DEF FNR(X)=INT(X*MO+.5)/MO:'  ROUNDING FUNCTION.
 160 DEF FNG(X)=X+PI*SGN(X)*(ABS(X)>PI2):'  LATITUDE ADJUST (-PI/2,PI/2)
 170 DEF FNACS(X)=ATN(SQR(1#-X*X)/(X-EPS*(X=0#)))-PI*(X<0#):'  ARCCOS
 180 DEF FNASN(X)=ATN(X/(SQR(1#-X*X)-EPS*(ABS(X)=1#))):'  ARCSIN
 190 'QATN (-PI,PI) FUNCTION:
 200 DEF FNATN2(Y,X)=ATN(Y/(X-EPS*(X=0#)))-PI*(X<0#)*(SGN(Y)-(Y=0#))
 210 'QATN (0,TWOPI) FUNCTION:
 220 DEF FNATNP(Y,X)=ATN(Y/(X-EPS*(X=0#)))-PI*(X<0#)+TP*(X>=0#)*(Y<0#)
 230 'CROSS PRODUCT: X(.)=X1(.)  X X2(.):
 240 DEF FNCX(X1,Y1,Z1,X2,Y2,Z2)=Y1*Z2-Z1*Y2
 250 DEF FNCY(X1,Y1,Z1,X2,Y2,Z2)=X2*Z1-Z2*X1
 260 DEF FNCZ(X1,Y1,Z1,X2,Y2,Z2)=X1*Y2-Y1*X2
 270 GOTO 2000
 280 '
1000 ' DECIMAL TO HH MM SS
1010 V$=" ":IF X<0 THEN V$="-":X=-X
1020 X=X+1/7200#:Y=INT(X):Z=LEN(STR$(Y))-1
1030 K%=0:IF Y<>0 THEN V$=V$+RIGHT$(" "+STR$(Y),Z)+"h":K%=1
1040 X=60#*(X-Y):Y=INT(X)
1050 IF Y<>0 OR K%=1 THEN X$=STR$(100#+Y):V$=V$+RIGHT$(X$,2)+"m"
1060 X=60#*(X-Y):Y=INT(X):X$=STR$(100#+Y):V$=V$+RIGHT$(X$,2)+"s":RETURN
1070 '
1080 ' DECIMAL TO DDD MM.F
1090 V$=" ":IF X<0 THEN V$="-":X=-X
1100 X=X+1/1200#:Y=INT(X):V$=V$+RIGHT$(" "+STR$(Y),3)+CHR$(248)
1110 X=600#*(X-Y):Y=INT(X):X$=STR$(1000+Y)
1120 V$=V$+MID$(X$,3,2)+"."+RIGHT$(X$,1)+"'":RETURN
1130 '
1140 ' DDD.MMSS TO DECIMAL
1150 IX=0:FOR Z!=1 TO LEN(V$):C$=MID$(V$,Z!,1):IF C$="."THEN IX=Z!
1160 NEXT:IF IX=0 THEN X=VAL(V$):RETURN
1170 X=VAL(LEFT$(V$,IX)):SN=1:IF X<0# THEN SN=-SN:X=-X
1180 V$=V$+"0000":Y=VAL(MID$(V$,IX+1,2)):Z=VAL(MID$(V$,IX+3,2))
1190 X=SN*((Z/60#+Y)/60#+X):RETURN
1200 '
1300 'DIRECT SOLUTION, SPHER EARTH-RECT COORD. ALL ANGLES MUST BE IN RADIANS.
1310 'INPUT: LATITUDE P1, LONGITUDE L1, FORWARD AZIMUTH A12 AND
1320 ' DISTANCE DD TO A POINT P2.  NOTE: DD IS IN RADIANS.
1330 'OUTPUT: LATITUDE P2, LONGITUDE L2 AND BACKWARD AZIMUTH A21.
1340 P=P1:L=-L1:GOSUB 1750:'CHNG SIGN OF L1 GIVES RIGHT-HANDED COORDS
1350 FOR I!=1 TO 3:POS1(I!)=POSI(I!):NORTH1(I!)=NORTH(I!):EAST1(I!)=EAST(I!):
     NEXT
1360 GOSUB 1810:CD=COS(DD):SD=SIN(DD)
```

```
1370  FOR I!=1 TO 3:POS2(I!)=POS1(I!)*CD+CVEC1(I!)*SD:NEXT
1380  L2=FNATN2(POS2(2),POS2(1)):P2=FNASN(POS2(3)):P=P2:L=L2:GOSUB 1750
1390  X=GVEC(1)*EAST(1)+GVEC(2)*EAST(2)+GVEC(3)*EAST(3)
1400  Y=-(GVEC(1)*NORTH(1)+GVEC(2)*NORTH(2)+GVEC(3)*NORTH(3))
1410  A21=FNATNP(Y,X):L2=-L2:RETURN:'CONVERT BACK TO LEFT-HANDED OUTPUT
1420  '
1500  'INVERSE SOLUTION, SPHER EARTH-RECT COORD. ALL ANGLES MUST BE IN RADIANS.
1510  'INPUT: LATITUDES P1 & P2, AND LONGITUDES L1 & L2.
1520  'OUTPUT: DISTANCE DD TO A POINT P2.  (NOTE: 0 <= DD <= PI RADIANS).
1530  ' FORWARD AZIMUTH A12, AND BACKWARD AZIMUTH A21.
1540  P=P1:L=-L1:GOSUB 1750:'  CHNG SIGN OF L1 GIVES RIGHT-HANDED COORDS
1550  FOR I!=1 TO 3:POS1(I!)=POSI(I!):NORTH1(I!)=NORTH(I!):EAST1(I!)=EAST(I!):
      NEXT
1560  P=P2:L=-L2:GOSUB 1750:'  CHNG SIGN OF L2 GIVES RIGHT-HANDED COORDS
1570  FOR I!=1 TO 3:POS2(I!)=POSI(I!):NORTH2(I!)=NORTH(I!):EAST2(I!)=EAST(I!):
      NEXT
1580  DD=FNACS(POS1(1)*POS2(1)+POS1(2)*POS2(2)+POS1(3)*POS2(3))
1590  X=FNCX(POS1(1),POS1(2),POS1(3),POS2(1),POS2(2),POS2(3))
1600  Y=FNCY(POS1(1),POS1(2),POS1(3),POS2(1),POS2(2),POS2(3))
1610  Z=FNCZ(POS1(1),POS1(2),POS1(3),POS2(1),POS2(2),POS2(3))
1620  A12=FNATNP(X*NORTH1(1)+Y*NORTH1(2)+Z*NORTH1(3),
      -(X*EAST1(1)+Y*EAST1(2)+Z*EAST1(3)))
1630  A21=FNATNP(-(X*NORTH2(1)+Y*NORTH2(2)+Z*NORTH2(3)),
      X*EAST2(1)+Y*EAST2(2)+Z*EAST2(3))
1640  RETURN
1650  '
1700  CA=COS(A):SA=SIN(A):T=Y*CA+Z*SA:Z=Z*CA-Y*SA:Y=T:RETURN:' X-AXIS ROT
1710  CA=COS(A):SA=SIN(A):T=Z*CA+X*SA:X=X*CA-Z*SA:Z=T:RETURN:' Y-AXIS ROT
1720  CA=COS(A):SA=SIN(A):T=X*CA+Y*SA:Y=Y*CA-X*SA:X=T:RETURN:' Z-AXIS ROT
1730  '
1740  'UNIT VECTORS: POSITION, NORTH & EAST.
1750  SL=SIN(L):CL=COS(L):SP=SIN(P):CP=COS(P)
1760  POSI(1)=CP*CL:POSI(2)=CP*SL:POSI(3)=SP
1770  NORTH(1)=-SP*CL:NORTH(2)=-SP*SL:NORTH(3)=CP
1780  EAST(1)=-SL:EAST(2)=CL:EAST(3)=0:RETURN
1790  '
1800  'VECTORS:CVEC=COURSE & GVEC=GREAT CIRCLE NORMAL
1810  X=0:Y=0:Z=1:A=A12:GOSUB 1700:A=P:GOSUB 1710:A=-L:GOSUB 1720
1820  CVEC1(1)=X:CVEC1(2)=Y:CVEC1(3)=Z
1830  GVEC(1)=FNCX(POS1(1),POS1(2),POS1(3),CVEC1(1),CVEC1(2),CVEC1(3))
1840  GVEC(2)=FNCY(POS1(1),POS1(2),POS1(3),CVEC1(1),CVEC1(2),CVEC1(3))
1850  GVEC(3)=FNCZ(POS1(1),POS1(2),POS1(3),CVEC1(1),CVEC1(2),CVEC1(3))
1860  RETURN
1870  '
2000  CLS:PRINT SPC(20);"ALGORITHM DEMO
2010  PRINT:PRINT:PRINT
2020  PRINT SPC(15);"1) DIRECT SOLUTION
2030  PRINT SPC(15);"2) INVERSE SOLUTION
2040  PRINT SPC(15);"3) FIND CPA
2050  PRINT SPC(15);"4) SPEED NEEDED TO INTERCEPT
2060  PRINT SPC(15);"5) TIME NEEDED TO INTERCEPT
2070  PRINT:PRINT SPC(15);"6) QUIT
2080  GOSUB 9010:C=VAL(C$):ON C GOSUB 3000,4000,5000,6000,7000,8000
```

```
2090 GOTO 2000
2100 '
3000 CLS:PRINT SPC(15);"DIRECT SOLUTION":PRINT:PRINT
3010 PRINT SPC(10);:PRINT "1st LATITUDE DD.MMSS (-S) ";
3020 INPUT V$:GOSUB 1150:P1=X*RD
3030 PRINT SPC(10);:PRINT "1st LONGITUDE DDD.MMSS (-E) ";
3040 INPUT V$:GOSUB 1150:L1=X*RD
3050 PRINT SPC(10);:PRINT "INITIAL COURSE DDD.MMSS ";
3060 INPUT V$:GOSUB 1150:A12=X*RD
3070 PRINT SPC(10);:INPUT "DISTANCE (n.  mi.)  ?  ",D
3080 D1=D*RD/60#
3090 MO=100
3100 DD=D1:GOSUB 1340:PRINT:PRINT SPC(8);"SPHERICAL EARTH DIRECT SOLUTION
3110 PRINT SPC(12);"2nd LATITUDE ";:X=P2/RD:GOSUB 1090:PRINT V$
3120 PRINT SPC(12);"2nd LONGITUDE ";:X=L2/RD:GOSUB 1090:PRINT V$
3130 PRINT SPC(12);"BACK AZIMUTH ";:X=A21/RD:GOSUB 1090:PRINT V$
3140 GOSUB 9000:GOTO 2000
3150 '
4000 CLS:PRINT SPC(15);"INVERSE SOLUTION":PRINT:PRINT
4010 PRINT SPC(10);:PRINT "1st LATITUDE DD.MMSS (-S) ";
4020 INPUT V$:GOSUB 1150:P1=X
4030 PRINT SPC(10);:PRINT "1st LONGITUDE DDD.MMSS (-E) ";
4040 INPUT V$:GOSUB 1150:L1=X
4050 PRINT SPC(10);:PRINT "2nd LATITUDE DD.MMSS (-S) ";
4060 INPUT V$:GOSUB 1150:P2=X
4070 PRINT SPC(10);:PRINT "2nd LONGITUDE DDD.MMSS (-E) ";
4080 INPUT V$:GOSUB 1150:L2=X
4090 P1=P1*RD:P2=P2*RD:L1=L1*RD:L2=L2*RD
4100 DD=D1:GOSUB 1540
4110 PRINT:PRINT SPC(8);"SPHERICAL INVERSE SOLUTION
4120 '
4130 PRINT SPC(12);"DISTANCE ";
4140 MO=100:PRINT FNR(60#*DD/RD);" n.mi.
4150 PRINT SPC(12);"FORWARD COURSE ";:X=A12/RD:GOSUB 1090:PRINT V$
4160 PRINT SPC(12);"BACK COURSE ";:X=A21/RD:GOSUB 1090:PRINT V$
4170 GOSUB 9000:GOTO 2000
4180 '
5000 CLS:PRINT SPC(20);"FIND CPA":PRINT:PRINT
5010 PRINT SPC(10);:PRINT "1st LATITUDE DD.MMSS (-S) ";
5020 INPUT V$:GOSUB 1150:P1=X*RD
5030 PRINT SPC(10);:PRINT "1st LONGITUDE DDD.MMSS (-E) ";
5040 INPUT V$:GOSUB 1150:L1=X*RD
5050 PRINT SPC(10);:PRINT "INITIAL COURSE DDD.MMSS ";
5060 INPUT V$:GOSUB 1150:A1=X*RD
5070 PRINT SPC(10);:INPUT "SPEED (knots) ?  ",S1
5080 PRINT SPC(10);:PRINT "2nd LATITUDE DD.MMSS (-S) ";
5090 INPUT V$:GOSUB 1150:P2=X*RD
5100 PRINT SPC(10);:PRINT "2nd LONGITUDE DDD.MMSS (-E) ";
5110 INPUT V$:GOSUB 1150:L2=X*RD
5120 PRINT SPC(10);:PRINT "INITIAL COURSE DDD.MMSS ";
5130 INPUT V$:GOSUB 1150:A2=X*RD
5140 PRINT SPC(10);:INPUT "SPEED (knots) ?  ",S2
5150 B1=S1/AE:B2=S2/AE
```

```
5160  '
5170  P=P1:L=-L1:GOSUB 1750:'  CHNG SIGN OF L1 GIVES RIGHT-HANDED COORDS
5180  FOR I!=1 TO 3:X1(I!)=POSI(I!):NEXT
5190  X=0:Y=0:Z=1:A=A1:GOSUB 1700:A=P:GOSUB 1710:A=-L:GOSUB 1720
5200  C1(1)=X:C1(2)=Y:C1(3)=Z
5210  P=P2:L=-L2:GOSUB 1750:'  CHNG SIGN OF L2 GIVES RIGHT-HANDED COORDS
5220  FOR I!=1 TO 3:X2(I!)=POSI(I!):NEXT
5230  X=0:Y=0:Z=1:A=A2:GOSUB 1700:A=P:GOSUB 1710:A=-L:GOSUB 1720
5240  C2(1)=X:C2(2)=Y:C2(3)=Z
5250  '
5260  X1C2=X1(1)*C2(1)+X1(2)*C2(2)+X1(3)*C2(3)
5270  C1X2=C1(1)*X2(1)+C1(2)*X2(2)+C1(3)*X2(3)
5280  X1X2=X1(1)*X2(1)+X1(2)*X2(2)+X1(3)*X2(3)
5290  C1C2=C1(1)*C2(1)+C1(2)*C2(2)+C1(3)*C2(3)
5300  BA=-B1*X1C2 - B2*C1X2
5310  BB= B1*C1X2 + B2*X1C2
5320  BC=-B1*X1X2 + B2*C1C2
5330  BD= B1*C1C2 - B2*X1X2
5340  '
5350  T=1:IT!=0:'  ITERATE WITH NEWTON-RAPHSON
5360  B1T=B1*T:B2T=B2*T:S1=SIN(B1T):C1=COS(B1T):S2=SIN(B2T):C2=COS(B2T)
5370  S1S2=S1*S2:C1C2=C1*C2:S1C2=S1*C2:C1S2=C1*S2
5380  F=BA*S1S2+BB*C1C2+BC*S1C2+BD*C1S2
5390  FP=-(BC*B2+BD*B1)*S1S2+(BD*B2+BC*B1)*C1C2+(BA*B2-BB*B1)*S1C2-
      (BB*B2-BA*B1)*C1S2
5400  IT!=IT!+1:CORR=F/FP:T=T-CORR:IF ABS(CORR)<.00001 THEN 5440
5410  IF IT!>50 THEN PRINT "NO CONVERGENCE":STOP
5420  GOTO 5360
5430  '
5440  B1T=B1*T:B2T=B2*T:CB1T=COS(B1T):SB1T=SIN(B1T):CB2T=COS(B2T):SB2T=SIN(B2T)
5450  FOR I!=1 TO 3:POS1(I!)=X1(I!)*CB1T+C1(I!)*SB1T
5460  POS2(I!)=X2(I!)*CB2T+C2(I!)*SB2T:NEXT I!
5470  P1=FNASN(POS1(3)):L1=-FNATN2(POS1(2),POS1(1))
5480  P2=FNASN(POS2(3)):L2=-FNATN2(POS2(2),POS2(1))
5490  GOSUB 1540
5500  X=T:GOSUB 1010:PRINT:PRINT SPC(10);"TIME TO CPA = ";V$
5510  MO=100#:PRINT SPC(10);"DISTANCE AT CPA = ";FNR(60#*DD/RD);" n.mi.
5520  PRINT SPC(10);"BEARING AT CPA = ";:X=A12/RD:GOSUB 1090:PRINT V$
5530  PRINT SPC(10);"NO. ITERATIONS = ";IT!
5540  TCPA=T:GOSUB 9000
5550  '
5560  CLS:PRINT SPC(22);"FIND CPA":PRINT
5570  PRINT " TIME LAT 1 LONG 1 DISTANCE BEARING(1->2)"
5580  DT=TCPA/6#:T=0:FOR T!=1 TO 13:B1T=B1*T:B2T=B2*T
5590  FOR I!=1 TO 3:POS1(I!)=X1(I!)*COS(B1T)+C1(I!)*SIN(B1T)
5600  POS2(I!)=X2(I!)*COS(B2T)+C2(I!)*SIN(B2T):NEXT I!
5610  P1=FNASN(POS1(3)):L1=-FNATN2(POS1(2),POS1(1))
5620  P2=FNASN(POS2(3)):L2=-FNATN2(POS2(2),POS2(1))
5630  GOSUB 1540
5640  X=T:GOSUB 1010:PRINT V$;
5650  LOCATE CSRLIN,12:X=P1/RD:GOSUB 1090:PRINT V$;
5660  LOCATE CSRLIN,23:X=L1/RD:GOSUB 1090:PRINT V$;
5670  LOCATE CSRLIN,37:PRINT FNR(60#*DD/RD);
```

```
5680 LOCATE CSRLIN,52:X=A12/RD:GOSUB 1090:PRINT V$
5690 T=T+DT:NEXT T!
5700 GOSUB 9000:GOTO 2000
5710 '
6000 CLS:PRINT SPC(10);"SPEED NEEDED TO INTERCEPT (1->2)":PRINT:PRINT
6010 CLN%=0:GOSUB 6510:'  GET INPUT
6020 CLN%=10:LOCATE CLN%,1:FOR I!=1 TO 11:PRINT SPC(79):NEXT:' CLEAR SCREEN
6030 LOCATE CLN%,11:PRINT "TIME TO INTERCEPT (SECONDS) ";:INPUT TMI
6040 TMI=TMI/3600#:TM=TMI:P1=P1S:L1=L1S:P2=P2S:L2=L2S:A2=A2S
6050 '
6060 ' COMPUTE SPEED
6070 P=P1S:L=-L1S:GOSUB 1750:'  CHNG SIGN OF L1 GIVES RIGHT-HANDED COORDS
6080 FOR I!=1 TO 3:X1(I!)=POSI(I!):NEXT
6090 P=P2S:L=-L2S:GOSUB 1750:'  CHNG SIGN OF L2 GIVES RIGHT-HANDED COORDS
6100 FOR I!=1 TO 3:X2(I!)=POSI(I!):NEXT
6110 X=0:Y=0:Z=1:A=A2:GOSUB 1700:A=P:GOSUB 1710:A=-L:GOSUB 1720
6120 C2(1)=X:C2(2)=Y:C2(3)=Z
6130 B2T=B2*TM:CB2T=COS(B2T):SB2T=SIN(B2T):CS=0#
6140 FOR I!=1 TO 3:POS2(I!)=X2(I!)*CB2T+C2(I!)*SB2T
6150 CS=CS+X1(I!)*POS2(I!):NEXT I!
6160 S=FNACS(CS):SPD=S*AE/(TM-EPS*(TM=0))
6170 '
6180 ' GET INVERSE SOLN
6190 P2=FNASN(POS2(3)):L2=-FNATN2(POS2(2),POS2(1)):GOSUB 1540
6200 '
6210 MO=10:LOCATE CLN%+2,11:PRINT "SPEED REQUIRED = ";FNR(SPD);" knots"
6220 GOSUB 6810:'  PRINT OUT BEARING, RANGE, LAT & LONG
6230 LOCATE CLN%+8,15:PRINT "1) CHANGE TIME OF INTERCEPT"
6240 LOCATE CLN%+9,15:PRINT "2) NEW PROBLEM"
6250 LOCATE CLN%+10,15:PRINT "3) MASTER MENU"
6260 GOSUB 9010:C=VAL(C$):ON C GOTO 6020,6000,2000
6270 GOTO 6260
6280 '
6500 ' INPUT ROUTINE
6510 LOCATE CLN%+3,11:PRINT "1st LATITUDE DD.MMSS (-S) ";
6520 INPUT V$:GOSUB 1150:P1S=X*RD
6530 LOCATE CLN%+4,11:PRINT "1st LONGITUDE DDD.MMSS (-E) ";
6540 INPUT V$:GOSUB 1150:L1S=X*RD
6550 LOCATE CLN%+5,11:PRINT "2nd LATITUDE DD.MMSS (-S) ";
6560 INPUT V$:GOSUB 1150:P2S=X*RD
6570 LOCATE CLN%+6,11:PRINT "2nd LONGITUDE DDD.MMSS (-E) ";
6580 INPUT V$:GOSUB 1150:L2S=X*RD
6590 LOCATE CLN%+7,11:PRINT "2nd COURSE DDD.MMSS ";
6600 INPUT V$:GOSUB 1150:A2S=X*RD
6610 LOCATE CLN%+8,11:INPUT "2nd SPEED (knots) ?  ",S2
6620 B2=S2/AE:RETURN
6630 '
6800 ' OUTPUT BEARING, RANGE, LAT & LONG
6810 LOCATE CLN%+3,11:PRINT "BEARING TO INTERCEPT = ";
6820 X=A12/RD:GOSUB 1090:PRINT V$
6830 LOCATE CLN%+4,11:PRINT "RANGE TO INTERCEPT = ";FNR(60#*DD/RD);" n.mi."
6840 LOCATE CLN%+5,11:PRINT "LATITUDE OF INTERCEPT = ";
6850 X=P2/RD:GOSUB 1090:PRINT V$
```

```
6860 LOCATE CLN%+6,11:PRINT "LONGITUDE OF INTERCEPT = ";
6870 X=L2/RD:GOSUB 1090:PRINT V$
6880 RETURN
6890 '
7000 CLS:PRINT SPC(10);"TIME NEEDED TO INTERCEPT (1->2)":PRINT:PRINT
7010 CLN%=0:GOSUB 6510:'  GET INPUT
7020 '
7030 ' FIND Vmin AND Tmin vel.
7040 ' Compute arc distance S
7050 P=P1S:L=-L1S:GOSUB 1750:'  CHNG SIGN OF L1 GIVES RIGHT-HANDED COORDS
7060 FOR I!=1 TO 3:X1(I!)=POSI(I!):NEXT
7070 P=P2S:L=-L2S:GOSUB 1750:'  CHNG SIGN OF L2 GIVES RIGHT-HANDED COORDS
7080 FOR I!=1 TO 3:X2(I!)=POSI(I!):NEXT
7090 X=0:Y=0:Z=1:A=A2S:GOSUB 1700:A=P:GOSUB 1710:A=-L:GOSUB 1720
7100 C2(1)=X:C2(2)=Y:C2(3)=Z
7110 '
7120 ' INITIALIZE
7130 X1X2=X1(1)*X2(1)+X1(2)*X2(2)+X1(3)*X2(3)
7140 X1C2=X1(1)*C2(1)+X1(2)*C2(2)+X1(3)*C2(3)
7150 '
7160 ' BEGIN ITERATION
7170 S=FNACS(X1X2):T=S*AE/S2:IT!=1
7180 B2T=B2*T:CB2T=COS(B2T):SB2T=SIN(B2T)
7190 CS=X1X2*CB2T+X1C2*SB2T:S=FNACS(CS):SS=SIN(S)
7200 DSDT=(X1X2*SB2T-X1C2*CB2T)*B2/SS
7210 F=T*DSDT-S
7220 FP=T*(B2*B2*(X1X2*CB2T+X1C2*SB2T)-CS*DSDT*DSDT)/SS
7230 IT!=IT!+1:CORR=F/FP:T=T-CORR:IF ABS(CORR)<.00001 THEN 7270
7240 IF IT!>50 THEN PRINT "NO CONVERGENCE":STOP
7250 GOTO 7180
7260 '
7270 TMS=T:VMIN=S*AE/T
7280 LOCATE CLN%+10,11:MO=10
7290 PRINT "MINIMUM SPEED REQUIRED TO INTERCEPT = ";FNR(VMIN);" knots"
7300 LOCATE CLN%+11,11:X=TMS:GOSUB 1010
7310 PRINT "TIME REQ'D TO INTERCEPT AT MIN SPEED = ";V$
7320 '
7330 CLN%=13:LOCATE CLN%,1:FOR I!=1 TO 11:PRINT SPC(79):NEXT:' CLEAR SCREEN
7340 LOCATE CLN%,11:PRINT "INTERCEPTOR SPEED (knots) ";:INPUT SPD
7350 IF SPD>=VMIN THEN 7390
7360 LOCATE CLN%+2,11:PRINT "SPEED TOO LOW, CANNOT INTERCEPT"
7370 GOSUB 9000:GOTO 7330
7380 '
7390 B1=SPD/AE:T=TMS/5:  T=.1 :IT!=1
7400 B2T=B2*T:CB2T=COS(B2T):SB2T=SIN(B2T)
7410 CS=X1X2*CB2T+X1C2*SB2T:S=FNACS(CS):SS=SIN(S)
7420 DSDT=(X1X2*SB2T-X1C2*CB2T)*B2/SS
7430 F=S/T-B1:FP=(DSDT-S/T)/T
7440 IT!=IT!+1:CORR=F/FP:T=T-CORR:IF ABS(CORR)<.00001 THEN 7500
7450 IF IT!>50 THEN PRINT "NO CONVERGENCE":STOP
7460 IF ABS(CORR)<1000000000# THEN 7400
7470 LOCATE CLN%+2,11:PRINT "SPEED TOO HIGH, NO CONVERGENCE"
7480 LOCATE CLN%+4:GOSUB 9000:GOTO 7330
```

```
7490  '
7500  X=T:GOSUB 1010
7510  LOCATE CLN%+2,11:PRINT "TIME REQUIRED = ";V$
7520  '
7530  ' GET INVERSE SOLN
7540  FOR I!=1 TO 3:POS2(I!)=X2(I!)*CB2T+C2(I!)*SB2T:NEXT I!
7550  P1=P1S:L1=L1S:P2=FNASN(POS2(3)):L2=-FNATN2(POS2(2),POS2(1)):GOSUB 1540
7560  '
7570  GOSUB 6810:'  PRINT OUT BEARING, RANGE, LAT & LONG
7580  LOCATE CLN%+8,15:PRINT "1) CHANGE INTERCEPTOR SPEED"
7590  LOCATE CLN%+9,15:PRINT "2) NEW PROBLEM"
7600  LOCATE CLN%+10,15:PRINT "3) MASTER MENU"
7610  GOSUB 9010:C=VAL(C$):ON C GOTO 7330,7000,2000
7620  GOTO 7610
7630  '
8000  CLS:END
8010  '
9000  PRINT:PRINT SPC(10);"PRESS ANY KEY TO CONTINUE
9010  FOR I!=1 TO 9:C$=INKEY$:NEXT
9020  C$=INKEY$:IF C$="" THEN 9020
9030  RETURN
```

DISTRIBUTION LIST

	NO. OF COPIES
Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5000	2
Office of Research Administration Code 012A Naval Postgraduate School Monterey, CA 93943-5000	1
Library, Code 55 Naval Postgraduate School Monterey, CA 93943-5000	1
Office of Naval Research Fleet Activity Support Division Code ONR-230 800 North Quincy Street Arlington, VA 22217	2
Chief of Naval Operations Attn: Code OP-953C2 Washington, D.C. 20350	1
Navy Tactical Support Activiity Attn: C. Earp, C. Reberkenny P.O. Box 1042 Silver Springs, MD 20910	2
COMPATWINGSPAC Attn: Code 51 and Code 532 Naval Air Station Moffett Field, CA 94035	2
COMPATWINGSLANT Attn: Code N7 Naval Air Station Brunswick, ME 04011	2

Commander 2
Surface Warfare Development Group
Naval Amphibious Base, Little Creek
Norfolk, VA

COMSUBDEVRON TWELVE 2
Attn: Code 221
Submarine Base New London
Groton, CT 06349

Mr. James Grant 1
Code 18, Fleet Readiness Office
Naval Oceanographic Systems Center
San Diego, CA 92152

Dr. Martin Leonardo 1
Code 2031
Naval Air Development Center
Warminster, PA 18974-5000

Prof. R.H. Shudde, Code 55Su 50
Naval Postgraduate School
Monterey, CA 93943-5000