FALCONFIX:
A MULTI-MODAL APPROACH
TO FIX COMPUTATION

WILLIAM T. HODSON III, LT COLONEL, USAF
JOSEPH C. H. SMITH, MAJOR, USAF

DEPARTMENT OF MATHEMATICAL SCIENCES
USAF ACADEMY, COLORADO 80840

JUNE 1979
FINAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DEAN OF THE FACULTY
UNITED STATES AIR FORCE ACADEMY

79 08 9 013
This research report is presented as a competent treatment of the subject, worthy of publication. The United States Air Forces Academy vouches for the quality of the research, without necessarily endorsing the opinions and conclusions of the author.

This report has been cleared for open publication and/or public release by the appropriate Office of Information in accordance with AR 190-17 and DODD 5230.9. There is no objection to unlimited distribution of this report to the public at large, or by DCSO to the National Technical Information Service.

This research report has been reviewed and is approved for publication.

[Signature]

John D. Muetzy, Colonel, USAF
Vice Dean of the Faculty
FALCONFIX: A Multi-Modal Approach to Fix Computation

**Title (and Subtitle)**
FALCONFIX: A Multi-Modal Approach to Fix Computation

**Author**
William T. Hedin, III
Joseph C.H. Smith

**Performing Organization Name and Address**
Department of Mathematical Sciences
United States Air Force Academy, CO 80840

**Report Date**
June 1979

**Distribution Statement (of the report)**
Approved for Public Release; Distribution Unlimited

**Abstract**
FALCONFIX is a new algorithm for computing the location of an emitter given the bearings from a number of direction-finding stations located worldwide. This approach assumes that each reported bearing may be considered as a sample from a composite normal/uniform distribution, thereby explicitly acknowledging that the bearing may be either "true" or "wild." These distributions are used to create a likelihood function defined over the unit sphere. Non-linear programming techniques are then employed to identify all relative maxima (modes) of
20. Abstract (continued)

this function. The modes (which coincide with concentrations of bearings) are then ordered by value and reported as estimates of the emitter location.
ABSTRACT

FALCONFIX is a new algorithm for computing the location of an emitter given the bearings from a number of direction-finding stations located worldwide. This approach assumes that each reported bearing may be considered as a sample from a composite normal/uniform distribution, thereby explicitly acknowledging that the bearing may be either "true" or "wild." These distributions are used to create a likelihood function defined over the unit sphere. Non-linear programming techniques are then employed to identify all relative maxima (modes) of this function. The modes (which coincide with concentrations of bearings) are then ordered by value and reported as estimates of the emitter location.
The authors gratefully acknowledge the assistance rendered by LCDR Eric S. Hodson and CWO Anthony Untied of the Naval Security Group Command in the development of the FALCONFIX method. LCDR Hodson provided many valuable insights into the shortcomings of existing fix computation programs and consequently what areas should receive attention in the development of a new approach. CWO Untied translated the FORTRAN code into NEILAC for use on NSGC computers, thereby enabling the code to be tested against other programs using real operational data.

This work was supported in part by funding received from the Naval Security Group Command.
**TABLE OF CONTENTS**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>i</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>ii</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>iii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>iv</td>
</tr>
<tr>
<td>Chapter 1 - Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Background</td>
<td>1</td>
</tr>
<tr>
<td>Overview of the FALCONFIX Method</td>
<td>3</td>
</tr>
<tr>
<td>Chapter 2 - Mathematical Description</td>
<td>4</td>
</tr>
<tr>
<td>Distribution of Bearing Error</td>
<td>4</td>
</tr>
<tr>
<td>The Likelihood Function</td>
<td>8</td>
</tr>
<tr>
<td>Vector Representation</td>
<td>13</td>
</tr>
<tr>
<td>Modes of $L(\phi, \theta)$</td>
<td>18</td>
</tr>
<tr>
<td>Standard Deviation of Bearing Error</td>
<td>23</td>
</tr>
<tr>
<td>The REFIX Procedures</td>
<td>26</td>
</tr>
<tr>
<td>The Confidence Region</td>
<td>30</td>
</tr>
<tr>
<td>Appendix A - Computer Program Flow Chart</td>
<td></td>
</tr>
<tr>
<td>Appendix B - Computer Program Listing</td>
<td></td>
</tr>
<tr>
<td>Appendix C - Sample Output</td>
<td></td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Typical Composite Normal-Uniform PDF</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Transition Point Curves</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>$L(\phi, \theta)$; Two Bearing Cross</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>$L(\phi, \theta)$; Three Bearing Case #1</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>$L(\phi, \theta)$; Three Bearing Case #2</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>$L(\phi, \theta)$; Four Bearing Case</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>$L(\phi, \theta)$; Bimodal Case</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>Position Vector</td>
<td>13</td>
</tr>
<tr>
<td>9</td>
<td>Bearing Vector</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>Dihedral Angle</td>
<td>16</td>
</tr>
<tr>
<td>11</td>
<td>Example of Mode Determination</td>
<td>21</td>
</tr>
<tr>
<td>12</td>
<td>Graph of $h(d)$</td>
<td>24</td>
</tr>
<tr>
<td>13</td>
<td>Inclusion Region</td>
<td>25</td>
</tr>
<tr>
<td>14</td>
<td>Coordinate System for the REFIX/2 Procedure</td>
<td>27</td>
</tr>
</tbody>
</table>
Chapter 1
INTRODUCTION

Background

A direction-finding system consists of both a net control station and a set of direction-finding sites at widely dispersed geographic locations. When instructed by the net control station, each site tunes its receiver to a specified frequency and, using extremely directional antennas, obtains a bearing on the signal it receives. These bearings are then reported to the net control station where they are processed to estimate the true location of the emitter (obtain a "fix").

Should all the bearings intersect at a single point, the estimate is, of course, obvious. Such a situation rarely occurs in practice since error is induced in the direction of the received signal by both the equipment at each site and by the propagation medium between the sites and the emitter. On the basis of empirical evidence, these errors appear to be approximately normally distributed. Thus, the reported bearings may be thought of as statistical samples taken from normal distributions with unknown true means. Various statistical techniques may be used to estimate these means, thereby providing an estimate of the true location of the emitter.

A particularly vexing aspect of the problem, however, is that of so-called "wild bearings." When a site tunes up on the specified frequency, it might not hear the same signal that other sites are receiving. This can happen both as a result of propagation conditions and of slight differences in the times the various sites listen. In the
latter case a site might report a bearing on the wrong emitter. From a statistical point of view, a wild bearing cannot properly be considered to be a sample from a normal distribution with mean bearing on the true emitter location. Unfortunately, it is impossible to distinguish between bearings taken on the correct emitter from bearings taken on another emitter prior to processing. All fix computation algorithms must deal with this problem in one way or another.

A common method is to consider the problem in two phases. In the first phase, the "most likely" set of true bearings is identified and in the second phase only these bearings are used in making the estimate. The difficulty with this approach is that in deciding which bearings are wild, it is necessary to establish a preliminary estimate of the true location. Then bearings which are not close to this position are considered wild and are rejected. If the preliminary estimate is not close to the true location, it is quite likely that good bearings will be thought to be wild and wild bearings good. It is also possible that enough bearings will be rejected so as to create a "no-fix" situation, i.e., not enough bearings remaining to reliably compute a fix.
Overview of the FALCONFIX Method

The method developed in this paper takes a considerably different approach to the problem—no bearing is ever rejected. Instead, the bearing reported from each site is considered to be a sample from a normal distribution with probability $p$ (a good bearing) and from a uniform distribution with probability $1-p$ (a wild bearing), where $p$ is determined empirically by observing the fraction of good bearings over a period of time. Then a likelihood function, the joint probability distribution of bearing errors assuming this composite normal/uniform distribution, is defined over a spherical earth model. The estimate of true emitter location is then taken as the location which maximizes the likelihood. Because of the nature of each of the bearing error distributions, this likelihood function may have numerous relative maxima (modes), corresponding to points on the earth where intersections of the reported bearings are concentrated.

Non-linear programming and regression techniques are then used to find the location and value of each mode of the likelihood function. The algorithm generates the three most likely modes and orders them according to the probability that they are associated with the true emitter location. The net control station operator may then determine which position to report, on the basis of both the algorithm's ordering as well as other available information.

Finally, the algorithm generates an elliptical confidence region with a specified probability of containing the true position of the emitter.
Chapter 2

MATHEMATICAL DESCRIPTION

Distribution of Bearing Error

The bearing which is reported from a direction-finding site may be taken on either the true target or, inadvertently, on a false target. If it is taken on a true target, we assume that the angular error ($\epsilon$) between the reported bearing and the true bearing is a random variable with a normal distribution having mean zero and variance $\sigma^2$. On the other hand, if the bearing is taken on a false target, no information is conveyed by the reported bearing concerning the bearing to the true target. Therefore, we assume that the angular error is a random variable with a uniform distribution. Further, we assume that the bearing is taken on the true target with probability $p$ and on the false target with probability $(1-p)$; the value for $p$ may be obtained from empirical evidence. For small values of $\epsilon$, therefore, we assume the distribution to be normal and for larger values we assume it uniform. The transition points are designated $\pm c$, where $c$ is a constant which depends upon both $p$ and $\sigma$. Lastly, it is assumed that the probability density function of the bearing error is a continuous function of $\epsilon$. The resulting distribution, which we shall call "composite normal-uniform" is represented by

$$h(\epsilon) = \begin{cases} \frac{1}{K\sqrt{2\pi}} \exp(-\epsilon^2/2\sigma^2), & -\sigma \leq \epsilon \leq \sigma \\ \frac{1}{K\sqrt{2\pi}} \exp(-c^2/2), & -\pi < \epsilon < -\sigma \text{ or } \sigma < \epsilon < \pi \end{cases}$$
where $\epsilon$ and $\sigma$ are in radians and the normalizing constant $K$ is given by

$$K = [2\Phi(c)-1] + 2[\pi-\sigma] \left[ \frac{\exp(-c^2/2)}{\sqrt{2\pi\sigma}} \right]$$

where $\Phi$ is the standard normal cumulative distribution function.

The graph of a typical composite normal-uniform is depicted in Figure 1 below. The fact that the ordinate is constant for values of $\epsilon$ greater than $\pm \sigma$ or less than $-\infty$ is extremely significant to the method, as shall become apparent in later sections.

![Figure 1. Typical Composite Normal-Uniform PDF](image-url)
To determine the transition point, we note that

\[ p = \frac{1}{K} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi \sigma}} \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right) d\epsilon \]

\[ = \frac{1}{K} [2\Phi(c)-1], \text{ or} \]

\[ p = \frac{2\Phi(c)-1}{[2\Phi(c)-1] + 2[\pi-\alpha]} \left[ \frac{\exp\left(-c^2/2\right)}{\sqrt{2\pi \sigma}} \right] \]

In Figure 2 below, this relationship between \( p \) and \( \sigma \) is plotted for various values of \( c \). Then, given values of \( p \) and \( \sigma \), \( c \) can be obtained by interpolating between the curves.

![Figure 2. Transition Point Curves](image)
One final approximation for the density function is required to make the procedure more efficient computationally. The $\epsilon^2$ term is difficult to work with using vector methods, so the Maclaurin approximation for $\cos \epsilon$ is used to derive the relationship

$$\epsilon^2 \approx 2(1 - \cos \epsilon).$$

This approximation is correct to within 0.4% for values of $\epsilon$ up to 0.2 radians, a value greater than will occur in practice.

Thus, $h(\epsilon)$ is approximated by

$$f(\epsilon) = \left\{ \begin{array}{ll}
\frac{1}{k\sqrt{2\pi}} \exp \left[ \frac{-1}{2} \left( 1 - \cos \epsilon \right) \right], & -\infty \leq \epsilon \leq -\epsilon_0 \\
\frac{1}{k\sqrt{2\pi}} \exp \left[ \frac{-1}{2} \left( 1 - \cos (\epsilon) \right) \right], & -\pi < \epsilon < -\epsilon_0 \text{ or } \epsilon_0 < \epsilon < \pi
\end{array} \right.$$
The Likelihood Function

The likelihood function for the emitter location \((\phi, \theta)\) is defined to be:

\[
F(\phi, \theta) = G(e_1, e_2, \ldots, e_n) = f_1(e_1)f_2(e_2)\cdots f_n(e_n)
\]

where \(e_i\) is the angular error between the reported bearing from station \(i\) and the true bearing from station \(i\) to \((\phi, \theta)\). Thus \(e_i\) is a random variable with density function

\[
f_1(e_1) = \begin{cases} 
\frac{1}{K_1\sqrt{2\pi}\sigma_1} \exp \left( -\frac{1}{2} \left( \frac{e_1^2}{\sigma_1^2} \right) \right), & -c_1\sigma_1 \leq e_1 < c_1\sigma_1 \\
\frac{1}{K_1\sqrt{2\pi}\sigma_1} \exp \left( -\frac{1}{2} \left( 1-\cos(c_1\sigma_1) \right) \right), & -\pi \leq e_1 < -c_1\sigma_1 \text{ or } c_1\sigma_1 \leq e_1 < \pi
\end{cases}
\]

Therefore,

\[
\sum'(\phi, \theta) = \frac{1}{(2\pi)^{n/2}} \prod_{i=1}^{n} \frac{1}{K_i\sigma_i} \prod_{i \in W} \exp \left( -\frac{1}{2} \left( 1-\cos(c_i\sigma_i) \right) \right) \prod_{i \notin W} \exp \left( -\frac{1}{2} \left( \frac{e_i^2}{\sigma_i^2} \right) \right)
\]

where \(W = \{i : |c_i| < c_i\sigma_i\}\)

As will be made clear subsequently, our approach to estimating emitter location will be to find the relative maxima—which we call "modes"—of this function.

Since \(\frac{1}{(2\pi)^{n/2}} \prod_{i=1}^{n} K_i\sigma_i\) is not a function of the \(e_i\), \(F(\phi, \theta)\) is maximized whenever
\[ F^*(\phi, \theta) = \prod_{i \in W} \exp \left( \frac{1}{\sigma_i} (1 - \cos \epsilon_i) \right) \prod_{i \notin W} \exp \left( \frac{1}{\sigma_i} (1 - \cos (c_i \sigma_i)) \right) \]

is a maximum. \( F^*(\phi, \theta) \), in turn, is maximized whenever its logarithm

\[ L(\phi, \theta) = \sum_{i \in W} \frac{1}{\sigma_i} (1 - \cos \epsilon_i) + \sum_{i \notin W} \frac{1}{\sigma_i} (1 - \cos (c_i \sigma_i)) \]

is, and this is the function to which we shall direct our attention.

Before proceeding with the development of the method, we shall pause to consider exactly what this function \( L(\phi, \theta) \) looks like. In Figures 3 through 7 are depicted five examples of a set of bearings and the resulting function \( L \).

In the figure below the case in which two lines of bearing intersect at a single point is illustrated. Notice that the function has a single mode at the point of intersection. (As we shall see later, the FALCONFIX algorithm does not consider two bearing crosses as potential position estimates.)

Figure 3. \( L(\phi, \theta) \); Two Bearing Cross
In Figure 4 below, the function $L(\phi, \theta)$ in which three bearings intersect at a point is illustrated. Here the mode is somewhat more sharply peaked than in the previous case.

![Figure 4. $L(\phi, \theta)$; Three Bearing Case #1](image)

In Figure 5, the three bearings do not meet at a single point, and, in fact, the "cocked hat" is so large compared to the standard deviation of the bearing errors, that the function does not have a single mode in the center. Instead, there are three individual two bearing crosses at each of the intersections.
In Figure 6 below, a four bearing case is depicted in which the bearings do not intersect at a single point. Notice that this figure is considerably less regular than the previous figures, and is more typical of what one finds in real cases.
In Figure 7 the bimodal case is illustrated. In this five bearing example it is not likely that all bearings were taken on the same emitter. Rather, it is much more plausible that either Bearings 1, 2 and 3 are good bearings (with 4 and 5 wild) or Bearings 1, 4 and 5 are good bearings (with 2 and 3 wild). The two modes in the figure correspond to these two possibilities.

Figure 7. $L(\phi, \theta)$; Bimodal Case

Before going on to the methodology of identifying these modes, it will first be necessary to express each of the $e_i$ in the equation for $L$ explicitly as functions of $\phi$ and $\theta$, the latitude and longitude of the position estimate. To accomplish this we will rely heavily on the use of vectors as described in the next section.
Vector Representation

A position vector $P$ from the center of the earth to a point $(\phi, \theta)$ on the surface may be represented as

$$P = (\cos \theta \cos \phi) \mathbf{i} + (\sin \theta \cos \phi) \mathbf{j} + (\sin \phi) \mathbf{k}$$

This vector $P$ is depicted in Figure 8 below in a right-hand coordinate system in which $\mathbf{i}$ points to the prime meridian on the equator and $\mathbf{k}$ points to the north pole. Thus, east longitude is represented by positive values of $\theta$, west longitude by negative values, north latitude by positive values of $\phi$ and south latitude by negative values.

Figure 8. Position Vector
If \((\phi, \theta)\) is the location of the target and \((\phi_i, \theta_i)\) is the location of the \(i^{th}\) station, then the corresponding position vectors are:

\[
T = (\cos \theta \cos \phi)\vec{i} + (\sin \theta \cos \phi)\vec{j} + (\sin \phi)\vec{k}
\]

\[
S_i = (\cos \theta_i \cos \phi_i)\vec{i} + (\sin \theta_i \cos \phi_i)\vec{j} + (\sin \phi_i)\vec{k}
\]

Let \(\beta_i\) be the bearing reported by the \(i^{th}\) station. Then in terms of the local coordinate system \((\vec{i}^*, \vec{j}^*, \vec{k}^*)\) in which \(\vec{i}^*\) points east, \(\vec{j}^*\) points north and \(\vec{k}^*\) points to the local zenith, a vector pointing in the direction of \(\beta_i\) may be represented as

\[
B_i^* = (\sin \beta_i)\vec{i}^* + (\cos \beta_i)\vec{j}^*
\]

which is shown in the figure below.

![Figure 9. Bearing Vector](image)

To be useful, however, it is necessary to express each bearing vector in the same coordinate system as the \(T\) and \(S_i\) vectors. This is accomplished by rotating the local frame with transformation \(R_i\) so that it coincides with the geocentric frame shown in Figure 8.
Specifically,

\[ B_i = R_i B_i^* = A^i + B_j + Ck \]

where

\[ A = (\sin \phi_i \cos \theta_i \cos \beta_i + \sin \theta_i \sin \beta_i) \]

\[ B = \cos \theta_i \sin \beta_i - \sin \phi_i \sin \theta_i \cos \beta_i \]

\[ C = \cos \phi_i \cos \beta_i \]

Now, \( T \), the \( S_i \) and the \( B_i \) are all unit vectors expressed in the same geometric frame.

Then, the vector \( \left( \frac{S_i \times T}{|S_i \times T|} \right) \) is a unit vector normal to the "station-target plane," the plane containing \( S_i \) and \( T \). Similarly, \( \left( \frac{S_i \times B_i}{|S_i \times B_i|} \right) \) is a unit vector normal to the "station-bearing plane," the plane containing \( S_i \) and \( B_i \).

Clearly, the angle between these two normal vectors is equal to the dihedral angle between the two planes, which in turn is equal to the angle \( \epsilon_i \) between the reported bearing and the true bearing from the \( i^{th} \) station to the target. This relationship is shown in Figure 10.

From the familiar dot product relationship we have

\[ \cos \epsilon_i = \left( \frac{S_i \times T}{|S_i \times T|} \right) \cdot \left( \frac{S_i \times B_i}{|S_i \times B_i|} \right) \]

Since \( S_i \) and \( B_i \) are orthonormal, it is true that \( |S_i \times B_i| = 1 \), which means that

\[ \cos \epsilon_i = \frac{(S_i \times T) \cdot (S_i \times B)}{|S_i \times T|} \]
Employing a little-used identity from vector calculus on the numerator of this expression yields

\[(S_i \times T) \cdot (S_i \times B_i) = (B_i \cdot T)(S_i \cdot S_i) - (S_i \cdot T)(S_i \cdot B_i)\]

But \(S_i \cdot S_i = 1\) since \(S_i\) is a unit vector and \(S_i \cdot B_i = 0\) since \(S_i\) and \(B_i\) are orthogonal.

Thus \((S_i \times T) \cdot (S_i \times B_i) = B_i \cdot T\), which means that

\[\cos \epsilon_i = \frac{B_i \cdot T}{|S_i \times T|}\]

Now \(|S_i \times T|\) is equal to the sine of the angle between \(S_i\) and \(T\). Since \(S_i \cdot T\) equals the cosine of the same angle, we can represent \(|S_i \times T|\) as
\[|S_i \cdot T| = \sqrt{1-(S_i \cdot T)^2}\]

which is somewhat easier to deal with computationally. The relationship

\[\cos c_i = \frac{B_i \cdot T}{\sqrt{1-(S_i \cdot T)^2}}\]

will be used in subsequent calculations. Making this substitution, the logarithm of the likelihood function becomes:

\[L(\phi, \theta) = \sum_{i \in W} \frac{-1}{2} \left(1 - \frac{B_i \cdot T}{\sqrt{1-(S_i \cdot T)^2}}\right) + \sum_{i \in W} \frac{-1}{2} (1-\cos(c_i \cdot \theta))\]

In this equation the \( c_i, \theta_i, B_i \) and \( S_i \) are all constants and \( T \) is a function of \( \phi \) and \( \theta \). We now have \( L(\phi, \theta) \) in a form where it is possible to find the modes by moving \( T \) in such a way as to progressively increase the value of \( L \). In the next section we will illustrate exactly how this is done.
Modes of $L(\phi, \theta)$

$L(\phi, \theta)$ can be written in the form:

$$L(\phi, \theta) = \sum_{i \in W} \frac{B_i \cdot T}{\sigma_i^2 \sqrt{1 - (S_i \cdot T)^2}} + K$$

where $K = a$ constant

$$B_i = b_{i1} \bar{I} + b_{i2} \bar{J} + b_{i3} \bar{K}$$

$$S_i = s_{i1} \bar{I} + s_{i2} \bar{J} + s_{i3} \bar{K}$$

$$T = t_{i1} \bar{I} + t_{i2} \bar{J} + t_{i3} \bar{K}$$

$$= \cos \theta \cos \phi \bar{I} + \sin \theta \cos \phi \bar{J} + \sin \phi \bar{K}$$

and $|B_i| = |S_i| = |T| = 1$

A necessary condition for the point $(\phi^*, \theta^*)$ to be a mode (relative maximum) of $L(\phi, \theta)$ is that both

$$\frac{\partial L}{\partial \theta} \bigg|_{\theta=\theta^*} = 0 \quad \text{and} \quad \frac{\partial L}{\partial \phi} \bigg|_{\phi=\phi^*} = 0$$

After considerable calculation it can be shown that:

$$\frac{\partial L}{\partial \theta} = \sum_{i \in W} F_i(T) \left( \frac{-b_{i1} \sin \theta \cos \phi + b_{i2} \cos \theta \cos \phi}{B_i \cdot T} \right) + \frac{(S_i \cdot T)(-s_{i1} \sin \theta \cos \phi + s_{i2} \cos \theta \cos \phi)}{1 - (S_i \cdot T)^2}$$

and,

$$\frac{\partial L}{\partial \phi} = \sum_{i \in W} F_i(T) \left( \frac{-b_{i1} \cos \theta \sin \phi - b_{i2} \sin \theta \sin \phi + b_{i3} \cos \phi}{B_i \cdot T} \right) + \frac{(S_i \cdot T)(-s_{i1} \cos \theta \sin \phi - s_{i2} \sin \theta \sin \phi + s_{i3} \cos \phi)}{1 - (S_i \cdot T)^2}$$
where \( F_i(T) = \frac{B_i \cdot T}{c_i^2 \sqrt{1 - (S_i \cdot T)^2}} \)

Setting these partial derivatives equal to zero and then solving for \( \theta \) and \( \phi \) respectively, we obtain:

\[
\theta = J(\phi, \theta) = \tan^{-1} \left( \frac{\sum_{i \in W} F_i(T) \left( \frac{b_{i2} + (S_i \cdot T) s_{i2}}{1 - (S_i \cdot T)^2} \right)}{\sum_{i \in W} F_i(T) \left( \frac{b_{i1} s_{i1} + (S_i \cdot T) s_{i1}}{1 - (S_i \cdot T)^2} \right)} \right)
\]

and

\[
\phi = K(\phi, \theta) = \tan^{-1} \left( \frac{\sum_{i \in W} F_i(T) \left( \frac{b_{i3} + (S_i \cdot T) s_{i3}}{1 - (S_i \cdot T)^2} \right)}{\sum_{i \in W} F_i(T) \left( \frac{b_{i1} \cos \theta + b_{i2} \sin \theta}{B_i \cdot T} + \frac{(S_i \cdot T) (s_{i1} \cos \theta + s_{i2} \sin \theta)}{1 - (S_i \cdot T)^2} \right)} \right)
\]

Thus, at the point \((\phi^*, \theta^*)\) these equations will be satisfied. These equations are used to establish the following recursion relationships:

\[
\theta_{k+1} = J(\phi_k, \theta_k) \quad \text{and} \quad \phi_{k+1} = K(\phi_k, \theta_k), \quad k = 0, 1, 2, \ldots
\]

The FALCONFIX algorithm for estimating \((\phi^*, \theta^*)\) proceeds as follows:

1. Select a starting point \((\phi_0, \theta_0)\). (The selection procedure is discussed later.)
2. \(T_0 = T(\phi_0, \theta_0)\) : Initialize the target vector
3. \(k = 0\) : Initialize \(k\)
4. \(\delta = \delta_1\) : Specify the stopping rule
5. \( \theta_{k+1} = J(\phi_k, \theta_k) \) : Calculate the next longitude
6. \( \phi_{k+1} = K(\phi_k, \theta_{k+1}) \) : Calculate the next latitude
7. \( T_{k+1} = T(\phi_{k+1}, \theta_{k+1}) \) : Calculate the next target vector
8. If \( k > M \) then \( \delta = \delta_2 \) : Relax, the stopping rule if \( k \) is large
9. If \( |T_{k+1} - T_k| < \delta \) or \( k > N \) then go to 12 : stopping rule
10. \( k = k+1 \) : Increment \( k \)
11. Go to 5 : Perform the next iteration
12. \( (\phi_{k+1}, \theta_{k+1}) \) is an estimate for \((\phi^*, \theta^*)\)

Note: \( N > M \) and \( \delta_2 > \delta_1 \)

The intersections of pairs of bearings are used as the starting points for the algorithm. Those pairs which intersect at an angle of greater than 20° are considered first—those intersecting at smaller angles are deferred for later consideration. The first check in determining whether an intersection constitutes a valid starting point is if that particular pair is not a subset of a bearing set \( W \) from a previously generated mode. If it is, the intersection is rejected since the iteration procedure would yield the same mode. The second check is to determine if at least one additional bearing is contained in the set \( W \) at the point of intersection. If not, the intersection is also rejected as a starting point since in this case no iteration will take place and two-bearing modes are not allowable, except in the special case where only two bearings are input.

The following example illustrates the algorithm and the selection of starting points. Suppose the reported bearings are as shown in
Figure 11. Example of Mode Determination

Figure 11. The intersection (1,2) is temporarily set aside because the angle is too small. (1,3) satisfies this criterion and also that at least one additional bearing, (2), is contained in the bearing set at this point. Therefore (1,3) is a valid starting point. The algorithm commences and after several iterations converges to the point A. Since bearings 1, 2 and 3 are in this bearing set, the intersections (1,2), (1,3) and (2,3) are removed from further considerations. (1,4) is the next candidate and since it meets both criteria it is selected as a starting point. The algorithm begins there and converges to point B. In the same way as before, (1,2), (1,4) and (2,4) are eliminated. (3,4) is the only intersection remaining. It satisfies the angular
criterion, but no other bearing is close enough to be a member of the bearing set at the point of intersection. Therefore (3,4) is eliminated. Since no intersections remain as candidates, all modes have been found and the problem is terminated.

When the algorithm generates more than a single mode, we order them with respect to the probability that they coincide with the true location of the emitter. Since each bearing has a probability \( p \) of being a true bearing and for realistic direction-finding systems, this value is greater than 0.5, the mode having the greatest number of forward bearings in its bearing set is considered the most likely. In the case where two or more modes involve the same number of forward bearings, the algorithm uses the degree of dispersion of the bearings as a secondary criterion. Specifically, the average squared perpendicular distance from the position estimate to each line of bearing is computed and the modes are ordered so that the one with the smallest such value is first.
Standard Deviation of Bearing Error

The standard deviation of the bearing error (\( \sigma \)) is used in the calculation of the likelihood function and its partial derivatives. While the particular method used for obtaining \( \sigma \) is not crucial to the FALCONFIX algorithm, for the sake of completeness its computation will be discussed.

The standard deviation is taken to be a function of both the characteristics of the particular receiving site (antenna, equipment, location, etc.) and of the distance between the site and the emitter location. In particular, we assume the multiplicative relationship \( \sigma = h(d) \bar{\sigma} \) where \( \bar{\sigma} \) is a constant representing the normalized standard deviation of the bearing error from the site and \( h(d) \) is an adjustment for distance.

If \( s = \sin \delta \), where \( \delta \) is the central angle between the target vector \( T \) and the station vector \( S \), then

\[
d = 3444 \sin^{-1}(s) \quad \text{, d in nautical miles}
\]

or

\[
s = \sin \left( \frac{d}{3444} \right)
\]

The function \( h(d) \) below is motivated by the Ross Range Curve, although certain modifications have been made.

\[
h(d) = \begin{cases} 
60 & \text{, if } d \leq 60 \\
60 - 3152.387(s - 0.0174524) & \text{, if } 60 < d \leq 120 \\
0.174524/s & \text{, if } 120 < d \leq 600 \\
s + 0.826352 & \text{, if } 600 \leq d \leq 5410 \\
1.826352 & \text{, if } d > 5410 
\end{cases}
\]

A graph of \( h(d) \) is provided in Figure 12 below.
The extremely large standard deviation for distances less than 100 nautical miles or so, is essentially an artifice to assure inclusion of bearings from sites which are close to a potential mode; such bearings are almost invariably good bearings. For example, if $\sigma$ is equal to 1.0 degrees and if the transition point $c$ (referred to in Figure 2) is set equal to 3.0, then any estimate of emitter location within the "keyhole" in Figure 13 will include the bearing from site $s$. (In this figure the origin represents the site and the horizontal axis is the direction of the reported bearing). Notice that the bearing is included whenever the site is within 60 nautical miles of the estimate.
Figure 13. Inclusion Region
The REFIX Procedures

One final procedure is used to refine the position estimates generated by the previously described algorithm. The need for such a termination procedure is two-fold.

- If one or more of the bearings in the included set is from a station which is reasonably close to the position estimate, the artificially large standard deviation associated with such bearings will tend to diminish its rightful impact on the position estimate.
- If the likelihood function happens to be unusually flat in the neighborhood of the mode, it is possible that the stopping rule may terminate the algorithm prematurely, resulting in a reported position estimate which is not coincident with the point at which the relative maximum occurs.

Two different termination procedures have been developed, but both have the following features in common.

- The bearing set is "frozen" to include only those bearings which were included in the final iteration of the algorithm previously described.
- The standard deviations of bearings from stations within 600 nautical miles of the position are modified by changing \( h(d) \) to \( h^*(d) = 0.75 + d/2400 \) where \( d \) is the distance in nautical miles.

REFIX/1. In this procedure, after the bearing set has been frozen and the standard deviations of bearings from close-in stations have been modified, the iterative algorithm is employed again.
REFIX/2. This termination procedure uses techniques employed in multiple regression analysis. Most of the computation done here is required for the confidence region calculations which are discussed in the next section. In this procedure a cartesian coordinate system is set up in a plane tangent to the spherical earth model, such that the origin is coincident with the position estimate produced by the iterative procedure. The lines of bearing are assumed to be straight lines in this plane.

A single representative line of bearing is shown in Figure 14 with \((x_0, y_0)\) being the true target location.

![Figure 14. Coordinate System for the REFIX/2 Procedure](image)

From elementary algebra, the equation of the solid line in Figure 14 is 

\[ (-\cos \beta_i)x + (\sin \beta_i)y = \delta_i \]

The equation of the dashed line through
the true target location \((x_o', y_o')\) is 
\(-\cos \beta_1)x + (\sin \beta_1)y = \delta_1 - e_1\n
Then, since \((x_o', y_o')\) is on the dashed line, we have:

\[
\delta_1 = (-\cos \beta_1)x_o + (\sin \beta_1)y_o - e_1
\]

This equation is in the form of a general multiple regression model.

Both the \(\beta_1\) and the \(\delta_1\) can be calculated since the origin is known. The \(\delta_1\) is approximately the bearing from the origin to the station for distant stations and approximately the reported bearing for near stations.

The \(\delta_1\) is approximately \(\gamma_i \sin \delta_i\) where \(\gamma_i\) is the angular difference between the reported bearing and the bearing from the station to the origin, and where \(\delta_i\) is the distance from the station to the origin.

Also, \(e_i\), the perpendicular distance from the true target location to the bearing, is a normal random variable since \(e_i = \epsilon_i \sin \delta_i\) and \(\epsilon_i\) (the angular error) is a normal random variable. Furthermore, since the \(\epsilon_i\) are normal and independent with mean zero and variance \(\sigma_i^2\), the \(e_i\) will be normal and independent with mean zero and variance \((\sin^2 \delta_i)\sigma_i^2\). Thus, the problem reduces to the simple multiple regression problem of estimating \(\beta\) when

\[
\delta = Ab + e
\]

where

\[
\begin{bmatrix}
\delta_1 \\
\vdots \\
\delta_k
\end{bmatrix} = 
\begin{bmatrix}
-x_1 \\
\vdots \\
x_k
\end{bmatrix} \begin{bmatrix}
-\cos \beta_1 & \sin \beta_1 \\
\vdots & \vdots \\
-\cos \beta_k & \sin \beta_k
\end{bmatrix}
\]

\[
b = \begin{bmatrix}
x_o \\
y_o
\end{bmatrix} \text{ and } e = 
\begin{bmatrix}
e_1 \\
\vdots \\
e_k
\end{bmatrix}
\]
Further, we have that $e$ is distributed as a multivariate random variable with $\mu = \alpha_{kx1}$ and diagonal covariance matrix

$$\Sigma = \sigma_o^2 \begin{bmatrix} \sin^2 d_{w1} & 0 & \cdots & 0 \\ 0 & \sin^2 d_{w2} & \cdots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \sin^2 d_{k} & \sin^2 d_{w_k} \end{bmatrix} = \sigma_o^2 W$$

with $\sigma_i^2 = (w_i\sigma_o)^2$ for each station.

Using the above notation, the maximum likelihood estimate of $b = (x_0, y_0)$, the true target location, is given by

$$\hat{b} = (A^T \Sigma^{-1} A)^{-1} A^T \Sigma^{-1} \delta$$

Once $\hat{b}$ is available, we update the target vector to give the refined position estimate.

**Relative Advantages.** Either termination procedure works quite adequately in practice. The REFIX/1 procedure has a slight advantage when the position estimate is very close to one station since the assumption of parallel movement of bearings which is made in REFIX/2 is not really true in this case. On the other hand, REFIX/2 is quicker in some cases since on occasion REFIX/1 requires a number of additional iterations.
The Confidence Region

The calculation of the confidence region follows directly from the REFIX/2 algorithm. The estimate $\hat{b}$ is distributed multivariate normal with mean $\mu^* = [x_0, y_0]^T$ and covariance matrix

$$
\Sigma^* = (A^T \Sigma^{-1} A)^{-1} = \frac{1}{\sigma^2} (A^T W^{-1} A)^{-1}
$$

The minimum area 100 (1-\alpha)% confidence region is given by

$$
\frac{1}{s_o^2} [x, y]^T \Sigma^* \begin{bmatrix} x \\ y \end{bmatrix} \leq F_{\alpha, 2, n-2}
$$

where $s_o^2$ is the estimate of $\sigma^2$ based on $n$ data points and $F_{\alpha, 2, n-2}$ is the appropriate $F$ distribution percentage point. In this algorithm, the $\sigma^2$ is assumed known so $F_{\alpha, 2, \infty}$ is used. This value is approximated by $-\log_e (\alpha)$.

The ellipse which is generated is given by the direction of the major axis and the magnitudes of the semi-major and semi-minor axes.
Appendix A

Computer Program Flow Chart
Appendix B

Computer Program Listing
RESET FREE

FILE 3=HDF/STATION, UNIT=DISK, RECORD=14, BLOCKING=30

FILE 3 CONTAINS STATION INFORMATION. FORMAT: CARD #1 HAS

THE NUMBER OF STATIONS IDENTIFIED, STATION

STATION LONGITUDE AND STATION STANDARD DEVIATION USING

AZ, BZ (10.0) AS THE INPUT FORMAT. LATITUDE AND LONGITUDE

ARF IN DEGREES WITH NEGATIVE BEING WEST LONGITUDE

FILE 7=7TJDM/PRINTER, RECORD=22

FILE 7 IS A OUTPUT FILE.

FILE 4 IS THE CASE INPUT FILE. DATA IS SET UP AS FOLLOWING:

CARD #2 (12) = 72 CHARACTER TITLE

LATITUDE AND LONGITUDE OF THE LATITUDE AND LONGITUDE OF

THE UNIVERSAL HUMAN STARTING POINT (LATITUDE AND LONG

ITUDE INPUT IN DEGREES AND MINUTES). A SECOND OPTIONS INCLUDES

LATITUDE AND LONGITUDE IN DEGREES AND MINUTES (E.G.,

72, 15 MEANS 72 DEGREES AND 15 MINUTES) USING AN

CARD 3 (12, 24 FB 12) FORMAT.

CARD 3=24 FB 12, STATION ID, REPORTED BEARING.

CARD #3#1 THROUGH #3# FOR EACH CASE.

*********** FALCONFIX **********************

CURRENT VERSION 79710

FALCONFIX IS A FIX COMPUTATION ALGORITHM DEVELOPED BY

LT COL WILLIAM T. BESSON AND MAJOR JOSEPH C. H. SMITH OF

THE DEPARTMENT OF MATHEMATICS, U.S. AIR FORCE

ACADEMY, COLORADO 80840 (AVV 256-4070). THIS ALGORITHM WAS

DEVELOPED UNDER THE JI. J. SPONSORSHIP OF THE NAVAL SECURITY

GROUP COMMAND AND THE US AIR FORCE. IT IS INTENDED FOR USE IN

THIS COUNTRY'S WORDWIDE SYSTEM OF HIGH FREQUENCY DIRECTION

NETS. ADDITIONAL DOCUMENTATION IS PROVIDED IN USAF ACADEMY

TECHNICAL REPORT 79-5. SPECIFIC PAGE REFERENCES IN THE TECH

REPORT GIVEN IN BRACKETS THROUGHOUT THE COMMENTS BELOW (E.G.,

(1) T 79-6:13) REFERS TO PAGE 13 IN THE TECH REPORT.

THE FOLLOWING COMMENT CARD WILL APPEAR THROUGHOUT THE PROGRAM:

### THIS CARD MARKS POINTS IN THE PROGRAM IN WHICH VARIABLES HAVE BEEN
**ASSIGNED SPECIFIC VALUES**

The user may change these values if appropriate.

**OPERATING CONDITIONS**

Indicate that the current values are not appropriate.

**VARIABLE DEFINITIONS:**

**COMMON DATA:**

Station parameters with NST the number of stations.

**ID:**

Station two-letter identifier ALA and ALU station.

**LATITUDE AND LONGITUDE:**

And the long range standard.

**DEVIATION OF ANGULAR ERROR IN DEGREES:**

This area is used to keep track of the data for each individual best point estimate (BPP).

**PREVIOUS POSSIBLE SPECIES HAVE BEEN FOUND:**

LAD, AND AS, AND LOG BY LOG SECONDS FOR BPP LATITUDE AND ANGLE.

**LONGITUDE, MS, AND USE SPECIFY NORTH-SOUTH AND EAST-WEST FOR EACH BEST POINT ESTIMATE (INTERIAL TO THE PROGRAM, THE CONVENTION THAT SOUTH AND WEST ARE NEGATIVE IS USED).**

**AA, AXES, AND AXES:**

Are the search areas major and minor axes, and major axis.

**PROCEDURE:**

Orientation for the confidence ellipse based on the assumption that the variance is known. The value associated with each BPP is the size included bearings.

**DISTANCE FROM THE STATION TO THE BPP, ERR, ABS:**

The absolute value of the angular error for the reported bearing.

**ADD AND AND ARE THE STATIONS REPORTING AND THE REPORTED BEARINGS:**

Last axis is not used in this version.

**COMMON BEARING VECTORS:**

S = STATION VECTORS; T = CURRENT BPP VECTOR.

**DST, S DOT T:**

S = DOT T; FIT = COSINE ERROR/VARIANCE.

**D = SQUARES:**

Of the angle between the BPP and the STATION.

**AST = CURRENT ANGULAR STANDARD DEVIATION:**

And the bearing included in the current calculations. SAV and SIGSAVE are the original B-VECTORS and the standard deviation at 1000 KM. IREV is used to keep track of the reversed BPP.

**SUMMARY:**

Summary data. The dimension of 300 limits the number of bearings sets per run to 300. BSUM saves the first 12 characters of the case title. NSAUM is the number of nodes found.

**DSUM** is the distance to the CMT. CSM is the confidence ellipse in the direction of the CMT. ESUM is the search area.
ORDERING THE MODES. 

LSUM IS THE SIZE OF THE COCKED HAT.

THE THREE BEST THREE MODES OF EACH CASE.

**SUBROUTINES**
BEGIN NIGIER SE \# NAME

57000 ASORT(X,AR,ORD)

52000 FIT(V,INC,3,1+VINC)

50000 DSTAR,DSTAR,ANGLE,PHI,PO

70000 FROSS(A,B,C1)

71000 SMS(A,AD,AM,AS,IM,IP,UIT)

74000 FLACK(V,JN,JN)

76000 PEAS (L,INC)

59100 JEFIX (THE,PHI,VRG,BRG)

33000 STRPTN(PHI,THETA,KC,INC)

**FUNCTIONS**
BEGIN NIGIER SE \# NAME

56900 ARCOS(A)

57400 ARSIN(A)

60500 ARW(L,DS,DIST,ERR,INN,JREV)

73500 DST(D,L)

77000 FSIGN(SIGN,SIGT)

80200 PHIHEM(ALY)

90300 SIGM(SIGN,SIGT)

10000 THE ANA(LA?)

**THE END**

00000 COMON/TS(20),DUT(20),O3(20),S3(20),T3(20),FIT(20),T(20)

01000 ACC(20),JHY(20),JRE(20),BSAV(3,20),SIGSN(20),EAY(20),INN(20)

01000 COMO/TS(20),ST(20),DEL(20),LAN(20),SIG(20),CGC(20),SIGT(20)

01000 COMO/FLAC/PLAC(20,20),ISTR,JSRT

01000 COMO/SAVE/LAD(20),LAM(20),LAD(20),LDD(20),LDM(20),LDS(20)

01100 *MDS(20),MEN(20),ARC(20),AXES(20),VAL(20)

01100 *ABTPTP,INS(20,20),DST(D,20,20),ERR(20,20),IND(20),BRGN(20),ADELG(2)

01100 **CO40/TSAPY/TSUM(300,2),NMSUM(300),BSUM(300,3),CSU

01100 *R(300,3),N(S(300,3),SUM(300,3),S(U(300,3))

01100 *3),BSUM(300,3),FSUM(300)

01100 DIDEWSI, INC(3),TT(3),TS(3),IND(20),IHEAL(4)
DATA 102, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20/ 
C++ READ AND D2P CONVERT FROM RADIANS TO DEGREES AND VICE VERSA. 
C++ REMARKS INCLUDE ERROR CODES 
C++ (+-) = 0 IF DATA REACHES THE TERMINAL 
C++ LIMITS LINES TO 72 CHARACTERS 
C++ LIMITS OUTPUT TO ESSENTIAL INFORMATION 
C++ KEND LIMITS OUTPUT TO ESSENTIAL INFORMATION 
C++ START IS A MANUAL START POINT OPTION IF KSTART IS 0, THE PROGRAM WILL 
C++ BE USED AS THE STARTING POINT, AND ONLY ONE MODE 
C++ KEND LIMITS OUTPUT WHEN DATA INFORMATION IS NOT AVAILABLE 
C++ KEND PROVIDES A TERMINATION OPTION IF KEND IS 0, THE PROGRAM 
C++ TERMINATES BY USING THE REGRESSION FIX COMPUTATION IF KEND IS 
C++ SET AND VARIANCE 
C++ END OF OPTION CONTROLS 
C++ READ IN STATION INFORMATION 
C++ READ IN STATION INFORMATION 
C++ READ CASE TITLE AND BEARING INFORMATION FOR INDIVIDUAL CASE 
C++ NCASE COUNTS THE NUMBER OF CASES, NTTOT COUNTS THE TOTAL NUMBER OF 
C++ ITERATIONS FOR EACH CASE 
C++ IF(KA.EQ.1) WRITE(LM=7075)
C++ IF YOU GET TO HERE THE STATION REPORTING THE BEARING IS
C++ NOT IN THE STATION FILE, THE PROGRAM DELETS THE BEARING

7000 FORMAT("STATION 'A2' IS NOT IN THE STATION FILE.'/)
N=1
GO TO 2

20 A=SIG*SIG
B=2*SIG
C=SIG*SIG

C++CALCULATE THE STATION VECTOR AND THE BEARING VECTOR FOR THE
C++STATION

SIG=SIG*SIG
S(I,1)=S(I,A)*COS(A)
S(I,2)=S(I,A)*SIN(A)
SIG=SIG*SIG
S(J,1)=SIG*SIG
S(J,2)=SIG*SIG
S(J,3)=SIG*SIG

C++IF YOU GET TO HERE, YOU ONLY HAVE TWO STATIONS REPORTING.
C++CALCULATE THE INTERSECTION AND REPORT IT AS A BKE.

C=1
J=3
DO 25 I=1,3
A(SIG(I))=SIG
DO 25 J=1,3
B(J,1)=SIG(J,1)
CALL PTPN(SIG(I),AD,1,THETO,IN)
CALL PTPN(SIG(J,1),SIG(J,2),T,NO)
MOS=1
INX=1
GO TO 25

C++ KC IS A PARAMETER USED TO PREVENT STARTING AT BEARING
C++ INTERSECTIONS WITH LESS THAN A 20 DEGREE INTERSECTION
C++ ANGLE THE FIRST TIME THROUGH, IF KC=3, IT WILL START WITH
C++ DEGREES; THE USE 20 DEGREES. THEN PICK UP THE REMAIN-
C++ DEGREES, IF KC>3, IT WILL START WITH (N-1)°20 DEGREES.
C++ THEN USE (1-2)*20 DEGREES, AND PROCEED TO ZERO DEGREES.
C++ THE FOLLOWING STATEMENT LIMITS THE TOTAL NUMBER OF ITERATIONS TO 20.
C++ THEN ARE INDICATORS TO ROUTE THE PROGRAM THROUGH THE
C++ ANTIPARALLEL POINT PRODUCED LBS 35960-40500.
C++ INITIALIZE THE ARRAY TO INDICATE IF BEARINGS HAVE BEEN REVERSED.
C++ INITIALIZE THE BEARING VECTORS (SOME MAY HAVE BEEN REVERSED).
C++ LIMIT THE TOTAL NUMBERS OF ITERATIONS FOR THIS BEARING SET.
C++ LIMIT USED TO TERMINATE THE ITERATIONS. IF ITMD IS GREATER THAN
C++ 120, ACK IS REFOCUSED. IF ITMD IS GREATER THAN 150, THE ITERATION
C++ 13  TERMINATE.

THE FOLLOWING STATEMENT Initializes the STOPPING CRITERION for the
C++ ITERATIVE PROCEDURE.

C++ FIND A STARTING POINT.

IF (STAD = 1) GO TO 37

C++ IF START= USE -4TLA AND CMTL as the starting point.

ATHT=20+CHTL

T(1)=COS(ATHET)*COS(APHI)

T(3)=SIN(APHI)

T(2)=SIN(APHI)

GO TO 3A

37 CALL STARTT(N,APHI,ATHT,KC,IND)

C++ THE SUBROUTINE STARTT RETURNS THE NEXT STARTING POINT.

C++ IF 4570 ALL STARTING POINTS HAVE BEEN USED

APHI=20+APHI

ATHT=20+ATHT

38 IF (K<70) WRITE(LD,7072)APHI,ATHT

7072 FORMAT(LA,2F10.3)

C++ REVERSE STARTINGS IF NECESSARY

CALL REVORG(N)

C++ A NEW ITERATION BEGINS AT LABEL 50.

C++ CFIT UPDATES THE ARRAYS IN COMMON/ITR/

50 CALL CFIT(N,IND,NCOUNT,END)

C++ COUNT is the NUMBER of USEABLE BEARINGS. IF <3 DO

C++ NOT ITERATE

IF (N=27) 37 TO 60

7071 FORMAT (L4,7071)

50 GO TO 3A

C++ CALCULATE THE NEW LATITUDE (APHI) AND LONGITUDE (ATHT)

60 APHI=PHI(N,ATHT)

APHI=20+APHI

ATHT=LETA(N)

ATHT=20+ATHT
IF (K <= 0.1) WRITE (L, 7073) UTHID, UTHETU, (LUN(1), 1 = 1, 4)
7073 FORMAT (4X, 22 CHARACTER, 13, 10, 38, 5X, 5X, 2512)

C++ CALCULATE THE VECTORS MAGNITUDE OF THE CHANGE AND
C++ TEST FOR STOPPING THE ITERATIONS

T1 = COS (ATHETU) * COS (APHI)
T2 = SIN (ATHETU) * COS (APHI)
T3 = SIN (APHI)
A1 = T1
A2 = T2
A3 = T3
C++ VARIABLE T1 BELOW IS THE MAGNITUDE OF THE DIFFERENCE BETWEEN THE NEW
C++ T-VECTOR AND THE OLD T-VECTOR.
C++ REPLACE THE OLD T-VECTOR WITH THE NEW T-VECTOR.

C++ THE STATEMENT BELOW LIMITS THE MAXIMUM NUMBER OF ITERATIONS FOR ANY
C++ MENU MODE TO 150.
C++ THE FOLLOWING STATEMENT REDUCES THE STOPPING CRITERION FOR THE
C++ ITERATIVE PROCEDURE WHEN THE NUMBER OF ITERATIONS EXCEEDS 120.

IF (ITR <= 150) GO TO 130
C++ THE ITERATIVE PROCEDURE HAS TERMINATED WHEN YOU GET TO LABEL 130.
C++ INITIATE THE PROPER REFIX PROCEDURE DEPENDING UPON THE VALUE OF NEND.
C++ ITR = 130

130 WRITE (M, 495) ITR
C++ APE \$F (\$F3) = \$F4.
C++ IF (VEN) GO TO 135
C++ IF (VEN) GO TO 140
500 IF (VEN) \$F7 \$F8 \$F9 \$F10 \$F11 \$F12
SOMO \$F42 (' TERMINATION INITIATED ')
JEND
GO TO 30
135 CALL REFIX (ATHETU, APHI, NBRIG)
C++ MOVE OF NUMBER
C++ A BEST POINT ESTIMATE HAS BEEN FOUND. CALCULATE AND SAVE THE
C++ NECESSARY INFORMATION.
140 CALL C34F2(I2, CSTAR, DSTAR, ANGLE, APHI, .95)
MOS=MOD(I, 2)
C++ CALCULATE THE DISTANCE FROM THE STATION TO THE BPE
DO 160 I=1, N
C++ SAVES THE INCLUDED BEARINGS ARRAY:
INS(MOS, 1)=STY(I)
APARSIN(SORT(I), 1)
IF(CST(I) .LT. 3) = 141.592654 = A
DIST(I, 1)=4344
C++ IF THE DISTANCE IS GREATER THAN 90 DEGREES AND THE BEARING
C++ HAS BEEN REVERSED, CONSIDER IT AN AROUND THE WORLD BEARING
IF(DIST(MOS, 1) .GT. 90) .AND. INE(I, 1) DIST(MOS, 1)=
*17561.50-360*INSTE(MOS, 1)
C++ CALCULATE THE BEARING ERRORS FOR EACH STATION
DEMON(I-1, 1) = STY(I)/SORT((I-1, 1), 1), I
DEMON(I)=4PCOS(DELETE(MON)+20)
IF(S2, 1) AT (P) = 1) .LT. 0) DEMON=360-DEMON
BPR(I-1, 1) = 1)
C++ ERRORS DEMON=
ERR(MOS, 1)=ERR(MOS, 1)
IF(A=0, 1) ERR(MOS, 1)=SIGN(360, -A, ERR(MOS, 1))
150 CONTINUE

C++SAVES INFORMATION FOR THIS MD:
CALL DMS(ANGLE, LAM(MOS), LAM(MOS), LAS(MOS), MNS(MOS), 0)
CALL DMS(ANGLE, LAM(MOS), LAM(MOS), LAS(MOS), MNS(MOS), 1)
A=2(MOS)=CSTAR+DSTAR*3.1416
AXEM(I-1, 1)=AXEM(I-1, 1)
AXEM(I)=AXEM(I-1, 1)
CSTAR=STAR-STAR
VAL(MOS)=LAM(MOS), LAM, LAS, MNS, LUD(M
ADEG(MOS)=2ADDEG
IF(XF, 1) = 1074, LAM(MOS), LAM(MOS), LAS(MOS), MNS(MOS), LUD(M
4) AT (P) = 1) .LT. 12, 13, 14, 12, 13, 14, 12, A, 12, 13, 12, 13, 14
110X, 110X, 110X, 110X, 110X
C++ |PROCEDURE TO INCLUDE THE ANTIPODAL POINT IF THERE ARE
C++ (I/2)=1 OR 4, IF REVERSED BEARINGS(LNS, 35900=40500)
C++ INCLUDES THE INCLUDED BEARINGS, S2 COUNTS THE BEARINGS WHICH ARE
C++ INCLUDED AND REVERSED. DISTSU IS THE SUM OF THE DISTANCES TO BPE.
S2=0
I=0
DISTSU=0
DO 153 1=1, 4
IF (I = N (I), F) 0) GO TO 153
11 = 1
DIST2 = DIST2 + DIST (NDS, I)
IF (J > F) THEN 150 TO 153
12 = 2

153 IF (STATUS < 1) GO TO 156
DIST = DIST + DIST (NDS, I)
IF (DIST > 7.5, 400) THEN 156
I = I + 1
IF (J = F) THEN 150

C++ CAUSES THAT TO BE THE ONLY REPORTED BPE.

C++ CAUSE THAT TO BE THE ONLY REPORTED BPE.

C++ CAUSE THAT TO BE THE ONLY REPORTED BPE.

C++ CAUSE THAT TO BE THE ONLY REPORTED BPE.

C++ CAUSE THAT TO BE THE ONLY REPORTED BPE.

C++ CAUSE THAT TO BE THE ONLY REPORTED BPE.

C++ CAUSE THAT TO BE THE ONLY REPORTED BPE.

C++ CAUSE THAT TO BE THE ONLY REPORTED BPE.

C++ CAUSE THAT TO BE THE ONLY REPORTED BPE.

C++ CAUSE THAT TO BE THE ONLY REPORTED BPE.

C++ CAUSE THAT TO BE THE ONLY REPORTED BPE.

C++ CAUSE THAT TO BE THE ONLY REPORTED BPE.

C++ CAUSE THAT TO BE THE ONLY REPORTED BPE.

C++ CAUSE THAT TO BE THE ONLY REPORTED BPE.

C++ CAUSE THAT TO BE THE ONLY REPORTED BPE.

C++ CAUSE THAT TO BE THE ONLY REPORTED BPE.

C++ CAUSE THAT TO BE THE ONLY REPORTED BPE.

C++ CAUSE THAT TO BE THE ONLY REPORTED BPE.
C++ POINTS... 165 FIND INTERSECTIONS OF USED BEARINGS AS POSSIBLE STARTING POINTS. CALL FLAGST(4,IV4) GO TO 36 167 IF KC < 0 ALL MODES HAVE BEEN FOUND. STOP.
C++ FIND THE THEFF REST BEE'S BASED ON THE NUMBER OF FORWARD
C++ BEARINGS AND THE HAT SIZE.
170 IF (IV3,NE,0) GO TO 115
170 WRITE(LW,7021) TITTLE
170 CALL SIS(SHIA,33,IA,II,IAS,JYS,0)
170 CALL NIS(SHIA,33,IA,II,IAS,JYS,0)
7035 FORMAT(IV, NUMBER OF BEARINGS; 1X,13)
171 CPT LALU; '12,1','12,'
172 WRITE(LW,7019)
7020 IF (4,STAF,1) WRITE(LW,7036)N,1D,1,IA,1AM,IAS,JYS,1UM,1OM,10S,1EM,MUS
7036 FORMAT(IV, NUMBER OF BEARINGS; 1X,13)
172 CPT LALU; '12,1','12,'
173 WRITE(LW,7020)
7021 IF (5,STAF,1) WRITE(LW,7037)
7022 IF (6,STAF,1) WRITE(LW,7057)
7023 WRITE(LW,7028)
7024 WRITE(LW,7057)
7025 WRITE(LW,7058)
7026 GO TO 112
7027 IF (4,STAF,1) GO TO 177
7028 WRITE(LW,7048)ID(1),BMNG(1)
7029 GO TO 112
7030 CALL ASORT (IV3,INN)
7031 NP F Y = N P E F M I N (3,IV3)
7032 IF (IV3,GT,1) GO TO 177
7033 INC(1)=100
7034 GO TO 316
7035 CALL ASORT (IV3,INN)
7036 NP F Y = N P E F M I N (3,IV3)
7037 IF (IV3,GT,1) GO TO 177
7038 INC(1)=100
7039 GO TO 316
175 CALL ASORT (IV3,INN)
7040 NP F Y = N P E F M I N (3,IV3)
7041 IF (IV3,GT,1) GO TO 177
7042 INC(1)=100
7043 GO TO 316
C++ BEGINNING OF A ROUTINE TO ELIMINATE MODES WITH THE SAME BEARING.
C++ SETS MULTIPLE MODES WITH THE SAME BEARING SET CAN OCCUR WHEN A
C++ STARTING POINT BEARING IS NOT IN THE FINAL BEARING SET.
0270 7014 FORMAT (/'AR4T/0,3(F12.0,2x))
0280 7015 WRITE(6,7015)((AXE6(NVH),J),J=1,2),AVG(INR(J)),I=1,NPHT)
0290 7016 FORMAT (/'DSR,3X, 'AR4T/0,3(F12.0,2x))
0300 7017 WRITE(6,7017)(I),AVG(INR(J),I),DISI(INR(J),1)
0310 CONTINUE
0320 IF (KACMT/ge,1) GO TO 7
0330 WRITE(6,7032)
0340 7040 IF (KACMT.ge,1) GO TO 275
0350 WRITE(6,7050) NCSUM(I),CNCSUM(NCSUM(I),I),NCSUM(I)
0360 7050 FORMAT (' / BSUM(INR(J)),J=1,2),NMSUM(I)
0370 CONTINUE
0380 IF (KACMT.ge,1) GO TO 275
0390 WRITE(6,7060) NCSUM(I),CNCSUM(NCSUM(I),I),NCSUM(I)
0400 7060 FORMAT (' / BSUM(INR(J)),J=1,2),NMSUM(I)
0410 CONTINUE
0420 IF (KACMT.ge,1) GO TO 275
0430 WRITE(6,7070) NCSUM(I),CNCSUM(NCSUM(I),I),NCSUM(I)
0440 7070 FORMAT (' / BSUM(INR(J)),J=1,2),NMSUM(I)
0450 CONTINUE
0460 IF (KACMT.ge,1) GO TO 275
0470 WRITE(6,7080) NCSUM(I),CNCSUM(NCSUM(I),I),NCSUM(I)
0480 7080 FORMAT (' / BSUM(INR(J)),J=1,2),NMSUM(I)
0490 CONTINUE
0500 IF (KACMT.ge,1) GO TO 275
0510 WRITE(6,7090) NCSUM(I),CNCSUM(NCSUM(I),I),NCSUM(I)
0520 7090 FORMAT (' / BSUM(INR(J)),J=1,2),NMSUM(I)
0530 CONTINUE
0540 IF (KACMT.ge,1) GO TO 275
0550 WRITE(6,7090) NCSUM(I),CNCSUM(NCSUM(I),I),NCSUM(I)
0560 7090 FORMAT (' / BSUM(INR(J)),J=1,2),NMSUM(I)
0570 CONTINUE
0580 IF (KACMT.ge,1) GO TO 275
0590 WRITE(6,7090) NCSUM(I),CNCSUM(NCSUM(I),I),NCSUM(I)
0590 7090 FORMAT (' / BSUM(INR(J)),J=1,2),NMSUM(I)
0600 CONTINUE
0610 CONTINUE
FUNCTION ARCCOS(A)
C++ THIS FUNCTION CALCULATES THE ARCCOS OF A. IT ALLOWS THE
C++ ARGUMENT TO BE GREATER THAN ONE SINCE ROUNDOFF ERRORS CAN
C++ GIVE SLIGHTLY WRONG VALUES OF A.
056900 ARCCOS0
056910 IF(ABS(A).LT.1)ARCCOS=ARCOS(A)
056920 RETURN
END

FUNCTION ARCSIN(A)
C++ THIS FUNCTION CALCULATES THE ARCSIN OF A. IT ALLOWS THE
C++ ARGUMENT TO BE GREATER THAN ONE SINCE ROUNDOFF ERRORS CAN
C++ GIVE SLIGHTLY WRONG VALUES OF A.
057400 ARCSIN=SIGN(1,570796327.A)
057500 IF(ABS(A).LT.1)ARCSIN=ARSIN(A)
057600 RETURN
END

SUBROUTINE ASORT(X,N,IORD)
C++ THIS ROUTINE SORTS THE ARRAY X AND RETURNS THE INDEX OF
C++ THE LARGEST IN IORD(1), THE SECOND LARGEST IN IORD(2), ETC.
C++ DIMENSION X(1),IORD(1),TEMP(25)
058000 DO 10 I=1,N
058100 IORD(I)=I
058200 10 IF(N.LE.1)RETURN
058300 NL=0
058400 DO 50 IPI=I+1,NPI
058500 TEMP(I)=X(I)
058600 50 IF(TEMP(I).GE.TEMP(J))GO TO 60
058700 IORD(J)=IORD(J)
058800 IORD(J)=ISAV
058900 ISAV=TEMP(J)
059000 60 NL=NL+1
059100 IF(NL.GE.25)GO TO 10
059200 059300 IORD(J)=IORD(J)
059400 IORD(J)=ISAV
059500 059600
059600       TEMP(1)=TPH(1)
059700       TEMP(1)=TIV
059800
059900       60 CONTINUE
060000       100 RETURN
060100       END

060200       FUNCTION AVA(L, M1, DS1, D1, R, INN, JREF)
060300       C++ THIS SUBROUTINE CALCULATES A VALUE TO DETERMINE THE "BEST" MODE.
060400       C++ THE VALUE USED IS THE NUMBER OF FORWARD BEARINGS PLUS 1/HAT SIZE.
060500       DIMENSION JREF(1), INN(1), DIST(20, 20), ERR(20, 20)
060600       I1=0
060700       DO 111=1,11
060800       IF (INN(I)=1) GO TO 111
060900       HATSZ=0
060100       IF (JREF(1,F11)<.1)30 TO 100
061200       HATSZ=HATSZ+(SIN(INN1(M1,1))/344.)*TAN(ERR(MUS, 11)/57.295781344.)*2
061300       I1=I1+1
061400       IF (JREF(1,F11)<.1)30 TO 100
061500       100 CONTINUE
061600       C++ HATSZ MUST BE LIMITED TO A NUMBER GREATER THAN 1 SO THAT IT DOES
061700       C++ OVERCOME THE NUMBER OF FORWARD BEARING AS THE PRIMARY CRITERION.
061800       C++ THE HATSZ IS FORCED TO BE SLIGHTLY GREATER THAN 1 SO THAT IT
061900       C++ WILL NOT AVERAGE A PRIMARY CRITERION FOR ORDERING THE MODES.
062000       HATSZ=MAX(1,0.0001,HATSZ/I1)
062100       AVAL=12+1.7/HATSZ
062200       RETURN
062300       END
SUBROUTINE CONF2(Y, CSTAR, DSTAR, ANGLE, PHI, PC)

C++ CALCULATE THE SEARCH REGION

C++ THE CONFIDENCE REGION IS CALCULATED UNDER THE ASSUMPTION

C++ THAT (XBAR - SIGMA) / SIGMA IS APPROXIMATELY NORMALLY

C++ DISTRIBUTED WITH 'MEAN 0 AND VARIANCE SIGMA**2'. ('D' IS THE

C++ DISTANCE FROM THE STATION TO THE BORE, AND SIGMA IS THE

C++ RATIO OF THE TRUE STANDARD DEVIATION TO SIGMA). SIGMA

C++ IS ASSUMED < MEAN AND EQUAL TO 1. (D 70.5 = 30)

COMMON/T2P/DST(20), FST(20), B(320), TS(320), T(20), D(20)

COMMON/SAVE FLAG(20), XSTAR(20), YSTAR(20), INN(20)

COMMON/SAVE FLAG(20), XSTAR(20), YSTAR(20), INN(20)

+4NS(20), M(20), A20(20), ASE(20), VAL(20)

+VAST(20), INS(2), DIST(20, 20), ERR(26, 20), IDO(20), ERNG(20), AOE(G(2

+0), TITLE(12)

A=0

BB=0

C=0

D=0

IF(INN(T), T, 0) GO TO 100

B=0(T)=O(1) =OST(20)*2 =ASTAR(T)*SIGMA(I)

A=0(T)=O(1) =OST(20)*2 =ASTAR(T)*SIGMA(I)

BB=0(T)=O(1) =OST(20)*2 =ASTAR(T)*SIGMA(I)

C=0(T)=O(1) =OST(20)*2 =ASTAR(T)*SIGMA(I)

CONTINUE

100 RETURN

A=472

C=0

C++ ALN IS PERPENDICULAR TO THE CRITICAL F VALUE FOR 2 AND

C++ INFINITE DEGREES OF FREEDOM WHICH IS USED IF THE VARIANCES

C++ ARE ASSUMED KNOWN

IF(A-C) NE 0)30 TO 200

ANFLE = 1583982

GO TO 300

200 ANGLE = ATAN(BB/(A-C))/2

300 CACOS(ANGLE)

SASIN(ANGLE)

CTARR = SQRT(ALN/(A*CA*CA+BB*CA*SA+C*SA*SA)) * 3444

DSTAR = SQRT(ABS(ANGLE)) ANGLE = 1.5707963 - ANGLE

IF(ANGLE.LT.DSTAR) ANGLE = ANGLE + 1.5707963

IF(ANGLE.LT.0) ANGLE = ANGLE + 1.5707963
070200    CSTAR=AMAX1 (CSTAR+1*)
070300    DSTAR=AMAX1 (DSTAR+1*)
070400    RETURN
070500    END

070800    SUBROUTINE CROSS(A,B,C)
070900    C++ CALCULATE THE CROSS PRODUCT A CROSS B AND RETURN THE
071000    C++ RESULT T3 C
071100    C++ THIS SUBROUTINE IS CALLED USING THE DIMENSIONAL ARRAYS WITH THE
071200    C++ FIRST INDEX REPRESENTING THE VECTOR COMPONENTS:
071300    DIMENSION A(3),B(3),C(3)
071400    C(1)=A(2)*B(3)-A(3)*B(2)
071500    C(2)=A(3)*B(1)-A(1)*B(3)
071600    C(3)=A(1)*B(2)-A(2)*B(1)
071700    RETURN
071800    END

072000    SUBROUTINE DMS(AA,AD,AM,AS,IM,IOPT)
072100    C++ CONVERTS A DECIMAL ANGLE TO DEGREES MINUTES AND SECONDS.
072200    INTEGER AD,AM,AS
072300    DATA N5,'S','E','W','N','S','E','W'/
072400    A=AA+.5729577951
072500    IF (I0PT .EQ. 1) GO TO 10
072600    TH=AA
072700    IF (A .LT. 0) H=-N5
072800    GO TO 20
072900    10    H=NE
073000    IF (A .LT. 0) H=-N4
073100    20    I=ABS(A)
073200    J=60*ABS(A)-1
073300    A=60*ABS(A)-1
073400    K=60*ABS(A)-1
073500    AD=I
073600    AM=J
073700    AS=K
073300   RETURN
073400   END

073500   FUNCTION DOT(A,C)
073600   C++ CALCULATE THE DOT PRODUCT OF TWO VECTORS
073700   C++ THIS FUNCTION IS CALLED USING TWO DIMENSIONAL ARRAYS WITH THE
073800   C++ FIRST INDEX REPRESENTING THE VECTOR COMPONENTS.
073900   DIMENSION A(3),C(3)
074000   DOT=A(1)*C(1)+A(2)*C(2)+A(3)*C(3)
074100   RETURN
074200   END

074300   SUBROUTINE FLAG:<(U:1JY)
074400   COMMON/FLAG1/FLAS(20,20),ISTRT,JSTR
074500   JLEN=-1
074600   IF(ISTR.GT.JLEN)GO TO 12
074700   DO 10 I=1,ISTRT+1
074800   IP1=IP1+1
074900   IF(IP1.GT.JSTR)IP1=JSTR
075000   DO 10 J=1,ISTRT+1
075100   IF(IFLAG(I,J).EQ.1)GO TO 10
075200   JNE=J
075300   ISTR=I
075400   IF(IP1.IEQ.IP1+1)RETURN
075500   ISTR=I+1
075600   IF(JSTR.GT.JSTR+1)RETURN
075700   JSTR=J+1
075800   RETURN
075900   CONTINUE
076000   10 CONTINUE
076100   JNE=0
076200   RETURN
076300   END
SUBROUTINE FLASND(IN, INC)

C++ SET THE IFLAG ARRAY VALUES TO 1 (INDICATING THE INTERSECTION
C++ CANNOT BE USED AS A STARTING POINT) FOR ALL INTERSECTIONS OF
C++ BEARINGS WHICH HAVE NOT BEEN USED IN THE CURRENT FIX.

C++ COMMON/FLAG/IFLAG(I, J), ISTRJ, JSTRT

C++ DIMENSION INC(20)

C++ IF(I@1(I) = INC) DO 10 I=1, INC

10 CONTINUE

 RETURN

END

FUNCTION FSIG(S2, SIGM, SDT)

C++ THIS FUNCTION CALCULATES THE STANDARD DEVIATIONS OF A BEARING AS
C++ A FUNCTION OF THE RANGE FROM THE STATION TO THE TARGET VECTOR.
C++ THE PROEDURE IS USED TO CALCULATE THE FINAL STANDARD DEVIATION
C++ FOR INPUT IN THE REFIX PROCEDURE. IT IS DIFFERENT FROM SIG IN
C++ THAT IT ELIMINATES THE KEYROLE AROUND THE STATION. [TR 79-5123, 26]

C++ SQ IS THE SQUARE OF THE DISTANCE FROM THE STATION TO THE TARGET.
C++ SDT IS THE COSINE OF THE DISTANCE. SIGM IS THE INPUT STANDARD
C++ DEVIATION AND IS USED AS THE LONG RANGE STANDARD DEVIATION.

C++ THE CONSTANT 1.826352 IS THE LONG RANGE MULTIPLIER OF THE STANDARD
C++ DEVIATION DERIVED FROM THE ROSS RANGE CURVE.

C++ SIGM = FSIG + SIGM

FINISH(S2)

DIST=FSIG(S2)

IF(DIST < .600) FSIG = .75 + DIST/2400.

IF(DIST > .800) AND (DIST < 5.010) FSIG = .826352 + S

FSIG=FSIG+S

RETURN

END
FUNCTION PHINEW (A,Y)
C++ CALCULATE A NEW ALTITUDE [TR 79-5119]
COMMON /TR/ (S1*20) , B1 (20), B3 (20) , S (3,20) , T (3) , FI (20) , D1 (20)
* , CC (20) , IPEV (20) , JR (20) , BSAD (3,20) , SIGSAV (20) , ASIG (20) , IN (20)

S1 = 0
DB = 0
IF (S1 .EQ. 0) GO TO 10
S2 = S1 + IN (I) * FIT (I) + ( D1 (1) + CO (A) + D2 (1) + SIN (A) ) / DT (1) + DST (1)
* ( S (1,1) * COS (A) + S (2,1) * SIN (A) ) / DT (1)
10 CONTINUE
IF (S2 .EQ. 0) GO TO 20
PHINEW = AT TV (SI/S2)
RETURN
20 PHINEW = 1 .* 70795327
IF (SI .EQ. 0) PHINEW = -PHINEW
RETURN
END

SUBROUTINE REFIX (THETA, PHI, NBRG, BRNG)
C++ SUBROUTINE REFIX IS A PROCEDURE USING A STANDARD REGRESSION
C++ RECURRING SET. THE THEA AND PHI ARE INITIAL ESTIMATES OF THE
C++ X AND Y ARE USED AS THE ORIGIN FOR THE RECURRING PROBLEM.
C++ (TR 79-5127)
COMMON /TR/ (S1*20) , B1 (20) , B3 (20) , S (3,20) , T (3) , FI (20) , D1 (20)
* , CC (20) , IPEV (20) , JR (20) , BSAD (3,20) , SIGSAV (20) , ASIG (20) , IN (20)

D2 = 0 .17453293
C++THE FOLLOWING ARE COMPONENTS OF THE X = TRANSPOSE-X AND X = TRANSPOSE-Y
C++ MATRICES

X11 = 0.
X12 = 0.
X13 = 0.
X14 = 0.
DO 100 I = 1, NBRG
SIGMA = SIGA (I) , SIGSAV (I) , DST (I)
ASIG (I) = SIGMA
DO 100 100
RETURN
END
088200  T(2)=T(2)/C_T
088300  T(3)=T(3)/C_T
088400  PRINT 99999(T(1)/COS(PHI))
088500  IF (T(7).LT.0) TETA=TETA
088600  END

088900  C++ SUBROUTINE REV33(V)
089000  C++ REVERSE BEARING. IREV INDICATES WHICH BEARINGS HAVE BEEN
089100  C++ REVISED.
089200  C++ REVISED.
089300  CC13N,TT,ST(20),DBT(20),Bl(20),S(3,20),T(3),F1T(20),J(20)
089400  C++ REVISED.
089500  DO 200 1=1,1
089600  DO 99 9J=1,1
089700  C++ REVISED.
089800  CONTINUE
089900  CONTINUE
090000  200 CONTINUE
090100  RETURN
090200  END

090300  FUNCTION SIGC(G2, SIGM, D01)
090400  C++ THIS FUNCTION CALCULATES THE STANDARD DEVIATIONS OF A BEARING AS
090500  C++ A FUNCTION OF THE RANGE FROM THE STATION TO THE TARGET VECTOR.
090600  C++ THE PROCEDURE USES AN APPROXIMATION TO THE RUSS RANGE CURVE.
090700  C++ TR 79-1231
090800  C++ TR 79-1231
090900  C++ "G" IS THE SIGN OF THE DISTANCE FROM THE STATION TO THE TARGET.
091000  C++ "G" IS THE COSINE OF THE DISTANCE. SIGM IS THE INPUT STANDARD.
091100  C++ DEVIATION AND IS USED AS THE LONG RANGE STANDARD DEVIATION.
091200  C++ FESTM(I,23532) IS THE LONG RANGE MULTIPLIER OF THE STANDARD
091300  C++ THE CONSTANT 1.325352 IS THE LONG RANGE MULTIPLIER OF THE STANDARD
C++ DEVICATION DERIVED FROM THE ROSS RANGE CURVE.

C++ IF S02 IS LESS THAN 0, THE DISTANCE IS GREATER THAN 90 DEGREES.

C++ THE DISTANCES BELOW ARE DERIVED FROM THE ROSS RANGE CURVE. FOR
C++ CLOSE TO DISTANCES, THE VARIANCE HAS BEEN INCREASED TO PREVENT
C++ UNAERFIFIED ELIMINATION OF BEARINGS FROM NEAR STATIUS.

C++ FINDING STARTING POINT.

C++ SUBROUTINE STRTPT(\NPHI, \NTHETA, \NK, \NNO)

C++ DIMENSION 3=20, TT(3), TS(3)

C++ WRITE CONTROL ***

C++ CALL CLAGG(7, \NN, I4, J4)
035100 IF (K..E.1) RETURN
09400  \[
\text{\texttt{COD}} = \cos(\text{\texttt{PHI}})
\]
09500  \[
\text{\texttt{THETA}} = \text{\texttt{ATAN2}}(\text{\texttt{T}(2)}/\text{\texttt{COD}}, \text{\texttt{T}(1)}/\text{\texttt{COD}})
\]
09600  \[
\text{\texttt{IFLAG(IN\_JN)}} = 1
\]
09700  \[
\text{\texttt{IFLAG(JN\_IN)}} = 1
\]
09800  \[
\text{\texttt{RETURN}}
\]
09900  \[
\text{\texttt{END}}
\]

100000  \[
\text{\texttt{FUNCTION}} \text{\texttt{THETAN}\_\texttt{(JN, IN)}} \text{\texttt{)}}
\]
100100  \[
\text{\texttt{C++ Calculates a \texttt{Phi} Longitude (TR 79-519)}}
\]
100200  \[
\text{\texttt{COMMON}} \text{\texttt{AI}} \text{\texttt{R,S,T}}(20) \text{\texttt{,A,B}}(3,20) \text{\texttt{,C,S}}(3,20) \text{\texttt{,T}}(3) \text{\texttt{,FIT20}} \text{\texttt{,D20)}}
\]
100300  \[
\text{\texttt{,CC(20),JREV(20),JREV(20),BSAV(3,20),SIGSAV(20),AS13(20),JN\_IN(20)}}
\]
100400  \[
\text{AA} = 150 * \text{\texttt{.017453293}}
\]
100500  \[
\text{\texttt{BI}} = 0
\]
100600  \[
\text{\texttt{SE}} = 0
\]
100700  \[
\text{\texttt{DD(10)} = 1\text{\texttt{AN}}}
\]
100800  \[
\text{\texttt{IF}}(\text{\texttt{O}}(1) \text{\texttt{,DCO}}) \text{\texttt{,SN}} \text{\texttt{,T}}(1) \text{\texttt{,EW}}) \text{\texttt{,GN]]) \text{\texttt{GO TO 10}}
\]
100900  \[
\text{\texttt{SE = SE + \text{\texttt{SIN}}(1) * \text{\texttt{FT}(1)} * (\text{\texttt{B}}(1) \text{\texttt{,S}}(1)) \text{\texttt{,D}}(1)/\text{\texttt{D}}(1))}}
\]
101000  \[
\text{\texttt{SE = SE + \text{\texttt{SIN}}(1) * \text{\texttt{FT}(1)} * (\text{\texttt{B}}(1) \text{\texttt{,S}}(1)) \text{\texttt{,D}}(1)/\text{\texttt{D}}(1))}}
\]
101100  \[
\text{\texttt{GO TO 10}}
\]
101200  \[
\text{\texttt{C++ Put the \texttt{THETA} in the correct quadrant.}}
\]
101300  \[
\text{\texttt{THETAN}} = \text{\texttt{THETAN}} \text{\texttt{(I)}} \text{\texttt{(N)}}
\]
101400  \[
\text{\texttt{IF}}(\text{\texttt{THETAN}} \text{\texttt{C\_GE}}} \text{\texttt{,A\_T\_LT}} \text{\texttt{,B}} \text{\texttt{,OR A\_T\_LT}}} \text{\texttt{,C}} \text{\texttt{,THETAN}} \text{\texttt{AA}}
\]
101500  \[
\text{\texttt{IF}}(\text{\texttt{THETAN}} \text{\texttt{C\_GE}}} \text{\texttt{,A\_T\_LT}} \text{\texttt{,B}} \text{\texttt{,OR A\_T\_LT}}} \text{\texttt{,C}} \text{\texttt{,THETAN}} \text{\texttt{AA}}
\]
101600  \[
\text{\texttt{RETURN}}
\]
101700  \[
\text{\texttt{THETAN = \text{\texttt{THETAN}}}}
\]
101800  \[
\text{\texttt{IF}}(\text{\texttt{THETAN}} \text{\texttt{C\_LT}}} \text{\texttt{,A\_T\_LT}}} \text{\texttt{,B}} \text{\texttt{,OR A\_T\_LT}}} \text{\texttt{,C}} \text{\texttt{,THETAN}} \text{\texttt{AA}}
\]
101900  \[
\text{\texttt{RETURN}}
\]
102000  \[
\text{\texttt{THETAN = \text{\texttt{THETAN}}}}
\]
102100  \[
\text{\texttt{IF}}(\text{\texttt{A\_T\_LT}}} \text{\texttt{,THETAN}} \text{\texttt{-THETAN}}
\]
102200  \[
\text{\texttt{RETURN}}
\]
102300  \[
\text{\texttt{END}}
\]
Appendix C

Sample Output
**TEST CASE NR 1**

**NUMBER OF BEARINGS:** 6  
**CMT LADL:** 39-0-04  
**105-0-04**  
**NUMBER OF OJES:** 1

**BPE:**  
39-29-294  
104-57-04

**AREA:**  
34.6 x 3300

**AXES:**  
30.4 (53)

<table>
<thead>
<tr>
<th>BEARINGS</th>
<th>INC</th>
<th>DISTANCE</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBR</td>
<td>1</td>
<td>651.7</td>
<td>0.4</td>
</tr>
<tr>
<td>BRR</td>
<td>1</td>
<td>914.7</td>
<td>0.4</td>
</tr>
<tr>
<td>BBRR</td>
<td>1</td>
<td>792.1</td>
<td>0.4</td>
</tr>
<tr>
<td>BRRR</td>
<td>1</td>
<td>533.5</td>
<td>0.4</td>
</tr>
<tr>
<td>MB</td>
<td>1</td>
<td>1564.5</td>
<td>0.4</td>
</tr>
<tr>
<td>MA</td>
<td>1</td>
<td>1364.5</td>
<td>0.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BEARING TO CAT ECLIPSE</th>
<th>DIST TO CAT CMT</th>
<th>RATIO</th>
<th>SEARCH AREA</th>
<th>INCLUDED BEARINGS</th>
<th>FORWARD BEARINGS</th>
<th>CMT SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>184.6</td>
<td>32.0</td>
<td>29.1</td>
<td>0.911</td>
<td>3300.0</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>
**TEST CASE NR 2**

**NUMBER OF BEARINGS:** 7  
**CMT LAGO:** 38-13- ON  85-54- ON  **NUMBER OF NODES:** 2

**AREA:**  
**AXES:** 47.0 x 45.8 (55.0)  
41.0 x 21.6 (21.6)

<table>
<thead>
<tr>
<th>BEARINGS</th>
<th>INC</th>
<th>DISTANCE</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBR 216.7</td>
<td>1</td>
<td>853.1</td>
<td>0.2</td>
</tr>
<tr>
<td>MBR 106.7</td>
<td>1</td>
<td>1730.1</td>
<td>5.1</td>
</tr>
<tr>
<td>SBR 106.7</td>
<td>1</td>
<td>1981.5</td>
<td>37.5</td>
</tr>
<tr>
<td>OSBR 281.1</td>
<td>0</td>
<td>1485.5</td>
<td>31.3</td>
</tr>
<tr>
<td>NMP 281.1</td>
<td>0</td>
<td>1480.1</td>
<td>15.3</td>
</tr>
<tr>
<td>LI 261.0</td>
<td>1</td>
<td>1585.5</td>
<td>0.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BEARING TO CMT</th>
<th>DIST TO ECLIPSE</th>
<th>DIST TO CMT</th>
<th>RATIO</th>
<th>SEARCH AREA</th>
<th>INCLUDED BEARINGS</th>
<th>FORCADO BEARINGS</th>
<th>CKR</th>
<th>SIZE MAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>88.4</td>
<td>39.7</td>
<td>891.7</td>
<td>22.476</td>
<td>4548.0</td>
<td>4</td>
<td>4</td>
<td></td>
<td>28,936</td>
</tr>
<tr>
<td>135.6</td>
<td>27.5</td>
<td>12.7</td>
<td>0.452</td>
<td>2794.0</td>
<td>4</td>
<td>4</td>
<td></td>
<td>32,191</td>
</tr>
</tbody>
</table>
### Test Case NR 3

**Number of Bearings:** 5  
**CMT LALO:** 35-50-0N  
**75-08-0N**  
**Number of Nodes:** 2

<table>
<thead>
<tr>
<th>BPE</th>
<th>30-46-17H</th>
<th>75-56-59H</th>
<th>36-22-5H</th>
<th>77-20-17H</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREA AXES</td>
<td>50.9 x 159.0 (77.0)</td>
<td>28.8 x 238.0 (69.0)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BEARINGS</th>
<th>INC</th>
<th>DISTANCE</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>PT</td>
<td>211.0</td>
<td>0.153</td>
<td>0.0</td>
</tr>
<tr>
<td>LI</td>
<td>224.6</td>
<td>1.367</td>
<td>0.3</td>
</tr>
<tr>
<td>MI</td>
<td>73.3</td>
<td>1.00</td>
<td>1.0</td>
</tr>
<tr>
<td>PB</td>
<td>120.8</td>
<td>1.186</td>
<td>0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BEARING TO CMT ELLIPSE</th>
<th>DIST TO ELLIPSE CMT</th>
<th>CMT TO CSE</th>
<th>SEARCH AREA</th>
<th>INCLUDED BEARINGS</th>
<th>FORWARD BEARINGS</th>
<th>CUCKED MAI SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>74.8</td>
<td>12.6</td>
<td>7.0</td>
<td>0.573</td>
<td>159.0</td>
<td>3</td>
<td>154,006</td>
</tr>
<tr>
<td>68.8</td>
<td>28.8</td>
<td>79.1</td>
<td>0.752</td>
<td>230.0</td>
<td>4</td>
<td>232,104</td>
</tr>
<tr>
<td>Test Case NA 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Bearings: 8</td>
<td>CM 53-29-23N</td>
<td>85-50-21N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AREA:</td>
<td>41.0 x 31.6 (92.)</td>
<td>90.0 x 39.8 (90.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEARINGS</td>
<td>INC DISTANCE ERROR</td>
<td>INC DISTANCE ERROR</td>
<td>INC DISTANCE ERROR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MR 105.9</td>
<td>0</td>
<td>133.4</td>
<td>13.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WR 51.2</td>
<td>10</td>
<td>106.6</td>
<td>10.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MA 54.9</td>
<td>0</td>
<td>106.6</td>
<td>10.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB 26.3</td>
<td>0</td>
<td>106.6</td>
<td>10.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB 120.0</td>
<td>0</td>
<td>106.6</td>
<td>10.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEARING TO CIRCLE</td>
<td>DIST TO CM</td>
<td>DIST TO DE/DE</td>
<td>SEARCH AREA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>125.7</td>
<td>30.6</td>
<td>13.6</td>
<td>2789.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80.3</td>
<td>49.6</td>
<td>880.6</td>
<td>11265.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>213.0</td>
<td>43.6</td>
<td>239.0</td>
<td>10085.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INCLUDED BEARINGS</td>
<td>FORWARD BEARINGS</td>
<td>COCKED MAT SIZE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>26.191</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>9.701</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1983.531</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### HFDF/IN (06/18/79)

<table>
<thead>
<tr>
<th>Station</th>
<th>Code</th>
<th>Value</th>
<th>Test Case No.</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td></td>
<td></td>
<td>100</td>
<td>105.00</td>
</tr>
<tr>
<td>200</td>
<td>PR</td>
<td>48.9</td>
<td></td>
<td>123.0</td>
</tr>
<tr>
<td>300</td>
<td>VC</td>
<td>49.2</td>
<td></td>
<td>-124.0</td>
</tr>
<tr>
<td>400</td>
<td>WR</td>
<td>45.4</td>
<td></td>
<td>-190.1</td>
</tr>
<tr>
<td>500</td>
<td>SR</td>
<td>34.0</td>
<td></td>
<td>-97.5</td>
</tr>
<tr>
<td>600</td>
<td>BR</td>
<td>26.0</td>
<td></td>
<td>25.8</td>
</tr>
<tr>
<td>700</td>
<td>MI</td>
<td>25.8</td>
<td></td>
<td>-94.0</td>
</tr>
<tr>
<td>800</td>
<td>NF</td>
<td>35.7</td>
<td></td>
<td>-96.2</td>
</tr>
<tr>
<td>900</td>
<td>LI</td>
<td>41.0</td>
<td></td>
<td>-72.0</td>
</tr>
<tr>
<td>1000</td>
<td>PI</td>
<td>47.2</td>
<td></td>
<td>-68.2</td>
</tr>
</tbody>
</table>