[redacted]

DECLASSIFIED

NRL Report 5302

[redacted]

HF-DF FIXING

[Unclassified Title]

B. Wald and W. D. Googe

Countermeasures Branch
Radio Division

UNCLASSIFIED

[redacted]

May 22, 1959

[redacted]

UNCLASSIFIED

U. S. NAVAL RESEARCH LABORATORY
Washington, D.C.

[redacted]

DECLASSIFIED

[redacted]

[redacted]

DECLASSIFIED

NRL Report 5302

UNCLASSIFIED

HF-DF FIXING

[Unclassified Title]

HF-DF FIXING ON A SMALL DIGITAL COMPUTER

[redacted]]

B. Wald and W. D. Googe

Countermeasures Branch
Radio Division

May 22, 1959

UNCLASSIFIED

U. S. NAVAL RESEARCH LABORATORY
Washington, D.C.

[redacted]

DECLASSIFIED

CONTENTS

ABSTRACT

[redacted]

At the present time, U. S. techniques for the processing of high-frequency direction finder (hf-df) data lag far behind the collection capabilities of the U. S. Navy hf-df equipment. One striking deficiency is the lack of an automatic df fixing method. Although a special purpose digital computer has been developed for this purpose in the United Kingdom, it has been deemed desirable to explore the feasibility of fix computation on an inexpensive drum-type general purpose digital computer.

The input to the computer consists of a set of bearings on the target and the estimated variance associated with each of these bearings; the desired output consists of a "best point estimate" of the target location and a "search area" which is the smallest area having some predetermined fiducial probability of containing the actual target locations.

The solution of the equations that explicitly yield the best point estimate would be difficult on a large computer, and completely impractical on a small drum machine. Instead, a series expansion and approximation method is used which yields a procedure that can be applied iteratively. In order not to strain the weak convergence properties of this method, another procedure is necessary to compute an initial point for the series expansion procedure. This is accomplished by taking the weighted average of the vertices of the polygon formed by the reported bearing circles. The computation of an approximation to the search area is relatively simple once the series expansion method has yielded certain necessary parameters.

These procedures have been programmed for a LGP-30, one of the simplest and most inexpensive of the general purpose digital computers. Best point estimates and search areas are produced in times commensurate with the tasking rates of df nets. Although certain refinements in the program would probably be desirable, the feasibility of df fixing on such a minimal computer has been demonstrated.

In view of the comparatively modest cost of the type computer proposed it is felt the Navy should seriously consider adopting machine computation of df fixes as a part of the ASW modernization program.

PROBLEM STATUS

This is an interim report; work on the problem is continuing.

AUTHORIZATION

NRL Problem RO6-06
Project NE 070-154-8
BuShips No. S-1283

Manuscript submitted March 5, 1959

HF-DF FIXING ON A SMALL DIGITAL COMPUTER

[redacted]

INTRODUCTION

In the years since World War II, significant advances have been made in U.S. hf-df equipment. Unfortunately, data processing techniques for the information produced by this equipment have not undergone comparable improvement. The Intercept and Data Handling Section of the Countermeasures Branch at NRL has therefore become involved in the development of modern df data processing techniques.

The general df data handling problem is discussed elsewhere.* This report is confined to the description of research directed toward determining the feasibility of df fixing with a small, general purpose digital computer.

One of the most striking deficiencies in present df practice is the lack of a fixing method superior to the "string plot" of two decades ago. The feasibility of df fixing on a digital computer has been forcefully demonstrated by the development in the United Kingdom of a special purpose digital computer for this task.† Although the performance of this computer has been eminently satisfactory, the employment of a small, general purpose computer would grant a ten-to-one cost advantage and a reliability advantage of a similar order of magnitude.

STATEMENT OF THE FIXING PROBLEM

For any fixing task, the input to the computer consists of a set of reports from the stations comprising the df net. Each station will either report its inability to take a bearing on the target transmitter or else report a measured bearing and an estimate of the reliability of this measurement. The computer is assumed to contain information as to the locations of the df stations.

The desired output consists of the most probable location of the target and a measure of the reliability of this computed location. This location will be referred to as the "best point estimate" or BPE.‡ A useful mode of specification of the utility of the BPE is to designate a search area which may be defined as the smallest area on the surface of the earth which has some predetermined fiducial probability of containing the actual target location.

* B. Wald and R.D. Misner, "Bearing Readout Systems for Goniometer Direction Finders," NRL Memorandum Report 830 ([redacted]), Aug. 1958; B. Wald, "HF-DF Data Transmission and Processing," NRL Report in preparation

† L.H.F. Nichols, "153 Computer D.F. Plotting System," Admiralty Research Laboratory, Report A.R.L./R2/M 4.53 ([redacted] Mar. 1957

‡ Many mathematicians would object to calling the BPE a "most probable point." A more satisfactory definition of the BPE might be "that point which maximizes the fiducial probability that a very small circle centered about it contains the actual target location."

MATHEMATICAL APPARATUS

Location of Points on the Surface of the Earth

Three general methods will be employed to describe the location of a point on the surface of the earth. The first is the conventional description by latitude and longitude. Latitude will be denoted by α and longitude by β with measurements of both quantities made in degrees. North latitudes and east longitudes will be considered positive.

The second method of location employs the Cartesian coordinates x, y, z. The coordinate system is assumed to be right-handed, centered at the center of the earth, and so oriented that the z axis passes through the North Pole and the x axis through the point $(0^\circ, 0^\circ)$. The earth is assumed to be spherical and of unit radius, so that for any point on its surface

$$x^2 + y^2 + z^2 = 1.$$

The connection between these two notations can be deduced by inspecting Fig. 1 and using the simple trigonometric relations to yield

$$x = \cos \alpha \cos \beta, \quad y = \cos \alpha \sin \beta, \quad z = \sin \alpha .$$

The third method of location, which is applicable only to the Northern Hemisphere, involves imagining a plane tangent to the earth at the North Pole. As is shown in Fig. 2, the point (x, y, z) can be characterized by passing a line from $(0, 0, 0)$ through (x, y, z) and noting the coordinates $(r, s, 1)$ of the intersection of this line and the tangent plane.

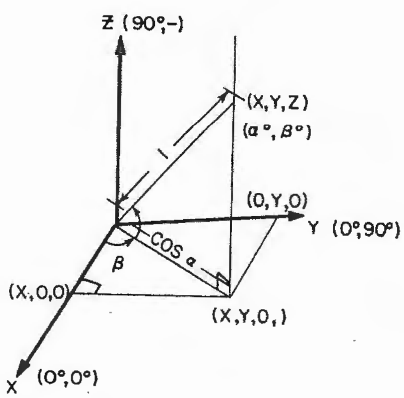


Fig. 1 - Geometry of point location

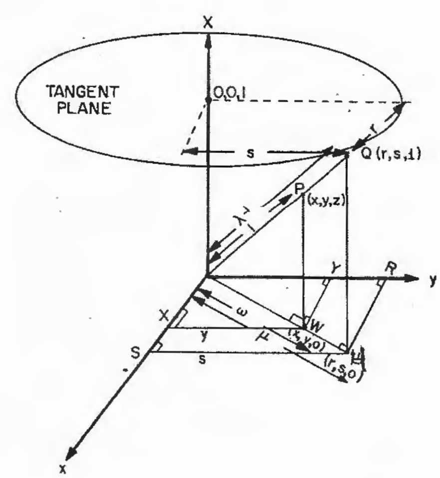


Fig. 2 - Projection on tangent plane

DECLASSIFIED

[redacted]

UNCLASSIFIED

Referring to Fig. 2, the length of the line segment OW, which is equal to $(x^2 + y^2)^{1/2}$, is denoted by ω, while the length of the line segment OU, which is equal to $(r^2 + s^2)^{1/2}$, is denoted by μ. The length of the line segment OQ, which is equal to $(1 + r^2 + s^2)^{1/2}$ is denoted by $1/\lambda$. Then, since $\triangle OPW \sim \triangle OQU$, $1 : 1/\lambda = z : 1 = \omega : \mu$. But, since $\triangle OWY \sim \triangle OUR$ and $\triangle OWX \sim \triangle OUS$, $\omega : \mu = x : r = y : s$. Then, combining these relations,

$$x = r\lambda, \quad y = s\lambda, \quad z = \lambda.$$

Concept of the Bearing Plane

Let τ_i be the bearing measured in the clockwise direction from north at the station located at (x_i, y_i, z_i). This bearing defines a direction circle which is the great circle passing through the station and making the angle τ_i with north. The bearing plane is defined as that plane whose intersection with the surface of the sphere is the bearing circle.

Two points on this plane are $(0, 0, 0)$ and (x_i, y_i, z_i) so only one other (x_i', y_i', z_i'), is needed to completely specify the plane. Any point lying on the direction circle except (x_i, y_i, z_i) and its antipode could be used. It should be noted that the plane produced by the bearing τ_i is identical to that produced by the bearing τ_i - 180°. It will be convenient to insist that z_i' be positive. Finally (x_i', y_i', z_i'), should not lie too close to (x_i, y_i, z_i) to avoid loss of significance in further calculations. These requirements are met by choosing (x_i', y_i', z_i') to be that point which lies on the direction circle to the north of (x_i, y_i, z_i) and whose longitude differs from that of (x_i, y_i, z_i) by 90°.

When τ_i is less than 90°, the situation depicted in Fig. 3 prevails. Napier's rules* immediately yield

$$\cos \delta_i = \cos(90° - \alpha_i) \sin \tau_i = \sin \alpha_i \sin \tau_i$$

$$\sin \delta_i = (1 - \cos^2 \delta_i)^{1/2} = (1 - \sin^2 \alpha_i \sin^2 \tau_i)^{1/2}.$$

Now, from the law of sines,

$$\sin C_i / \sin \tau_i = \sin(90° - \alpha_i)/\sin \delta_i$$

or

$$\sin C_i = \sin \tau_i \cos \alpha_i / \sin \delta_i$$

and

$$\cos C_i = (1 - \sin^2 C_i)^{1/2}.$$

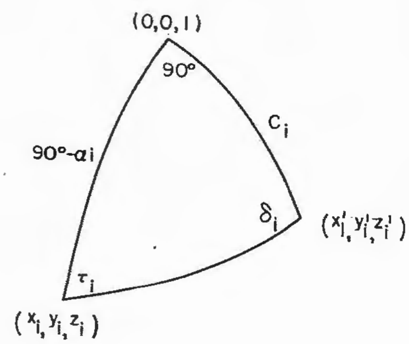


Fig. 3 - Right spherical triangle for bearings less than 90°

* See, for example, R.W. Brink, "Plane and Spherical Trigonometry," New York: Appleton-Century, pp. 8ff. (Book 2), 1942

[redacted]

DECLASSIFIED

Since (x_i', y_i', z_i') is in the Northern Hemisphere, C_i is less than 90° and its sine and cosine must be positive. This demands the use of the positive square roots in the expressions for $\cos C_i$ and $\sin \delta_i$.

For the point (x_i', y_i', z_i'), $\alpha_i' = 90° - C_i$ and $\beta_i' = \beta_i + 90°$. It therefore follows that

$$x_i' = \cos(90° - C_i)\cos(\beta_i + 90°) = -\sin C_i \sin \beta_i$$

$$y_i' = \cos(90° - C_i)\sin(\beta_i + 90°) = \sin C_i \cos \beta_i$$

$$z_i' = \sin(90° - C_i) = \cos C_i .$$

When τ lies between 180° and 270°, the reciprocal bearing, $\tau_i' = \tau_i - 180°$, is employed. The geometry is again that of Fig. 3 and the equations for x_i', y_i', and z_i' are those derived above with τ_i' replacing τ_i.

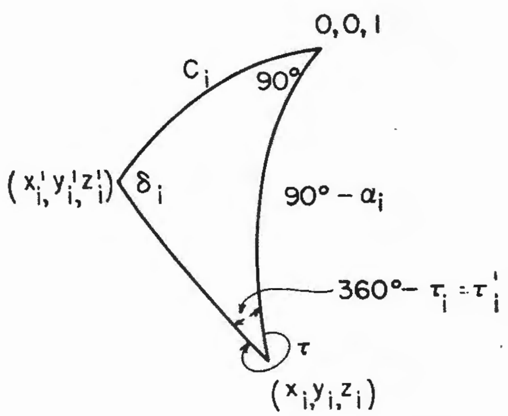


Fig. 4 - Right spherical triangle for bearings greater than 270°

When τ_i lies between 270° and 360°, the situation shown in Fig. 4 prevails. It should be noted that τ_i is the exterior angle at (x_i, y_i, z_i). If the interior angle $360° - \tau_i$ is denoted by τ_i', the same equations derived above can be used to find C, provided that τ_i' is substituted for τ_i. In this case, however, $\beta_i' = \beta_i - 90°$, so that

$$x_i' = \cos(90° - C_i)\cos(\beta_i - 90°) = \sin C_i \sin \beta_i$$

$$y_i' = \cos(90° - C_i)\sin(\beta_i - 90°) = -\sin C_i \cos \beta_i$$

$$z_i' = \sin(90° - C_i) = \cos C_i .$$

When τ_i lies between 90° and 180°, the reciprocal bearing, $\tau_i + 180°$, is used in the same analysis. In this case, however,

$$\tau_i' = 360° - (\tau_i + 180°) = 180° - \tau_i .$$

The overall procedure for finding (x_i', y_i', z_i') can be summarized as follows:

$0° \le \tau_i < 90°$	$\tau_i' = \tau_i$	use procedure A
$90° \le \tau_i < 180°$	$\tau_i' = 180° - \tau_i$	use procedure B
$180° \le \tau_i < 270°$	$\tau_i' = \tau_i - 180°$	use procedure A
$270° \le \tau_i < 360°$	$\tau_i' = 360° - \tau_i$	use procedure B

[redacted]

[redacted]

where procedures A and B are as follows. For both procedures

$$\cos \delta_i = \sin \alpha_i \sin \tau_i'$$

$$\sin \delta_i = +(1 - \cos^2 \delta_i)^{1/2}$$

$$\sin C_i = \sin \tau_i' \cos \alpha_i / \sin \delta_i$$

$$\cos C_i = +(1 - \sin^2 C_i)^{1/2} .$$

For procedure A let $Q_i = \sin C_i$, and for procedure B let $Q_i = -\sin C_i$; then

$$x_i' = -Q_i \sin \beta_i$$

$$y_i' = Q_i \cos \beta_i$$

$$z_i' = \cos C_i .$$

A fully equivalent procedure somewhat more useful for automatic computation involves

$$\tau_i'' = \tau_i - n90^\circ$$

where n is an integer such that

$$-90^\circ \le \tau_i'' < 0^\circ .$$

Then, if

$n = 1$	$\sin \tau_i' = \cos \tau_i''$	use procedure A
$n = 2$	$\sin \tau_i' = -\sin \tau_i''$	use procedure B
$n = 3$	$\sin \tau_i' = \cos \tau_i''$	use procedure A
$n = 4$	$\sin \tau_i' = \cos(\tau_i'' + 90^\circ)$	use procedure B .

Having found the location of (x_i', y_i', z_i'), it remains to determine the bearing plane. The general equation of a plane passing through the origin of coordinates is

$$ax + by + cz = 0.$$

[redacted]

For a plane passing through (x_i, y_i, z_i) and (x_i', y_i', z_i') the coefficients are given by

$$a_i = z_i y_i' - z_i' y_i$$

$$b_i = x_i z_i' - x_i' z_i$$

$$c_i = y_i x_i' - y_i' x_i.$$

These expressions may be verified by substituting the three known points into the equation for the bearing plane.

Intersection and Distance Formulas

The BT Lines - The lines formed by the intersections of the bearing planes with the tangent plane will be designated as BT lines. The general equation of a bearing plane is

$$a_i x + b_i y + c_i z = 0$$

while the tangent plane is that plane for which $z = 1$ (the plane tangent to the earth at the North Pole). Then using the notation (r, s) for a point $(r, s, 1)$ in the tangent plane, the equation for the BT line is simply

$$a_i r + b_i s = -c_i \; .$$

Intersections of BT Lines - In general, each station will generate a different bearing plane and hence a different BT line. If one station is designated by the subscript i and another by the subscript j, the equations of the two bearing planes are

$$a_i x + b_i y + c_i z = 0$$

$$a_j x + b_j y + c_j z = 0$$

and the equations of the BT lines are

$$a_i r + b_i s = -c_i$$

$$a_j r + b_j s = -c_j \; .$$

The point of intersection of the two BT lines may be denoted as (r_{ij}, s_{ij}) and its coordinates may be found by solving the two equations simultaneously to yield

$$r_{ij} = (c_j b_i - c_i b_j)/(a_i b_j - a_j b_i)$$

$$s_{ij} = (c_i a_j - c_j a_i)/(a_i b_j - a_j b_i) \; .$$

[redacted]

The significance of this intersection may be appreciated by recalling that the bearing plane passes through the center of the earth and through the direction circle. The BT line is therefore the projection of the direction circle on the tangent plane. It follows, then, that the intersection of two BT lines is the projection of the intersection of the corresponding two direction circles.

Distance from Point on Tangent Plane to Bearing Plane - The general expression for the distance from a plane $a_i x + b_i y + c_i z = 0$ to a point (X, Y, Z) is

$$d_i = (a_i X + b_i Y + c_i Z)/(a_i^2 + b_i^2 + c_i^2)^{1/2}.$$

In the special case that the plane is a bearing plane and the point lies on the tangent plane, the symbol Δ is applied to the quantity $a^2 + b^2 + c^2$ and the notation $h_i(r,s)$ is used for the distance, yielding

$$h_i(r,s) = (a_i r + b_i s + c)/\Delta^{1/2} .$$

Distance from Point on Tangent Plane to S Line - The line passing through the origin and the station will be denoted as the S line. It will be recalled that this line passes through the point $(r_i, s_i, 1)$ as well as through the station, i.e., the point (x_i, y_i, z_i) or $(r_i\lambda_i, s_i\lambda_i, \lambda_i)$. Since the S line passes through the origin and the station it must lie in the bearing plane. The parametric equations for the S line are

$$x = x_i k, \quad y = y_i k, \quad z = z_i k \quad .$$

Now let $(r,s,1)$ be a general point in the tangent plane. The distance from the point to the S line will be denoted as $g_i(r,s)$.

To find g_i, first write the expression for the square of the distance from $(r,s,1)$ to a general point on the S line:

$$d_i^2 = (x_i k - r)^2 + (y_i k - s)^2 + (z_i k - 1)^2 .$$

Since the minimum value of d_i^2 is g_i^2, the value of k is found by setting

$$d(d^2)/dk = 2x_i(x_i k - r) + 2y_i(y_i k - s) + 2z_i(z_i k - 1)$$

equal to zero, yielding

$$k = (x_i r + y_i s + z_i)/(x_i^2 + y_i^2 + z_i^2) = x_i r + y_i s + z_i \quad .$$

[redacted]

Substituting this value in the expression for d_i^2 yields

$$g_i^2 = [(x_i^2 - 1)r + x_i y_i s + x_i z_i]^2 + [x_i y_i r + (y_i^2 - 1)s + y_i z_i]^2 + [x_i z_i r + y_i z_i s + (z_i^2 - 1)]^2$$

$$= r^2[x_i^2(x_i^2 + y_i^2 + z_i^2 - 2) + 1] + s^2[y_i^2(x_i^2 + y_i^2 + z_i^2 - 2) + 1]$$

$$+ z_i^2(x_i^2 + y_i^2 + z_i^2 - 2) + 1 + 2rsx_i y_i(x_i^2 - 1 + y_i^2 - 1 + z_i^2)$$

$$+ 2rx_i z_i(x_i^2 - 1 + y_i^2 + z_i^2 - 1) + 2sy_i z_i(x_i^2 + y_i^2 - 1 + z_i^2 - 1)$$

$$= r^2(1 - x_i^2) + s^2(1 - y_i^2) + 1 - z_i^2 - 2rsx_i y_i - 2rx_i z_i - 2sy_i z_i \ .$$

Rotation of Coordinates

Given a point (x, y, z) and its projection (r, s), it will often be found necessary to rotate the coordinate system so that the point (x, y, z) becomes $(0, 0, 1)$ in the new coordinate system. The necessary transformation equations are

$$\bar{x} = sx/\mu - ry/\mu$$

$$\bar{y} = \lambda rx/\mu + \lambda sy/\mu - \lambda\mu z$$

$$\bar{z} = \lambda rx + \lambda sy + \lambda z \quad .$$

It should be remarked that there is no unique transformation as there is an unspecified degree of freedom; i.e., these equations perform a rotation about the $\bar{z}$ axis after the point (x, y, z) is brought to the new North Pole by the shortest route. The validity of the transformation as a rotation of coordinates can be verified by noting that the six standard relations among the direction cosines are satisfied.

If it is desired to transform the coordinates of a point from the new to the old system, the applicable equations are

$$x = s\bar{x}/\mu + \lambda r\bar{y}/\mu + \lambda r\bar{z}$$

$$y = -r\bar{x}/\mu + \lambda s\bar{y}/\mu + \lambda s\bar{z}$$

$$z = -\lambda\mu\bar{y} + \lambda\bar{z} \ .$$

If a number of transformations are to be performed, it is instructive to consider the matrix

$$A = \begin{bmatrix} s/\mu & \lambda r/\mu & r\lambda \\ -r/\mu & \lambda s/\mu & s\lambda \\ 0 & -\lambda\mu & \lambda \end{bmatrix} .$$

[redacted]

Then, if P is the row vector $[x, y, z]$,

$$\bar{P} = PA \text{ and } P = \bar{P}A^{-1}$$

where the inverse of this matrix is simply its transpose A'. Suppose three transformations A_1, A_2, and A_3 are to be performed in that order. Then,

$$\bar{P} = PA_1A_2A_3 \text{ and } P = \bar{P}A_3^{-1}A_2^{-1}A_1^{-1} \quad .$$

The latter equation implies that the inverse transformations should be applied in the reverse order of the original transformations. If the transformations are intermediate results of computer calculations, this would seem to require that the intermediate results be stored throughout the calculation. Actually all that is necessary is to multiply each new inverse as it is calculated from the <u>left</u> on to the product of the previous inverses. If this is done only one inverse matrix need be stored and the retransformation can be accomplished by multiplying the inverse product from the right on to $\bar{P}$.

CALCULATION OF THE BPE

Polygon Center Method

If only two stations report bearings, the BPE is simply the intersection of their direction circles. A method of finding this point of intersection by determining the intersection of the BT lines has already been discussed.

If n stations report bearings, there will be $(n^2 - n)/2$ intersections. At first glance it might be thought that the BPE could be found by somehow averaging the coordinates of these intersections. The fallacy of this idea is exposed in Fig. 5. For three stations there are three intersections, 1,2; 1,3; and 2,3. Averaging the intersections would place the BPE in the vicinity of A. However the BPE lies in the vicinity of B. Note that if the target actually were at B, the observed bearings would have relatively small errors; but if it were at A, station 3 would have an unreasonably large error.

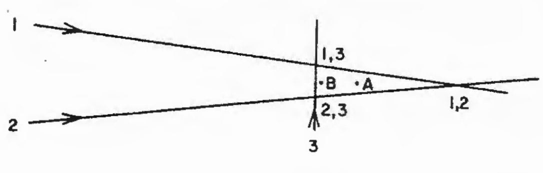


Fig.5 - A polygon center which is not a BPE

Nevertheless, this method quickly locates the "best" center of the polygon of error or "cocked hat." Although it cannot produce a BPE, it can provide an initial point from

[redacted]

which the more refined calculations can start. A center of the polygon can be determined in terms of the means of the r_{ij}'s and s_{ij}'s (defined in the discussion of BT lines) as

$$r = \left(\frac{2}{n^2 - n}\right) \sum_{\substack{i,j=1 \\ j>i}}^{n} r_{ij}$$

$$s = \left(\frac{2}{n^2 - n}\right) \sum_{\substack{i,j=1 \\ j>i}}^{n} s_{ij} .$$

If each observation is associated with an estimated variance σ_i^2, a more accurate determination of the polygon center might be

$$r = \left(\sum_{\substack{i,j=1 \\ j>i}}^{n} \frac{r_{ij}}{\sigma_i^2 \sigma_j^2}\right) \left(\sum_{\substack{i,j=1 \\ j>i}}^{n} \frac{1}{\sigma_i^2 \sigma_j^2}\right)^{-1}$$

$$s = \left(\sum_{\substack{i,j=1 \\ j>i}}^{n} \frac{s_{ij}}{\sigma_i^2 \sigma_j^2}\right) \left(\sum_{\substack{i,j=1 \\ j>i}}^{n} \frac{1}{\sigma_i^2 \sigma_j^2}\right)^{-1}$$

Two remarks should be made in connection with this latter expression. First, the weighting factor used is far from ideal. A better weighting factor might well be $(\sigma_i \sigma_j)^{-1}$, but it is assumed that the input to the computer will be in the form of inverse variances. Thus, the use of the latter weighting factor would involve taking n square roots. Second, it is not worth while to devote too much computer time to this process, for even if an ideal weighting factor could be found, the result would still be subject to the geometrical biases of the type illustrated in Fig. 5.

Series Expansion Method

Distribution Assumptions - It will be assumed that the error at each station, i.e., the difference between the measured bearing and the true bearing to the target, has a normal probability distribution with zero mean and a variance of σ_i^2. The assumption of zero mean implies simply that systematic errors of the type usually expressed in a calibration chart have been corrected at some prior stage of data processing.

[redacted]

In practice, it is found* that the distribution is not normal. At this point, students of the subject engage in violent disagreements, some maintaining that the distribution is inherently not Gaussian and others maintaining with equal vigor that it consists of a Gaussian distribution of smaller variance than that calculated by the straightforward method superimposed on a base which is linearly distributed or else normally distributed with a large variance. If the latter view is accepted, Gaussian statistics should be used and errors of several times the sample standard deviation rejected as coming from the base.

The latter philosophy will be employed, not out of any deep conviction for its theoretical soundness but merely because of its simpler implementation. The general method will be to assume Gaussian statistics and then to reject data which is likely to have come from the tails of the distribution.

The other major assumption is that the error at any station is independent of the errors at the other stations. When it is considered that as instrumental errors are decreased such effects as ionospheric tilt become relatively larger contributors to the total error, it is not unreasonable to suppose that there might be a positive correlation between the errors at nearby stations. Indeed, recent work in the United Kingdom† would seem to indicate that such a positive correlation exists, although its cause is not definitely established. Fortunately, this correlation has only a minor effect on the size of the search area, and it is reasonable to expect that in the U.S. nets with their longer baselines the correlation would be less significant.

Distribution Function - Under these assumptions it is possible to write the distribution function. The error at the ith station will be denoted as η_i, making the unnormalized distribution function

$$e^{-\eta_i^2/2\sigma_i^2} .$$

By this is meant that if a bearing of τ_i is observed, the probability that the true bearing lies between a and b is given by

$$\frac{\int_{a-\tau_i}^{b-\tau_i} e^{-\eta_i^2/2\sigma_i^2}\, d\eta_i}{\int_{-\infty}^{\infty} e^{-\eta_i^2/2\sigma_i^2}\, d\eta_i} .$$

*J.J. Cummings and H.J. Davis, "Accuracy of High Frequency Radio Direction Finder Bearings Against Operational Targets" (Secret), Office of Chief of Naval Operations, OP 202, Oct. 28, 1954; but see also a possible explanation in "Factor Analysis of HF DF Bearing Accuracy - Atlantic Net - 1954" (Secret), Royal Canadian Navy Directorate of Supplementary Radio Activities, Feb. 1957

†P.J.D. Gething, "DF Plotting and Analysis, Further Investigation of Probability Rectangles," General Communications Headquarters, Cheltenham, England, Reference MA/1427/117/4 (Secret), June 5, 1958

[redacted]

Under the assumption of independence of errors it is possible to write a joint distribution function by simply multiplying the individual functions

$$\prod_{i=1}^{n} e^{-\eta_i^2/2\sigma_i^2} = e^{-\frac{1}{2}\sum_{i=1}^{n}\frac{\eta_i^2}{\sigma_i^2}}$$

Then, recalling that $\eta_i(x,y,z)$ is equal to τ_i minus the bearing from (x_i, y_i, z_i) to (x,y,z), i.e., the hypothetical error if (x,y,z) were the target location, the overall probability density function can be written which is defined for all points on the sphere except the stations and their antipodes, namely,

$$f(x,y,z) = e^{-\frac{1}{2}\sum_{i=1}^{n}\frac{[\eta_i(x,y,z)]^2}{\sigma_i^2}}$$

It should be noted that although it has been tacitly assumed that (x,y,z) lies on the surface of the sphere, the bearing to any point, and hence the value of the density function, is the same to any point lying on the line through the given point and the center of the sphere. Therefore the value of $f(x,y,z)$ is equal to the value of $f(r,s,1)$ which can be written as $f(r,s)$. The BPE can be found by maximizing this function and projecting $(r,s,1)$ back onto the surface of the sphere.

Geometry of the Bearing Error - The basis of the fixing process is illustrated in Fig. 6. Suppose a certain bearing is observed at the station located at (x_i, y_i, z_i). This bearing determines a bearing plane and a bearing circle at the intersection of this plane and the surface of the sphere. Now let (x,y,z) be the target point on the surface of the sphere whose projection onto the $z = 1$ plane is $(r,s,1)$. A direction circle may be drawn to (x,y,z) and a new plane, is determined.

What is desired is the angle η_i between the two direction circles. Since the direction circles lie on the surface of the sphere, they must both be normal to the s line at the station, since this line is normal to the surface. The s line, however, is the line of intersection of the two planes in which the direction circles lie. Now if two lines, each lying in a plane, meet at the intersection of the planes and both lines are normal to the line of intersection, it follows from a fundamental theorem of solid geometry that the angle between the lines is the angle between the planes.

Next, turning attention to the middle one of the three small triangles, the shortest leg of this triangle has been formed by dropping a perpendicular from $(r,s,1)$ to the bearing plane and the longest leg has been formed by dropping a perpendicular from $(r,s,1)$ to the s line. This line obviously lies in the plane formed by $(0,0,0)$, (x,y,z), and (x_i, y_i, z_i). In order for the triangle to close, the third leg must be normal to the s line, and therefore the angle ρ is equal to the angle between the planes and hence to η_i. The angle ρ, however, can be expressed in terms of the previously derived lengths of the legs of the triangle.

[redacted]

[redacted]

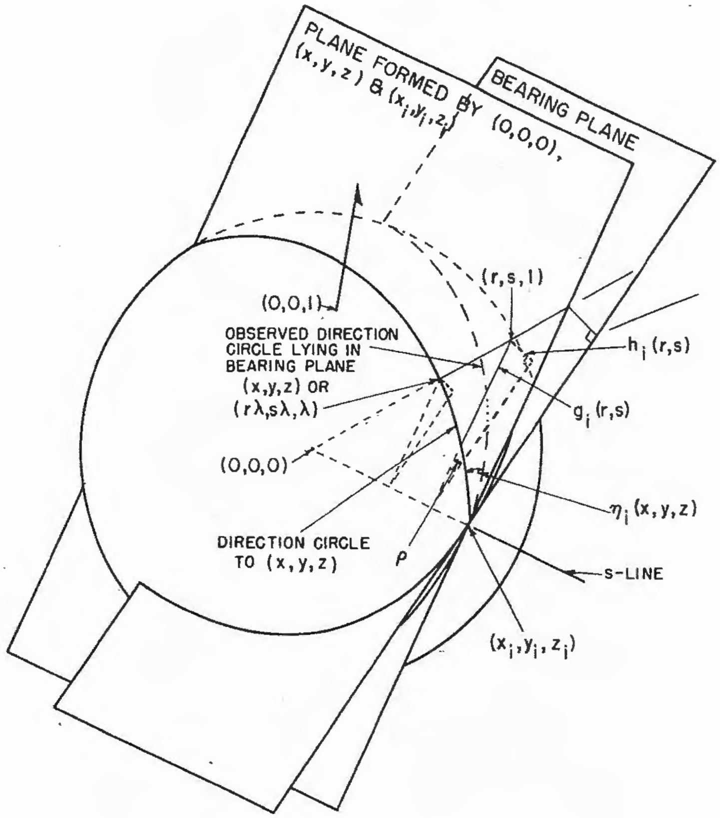


Fig. 6 - Geometry of bearing error

Therefore

$$\eta_i = \sin^{-1} \frac{h_i(r,s)}{g_i(r,s)}$$

and, after substitution of this expression, the probability density function becomes

$$f(r,s) = e^{-\frac{1}{2} \sum_{i=1}^{n} \left[\sin^{-1} \frac{h_i(r,s)}{g_i(r,s)}\right]^2 \frac{1}{\sigma_i^2}}$$

[redacted]

Approximations and Expansion - In theory, the BPE could be found by taking the partial derivatives of $f(r,s)$ with respect to r and s and setting them equal to zero to find r_o and s_o, the values of r and s that maximize $f(r,s)$. Unfortunately the expressions are rather complex, so some approximations are necessary. The approximations that will be made are based on the assumption that η_i, r_o, and s_o are all small. The first approximation will be to replace $\sin^{-1}\eta_i$ by η_i. For small errors this replacement is accurate to within a few parts in ten thousand. For an error of 4°, the replacement is accurate only to about one part in one thousand, but stations likely to have errors much larger than this will be heavily discriminated against in the weighting process.

The assumption that r_o and s_o are small appears indefensible until it is recalled that a coordinate rotation scheme was developed. After each calculation of the BPE a rotation will be performed that will bring the BPE to the North Pole. Repetition of the calculation should result in asymptotic refinement of the BPE. In order to ensure a reasonably small r_o and s_o for the first calculation, the polygon center method can be used to obtain the parameters of the first rotation.

Under the assumption of small η_i,

$$f(r,s) = e^{-\frac{1}{2}\sum_{i=1}^{n}\frac{1}{\sigma_i^2}\left[\frac{h_i(r,s)}{g_i(r,s)}\right]^2}.$$

Now, recalling the previously derived values for $h_i(r,s)$ and $g_i^2(r,s)$,

$$\left[\frac{h_i(r,s)}{g_i(r,s)}\right]^2 = \frac{(a_i r + b_i s + c_i)^2}{\Delta[r^2(1 - x_i^2) + s^2(1 - y_i^2) + 1 - z_i^2 - 2x_iy_irs - 2x_iz_ir - 2y_iz_is]}$$

$$= \frac{c_i^2 + 2a_ic_ir + 2b_ic_is + 2a_ib_irs + a_i^2r^2 + b_i^2s^2}{\Delta(1 - z_i^2)(1 + k_{i10}r + k_{i01}s + k_{i11}rs + k_{i20}r^2 + k_{i02}s^2)}$$

where

$$k_{i10} = -2x_iz_i/(1 - z_i^2),\quad k_{i01} = -2y_iz_i/(1 - z_i^2)$$

$$k_{i11} = -2x_iy_i/(1 - z_i^2),\quad k_{i20} = (1 - x_i^2)/(1 - z_i^2)$$

$$k_{i02} = (1 - y_i^2)/(1 - z_i^2).$$

[redacted]

But, considering terms up to quadratics,

$$(1 + k_{i10}r + k_{i01}s + k_{i11}rs + k_{i20}r^2 + k_{i02}s^2)^{-1}$$

$$= 1 - k_{i10}r - k_{i01}s + (2k_{i10}k_{i01} - k_{i11})rs + (k_{i10}{}^2 - k_{i20})r^2 + (k_{i01}{}^2 - k_{i02})s^2$$

and therefore

$$\left[\frac{h_i(r,s)}{g_i(r,s)}\right]^2 = a_{i10}r + a_{i01}s + a_{i20}r^2 + a_{i02}s^2 + a_{i11}rs + a_{i00}$$

where

$$a_{i10} = \frac{1}{\Delta(1 - z_i^2)}(2a_ic_i - c_i^2k_{i10}) = \frac{2}{\Delta(1 - z_i^2)^2}\left[a_ic_i(1 - z_i^2) + x_iz_ic_i^2\right]$$

$$a_{i01} = \frac{1}{\Delta(1 - z_i^2)}(2b_ic_i - c_i^2k_{i01}) = \frac{2}{\Delta(1 - z_i^2)^2}\left[b_ic_i(1 - z_i^2) + y_iz_ic_i^2\right]$$

$$a_{i20} = \frac{1}{\Delta(1 - z_i^2)}\left[a_i^2 - 2a_ic_ik_{i10} + c_i^2(k_{i10}{}^2 - k_{i20})\right]$$

$$= \frac{1}{\Delta(1 - z_i^2)^3}\left[a_i^2(1 - z_i^2)^2 + (4a_ic_ix_iz_i + c_i^2x_i^2 - c_i^2)(1 - z_i^2) + 4c_i^2x_i^2z_i^2\right]$$

$$a_{i02} = \frac{1}{\Delta(1 - z_i^2)}\left[b_i^2 - 2b_ic_ik_{i01} + c_i^2(k_{i01}{}^2 - k_{i02})\right]$$

$$= \frac{1}{\Delta(1 - z_i^2)^3}\left[b_i^2(1 - z_i^2)^2 + (4b_ic_iy_iz_i + c_i^2y_i^2 - c_i^2)(1 - z_i^2) + 4c_i^2y_i^2z_i^2\right]$$

$$a_{i11} = \frac{1}{\Delta(1 - z_i^2)}\left[2a_i b_i - 2b_i c_i k_{i10} - 2a_i c_i k_{i01} + c_i^2(2k_{i10}k_{i01} - k_{i11})\right]$$

$$= \frac{2}{\Delta(1 - z_i^2)^3}\left[a_i b_i (1 - z_i^2)^2 + (2b_i c_i x_i z_i + 2a_i c_i y_i z_i + c_i^2 x_i y_i)(1 - z_i^2)\right.$$

$$\left. + 4c_i^2 x_i y_i z_i^2\right]$$

The value of a_{i00} is unimportant, since a derivative will be taken. It will now be convenient to define

$$t_{pq} = \sum_{i=1}^{n} a_{ipq}/\sigma_i^2 .$$

Then,

$$\sum_{i=1}^{n}\left[\frac{h_i(r,s)}{g_i(r,s)}\right]^2 \frac{1}{\sigma_i^2} = t_{10}r + t_{01}s + t_{20}r^2 + t_{02}s^2 + t_{11}rs + t_{00}$$

and the distribution function which is to be maximized becomes

$$f(r,s) = e^{-\frac{1}{2}\sum_{\substack{p,q=0 \\ p+q<3}}^{2} t_{pq} r^p s^q}$$

This function can be maximized by minimizing the exponent. The values of r and s (r_o and s_o) which minimize the exponent are found by taking the partial derivatives with respect to r and s and setting them equal to zero, yielding

$$t_{10} + 2t_{20}r + t_{11}s = 0$$

$$t_{01} + t_{11}r + 2t_{02}s = 0$$

and solving these equations simultaneously to yield

$$r_o = \frac{t_{11}t_{01} - 2t_{02}t_{10}}{4t_{20}t_{02} - t_{11}^2}$$

$$s_o = \frac{t_{11}t_{10} - 2t_{20}t_{01}}{4t_{20}t_{02} - t_{11}^2} \quad .$$

Summary

The procedure for the calculation of the BPE may be summarized as follows.

1. The station locations and bearings are used to compute the constants that determine the bearing plane.

2. The polygon center method is used to obtain an initial estimate of the BPE.

3. Coordinates are rotated to bring this estimate to the point $(0,0,1)$. The inverse of this rotation is multiplied by an inverse accumulator which is initially a unit matrix.

4. The series expansion method is used to find r_o and s_o. If these are both small, step 5 is used. Otherwise $(r_o\lambda, s_o\lambda, \lambda)$ becomes the new estimate and step 3 is repeated.

5. The accumulated inverse product is multiplied by $(r_o\lambda, s_o\lambda, \lambda)$ to yield the BPE.

CALCULATION OF THE SEARCH AREA

The search area has been defined as the smallest area on the surface of the earth which has some predetermined fiducial probability of containing the actual target location. From this definition it follows that the search area is bounded by a contour of constant value of the distribution function. The same approximations that were made in the calculation of the BPE will be made here. Thus the distribution function is given by

$$f(r,s) = e^{-\frac{1}{2}\sum\limits_{\substack{p,q=0 \\ p+q<3}}^{2} t_{pq} r^p s^q}$$

Setting this function equal to some constant k and taking the natural logarithm of the result yields the equation of the curve that bounds the search area:

$$t_{00} + t_{10}r + t_{01}s + t_{11}rs + t_{20}r^2 + t_{02}s^2 = -2 \ln k$$

which defines an ellipse.

[redacted]

It is convenient to apply the transformations

$$r' = r - r_o \qquad s' = s - s_o$$

yielding

$$t_{00} + t_{10}(r' + r_o) + t_{01}(s' + s_o) + t_{11}(r's' + r_os' + r's_o + r_os_o)$$

$$+ t_{20}(r'^2 + 2r'r_o + r_o^2) + t_{02}(s'^2 + 2s's_o + s_o^2) = -2 \ln k$$

or, collecting terms,

$$t_{00} + t_{20}r'^2 + t_{02}s'^2 + t_{11}r's' + r'(t_{10} + 2t_{20}r_o + t_{11}s_o)$$

$$+ s'(t_{01} + t_{11}r_o + 2t_{02}s_o) + t_{10}r_o + t_{01}s_o + t_{11}r_os_o$$

$$+ t_{20}r_o^2 + t_{02}s_o^2 = -2 \ln k \quad .$$

Noting that the two terms in parenthesis were proved to equal zero in the location of the BPE, and grouping all the other terms not containing r' and s' into one term, t_{00}', yields

$$t_{00}' + t_{20}r'^2 + t_{02}s'^2 + t_{11}r's' = -2 \ln k$$

which is the equation of an ellipse centered on $r' = 0$, $s' = 0$, i.e., on $r = r_o$, $s = s_o$.

The cross term can now be eliminated by applying the rotation

$$r' = (r'' + ps'')/(1 + p^2)^{1/2}$$

$$s' = (-pr'' + s'')/(1 + p^2)^{1/2}$$

where

$$p = (1/t_{11}) \left\{ -t_{02} + t_{20} \pm \left[(t_{02} - t_{20})^2 + t_{11}^2 \right]^{1/2} \right\}$$

in which the positive sign is chosen if $t_{20} - t_{02}$ is negative and the negative sign otherwise. This rotation yields

$$u_{00} + u_{20}r''^2 + u_{02}s''^2 = -2 \ln k$$

[redacted]

where

$$u_{20} = (1/1 + p^2)(t_{20} + p^2 t_{02} - p t_{11})$$

$$u_{02} = (1/1 + p^2)(t_{20} p^2 + t_{02} + p t_{11}) \,.$$

The cumulative probability that the target lies within the ellipse is given by

$$\frac{\displaystyle\iint_{u_{00} + u_{20} r''^2 + u_{02} s''^2 = -2 \ln k} e^{-\frac{1}{2}(u_{00} + u_{20} r''^2 + u_{02} s''^2)} \, dr'' ds''.}{\displaystyle\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-\frac{1}{2}[u_{00} + u_{20} r''^2 + u_{02} s''^2]} \, dr'' ds''}$$

$$= \frac{\displaystyle\iint_{u_{20} r''^2 + u_{02} s''^2 = -2 \ln k'} e^{-\frac{1}{2}(u_{20} r''^2 + u_{02} s''^2)} \, dr'' ds''}{\displaystyle\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-\frac{1}{2}[u_{20} r''^2 + u_{02} s''^2]} \, dr'' ds''}$$

where k' is such that $-2 \ln k' = u_{00} + 2 \ln k$.

The ellipse may now be transformed into a circle by making the substitutions

$$s'' = s'''(u_{02})^{-1/2} \qquad r'' = r'''(u_{20})^{-1/2}$$

yielding the cumulative probability

$$\frac{\displaystyle\iint_{r'''^2 + s'''^2 = -2 \ln k'} e^{-\frac{1}{2}(r'''^2 + s'''^2)} \, dr''' ds'''}{\displaystyle\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-\frac{1}{2}(r'''^2 + s'''^2)} \, dr''' ds'''} \,.$$

[redacted]

If polar axes are now inserted in the r''', s''' plane, and the 2π's resulting from the angular integration are cancelled, the probability function becomes

$$\frac{\int_0^{\rho_0} e^{-\frac{1}{2}\rho^2}\rho d\rho}{\int_0^{\infty} e^{-\frac{1}{2}\rho^2}\rho d\rho} = \frac{-e^{-\frac{1}{2}\rho^2}\Big|_0^{\rho_0}}{-e^{-\frac{1}{2}\rho^2}\Big|_0^{\infty}} = 1 - e^{-\frac{1}{2}\rho_0^2}$$

where the search area is a circle of radius ρ_0 in the dimensionless r''', s''' plane and $\rho_0^2 = -2 \ln k'$.

If it is desired that the search area include the target location with a probability of .9091, then

$$1 - e^{-\frac{1}{2}\rho_0^2} = .9091$$

$$\rho_0^2 = -2 \ln (1 - .9091) = 4.80$$

$$\rho_0 = 2.19 \ .$$

The final approximation is to replace this circle by a square of equal area. This square will define a search area of slightly lower probability since the area is not bounded by a curve of constant probability density. If the length of a side of the square is denoted by d, then

$$d^2 = 4.80\pi = 15.18 \text{ or } d = 3.897 \ .$$

Thus the square is bounded by the points

$$r''' = 1.95, s''' = 1.95$$

$$r''' = 1.95, s''' = -1.95$$

$$r''' = -1.95, s''' = -1.95$$

$$r''' = -1.95, s''' = 1.95 \ .$$

When this square is transformed to the r'', s'' plane, a rectangle centered on the origin results. When this rectangle is transformed to the r', s' plane, the rectangle is rotated but still centered on the origin. When the rotated rectangle is transformed to the r, s plane, its center is displaced to r_0, s_0. Finally, when this rectangle is projected on to the sphere, the result is an approximation to the desired search area.

[redacted]

DIGITAL COMPUTER PROGRAM

General Description

In order to test the validity of the computational methods derived above, and in order to determine the time necessary to obtain a fix, these methods were programmed for a LGP-30 digital computer. This computer is a general purpose, fixed point, single address, drum memory machine with a simple order code.

The standard computer word is 32 bits long and often expressed as eight hexadecimal digits of four bits each. The hexadecimal alphabet employs the characters f, g, j, k, q, and w to represent 10, 11, 12, 13, 14, and 15 respectively. The implied location of the binal point is between the most significant and next most significant bit, with the most significant bit serving as a sign place in complementary arithmetic. The least significant of the 32 bits is available only in the accumulator and not in the main memory.

The fixed location of the binal point requires scaling in most instances. It will be convenient, when dealing with scaled quantities, to designate the scaling as being at a certain "q." By this is meant that the quantity has been multiplied by 2^{-q} before storage. In effect, the binal point for that quantity is shifted q places to the right of the machine binal point.

The storage capacity of the drum is 4096 words. The addresses of these storage locations in machine language are formed by expressing the decimal quantities 0 through 4095 in binary at a q of 29, i.e., by the hexadecimals 00000000 through 00003wwj with intervals of 00000004. It is more convenient, however, to concoct a programmer's language in which storage locations are of the form ttss, where tt is a decimal number between 00 and 63 and is called the "track" and ss is a decimal number in the same range and is called the "sector," these names corresponding to the physical construction of the drum. Appendix A contains additional information about the computer.

Figure 7, which is a foldout, in the back of this report, is a storage allocation chart showing the location of the programs, routines, and subroutines which are used in the fixing process. The complete coding and flow charts are contained in Appendix B. The material immediately following describes the general nature of the computation and the functions of the computational components.

Details

<u>Station Loading Program</u> - The station loading program, which is located in 3800 through 3916, computes x_i, y_i, z_i, cos α_i, sin β_i, and cos β_i for the ith station and to store these quantities, all at a q of 1, in 4148+i, 4200+i, 4216+i, 4232+i, 4248+i, and 4300+i respectively. The flow chart and the coding of this program are given in Figs. B1 and B2 in Appendix B.

Upon transfer of machine control to 3800, the initial location of this program, the typewriter writes a note requesting i, the hexadecimal digit between 0 and w denoting the station number. When this is entered, the storage addresses are set up and a note requesting latitude is printed. This is entered as a decimal number to the nearest hundredth of a degree. Next a note requesting longitude is printed, and the station longitude is entered in the same manner. Negative quantities (south latitudes and west longitudes) are entered as a minus sign, followed by a string of zeros, followed by the quantity, the number of zeros being chosen to make the hundredths of degrees the eighth character of the group.

[redacted]

The program then computes the desired quantities, stores them in the proper locations, and then halts. Depressing the start switch on the computer control panel will now transfer control back to 3800, repeating the process.

This program is not used in every fix computation, but only when it is desired to change the stations comprising the net. It is also possible to store the parameters pertaining to the stations of more than one net, subject only to the limitation that there may not be more than 16 stations stored at any one time.

Fixing Program Assembly - The fixing program assembly, located in 1500 through 1763, switches control to the other components of the fixing program in the proper sequence. Its coding and its flow chart are given in Figs. B3 and B4.

Upon transfer of machine control to 1500, the initial location of this program, the input processor is invoked, which reads the data tape, reassembles storage, computes x_i', y_i', and z_i', and then transfers control back to the program assembly, which sets the iteration counter to zero and makes the inverse rotation matrix a unit matrix at a q of 0. At this point the computer stops unless the "break point 32" switch on the control panel is depressed.

An initial fix estimate is now required. Ordinarily this will come from the polygon center method, but occasionally it may be desirable to impose a different initial point. The "transfer control" switch on the computer provides this choice.

If the "transfer control" switch is depressed, the latitude and longitude of the initial point may be entered on the keyboard. From this information, the r and s of the initial point are calculated at a q of 4 and deposited in 1407 and 1414 respectively.

If the switch is in its normal position, control is transferred to the coefficient calculator, which computes a_i, b_i, and c_i and stores them between 4316 and 4363. Control is next transferred to the polygon center routine, which calculates r and s at a q of 4 and stores them in 2662 and 2663. The program assembly now transfers them to 1407 and 1414 and the two branches rejoin. Control is now transferred to the converter which uses the values of r and s to compute the eight elements of the rotation matrix which are stored in various locations in track 09 at a q of 0 and to compute the values of x, y, and z and store them in the position printer in locations 2761 through 2763 at a q of 1. At this point the computer stops unless the "break point 4" switch is depressed.

The condition of the "transfer control" switch is again sensed, and if the switch is depressed control is transferred to the task number printer, a subroutine which prints the words "task no." and the task number that identified the input tape. The words "quick fix" are printed, and control is then transferred to the position printer, which prints the latitude and longitude of the point whose coordinates are located in 2761 through 2763 to the nearest minute. Control is now returned to the main sequence, where it would have directly proceeded had not the "transfer control" switch been depressed.

Control is next transferred to the rotator, which applies a rotation that would bring the point formerly in the converter to the North Pole to x_i, y_i, z_i, x_i', y_i', and z_i'. Control is then transferred to the series expander which computes r and s and stores them in 3962 and 3963 at a q of 0. The program assembly restores them in the converter in 1407 and 1414. The value of μ^2 is examined. If it is extremely small, x is set at 0, y at 0, and z at 1 and control is transferred to the BPE output procedure; otherwise the converter is invoked. Again μ^2 is examined. If it is less than some preset tolerance (about 10 miles) control is transferred to the BPE output procedure; otherwise the iteration

[redacted]

counter is incremented and then examined. If the number of iterations that have been performed exceeds some preset limit, control is transferred to the BPE output procedure; otherwise control proceeds in the main sequence and the computer stops unless the "break point 8" switch is depressed.

If, at this point, the "transfer control" switch is depressed, the derotator is invoked to convert the coordinates in the position printer to the original coordinate system. Control is next transferred to the task number printer, which prints the task number, and then control is transferred to coding which prints the words "iteration no." followed by the iteration number. The position printer is invoked and then control is returned to the point where it would have directly proceeded had not the "transfer control" switch been depressed. This point is where the rotator was previously invoked, and thus this transfer of control initiates another iteration.

Upon transfer to the BPE output procedure, control is transferred to the task number printer, which prints the task number, and then the word "BPE" is printed. Finally the derotator and position printer are invoked, after which control is transferred to the search area program.

Input Processor - The input processor is located in 0500 through 0763, and is designed to read the data tape and perform preliminary calculations. Its coding and its flow chart are given in Figs. B5 and B6.

For each task, the input tape contains 17 words. The first is a word of two hexadecimal digits that identifies the task. The remaining 16 are reports from the df stations. The sequence of these reports must correspond to the numbers the stations were assigned when their coordinates were entered in the station loading program.

If a station fails to report on the task, or if there is no station corresponding to a given station number, the report is a zero word. If a station reports on the task, the report is a composite word, the first character of which is a minus sign which marks the word as a report, the next three characters of which contain the reported bearing to the nearest 1/8 degree, the next three characters of which contain the estimated inverse variance to ten binal places, and the last character of which contains a zero. The form of a report word is illustrated in Fig. 8. The bearing is seen to be contained in the input word at a q of 12 and the inverse variance at a q of 17. It should be noted that the assumption has been made that the maximum estimated inverse variance will be less than four inverse square degrees.

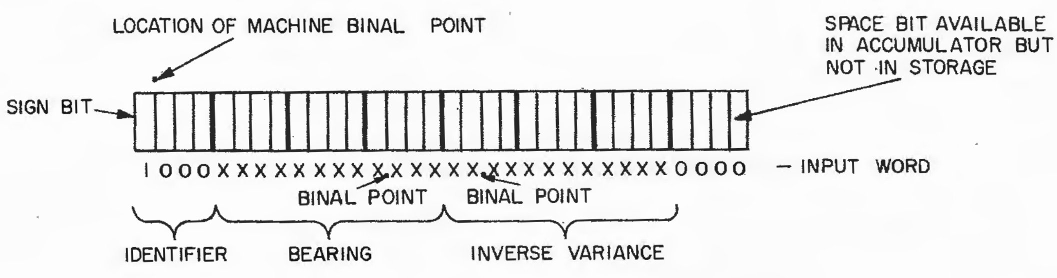


Fig. 8 - Form of input word

[redacted]

Upon transfer of control to 0500, the entrance location of this routine, the first word is read and the task number extracted and stored in 0730 at a q of 30. The next sixteen words are then read. If one of these, say the ith, is the rth valid report ($r = 0$ for the first valid report), the routine recalls from storage x_i, y_i, z_i, cos α_i, sin β_i, and cos β_i which were located at 4148+i, 4200+i, 4216+i, 4232+i, 4248+i, and 4300+i respectively. It stores x_i, y_i, and z_i in 4000+r, 4016+r, and 4032+r respectively, and stores the inverse variance at a q of 2 in 4132+r. The routine then computes x_i', y_i', and z_i' and stores them, at a q of 1, in 4048+r, 4100+r, and 4116+r respectively. The rationale behind this rearrangement is to allow the other computational components to consider quantities in successive storage locations without having to skip over quantities associated with stations not reporting bearings.

At the conclusion of this process, n, the number of stations reporting bearings, is contained in 0736 at a q of 29, and the number $n-1$ in 0731 at the same q. These are made available to the other components so they can run over the n values of i that correspond to stations reporting bearings. The program exits from 0725.

Coefficient Calculator - The coefficient calculator is located from 0800 to 0901, is entered at 0800, and exits from 0861. The coding and the flow chart are given in Figs. B7 and B8.

This routine extracts the n sets of x_i, y_i, z_i, x_i', y_i', and z_i' from storage, and for each set computes a_i, b_i, and c_i and stores them in 4316+i, 4332+i, and 4348+i respectively at a q of 1.

Polygon Center Routine - The polygon center routine is located from 2500 to 2663, is entered at 2500, and exits from 2652. Its coding and its flow chart are given in Figs. B9 and B10.

It may be recalled that the quantities r_{ij} and s_{ij} are functions of a_i, b_i, c_i, a_j, b_j, and c_j, where i runs from 0 to $n-2$ and j runs from $i+1$ to $n-1$ for each i. It may also be recalled that r_o and s_o are the sums of the r_{ij}'s and s_{ij}'s, each multiplied by a weighting factor, divided by the sum of the weighting factors. The routine accordingly sets up the various combinations of i and j, and, as each r_{ij} (or s_{ij}) is calculated, multiplies this quantity by the proper weighting factor and adds the weighted quantity to an accumulator located at 6354 (or 6355) and accumulates the sum of the weighting factors in 6341. After all combinations of i and j are computed, the division of 6354 by 6341 yields r_o, which is stored in 2662 at a q of 4, and the division of 6355 by 6341 yields s_o, which is stored in 2663 at the same q.

Since the number of operations required to perform this routine varies approximately as the square of n, and since the routine can not be trusted to yield anything more accurate than an initial point because of the geometric bias discussed earlier, all n's greater than six are arbitrarily reduced to six to ensure that the amount of time spent in this program is no greater than that warranted by the accuracy of the result obtained.

Converter - The converter is located between 1300 and 1463. Its coding and its flow chart are given in Figs. B11 and B12.

Before invoking this routine, the values of r and s to be converted must be located in 1407 and 1414 respectively. If these quantities are derived from the polygon center routine or from an externally imposed initial point, they will be at a q of 4, while if they are derived from the series expander, they will be at a q of 0. In the former case the converter is entered at 1303, in the latter case at 1300. In either case exit is from 1459.

[redacted]

[redacted]

The converter calculates the coordinates x, y, and z corresponding to the r, s description, and stores these coordinates in the position printer at 2861, 2862, and 2863 respectively at a q of 1. It also calculates the eight non-zero elements of the matrix that rotates this point to the North Pole, and stores these elements, λr_o, λs_o, λ, s_o/μ, $-r_o/\mu$, $-\lambda\mu$, $\lambda r_o/\mu$, and $\lambda s_o/\mu$, in the rotator at 0944, 0947, 0950, 0951, 0954, 0957, 0958, and 0961 respectively at a q of 0.

Position Printer - The position printer, which is located between 2700 and 2863, prints the latitude and longitude of a point whose x, y, and z coordinates are located at 2861, 2862, and 2863 at a q of 1. The routine is entered at 2700 and exits from 2754. Its coding and its flow chart are given in Figs. B13 and B14.

The routine first computes α and β, storing them in 2853 and 2855 respectively in degrees at a q of 9. It then applies the roundoff subroutine, which is located between 2755 and 2808, to α. At the conclusion of this procedure the angle is rounded off to the nearest minute of arc with the degrees stored in 6327 at a q of 9 and the minutes stored in 6330 at the same q.

The routine now recalls the degrees of α and invokes the output subroutine, which is located between 2809 and 2839. This subroutine converts the quantity to decimal and prints it.

The position printer now prints a hyphen, recalls the minutes of α, reinvokes the output subroutine, prints the letter "N", and spaces. The roundoff subroutine is applied to β and the process is repeated except that instead of printing an "N", either an "E" or a "W" is printed as required by the sign of y. Finally, a carriage return is executed and the position printer exits.

Rotator - The rotator routine, which is located between 0903 and 1115, applies a transformation that would rotate the point last processed by the converter to the North Pole. The transformation is not applied to this point, but rather to all other pertinent data. The routine is entered at 0903 and exits from 1112. Its coding and its flow chart are given in Figs. B15 and B16.

It may be recalled from the discussion on rotation of coordinates in the section on mathematical apparatus that this transformation may be accomplished by a matrix multiplication. The elements of the matrix are supplied at a q of 0 by the converter. This matrix is applied to the n vectors [x, y, z] and to the n vectors [x', y', z']. It may also be recalled that the inverse of this matrix is to be applied to an inverse accumulator matrix which keeps track of transformations and can be used to rotate coordinates back to the original coordinate system. Since the matrix is the matrix of a simple rotation, this can be accomplished by applying it to three vectors properly selected from the inverse accumulator. Thus the rotator is simply a device for performing 2n+3 multiplications of vectors by the same matrix.

Series Expander - The series expander routine, which is located between 1800 and 2463 and between 3916 and 3963, derives the values of r_o and s_o from the values of x_i, y_i, z_i, a_i, b_i, c_i, and the inverse variances according to the series expansion method derived earlier. The routine is entered at 1800 and exits from 3958. Its coding and its flow chart are given in Figs. B17 through B22.

The routine may logically be divided into four parts. The first part is designed to calculate the values of $\sigma_i^{-2} a_{ipq}$. There is a difficulty involved here deriving from the fact that a_{ipq} is a fraction whose denominator contains $1 - z_i$ to various powers.

[redacted]

As z_i approaches one, the magnitude of the fraction may become quite large. The computer is a fixed point machine and cannot handle quantities of magnitude greater than one. This problem is usually handled by scaling the quantities, but in this case a scaling factor sufficiently powerful to prevent overflow would cause serious loss of significance in many other cases. One way out of this dilemma is to program the machine for floating point arithmetic, but this procedure requires so many operations that the computing times become intolerable on a drum computer. The approach adopted has been to do as much of the arithmetic as possible at a fixed q, and then to use a procedure which resembles floating point arithmetic for only those processes which absolutely demand it.

Accordingly, Part 1 of the series expander computes and stores at a fixed q the numerators and denominators of $\sigma_i^{-2} a_{ipq}$. Although there are only two unique denominators for each i, all five are stored because of the nature of Part 2. These numerators and denominators are stored in locations from 4400 through 4631 as shown in Fig. 7. Part 2 of the series expander, located between 2200 and 2401, performs the minimum necessary scaling on these quantities and computes t_{pq}.

For each of the five pq combinations, t_{pq} may be found by performing the n divisions of numerator by denominator and adding the quotients. This is accomplished by first clearing zero into a t and a q accumulator located in 6307 and 6308 respectively. The first numerator and denominator are brought into working storage and examined. If the magnitude of the denominator is greater than twice that of the numerator, the division is performed yielding a quotient of magnitude less than 1/2. When this condition is not satisfied, the denominator in working storage is doubled, if this can be done without overflow, otherwise the numerator is halved. Since the q of the anticipated quotient is thereby increased by one, a rescaler is invoked to increment the q of the remaining i's of that pq. Control is returned to the original examination, and the process loops until sufficient scaling has occurred.

When a quotient is obtained, it is added to the t subtotal in 6307. This new subtotal is now examined to determine whether its magnitude is less than 1/2. If this condition is not satisfied, the rescaler is again invoked, thus assuring that the addition of the quotient to the subtotal will never cause an overflow.

This entire procedure is repeated for the next numerator-denominator pair until all n pairs have been processed. Location 6307 now contains t_{pq} and 6308 contains the q of this quantity, itself stored at a q of 29. These quantities are stored in the proper locations and the process is repeated for the next pq pair until all five have been computed. At the conclusion of the process, t_{10}, t_{01}, t_{20}, t_{02}, and t_{11} have been stored in 4632, 4634, 4636, 4638, and 4640 respectively, and the q of each of these quantities has been stored at a q of 29 in a location one higher than that of the quantity.

The rescaler is located between 2314 and 2360 with patches at 2153-2155 and 2211-2212. It halves the quantity in 6307 and adds 1 at a q of 29 to the quantity in 6308. It also increases the q of all quotients of the current pq pair of i greater than the current i up to n-1. It does this by doubling the denominator if this can be done without overflow, otherwise by halving the numerator. This branching assures scaling with minimum loss of significance. It should also be noted that sets of denominators originally identical will no longer be identical after restoring, thus justifying their separate storage.

Part 3 of the series expander, located between 2402 and 2463, converts all the t_{pq}'s to the same q. It searches the q's stored in alternate locations from 4633 to 4641 and determines the maximum value. It then converts all t_{pq}'s to this q through the use of the

[redacted]

table of shifts located between 3916 and 3931. At the conclusion of this process, this q is located in 2402 at a q of 29. This value is needed later by the search area program.

Part 4 of the series expander, located between 3932 and 3963, computes r_o and s_o. Since all terms in both the numerators and denominators of the expressions for r_o and s_o are quadratic in t_{pq}, the q of these latter quantities is immaterial provided they are all the same. At the conclusion of the routine, r_o and s_o are stored in 3962 and 3963 respectively at a q of 0 and the series expander exits from 3958.

Derotator - The derotator is located between 1116 and 1160. It contains within it the elements of the inverse accumulator matrix stored between 1123 and 1131. The derotator is entered at 1116 and exits from 1154. Its coding and its flow chart are given in Figs. B23 and B24.

The function of this routine is to perform a multiplication of the inverse accumulator matrix with the position vector stored in the position printer between 2861 and 2863, and thus to convert the coordinates of this position back to the original coordinate system. The inverse accumulator matrix was initially made a unit matrix by the program assembly coding, and was properly modified each time the rotator was invoked.

Search Area Program - The search area program is located between 3000 and 3232. Its function is to define the corners of the rectangle that approximates the search area. The program is entered at 3000 and exits from 3162 back into fixing program assembly at 1500 if "break point 16" switch is depressed, otherwise computation halts at 3161 after completion of the program. The coding and the flow chart of this program are given in Figs. B25 and B26.

The coding from 3000 to 3059 straightforwardly calculates ρ, $(1+\rho^2)^{-1/2}$, $(u_{20})^{1/2}$, and $(u_{02})^{1/2}$ from the stored values of t_{pq}. At this time a hitherto neglected factor must be taken into account. The scaling of t_{pq} was immaterial to the calculation of r_o and s_o since both numerators and denominators were quadratic in t_{pq}. Similar considerations apply to the calculation of ρ except that the factors are all linear. This neglect cannot be extended to u, however, since that quantity is itself linear in t_{pq}. It is therefore necessary to correct three sources of scaling in t_{pq}: the scaling of the $a_{i\,pq}$ numerators and denominators as produced by Part 1 of the series expander, the further scaling of Parts 2 and 3, and the scaling produced by expressing σ_i^{-2} in inverse square degrees instead of inverse square radians.

If u were properly scaled, s'' could be found by dividing 1.95 by $(u_{02})^{1/2}$. Since u is linear in t_{pq}, the resulting quotient should be multiplied by the square root of any factor that has previously scaled t_{pq}. Expressing σ_i^{-2} in inverse square degrees effectively multiplied t_{pq} by $(\pi/180)^2$. The constant in 3226 which is divided by the two square roots of u is accordingly made $1.95\pi/180$ or 0.0340339. The remaining scaling on t_{pq} is a multiplication by 2^{-q} where q is in 2402 at a q of 29 and a further halving of t_{pq} since Part 1 of the series expander calculated the $a_{i\,pq}$ numerators at a q of 3 and the denominators at a q of 2. Accordingly the quantity in 2402 is incremented by 1 at a q of 29 and now represents at that q twice the power of 2 by which the quotient should be divided. The quantity in 2402 is halved and then used to select an inverse power of 2 from the table of shifts which is used to multiply the factor in 3226 before division by the calculated u's. If the quantity in 2402 is odd, a special procedure employing $1/\sqrt{2}$ is used.

The result of this entire procedure is to place the correct values of $+r''$ and $+s''$ in 3246 and 3247 respectively. The note "search area corners" is now printed. The positive

r'' and s'' are transferred to 3248 and 3249 respectively and the output procedure is invoked to print the location of a corner. The process is repeated until the four combinations of positive and negative r'' and s'' have been used, whereupon two carriage returns are executed and the program halts.

The output procedure is located between 3200 and 3218. It uses the relations previously derived to compute the r and s of the corner from the properly signed r'' and s'' and deposits them in the converter. It then invokes the converter, derotator, and position printer to print the location of the corner and exits.

EXPERIMENTAL RESULTS

The complete system has been operating since mid December 1958 and some components were operating for several months prior to this. The following material summarizes observations made of program effectiveness through the first month of operation of the complete system.

Station Loading Program

The station loading program is a simple and straightforward program and is completely satisfactory. The time required to load a station is about 11 seconds, including the time required for the computer to print the notes but excluding the time required for the operator to type the coordinates on the keyboard.

Figure 9 is a reproduction of the printout produced during a loading operation. It was desired to establish the coordinates of station 2 as 26.98°N 80.02°W. The underlining was added afterwards to indicate what items were typed by the operator as distinguished from those typed by the computer. The operator presses a start switch after typing each word.

```
station no. 2
latitude? 2698
longitude? -0008002
```

Fig. 9 - Printout during station loading

The imprecision of this program is negligible compared with that involved in the rounding of coordinates to 0.01° of arc.

Fixing Program

In most of the experimentation with the fixing program, a standard set of stations was employed. These stations were selected from those plotted with bearing roses on H.O. Chart 5402 so that this chart could be used to assess the plausibility of fixes. The stations are given in Table 1.

Typical experiments involved picking a point on a chart and measuring the bearing to this point from selected stations. Tapes were prepared to simulate reports from these stations. In some cases the stations were imagined to have sent "correct" bearings; in

[redacted]

other experiments errors were systematically included. Estimated variances were varied to determine the characteristics of the weighting processes.

Table 1
Hypothetical Nets

Station No.	Station Name	Latitude	Longitude
	<u>Atlantic</u>		
0	Toro Point	9.35°N	79.95°W
1	South Pass	29.00°N	89.20°W
2	Jupiter	26.98°N	80.02°W
3	Norfolk	36.80°N	76.02°W
4	Bar Harbor	44.30°N	68.13°W
5	Cape Race	46.70°N	53.15°W
	<u>Pacific</u>		
6	Imperial Beach	32.60°N	117.15°W
7	Empire	43.45°N	124.30°W
8	Soapstone	58.01°N	136.52°W
9	St. Paul I.	57.10°N	170.30°W
f	Hilo	19.70°N	155.02°W
	<u>Special</u>		
g	Reserved for Special Experiments		
j			
k			
q			
w			

The stops and branches in the fixing program assembly were liberally used in the experimental work. If the program were to be used operationally, many of them would be eliminated. The stops provided opportunities for the printout of intermediate results through the use of the decimal memory printout, while some of the branches provided such printout without further intervention.

<u>Input Processor</u> - The input processor performs its function in a satisfactory manner. The time required for execution depends on the number of stations making valid reports and is given approximately by $t = 7.0 + 3.0n$ sec, where n is the number of stations reporting. Part of this time is required by the mechanical tape reader. It is estimated that the use of a photoelectric reader available as an accessory to the computer would reduce this to about $t = 5.1 + 2.2n$, but the advantage does not seem particularly significant.

Figure 10 is a reproduction of a printout of an input tape. It may be translated as reading, "On task 5 station 3 reports a bearing of 13.5° and an estimated inverse variance of 0.25 inverse square degrees (i.e., a standard deviation of 2°), station 4 reports a bearing of 5.75° and an estimated inverse variance of 1.0, station 5 reports a bearing of 336.5° and an estimated inverse variance of 1.0, and none of the other stations reported on the

[redacted]

task." The apostrophes are codes marking the end of an input word, and the spaces between adjacent marks therefore represent zero words or null reports.

```
05''''806j1000'802q4000'8f844000''''''''''
```

Fig. 10 - Typical data tape

The imprecision of this routine is negligible compared with that involved in rounding the bearing to the nearest 1/8 degree.

<u>Polygon Center Method</u> - The polygon center method employs two subsidiary routines--the coefficient calculator and the polygon center routine. The first of these is simple, straightforward, and rapid, requiring about 0.7n seconds.

The time required by the polygon center routine is given approximately by $t = 0.5 - 0.3n + 0.4n^2$. It should be remembered that n has been arbitrarily limited to six, although this can be modified by inserting $n_{max} + 1$ in 2654 at a q of 29.

In considering the effectiveness of the polygon center routine, two sources of "error" need to be considered. The first, which is a true error, is a possible imprecision caused by roundoff in the machine. A fairly detailed investigation indicates that provided at least two stations report reasonably large estimated inverse variances -- standard deviations of no worse than eight degrees or so -- the error due to roundoff will not be seen in the final result. The second source of "error" is the effect that was anticipated in the theoretical discussion -- the geometrical bias implicit in long, thin polygons.

This point can be clarified by giving the results of a typical experiment on the standard nets in which ten target locations were selected. Bearings were measured from the stations to the target locations and the "correct" bearings coded on the input tape. The break point and transfer control switches on the computer control panel were manipulated to print out a "quick fix," i.e., the result of the polygon center calculation. The results of this experiment are given in Table 2. The column headed "error" gives the distance in nautical miles between the hypothetical target location and the fix location.

Inspection of Table 2 reveals that of the ten tasks, admittedly not chosen quite at random, four of the quick fixes have errors of ten miles or less and two have errors of 16 miles. Four of the tasks are, however, considerably in error.

The cause of the difficulty with task 00 is not hard to see. Station 0, station 2, and the target lie on practically the same meridian. The due north bearing circle from station 0 and the almost due south circle from station 2 are nearly parallel, and a very slight "error" in rounding off the bearings has thrown the intersection of these circles hundreds of miles to the north of the target. Since this intersection constitutes 2/9 of the final result, the fix is too far north. The validity of this explanation can be easily tested by deleting either station 0 or station 2 from the report.

[redacted]

Table 2
Ten Quick Fixes with "Correct" Bearings

Task	Target	Reports			Fix	"Error" (naut mi)
		Stn.	Bearing	Inv. Var.		
00	15° 00′N	0	000.000°	1.00	16°53′N	
	080° 00′W	1	146.750°	1.00	080° 00′W	113
		2	180.500°	1.00		
		3	190.000°	0.50		
01	15° 00′N	0	074.750°	1.00	15° 06′N	
	055° 00′W	1	106.500°	0.50	055° 15′W	16
		2	112.250°	1.00		
		3	134.250°	0.25		
02	25° 00′N	0	039.750°	1.00	25° 04′N	
	065° 00′W	1	095.000°	0.50	065° 10′W	10
		2	095.000°	1.00		
		3	138.500°	1.00		
		4	171.250°	0.50		
		5	208.000°	0.50		
03	40° 00′W	1	050.500°	0.25	39° 52′N	
	070° 00′W	2	030.250°	1.00	070° 15′W	29
		3	054.000°	1.00		
		4	198.000°	1.00		
		5	247.500°	0.50		
04	50° 00′N	3	036.250°	0.25	50° 10′N	
	060° 00′W	4	041.500°	1.00	059° 00′W	40
		5	298.500°	1.00		
05	60° 00′N	3	013.500°	0.25	59° 56′N	
	065° 00′W	4	005.500°	1.00	65° 02′W	4
		5	336.500°	1.00		
06	20° 00′N	6	226.000°	1.00	20° 08′N	
	130° 00′W	7	193.500°	1.00	130° 04′W	9
		8	170.000°	0.25		
		f	085.000°	0.50		
07	35° 00′N	6	283.000°	1.00	34° 54′N	
	145° 00′W	7	249.250°	1.00	145° 03′W	7
		8	197.500°	1.00		
		9	131.500°	0.50		
		f	028.250°	1.00		
08	55° 00′N	7	311.500°	0.50	55° 20′N	
	155° 00′W	8	261.250°	1.00	156° 19′W	50
		9	097.250°	1.00		
09	40° 00′N	6	295.250°	0.50	40° 12′N	
		7	272.250°	1.00	154° 47′W	16
		8	223.500°	1.00		

A similar difficulty exists with task 04, although its presence is not quite so obvious. Station 3, station 4, and the target lie on the same great circle, although the bearings are different because the great circle is not a meridian. Again the validity of this explanation can be tested by deleting one of these stations.

Accordingly, the experiment was repeated with these deletions. The results (Table 3) verify the explanation.

Table 3
Modified Quick Fixes

Task	Target	Reports			Fix	"Error"
		Stn.	Bearing	Inv. Var.		
00	15°00'N 080°00'W	0	000.000°	1.00	16°53'N 080°00'W	113
		1	146.750°	1.00		
		2	180.500°	1.00		
		3	190.000°	0.50		
10	15°00'N 080°00'W	0	000.000°	1.00	14°51'N 079°56'W	10
		1	146.750°	1.00		
		3	190.000°	0.50		
04	50°00'N 060°00'W	3	036.250°	0.25	50°10'N 059°00'W	40
		4	041.500°	1.00		
		5	298.500°	1.00		
14	50°00'N 060°00'W	4	041.500°	1.00	49°59'N 059°56'W	3
		5	298.500°	1.00		

The conclusion from this discussion is that even with "correct" bearings the polygon center method may give a very poor fix. This fix is certainly good enough to serve as an initial point for the series expansion method, but not reliable enough to be trusted as a BPE even though it is available as a computer output. This is the result that was anticipated all along, and should not be surprising.

An important exception to this observation occurs in the case in which only two stations report. In this case the polygon center (the intersection of the bearing circles) is the BPE.

It is also interesting to consider the effectiveness of this routine on "incorrect" bearings. The results of one experiment will be given here. Referring to Table 2, task 06 is seen to have no geometrical difficulties and to yield a good quick fix. Errors of magnitude equal to the estimated standard deviation were inserted in the reports on this task. Some of the results of this experiment are presented in Table 4. In this table, "error" denotes the shift from the fix based on "correct" bearings.

Series Expansion Method - It will be recalled that the series expansion method is an iterative method, each iteration requiring the use of the converter, rotator, coefficient calculator, and series expander. After a sufficient number of iterations have brought

[redacted]

$r_o^2 + s_o^2$ within some preset tolerance, or after the preset maximum number of iterations have taken place, the converter, derotator, and position printer are invoked to print out the BPE.

Table 4
Effect of Bearing Errors

Task	Bearing and Inverse Variance				Fix	Error (naut mi)
	Station 6	Station 7	Station 8	Station f		
06	226.000° 1.00	193.500° 1.00	170.000° 0.25	085.000° 0.50	20°08′N 130°04′W	
16	227.000° 1.00	193.500° 1.00	170.000° 0.25	085.000° 0.50	20°23′N 130°07′W	15
26	225.000° 1.00	193.500° 1.00	170.000° 0.25	085.000° 0.50	19°52′N 130°02′W	16
36	226.000° 1.00	194.500° 1.00	170.000° 0.25	085.000° 0.50	20°00′N 130°23′W	20
46	226.000° 1.00	192.500° 1.00	170.000° 0.25	085.000° 0.50	20°12′N 129°46′W	18
56	226.000° 1.00	193.500° 1.00	172.000° 0.25	085.000° 0.50	19°27′N 130°22′W	43
66	226.000° 1.00	193.500° 1.00	168.000° 0.25	085.000° 0.50	20°34′N 129°50′W	29
76	226.000° 1.00	193.500° 1.00	170.000° 0.25	086.375° 0.50	19°54′N 130°11′W	15
86	226.000° 1.00	193.500° 1.00	170.000° 0.25	083.625° 0.50	20°22′N 129°58′W	15
96	227.000° 1.00	192.500° 1.00	172.000° 0.25	083.625° 0.50	19°45′N 130°01′W	23

Because it is impossible to determine in advance just how many iterations will be required, an explicit statement of the time required by this method cannot be made. A further difficulty arises from the fact that Part 2 of the series expander makes use of its internally contained rescaler an unpredictable number of times. Thus it is not even possible to accurately predict the time per iteration. Therefore some of the data in Table 5 must be taken as only approximate. In this Table, n represents the number of stations and m represents the number of iterations.

Roughly half the fixes investigated required only one iteration, most of the rest required two. A safe estimate of the median m would therefore be two. A typical running time for Part 2 of the series expander might be ten seconds. Under these assumptions, a typical operating time of the series expansion method would be $9.8n + 42$ seconds or slightly over a minute and a half for a five-station fix. No tasks will be performed in less than half this time, while a few tasks will take more than twice this time, thus raising the average times above these estimates. In estimating the total time required to obtain the BPE, the time required by the input processor and polygon center method must be included.

Table 5
Operating Time of the Series Expansion Method

Component	Time (sec)	Times Invoked	Total Time (sec)
Converter	2.1	$m + 1$	$2.1(m + 1)$
Rotator	$0.9n + 1.9$	m	$m(0.9n + 1.9)$
Coeffic. Calc.	$0.7n$	m	$0.7mn$
Series Exp. (1)	$3.3n$	m	$3.3mn$
Series Exp. (2)	5 to 60	m	m(5 to 60)
Series Exp. (3)	2.2	m	$2.2m$
Series Exp. (4)	1.0	m	m
Derotator	1.4	1	1.4
Position Printer	4 to 6	1	4 to 6
Program Assembly	4	1	4
Total Time: $4.9mn$ + (12 to 72) m + (7.5 to 9.5)			

An analytic investigation of the convergence properties of the series expansion method appears to be an extremely formidable task and accordingly was not attempted. It certainly seems plausible that an iteration which produces a near-zero value of r_o and s_o has produced an answer which is correct to the same extent that the distribution assumptions are correct. Conversely, a large value of one or both of these factors would seem to indicate the necessity of another iteration.

The experimental evidence does not seem to contradict these suppositions. Whenever an iteration does not significantly move the BPE, i.e., whenever it produces a small r_o and s_o, further iterations do not affect the BPE. Whenever an iteration moves the BPE more than ten miles or so, further iterations usually (but not always) cause further movement of the BPE. Thus the BPE acceptance rule in the program assembly would appear to be sound.

There remains the critical question of whether in every case a sufficient finite number of iterations will yield a BPE that is unaffected by further iterations. This question has been investigated over the first month that the program has been operating by fixing a wide variety of tasks. The answer, unfortunately, is in the negative, although a fix is obtained in the vast majority of cases.

In those cases where there are no geometrical difficulties, and where the reported bearings are reasonably "correct," i.e., converge to form a reasonably small polygon, the series expansion method performs quite well, In these cases it is necessary to impose an incorrect initial point in order to investigate convergence, as the polygon center method would yield an estimate that would be confirmed in the first iteration. Several different types of experiments of this general nature yield about the same results, which are displayed in Fig. 11. The abscissa represents the distance that one iteration should have

[redacted]

[redacted]

moved the BPE, i.e., the distance between the initial point and the target, while the ordinate represents the distance between the target location and the BPE after the iteration. The relationship summarized in Fig. 11 is most reproducible in the middle ranges, where it is approximately quadratic and almost always in the form of an overcorrection of the initial point.

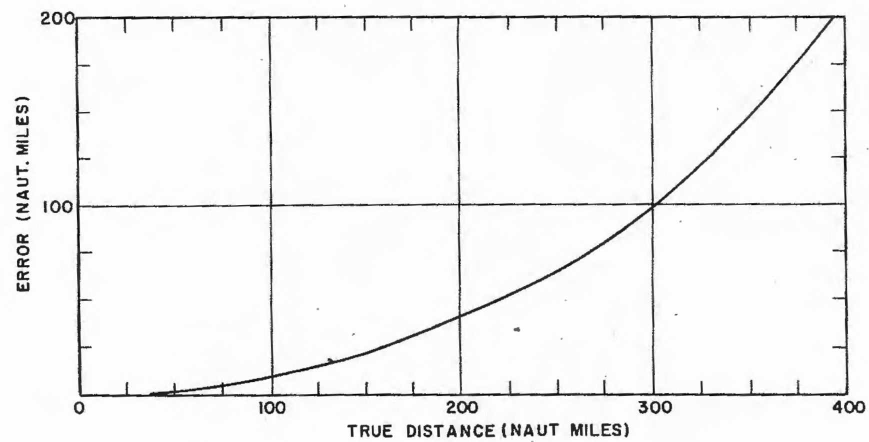


Fig. 11 - Convergence of series expander

Some thought was given to the use of this observation in performing a compensation or correction on the r_0 and s_0 produced by the first iteration. Unfortunately the compensation would be reliable only for well-behaved geometries where it would be unnecessary, since the initial point produced by the polygon center method would be quite good and the first iteration would produce the final BPE.

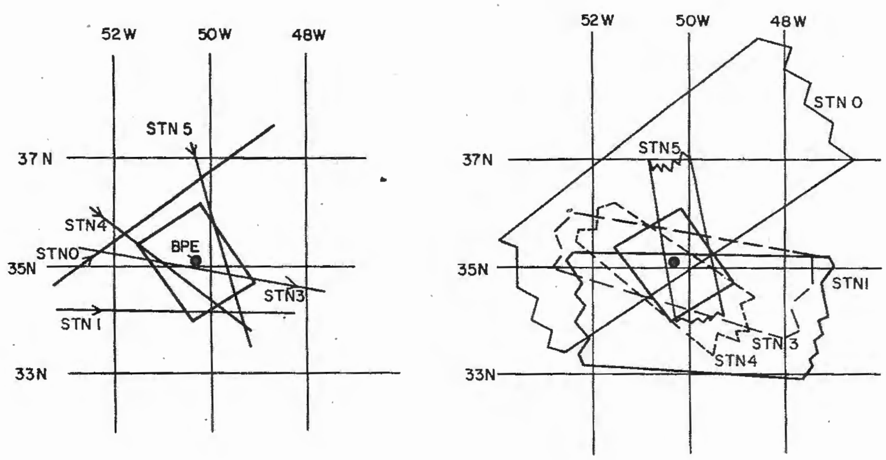
In those cases where there are geometrical difficulties such as a long thin polygon but where the bearings are again "correct," the series expansion method works well. Recalling tasks 00 and w0 from Table 3, BPE's within four miles of each other are produced. In cases of this type, the relationship of Fig. 11 holds reasonably well, although undershoots appear about as often as overshoots. The single failure to fix in cases of this type occurred with two bearings whose bearing circles were practically coincident. In this situation the polygon center method found the intersection of the bearing circles but the series expansion method diverged from this point. This single failure should not be too annoying, as any "fix" produced from such a pair of bearings would be meaningless.

An equally large number of cases in which the bearings contained considerable errors have been investigated, but the investigation is considered much less exhaustive becuase of the many variables in experiments of this kind. Thus far, there has been only one failure to fix which occurred in a three-station situation with a pair of coincident bearing circles. It is believed that in this case the BPE was sliding along this pair of circles and landed on a station location, thus producing an a_{ipq} denominator of zero. In all the other cases convergence to a fix proceeded smoothly, although occasionally at a slower rate than that of Fig. 11. It should be realized that in these cases with large polygons, the "correctness" of the fixes cannot be absolutely determined, they can only be judged for plausibility. Actually this is one of the reasons a computer fixing method is desirable! This question will be considered again in the section on future work.

[redacted]

Search Area Program

The search area program straightforwardly produces the corners of the search rectangle in about 54 seconds. The results have been judged only for plausibility and seem to meet this criterion. A typical case is shown in Fig. 12. In this five-station fix, all stations were assumed to have a standard deviation of two degrees, and errors were assigned all of the reported bearings in accordance with this assumption. The target was assumed to lie at 35°N 50°W. Figure 12a shows the bearing lines, the BPE, and the search rectangle. Fig. 12b shows instead of each bearing line a pair of lines located one standard deviation from the bearing line.



(a) (b)

Fig. 12 - Typical search area

Summary

In the first month in which all of the component programs have been running, a wide variety of tasks has been computed. In all cases but two, plausible BPE's and search areas were produced. The two exceptions were cases which could not have been made to yield a meaningful fix by any method. Insofar as the accuracy of the BPE's could be verified, they have all proved to be correct; others could be checked only for plausibility because no other simple method of obtaining the BPE exists in this country. The same observation applies to the search area.

The computing time for a task depends on several unpredictable factors. The average time for a task appears to be about three minutes with the median time somewhat less. The ten tasks of Table 2 together with the two additional tasks of Table 3 require 34 minutes of computer time if intermediate results are not printed out, 38 minutes if they are. Figure 13 is a reproduction of the output of the computer in the latter case. Print-out actually occurs in a single column; the output sheet was rearranged into two columns for ease of inclusion in this report.

[redacted]

[redacted]


```
task no. 00 quick fix 16-53N 080-00W
task no. 00 iteration no. 1 15-27N 080-00W
task no. 00 iteration no. 2 15-00N 079-59W
task no. 00 BPE 14-59N 079-59W
search area corners
13-49N 080-06W
16-07N 080-12W
16-08N 079-52W
13-50N 079-45W

task no. w0 quick fix 14-51N 079-56W
task no. w0 BPE 14-56N 079-57W
search area corners
13-46N 080-05W
16-05N 080-10W
16-06N 079-49W
13-47N 079-43W

task no. 01 quick fix 15-06N 055-15W
task no. 01 iteration no. 1 14-55N 054-59W
task no. 01 BPE 14-55N 054-59W
search area corners
13-59N 053-49W
14-43N 056-27W
15-50N 056-08W
15-06N 053-29W

task no. 02 quick fix 25-04N 065-10W
task no. 02 BPE 25-03N 065-07W
search area corners
25-02N 065-43W
25-32N 065-25W
25-04N 064-32W
24-34N 064-50W

task no. 03 quick fix 39-52N 070-15W
task no. 03 iteration no. 1 40-05N 069-52W
task no. 03 BPE 40-02N 069-56W
search area corners
40-14N 069-35W
39-42N 070-00W
39-50N 070-16W
40-22N 069-51W

task no. 04 quick fix 50-10N 059-00W
task no. 04 iteration no. 1 49-58N 059-46W
task no. 04 BPE 49-59N 059-56W
search area corners
49-58N 060-25W
50-13N 060-16W
49-59N 059-28W
49-45N 059-37W

task no. w4 quick fix 49-59N 059-56W
task no. w4 BPE 49-59N 059-56W
search area corners
49-56N 060-27W
50-15N 060-15W
50-02N 059-26W
49-43N 059-38W

task no. 05 quick fix 59-56N 065-02W
task no. 05 BPE 59-56N 065-02W
search area corners
58-56N 065-22W
60-47N 066-11W
60-57N 064-39W
59-05N 063-54W

task no. 06 quick fix 20-08N 130-04W
task no. 06 BPE 20-04N 130-07W
search area corners
19-38N 131-01W
21-01N 130-01W
20-30N 129-12W
19-07N 130-12W

task no. 07 quick fix 34-54N 145-03W
task no. 07 BPE 34-54N 145-03W
search area corners
34-45N 145-48W
35-30N 144-57W
35-00N 144-18W
34-15N 145-09W

task no. 08 quick fix 55-20N 156-19W
task no. 08 iteration no. 1 54-57N 154-24W
task no. 08 iteration no. 2 55-00N 154-51W
task no. 08 BPE 55-00N 154-51W
search area corners
55-16N 155-55W
55-12N 153-46W
54-43N 153-50W
54-48N 155-57W

task no. 09 quick fix 40-12N 154-47W
task no. 09 BPE 40-11N 154-55W
search area corners
39-08N 155-46W
39-49N 156-27W
41-12N 154-03W
40-31N 153-23W
```

Fig. 13 - Computer output

[redacted]

DECLASSIFIED

FUTURE WORK

Fixing and Search Area Programs

Minor Improvements - There are several minor improvements that will probably be made in the programs in the near future in order to eliminate certain hazards and decrease computing time.

At the present time there is no sure termination on the rescaling process in Part 2 of the series expander. If the result of an iteration produces a point coincident with a station location, the corresponding a_{ipq} denominators will be zero. The rescaler will keep doubling these denominators without effect. One solution might be to place a counter in the rescaler or else to modify the doubling process to include the addition of a very small constant that would not affect the large majority of calculations.

Another hazard involves the limited length of the table of shifts. It is theoretically possible that long shifts might be called for. The solution is to extend the table and to move Part 4 of the series expander elsewhere. Part 4 might also be rewritten to check the relative magnitudes of the r_o and s_o numerators and denominators to prevent overflow, although it seems inconceivable that the polygon center method could be in error by the 2500 miles necessary to cause overflow.

The techniques of optimization are considered in Appendix A. At the present time, some program

[redacted]

The major advantage of the second is the relative simplicity of such a comparison. The comparison again is not against the universe, but at least it is against another model.

The third method is the most attractive since it tests the program in the real environment in which it will have to work. The labor involved is reasonable. The only disadvantage is that if anomalous results occur it will not be known whether they are attributable to the statistical model or to the faithfulness of the approximation to the model in the program.

Accordingly, it is planned to run U.S. Navy check-target bearing data in the near future. At the time of this writing the transcription of the data is underway. Comparison of the fixes obtained against the fixes given by manual plotters should also prove most interesting. It is also hoped that sufficient data can be obtained from the United Kingdom to eventually utilize the second method.

Major Modifications - The program presented in this report was written mainly as a feasibility demonstration. Before operational use takes place, it would be worth while to consider several major modifications.

It is interesting to note that the program does away with the notion of sense, i.e., it treats a bearing and its reciprocal identically. This seems strange when it is considered that a knowledge of sense may greatly improve the fix. To concoct an extreme case, consider three stations on a rough north-south line. The southernmost station reports a northerly bearing and the northernmost station reports a southerly bearing yielding a weak convergence at the middle station. The search area could be cut in half if the sense of the bearing from the middle station were considered.

Another modification that might be necessary would be to allow simultaneous operation in both hemispheres. It would not be too surprising if the same modifications solved both of these problems.

Thus far no mention has been made of the matter of estimating the variances to be associated with each bearing as this is not a topic for this report. It is true, however, that the variance depends to some extent on the distance between the station and the target. It might be desirable, therefore, to have the computer correct the variances it is using after it has obtained a rough idea of the target location. Fortunately this is a rather simple modification.

A considerably more difficult problem involves the treatment of "wild" bearings. The use of normal statistics requires that we reject bearings likely to have come from another distribution. One possible solution would be to wait until late in the iterative process and then compute the hypothetical bearings from the stations to the calculated target location, rejecting the bearings from those stations whose hypothetical error exceeds a certain multiple of their assumed standard deviation.

It is necessary to wait until late in the process as otherwise good data might be rejected. In the case of task 00, for example, application of such a criterion too early would result in the rejection of the valid cross bearings and the retention only of the pair of coincident circles which, by themselves, can yield no useful information.

The calculation of the bearings would require a significant amount of time, but this time would probably be allowable. A more serious objection would be the time wasted in starting the computation over from the beginning after the rejection of a station. Since the inclusion of a wild bearing may so distort the tentative BPE as to encourage rejection of a valid bearing, it would be desirable to have some method of making a rapid check on the effects of rejecting a station.

[redacted]

The possibility of an elegant solution occurs when it is recalled that the hypothetical error is approximated by h_i/g_i, which itself was expanded in series. It appears, therefore, that at the end of an iteration the values of r and s could be substituted into a quadratic having the a_{ipq}'s as coefficients and the result examined. To test the effect of a station rejection, the a_{ipq}'s of that i could be omitted from the calculation of t_{pq} and the process repeated. Unfortunately, the action of the rescaler leaves these quantities in a form eminently unsuitable for further calculation. Thus if this scheme is to be employed the series expander will have to be largely rewritten.

Other Uses - The availability of the fixing program may make feasible the performance of other tasks besides the obtaining of fixes. One example might be the detection of systematic errors in the performance of a station on operational targets. Such a job requires a consistent method of fixing to ensure that bias blamed on a station is not actually the fault of a manual plotter.

Another question that may have some light shed upon it is that of the importance of an individual station to a net. By deleting the contributions of one station and recomputing the tasks it would be possible to quantitatively assess the contribution of that station. Again a consistent, unbiased method of fixing is required to make such an operation feasible. Some work on this latter problem is planned for the future.

Related Problems

Although this report is devoted to the actual computing process, sight should not be lost of the fact that the maximum benefits can be obtained from automatic fix computing only by careful examination of the entire direction finding activity. For this reason it is necessary that work be done in several areas including bearing storage systems, bearing readout systems, bearing transmission methods, report storage methods, station performance analysis, etc. Research in most of the necessary areas is underway within the Countermeasures Branch and will be coordinated so as to be of maximum utility when automatic fix computation is employed.

SUMMARY AND CONCLUSIONS

The feasibility of df fixing on an inexpensive digital computer has been demonstrated. A program has been developed which produces not only a best point estimate of the target location but also defines the smallest area that has a 90-percent probability of containing the target.

The employment of these techniques makes it possible for the first time in this country to have fixes free of human bias and blunder that express all the information available from a df bearing. It also makes available for the first time an accurate and meaningful search area, which is probably more important for tactical operations.

The average time required by the existing program to produce both the fix and the search area is about three minutes. Although faster results could be obtained from more sophisticated computers, the twenty-task-per-hour capability of the inexpensive computer exceeds the communications capability of the df nets and is therefore considered satisfactory.

Work is continuing on the use of the fixing program in other df problems, and in the refinement of related data handling techniques.

[redacted]

Whereas the cost of a computer serving an entire net is a few percent of the cost of one df station, the utilization of such a computer is considered to be one of the best bargains available to the Navy at this time.

For the U. S. Navy to best utilize the program developed in this report the following steps should be taken:

1. The Navy should install a small digital computer at a net control station for use as an auxiliary fixing device. The selection of an LGP-30 computer is recommended in order that use can be made of the program presented in this report.

2. After a suitable period of testing and program refinement, computers should be installed at all net control stations for use as primary fixing devices.

3. The collection of df station accuracy statistics should be continued and the analysis of this material intensified so that maximum advantage can be made of the mathematical precision afforded by a computer.

4. The development of df data handling equipment should be planned and coordinated so as to best utilize the speed, accuracy, reliability, and automaticity of a digital computer.

ACKNOWLEDGMENTS

The authors would like to express their gratitude to Mr. C. F. Sims and his associates of S59 GCHQ for enlightening discussions of U. K. df practices and problems; to LT H. J. Davis and Mr. Sterling Thrift of Naval Security Group Headquarters Activity for corresponding information from the U. S.; and to Mr. C. E. Corum of the Naval Research Laboratory for reviewing portions of this manuscript.

* * *

[redacted]

 [redacted]

APPENDIX A

LGP-30 COMPUTER
[Unclassified Appendix]

GENERAL DESCRIPTION

The LGP-30 is only one of the several small digital computers currently being marketed. It was chosen for this application because one was available within the Countermeasures Branch. Even if this factor were not present, however, the LGP-30 most probably would have been chosen, largely because no computer selling for less than $50,000 has a larger memory capacity and because of the excellent reliability it has demonstrated.

The computer is 44 inches long, 33 inches high, 26 inches deep, and weighs about 740 pounds. Power requirements are about 1.5 kw from a 115-volt, 60-cycle, single-phase source. Separate air conditioning is not required for the computer, but installation in an air-conditioned space is advisable.

The computer can be purchased or rented from the Royal McBee Corporation. The rental contract is listed in the Federal Supply Service catalog as GS-00S-19602 Item 132 costing approximately $1,200 per month on a single shift basis. The charge for continuous operation, however, is approximately $3,000 per month. Since the purchase price of the machine is approximately $45,000, outright purchase appears advisable. If funds for purchase cannot be obtained immediately, it would be feasible to rent a machine during the initial testing period and then to exercise the option contained in the rental contract to apply half the previously paid rent towards the purchase price sometime before the initiation of full-time operation.

Practically no corrective maintenance has been required on the machine installed at NRL, but thought should be given to the problem of providing maintenance for machines installed in the field. Free maintenance is provided for in the rental contract, if the machine is located not more than 100 miles from one of the vendor's service centers. Service contracts for purchased machines are available at approximately $150 per month subject to the same provision. Alternatively, the vendor will train individuals in the maintenance of the machine in a five-week course for approximately $500 per person.

The computer is a fixed point, single address machine operating in pure binary with complementary representation of negative numbers. The computer word is 32 bits long and is expressed for input-output purposes as a word of eight hexadecimal digits employing the alphabet 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, f, g, j, k, q, and w. The least significant bit is not available for all operations. In an operation on a word representing numerical data, the binal point is considered to be located between the two most significant bits with the most significant bit indicating the sign. Thus the range of valid computational numbers is $-1 < x < +1$ with -1 (i.e., 80000000) a valid computational number for some operations. The power of 2 by which a number has been divided to fit into this range is designated the q of the quantity.

The main memory of the machine consists of 4096 words located on a magnetic drum rotating at 4000 rpm. These locations are arranged in 64 tracks of 64 sectors each and are so designated in the material of this report. As actual addresses in an instruction, however, they must be expressed as the hexadecimals through www at a q of 29, i.e., by the words 00000000 through 00003wwj with intervals of 00000004. Located on the same drum

 [redacted]

are three tracks which are used as recirculating memory and constitute the control register, the instruction register, and the accumulator.

The standard input-output device is a tape-typewriter which is included in the purchase price, although a photoelectric reader is available as an extra-cost accessory. The tape code is a six-channel code allowing for 64 typewriter operations. Only four of the channels are ordinarily read into the machine, however, therefore causing certain pairs of typewriter functions to enter the computer identically. For example, the letter "b" and the digit "1" both enter the machine as the binary 0001, although they are distinguished in programming for mnemonic reasons. On output, however, all six channels are utilized and must be properly synthesized by the program.

Operations are usually read from successive memory locations and executed subject to programmed transfers of control until a stop order is reached. When such an order is read, computation halts unless the address contained in the stop order contains a bit corresponding to a depressed "break point" switch on the computer front panel. These four switches allow certain stops to be bypassed at the option of the operator.

A "transfer control" switch on the front panel allows certain variations of the program as will be explained in the following section. If an overflow occurs, i.e., if the result of a computation is outside the range of valid computational numbers, the machine halts. This eliminates the possiblity of undetected errors due to overflow, but introduces certain programming complications.

ORDER CODE

The order code of the computer is given in Table A1. The order is contained in bits 12 through 15 counting the most significant bit as bit 0. This is equivalent to stating that the order is stored as a hexadecimal 0 through w at a q of 15. The first column of Table A1 contains both the hexadecimal and the typewriter function that enters the machine identically and that is more often used in programming for mnemonic reasons.

OPTIMIZATION

The nominal speed of the drum is 4000 rpm or 15 milliseconds per revolution. The sectors are interlaced on the drum so that 7 "consecutive" locations pass under the read heads in one revolution. Thus "consecutive" orders can be examined at intervals of somewhat over 2 ms. Since the execution time for most orders is about 0.25 ms, it would appear that consecutive orders can be executed at the rate of 7 per revolution. Most orders, however, refer to an operand stored at some other storage location. Unless the operand can be read well before the next order passes under the reader, the cycle of reading the operand, executing the order, and executing the next order will require 1-1/7 instead of 1/7 drum revolution. Operands which can be read early enough are called optimally located operands.

The execution time of about 0.25 ms refers to the b, y, r, e, h, c, a, and s instructions. The difference between performing 467 optimum or 58 non-optimum instructions per second is significant, and every effort should be made to optimize these instructions. Optimization involves making the sector of the operand of an instruction located in sector ss equal to (ss + 7n + 1)modulo 64 where n is an integer from 2 through 7. Thus the optimum sectors for the operand of an instruction located in sector 40 would be sectors 55, 62, 05, 12, 19, and 26. The tracks are immaterial.

[redacted]

 [redacted]

Table A1
Order Code

Order		Address*	Interpretation
0	z	ttxx	Stop computing unless a break-point switch corresponding to a bit in tt is depressed.
1	b	ttss	Bring the contents of ttss into the accumulator.
2	y	ttss	Replace the address portion of ttss with the address portion of the accumulator.
3	r	ttss	Replace the address portion of ttss with the address of the r order increased by two.
4	i	xxxx	Let the input device feed into the accumulator. This order is always preceded by p00xx (start reader.)
5	d	ttss	Divide the contents of the accumulator by the contents of ttss and place the quotient in the accumulator.
6	n	ttss	Multiply the contents of the accumulator by the contents of ttss and place the low-order product in the accumulator.
7	m	ttss	Multiply the contents of the accumulator by the contents of ttss and place the high-order product in the accumulator.
8	p	ttxx	Perform the output function described by the bits of tt. Often followed by z0000 to stop computer which is restarted by output device after execution.
9	e	ttss	Extract the accumulator through ttss, i.e., where corresponding bits of the accumulator and ttss are 1 leave a 1 in the accumulator, otherwise a 0.
f	u	ttss	Transfer control to ttss.
g	t	ttss	Transfer control to ttss if the number in the accumulator is negative, otherwise proceed to the next order.
800g	800t	ttss	Transfer control to ttss if the number in the accumulator is negative or if the "transfer control" switch is down or both, otherwise proceed to the next order.
j	h	ttss	Hold the contents of the accumulator in ttss.
k	c	ttss	Hold the contents of the accumulator in ttss and clear the accumulator to zero.
q	a	ttss	Add the contents of the accumulator to the contents of ttss, retaining the sum in the accumulator.
w	s	ttss	Subtract the contents of ttss from the contents of the accumulator, retaining the difference in the accumulator.

*tt - Track (bits 18 through 23)
ss - Sector (bits 24 through 29)
xx - Immaterial

[redacted]

[redacted]

Execution time for the d, n, and m instructions is about one revolution, so the total time for optimum and non-optimum instructions respectively is 1-1/7 and 2-1/7 revolutions corresponding to 58 and 31 operations per second. The optimizing relation is the same except that n ranges from 4 through 7 for d, from 3 through 7 for m, and from 1 through 7 for n.

STANDARD SUBROUTINES

The program presented in Appendix B utilizes several standard subroutines distributed by the Royal McBee Corporation. A brief description of these subroutines follows.

The arcsine-arccosine subroutine interprets the quantity in the accumulator at the time of subroutine entry as either the arcsine or arccosine of an angle at a q of 1 if the subroutine is entered at 5600 or 5811 respectively. Upon exit, the subroutine leaves in the accumulator the angle in degrees at a q of 9, expressing arcsines in the range -90 to +90 degrees and arccosines in the range 0 to 180 degrees.

This subroutine contains within it a square root routine which can be invoked separately at 5800. Upon exit it leaves in the accumulator the square root of the quantity in the accumulator on entrance. The subroutine assumes these quantities to be unscaled, therefore if the initial quantity is scaled at some q the square root will be scaled at q/2.

The sine-cosine subroutine interprets the quantity in the accumulator at entrance as an angle in degrees at a q of 9. Depending on whether it is entered at 5900 or 5904, the accumulator will contain on exit the sine or cosine respectively of the angle at a q of 1.

The alphanumeric output subroutine is a device for saving storage locations when it is required to print headings and notes. Ordinarily the coding of such notes would require two storage locations per typewriter function, one for the print order and one for a halt to prevent jamming the typewriter. When the alphanumeric output subroutine is employed, each storage location following subroutine entry contains four pairs of hexadecimals. The central six bits of each pair are extracted by the subroutine and a print order synthesized and executed. This process continues until the pair 7q (or vq) is encountered, whereupon the subroutine returns control to the location following the one containing the 7q.

The coding sheets of Appendix B list the hexadecimal codes. They may be recognized as always being preceded by a r6000, u6000 pair. The flow charts translate the codes into the typewriter functions performed.

The data input subroutine reads data in binary coded decimal, converts it to binary, and stores it in the memory. Upon entry, it examines the quantity in the accumulator which must be a hexadecimal code word. The first three digits of this code word state how many words are to be stored, the next digit describes their scaling, and the last four state where they are to be stored. Whenever this subroutine is invoked by the program of Appendix B, the codeword is of the form 0015aaaa which is interpreted as "input one word, consider the decimal point to be located two places from the right, convert it to binary at a q of 17, and store it at the hexadecimal address aaaa."

[redacted]

 [redacted]

APPENDIX B

CODING AND FLOW CHARTS

[redacted]

This appendix presents the coding and flow charts of the fixing program. Instructions are presented with decimal addresses, data is presented in either decimal or hexadecimal as seems appropriate. Addresses of 0000 (except for $_z$0000) are always variable addresses. Decimal data may be recognized by a plus or minus sign; hexadecimal data is preceded by a comma. In the flow charts, solid lines indicate the path of control, dotted lines indicate address modification.

 [redacted]

[redacted]


```
3800  r6000 u6000 ,20207f5f ,725f2246
3804  ,32063246 ,2q067q00 c6356 p0000
3808  i0000 d3910 a3907 y3904 a3908 y3861 a3908 y3849
3816  a3908 y3853 a3908 y3858 a3908 y3901 r6000 u6000
3824  ,200j725f ,225f522f ,4f102608 ,067q0000
3828  b3912 r6221 u6204 b3913
3832  d3911 h3914 r6000 u6000
3836  ,200j4632 ,5j225f52 ,2f4f1026 ,08067q00
3840  b3912 r6221 u6204 b3913 d3911 h3913 b3914 r5949
3848  u5900 h0000 b3914 r5949 u5904 h0000 h3915 b3913
3856  r5949 u5900 h0000 m3915 d3909 h0000 b3913 r5949

3900  u5904 h0000 m3915 d3909 h0000 z0000 u3800 z4148
3908  z0016 +.50000000 +.25000000 ,00800000
3912  ,00152734 z0000 z0000 z0000
```

Fig. B1 - Station loading program - coding

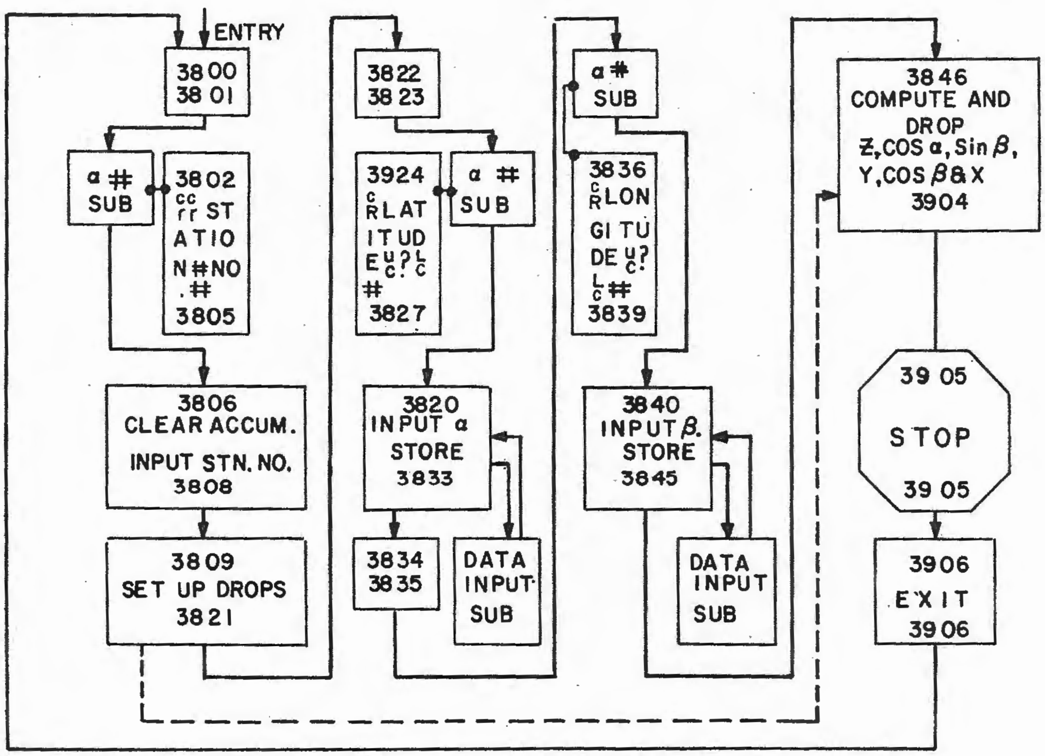


Fig. B2 - Station loading program - flow chart

[redacted]

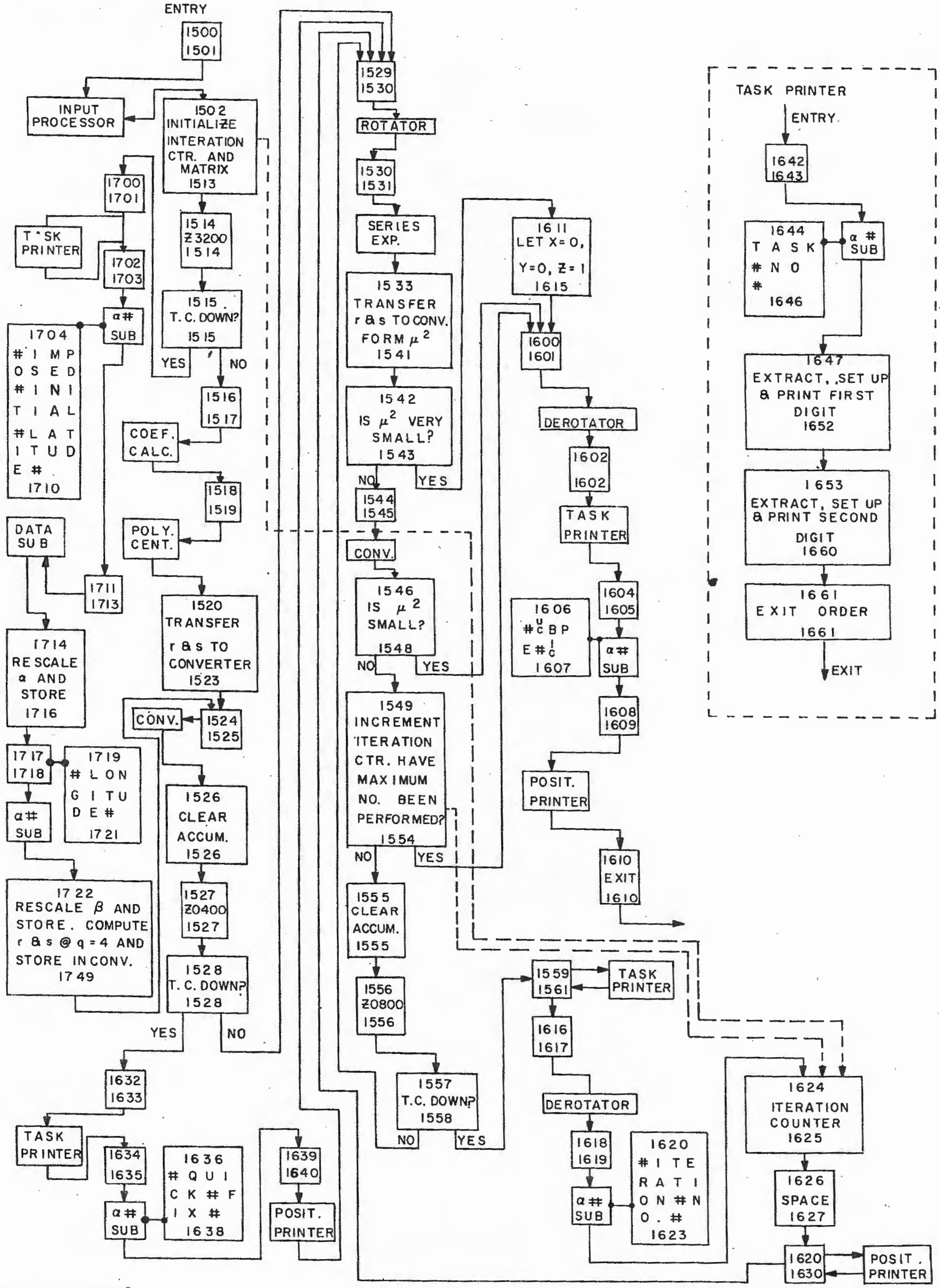
 [redacted]

```
1500  r0725 u0500 b1759 h1624 b3916 h1127 h1131 c1123
1508  h1130 h1124 h1125 h1126
1512  h1129 h1128 z3200 800t1700
1516  r0861 u0800 r2652 u2500 b2662 h1407 b2663 h1414
1524  r1459 u1303 c6362 z0400
1528  800t1632 r1112 u0903 r3958
1532  u1800 b3962 h1407 m1407 h1755 b3963 h1414 m1414
1540  a1755 h1755 s1757 t1611 r1459 u1300 b1755 s1756
1548  t1600 b1624 a1758 c1624 s1624 a1754 t1600 c6363
1556  z0800 800t1559 u1529 r1661
1560  u1642 u1616 z0000 z0000

1600  r1154 u1116 r1661 u1642
1604  r6000 u6000 ,06100f42 ,4f06087q
1608  r2754 u2700 u3000 b3917 c2863 h2861 h2862 u1600
1616  r1154 u1116 r6000 u6000
1620  ,06225f4f ,1f725f22 ,46320632 ,462q067q
1624  p0600 z0000 p0300 z0000 r2754 u2700 u1529 z0000
1632  r1661 u1642 r6000 u6000
1636  ,06745222 ,6f6j0654 ,224q067q r2754
1640  u2700 u1529 r6000 u6000
1644  ,5f727f6j ,0632462q ,067q0000 b0730
1648  e1752 d3921 a1759 h1652 p0200 b0730 e1753 d3925
1656  z0000 a1759 h1659 p0200 z0000 u0000 z0000 z0000

1700  r1661 u1642 r6000 u6000
1704  ,06223f42 ,467f4f2f ,06223222 ,5f22720j
1708  ,060j725f ,225f522f ,4f067q00 b1751
1712  r6221 u6204 b1763 d3924
1716  h1762 r6000 u6000 ,060j4632
1720  ,5j225f52 ,2f4f067q b1751 r6221
1724  u6204 b1763 d3924 h1763 b1762 r5949 u5900 h1761
1732  b1762 r5949 u5904 m3919 h1760 b1763 r5949 u5904
1740  m1760 d1761 h1407 b1763 r5949 u5900 m1760 d1761
1748  h1414 u1524 z0000 ,001511wj
1752  z0156 ,0000001q p2000 ,46q334wq
1756  ,00008000 z0032 z0400 p0200
1760  z0000 z0000 z0000 z0000
```

Fig. B3 - Fixing program assembly - coding

[redacted]

[redacted]



[redacted]

Fig. B4 - Fixing program assembly - flow chart

[redacted]

```
0500  c6329 p0030 i0031 d0739 c0730 h0736 b0756 h0742
0508  b0744 y0553 b0746 y0554 a0748 y0556 a0750 y0558
0516  a0752 y0706 a0754 y0702 a0749 y0658 a0751 y0552
0524  b0553 a0761 u0531 z0647 z0649 z0000 z0000 y0553
0532  a0754 y0555 a0749 y0557 a0751 y0602 a0753 y0561
0540  a0755 y0563 c6307 p0001 i0002 t0547 u0717 h0762
0548  e0763 d0760 u0552 z0000 h0000 b0000 h0000 b0000
0556  h0000 b0000 h0000 h0729 u0561 b0000 h0741 b0000

0600  h0743 u0602 b0000 h0530 b0762 e0734 d0747 s0735
0608  t0630 s0735 t0625 s0735 t0619 r5949 u5904 h0737
0616  b0527 y0646 u0633 r5949 u5904 h0737 b0528 y0646
0624  u0633 r5949 u5900 c6300 s6300 u0615 r5949 u5904
0632  u0621 b0729 m0737 d0739 c6316 s6316 m6316 a0727
0640  r5750 u5700 h6317 b0737 m0530 d6317 u0000 c6319
0648  s6319 h0757 u0652 z0000 c6318 s6318 m6318 a0727
0656  r5750 u5700 h0000 b0757 m0743 d0740 u0702 z0000

0700  z0000 z0000 c0000 s0757 m0741 d0740 h0000 b0736
0708  a0758 h0736 b0742 s0761 t0722 h0742 b0554 a0758
0716  u0511 b0742 s0761 t0722 h0742 u0524 b0736 s0759
0724  h0731 u0000 z0000 +.25000000
0728  z0000 z0000 z0000 z0000
0732  z0000 +.25000000 ,0www0000 +.17578125
0736  z0000 z0000 z0000 +.50000000
0740  +.50000000 z0000 z0000 z0000
0744  z4147 z0000 z4000 +.12500000
0748  z0016 z0016 z0016 z0016
0752  z0016 z0016 z0016 z0016
0756  z0015 z0000 z0001 z0001
0760  b0000 z0001 z0000 ,0000www0
```

Fig. B5 - Input processor - coding

[redacted]

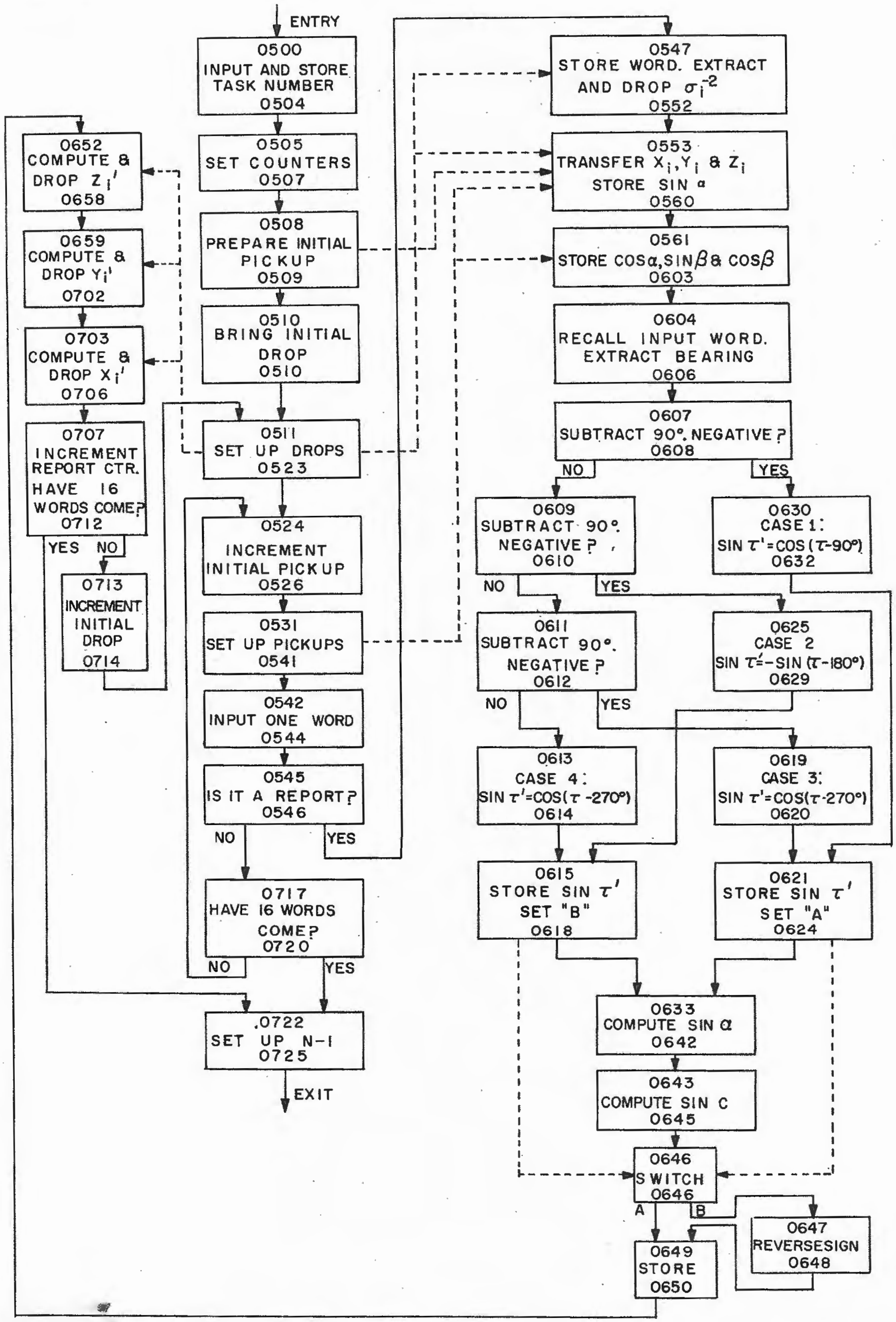


Fig. B6 - Input processor - flow chart

[redacted]

[redacted]

0800	b0736	s0761	h6345	b0746	y0847	y0831	a0749	y0836
0808	y0851	a0752	y0839	y0829	a0755	y0828	y0850	a0751
0816	y0838	y0846	a0754	y0832	y0835	a0843	y0844	a0752
0824	y0858	a0754	y0855	u0835	b0000	m0000	h6359	b0000
0832	m0000	s6359	u0857	b0000	m0000	h6302	b0000	m0000
0840	s6302	d0739	u0844	z0200	h0000	u0846	b0000	m0000
0848	h6363	u0850	b0000	m0000	s6363	d0739	u0855	h0000
0856	u0828	d0740	h0000	b6345	s0761	t0000	h6345	b0847
0900	a0761	u0804						

Fig. B7 - Coefficient calculator - coding

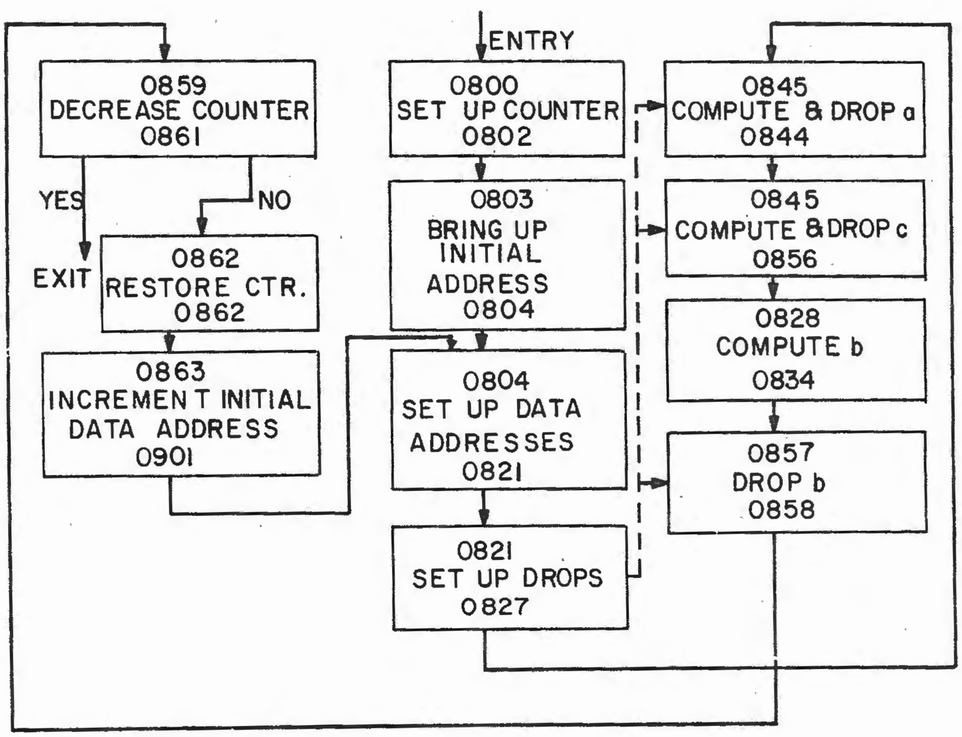


Fig. B8 - Coefficient calculator - flow chart

[redacted]

[redacted]

2500	c6350	c6354	c6355	c6341	b0736	s2654	t2509	b2655
2508	u2510	a2656	a2657	h6362	a2641	h6308	b2657	y2547
2516	a0752	y2532	a0754	y2534	s2658	y2536	b2547	a2659
2524	y2539	a0754	y2541	a0749	y2543	s2658	y2545	u2532
2532	b0000	h6332	b0000	h6350	b0000	h6316	u2539	b0000
2540	h6319	b0000	h6321	b0000	h6348	b0000	h6318	b0000
2548	h6347	m6321	h6322	b6316	m2645	m6318	h6311	a6341
2556	c6341	u2560	d6346	u2638	s6332	m6319	u2600	z0000
2600	a6322	h6346	u2603	b6332	m6348	c6323	s6321	m6350
2608	a6323	m2645	m6311	u2558	h6354	b6319	m6350	c6351
2616	u2618	z0000	s6347	m6348	u2622	z0000	a6351	m2645
2624	d6346	m6311	a6355	h6355	b2539	s6308	t2642	u2632
2632	b2547	s6362	t2636	u2646	a6308	u2515	a6354	u2612
2640	z0000	z0001	a2641	a6308				
2644	u2524	+.06250000	b6354	d6341				
2648	h2662	b6355	d6341	h2663	u0000	z0000	z0007	z0004
2656	z0005	b4316	z0216	z0001	z0000	z0000	z0000	z0000

Fig. B9 - Polygon center routine - coding

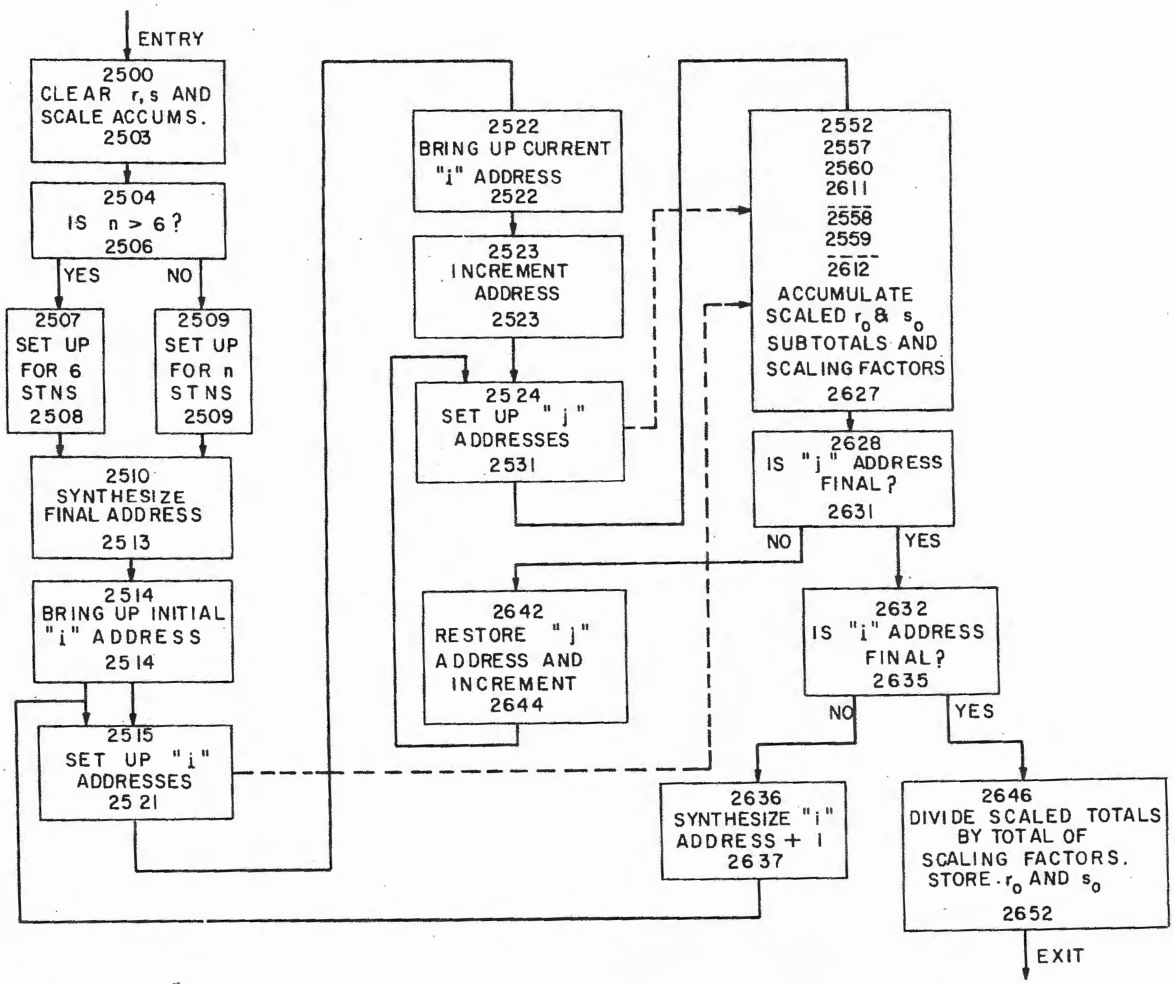


Fig. B10 - Polygon center routine - flow chart

[redacted]

[redacted]

1300	r1317	u1306	u1318	r1317	u1306	u1336	b1407	m1407
1308	h1401	b1414	m1414	h1402	a1401	h1403	r5750	u5700
1316	h1404	u0000	b1403	m3917	a3917	h1405	b3917	d1405
1324	r5750	u5700	h0950	m1407	h0944	b0950	m1414	c0947
1332	s0950	m1404	c0957	u1432	s3929	t1416	u1339	b1403
1340	a3924	h1400	b3924	d1400	r5750	u5700	h0950	b1407
1348	m0950	d3920	h0944	b1414	m0950	d3920	c0947	s0950
1356	m1404	d3920	c0957	u1432	z0000	z0000	z0000	z0000

1400	z0000	z0000	z0000	z0000	z0000	z0000	z0000	z0000
1408	s1401	s1415	u1443	s1402	s1415	u1453	z0000	z0001
1416	b3917	c2863	h2861	h2862	b3916	h0950	h0951	c0961
1424	h0944	h0947	h0954	h0957	h0958	u1459	z0000	z0000
1432	b0944	m3917	h2861	b0947	m3917	h2862	b0950	m3917
1440	h2863	b1403	u1408	t1460	b1407	d1404	h0954	m0950
1448	c0958	s0954	h0954	b1403	u1411	t1462	b1414	d1404
1456	h0951	m0950	h0961	u0000	b3916	u1446	b3916	u1456

Fig. B11 - Converter - coding

[redacted]

[redacted]

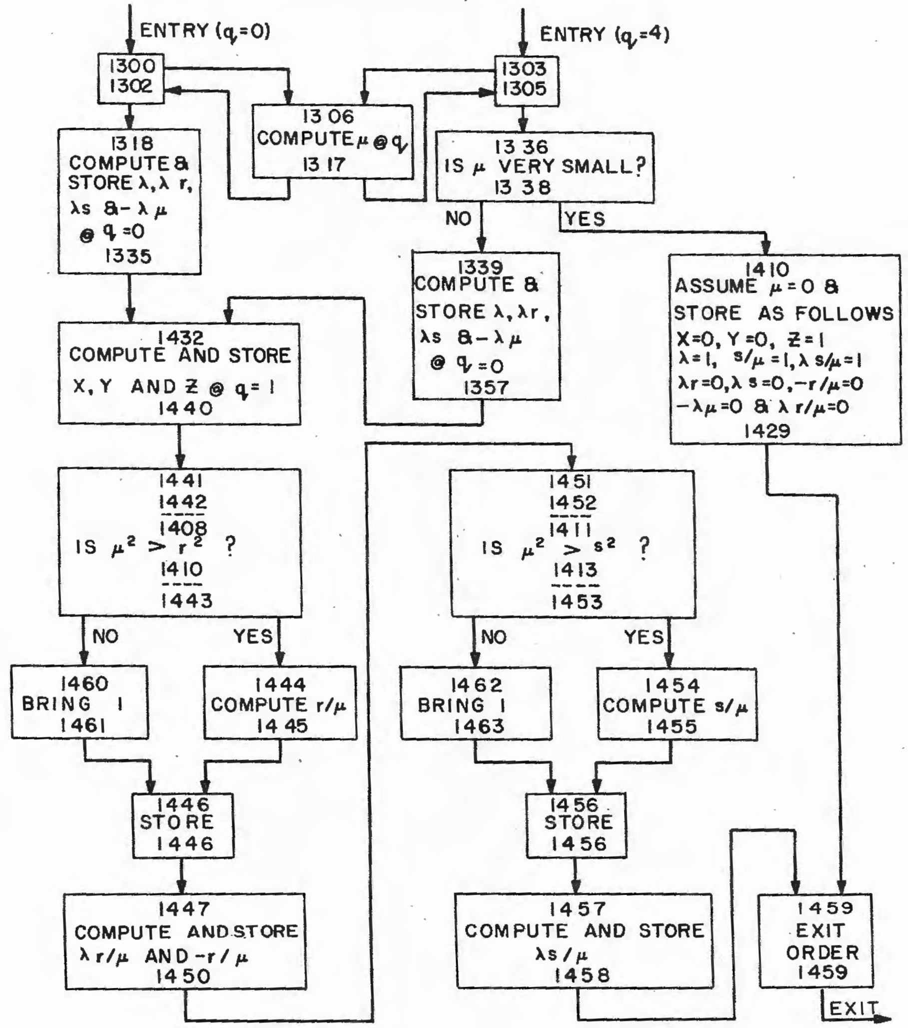


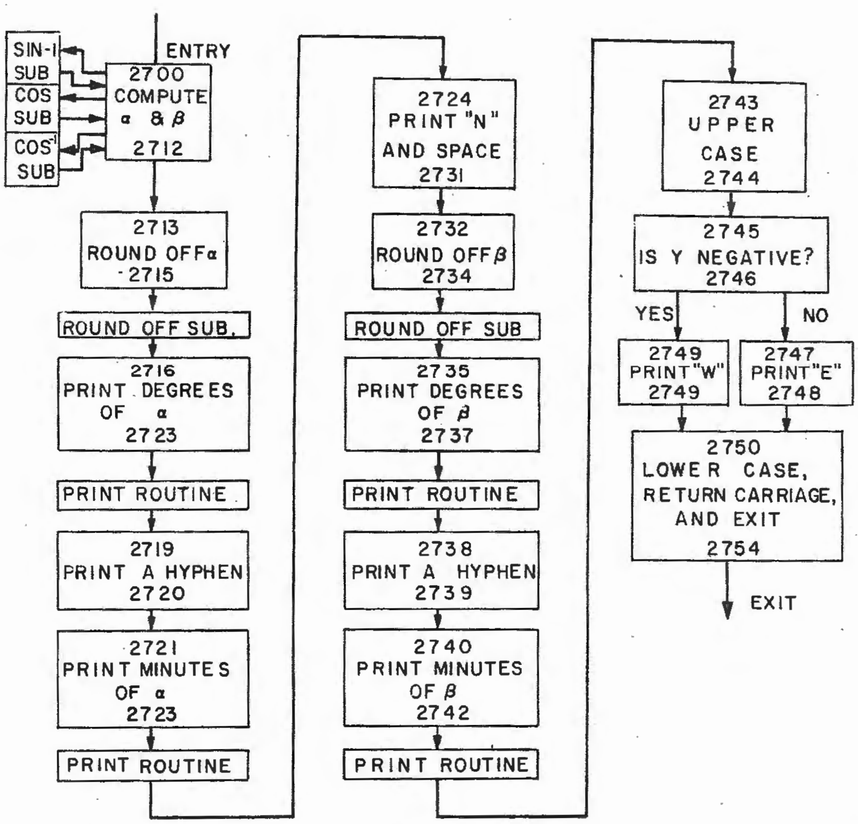
Fig. B12 - Converter - flow chart

[redacted]

[redacted]

```
2700  b2863 r5621 u5600 h2853 r5949 u5904 u2847 b2861
2708  m2858 d2856 r5621 u5811 h2855 b2853 r2808 u2755
2716  b6327 r2824 u2815 p0755 z0000 b6330 r2824 u2815
2724  p0860 z0000 p2562 z0000 p0400 z0000 p0302 z0000
2732  b2855 r2808 u2755 b6327 r2824 u2809 p0703 z0000
2740  b6330 r2824 u2815 p0808 z0000 b2862 t2749 p3712
2748  u2750 p6214 z0000 p0416 z0000 p1618 u0000 h6327
2756  e2842 d2843 h6330 e2845 s2846 t2808 b6330 a2849

2800  u2801 h6330 s2852 t2808 b6327 a2849 c6327 h6330
2808  u0000 h6363 b2860 h6307 r2839 u2825 u2816 h6363
2816  b2859 h6307 r2839 u2825 b2849 h6307 r2839 u2825
2824  u0000 b2840 h2837 b6363 s6307 t2835 h6363 b2837
2832  a2854 h2837 u2827 a6307 h6363 p3405 z0000 u0000
2840  p0205 z0000 ,003wwwwj +.01666666
2844  z0000 ,00200000 ,00100000 a2857
2848  u2850 ,00400000 h2856 u2707
2852  +.11718750 z0000 z0400 z0000
2856  z0000 z0001 +.50000000 +.01953125
2860  +.19531250 z0000 z0000 z0000
```

Fig. B13 - Position printer - coding



[redacted]

Fig. B14 - Position printer - flow chart

[redacted]

[redacted]

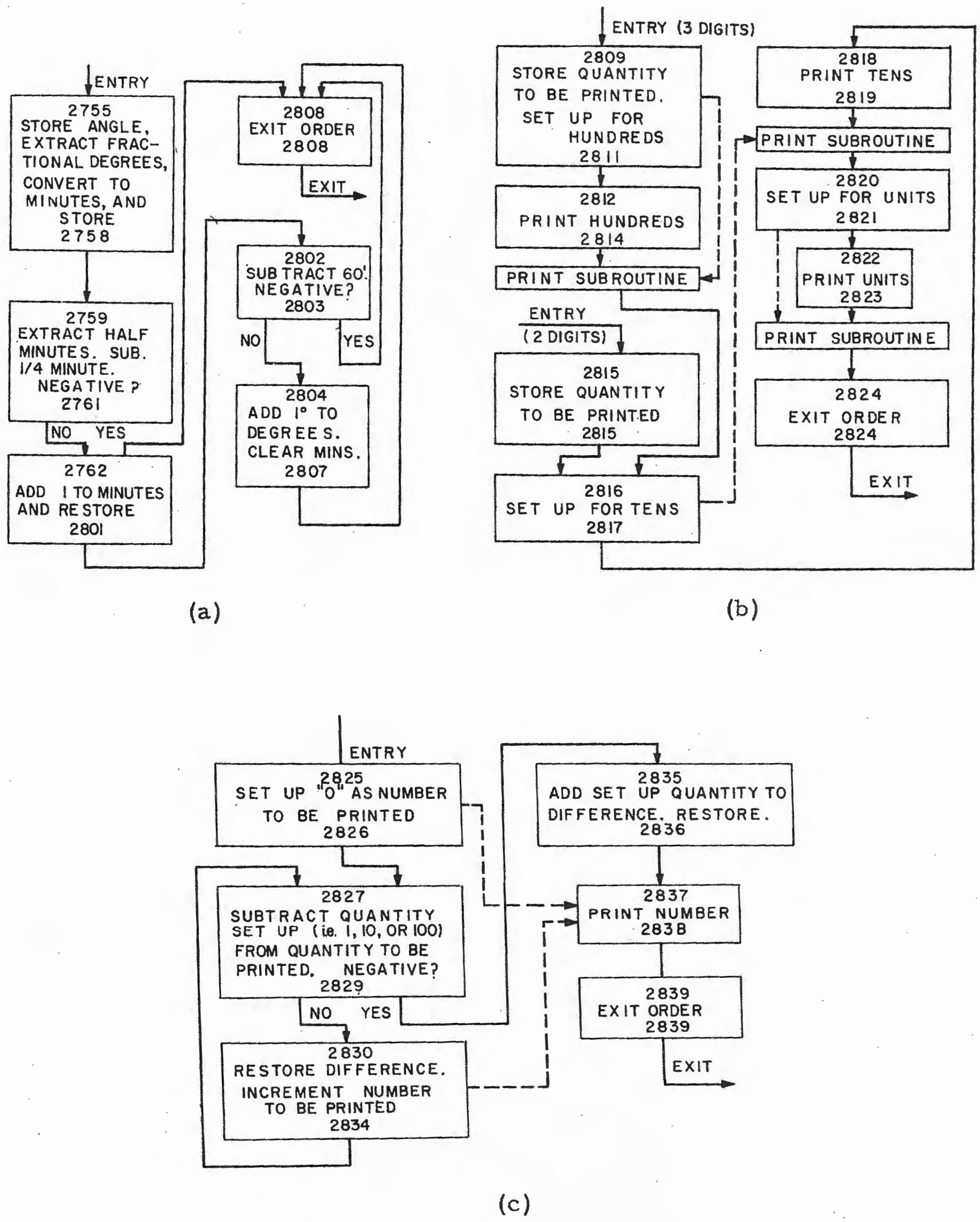


Fig. B14 - Position printer - flow chart

[redacted]

DECLASSIFIED [redacted]

0903				b0946	y1054	u0906	b0949	y1050
0908	b0731	h6334	b0960	y1026	y0927	a0749	y1036	y0930
0916	a0752	y1046	y0933	u0927	b0956	y1050	b0731	h6334
0924	b0939	u0911	z0000	b0000	h0943	u0930	b0000	h0953
0932	u0933	b0000	h0963	u1000	z0001	z0000	z0000	z4048
0940	z0000	z1113	z0002	z0000	z0000	z1123	z0911	z0000
0948	z0001	z0920	z0000	z0000	z0000	z0000	z0000	z0000
0956	z1055	z0000	z0000	z0000	z4000	z0000	z0000	z0000
1000	b0943	m0951	h6338	b0953	m0954	h6320	u1007	b0943
1008	m0958	h6352	b0953	m0961	h6355	u1014	b0943	m0944
1016	h6302	b0953	m0947	h6362	b0963	m0950	h6301	b6338
1024	a6320	u1026	h1129	b0963	m0957	u1031	z0001	a6352
1032	u1033	a6355	u1036	z0001	h0000	u1040	z0000	z0001
1040	b6362	u1043	z0000	a6301	a6302	u1046	h0000	u1048
1048	b6334	s1035	t0000	h6334	b1026	a1039	u0000	b0941
1056	y1054	b0942	h6334	b0945	y1104	y1026	a0948	y1106
1100	y1036	a1030	y1108	y1046	b0000	h0943	b0000	h0953
1108	b0000	h0963	r1050	u1000	u0000	b1108	a0936	u1060

Fig. B15 - Rotator - coding

[redacted]

DECLASSIFIED

[redacted]

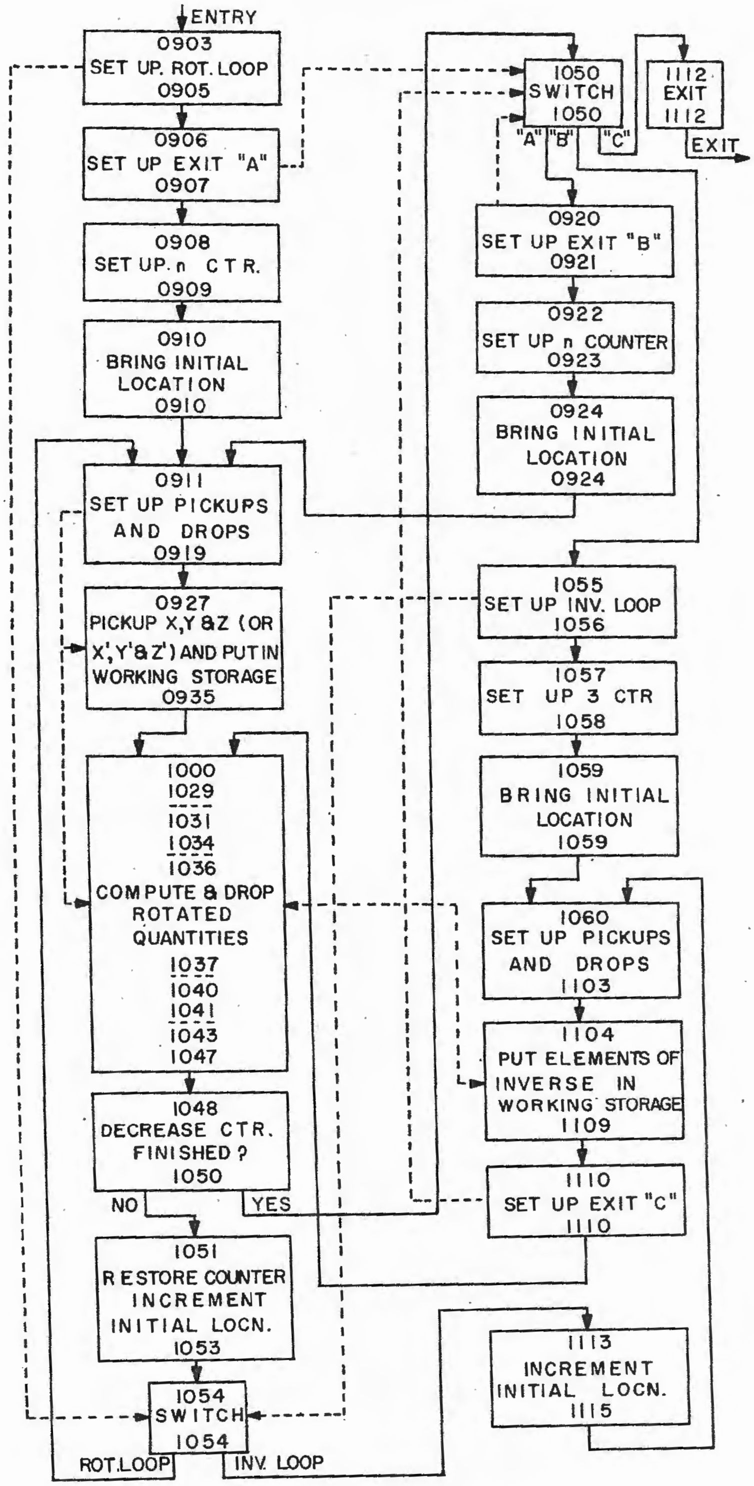


Fig. B16 - Rotator - flow chart

[redacted]

DECLASSIFIED [redacted]

```
1800  r0861 u0800 b0731 h6314 b2005 y1841 a0749 y1843
1808  a0751 y1845 a2006 y1847 a0755 y1849 a0750 y1851
1816  a0752 y1960 a0754 y1934 a0749 y2007 a0751 y1938
1824  a0753 y2040 a0755 y1942 a0750 y2101 a0752 y1946
1832  a0754 y2148 a0749 y1950 s2015 y1838 b0000 h6303
1840  u1841 b0000 h6307 b0000 h6311 b0000 h6351 b0000
1848  h6306 b0000 h6330 b0000 h6332 m6332 h6340 b6306
1856  m6306 h6308 b6330 m6330 a6308 a6340 h6304 b6351

1900  m6351 c6344 s6344 a2146 d2147 h6358 m6358 h6329
1908  m6358 h6338 u1932 m6303 u2007 b6306 m6332 d2144
1916  h6336 b6330 m6332 d2162 h6323 b6307 m6351 d2009
1924  h6346 b6311 m6351 d2163 h6300 b6332 m6332 u1953
1932  b6304 m6329 h0000 u1938 z0001 z0000 h0000 b6304
1940  m6338 u1942 h0000 u1946 z0001 z0000 h0000 u1950
1948  z0000 +.12500000 h0000 u1913
1952  z0000 h6339 m6346 d2141
1956  h6342 b6336 m6358 u2124 h0000 b6339 m6300 d2142

2000  h6343 b6323 m6358 a6343 u1911 z4000 z0248 h0000
2008  u2016 +.25000000 +.25000000 +.12500000
2012  m6303 u1960 +.25000000 z0448
2016  b6346 m6339 m6346 h6305 b6307 m6307 s2144 m6339
2024  h6344 u2026 b6336 m6346 m2014 a6344 h6302 b6306
2032  m6306 m2162 m6358 a6302 m6358 a6305 m6303 d2011
2040  h0000 b6300 m6300 m6339 h6302 b6311 m6311 s2162
2048  m6339 h6318 b6300 m6323 m2010 a6318 h6333 b2141
2056  m6330 m6358 m6330 a6333 m6358 a6302 m6303 d1949

2100  u2101 h0000 b6307 m6332 m6311 m6332 d2143 h6357
2108  m6351 m6351 d2160 h6340 b6336 m2161 m6300 h6344
2116  b6323 m6346 m2161 h6305 b6306 m6330 m2143 u2129
2124  a6342 m6303 u1960 z0000 z0000 m6358 u2133 z0000
2132  z0000 a6305 a6344 a6357 m6358 a6340 m6303 d2011
2140  u2148 +.25000000 +.25000000 +.50000000
2144  +.25000000 +.50000000 +.24999999 +.25000000
2148  h0000 b6314 s1936 t2200 u2156(b6307 m2361 u2211)
2156  h6314 b1841 a1944 u1805
2160  +.06250000 +.25000000 +.25000000 +.25000000
```

Fig. B17 - Series expander (Part 1) - coding

[redacted]

DECLASSIFIED

[redacted]

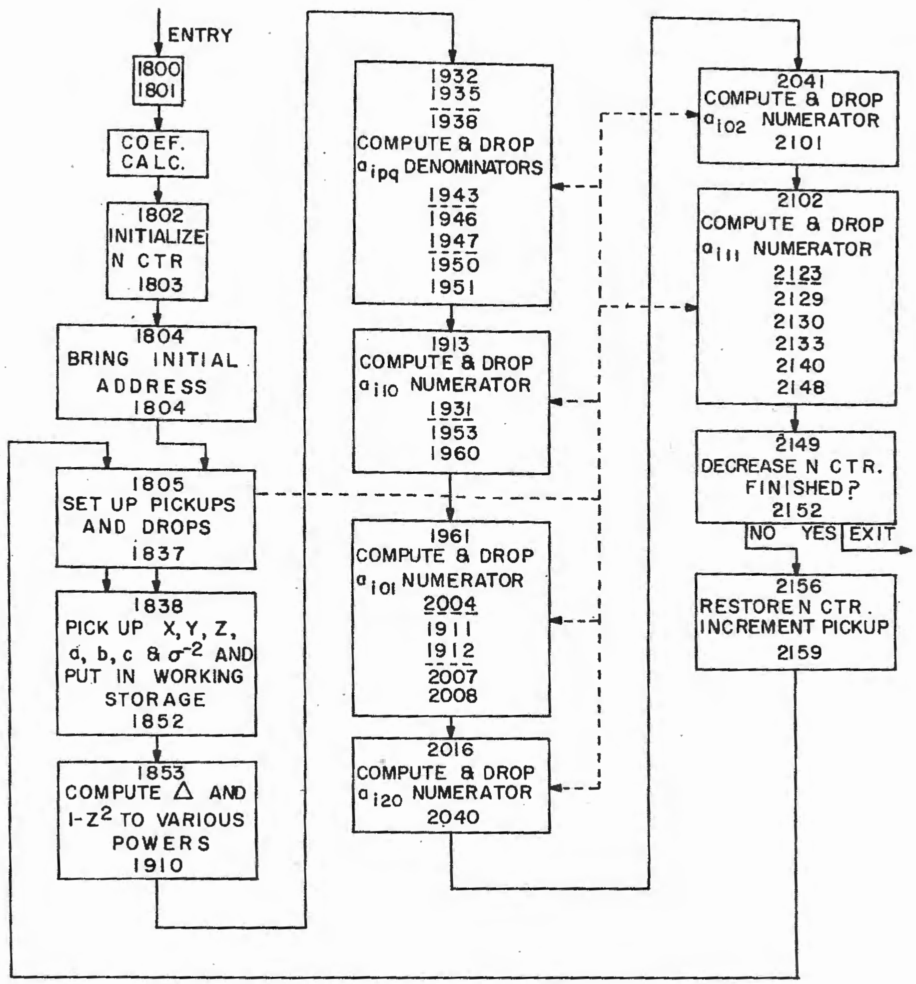


Fig. B18 - Series expander (Part 1) - flow chart

[redacted]

 [redacted]

```
2153                                            b6307 m2361 u2211

2200   b2214 c6320 h6321 h6307 h6308 b6320 a6321 y2222
2208   a0751 y2224 u2222 h6307 u2314 z4632 z4400 b6302
2216   m6302 h6303 b6306 m2362 m6306 u2230 b0000 h6302
2224   b0000 u2227 z4620 h6306 u2215 z0001 s0761 s6303
2232   t2344 b6302 d6306 a6307 h6307 m6307 s2362 t2242
2240   r2354 u2153 b6321 a2229 h6321 s0736 t2205 u2248
2248   b6320 s2214 m2343 a2213 y2256 a2331 y2258 b6307
2256   h0000 b6308 h0000 b6320 a2332 h6320 s2226 t2400

2300   u2434 m6306 s2362 t2310 b6302 m2361 h6302 r2354
2308   u2153 u2215 b6306 d2361 h6306 u2307 b6320 a0736
2316   h6314 u2358 y2333 y2341 a0749 y2336 y2346 u2333
2324   h6310 m6310 s2362 t2338 b6300 m2361 u2341 z0001
2332   z0032 b0000 u2335 h6300 b0000 u2324 b6310 d2361
2340   u2346 h0000 u2347 +.06250000
2344   b6306 u2301 h0000 b2333
2348   e2363 s6314 t2355 b6308 a2331 h6308 u0000 b2341
2356   a2331 u2318 b2222 a2331
2360   u2318 +.50000000 +.25000000 z6363

2400   c6350 u2202
```

Fig. B19 - Series expander (Part 2) - coding

[redacted]

[redacted]

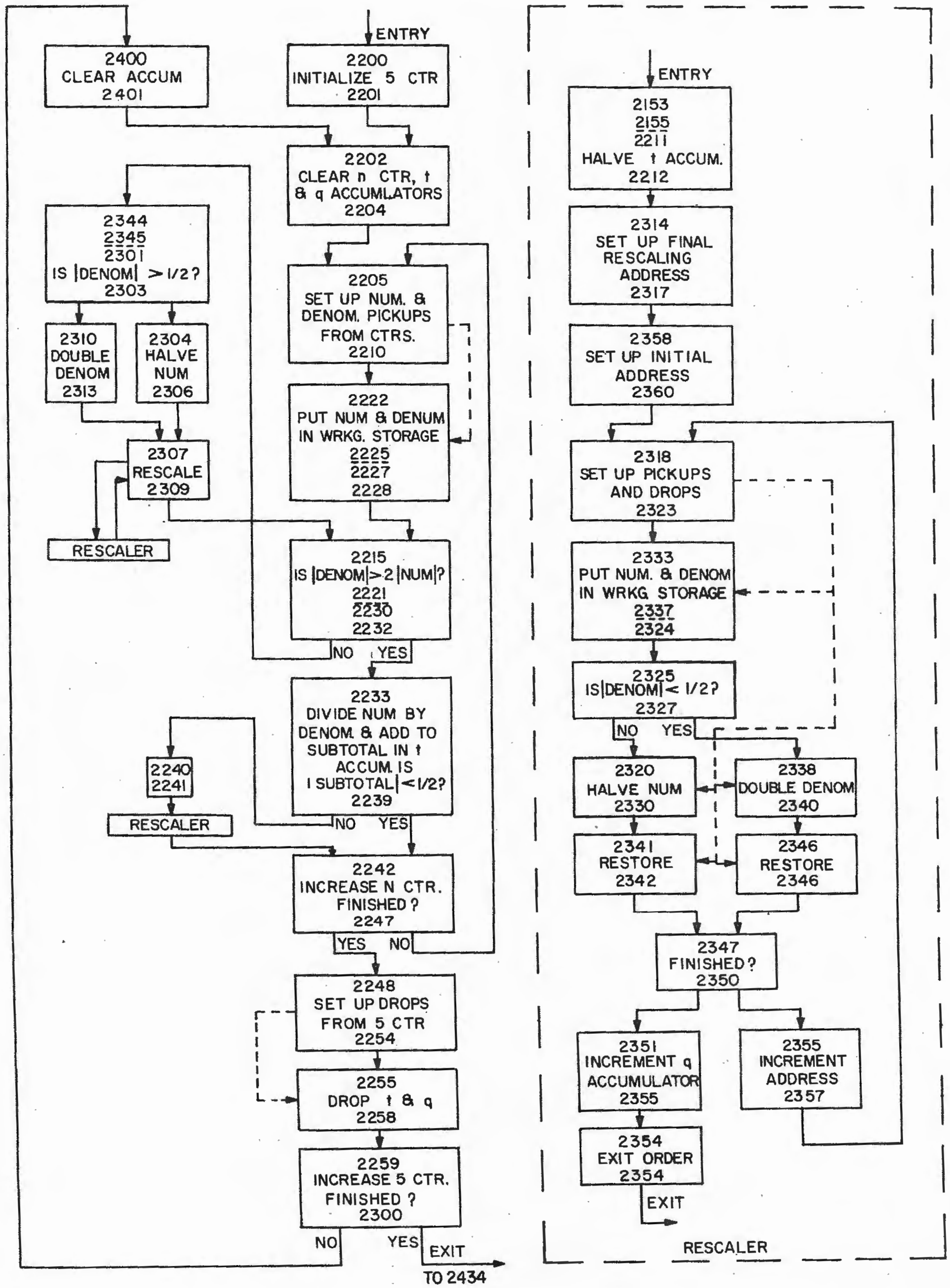


Fig. B20 - Series expander (Part 2) - flow chart

[redacted]

[redacted]

```
2402                   z0000 b2402 s4633 t2415 u2407 b2402
2408 s4637 t2415 u2411 b2402 s4641 t2415 u2419 b2436
2416 a2445 u2435 z4633 b2402 s4635 t2415 u2423 b2402
2424 s4639 t2415 u2442 b2463 y2450 u2430 y2452 a2440
2432 y2447 u2446 b2418 y2436 b4639 h2402 u2403 z3916
2440 z0001 z0004 b2441 h6358 u2427 z0002 b2402 s0000
2448 a2439 y2451 b0000 m0000 h0000 b6358 s2440 t3932
2456 h6358 u2458 b2447 a2440 u2428 z0000 z0000 z4632


3916 +.99999999 +.50000000 +.25000000 +.12500000
3920 +.06250000 +.03125000 +.01562500 +.00781250
3924 +.00390625 +.00195312 +.00097656 +.00048828
3928 p0000 i0000 y0000 b0000
3932 b4636 m4638 d3918 c3959 s4640 m4640 a3959 h3960
3940 b4640 m4634 c3959 s4638 m4632 d3917 a3959 d3960
3948 h3962 b4640 m4632 c3959 s4636 m4634 d3917 a3959
3956 d3960 h3963 u0000 z0000 z0000 z0000 z0000 z0000
```

Fig. B21 - Series expander (Parts 3 and 4) - coding

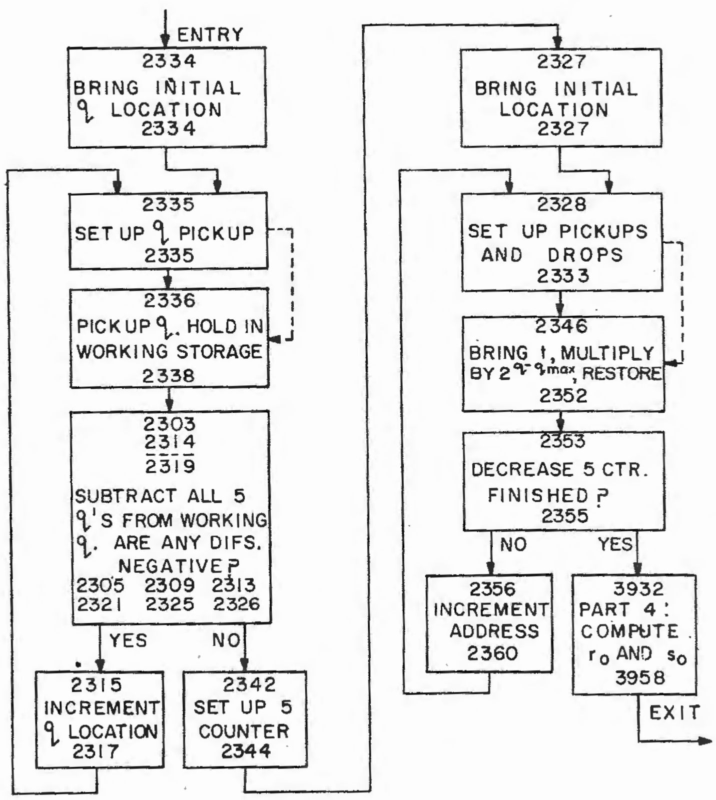


Fig. B22 - Series expander (Parts 3 and 4) - flow chart

[redacted]

[redacted]


```
1116  b2861 h6328 b2862 h6331 b2863 h6334 u1132

1132  b1160 h1151 b1159 a0759 y1141 a0759 y1144 a0761
1140  y1147 b0000 m6328 h6322 b0000 m6331 h6332 b0000
1148  m6334 a6332 a6322 h0000 b1151 s1158 t0000 h1151
1156  b1147 u1135 ,00wwwwwj z1122 ,020j1jw4
```

Fig. B23 - Derotator - coding

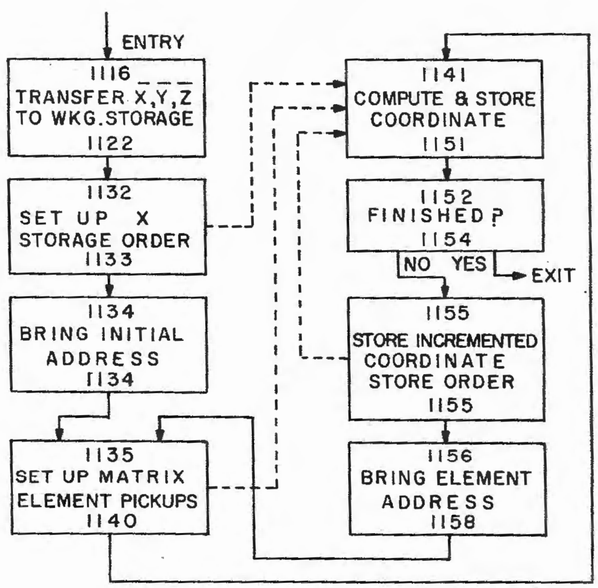


Fig. B24 - Derotator - flow chart

[redacted]

 [redacted]

```
3000  b4636 s4638 h3233 t3007 b3231 y3019 u3009 b3230
3008  y3019 b3233 m3233 h3234 b4640 m4640 a3234 r5750
3016  u5700 h3235 b3233 u0000 a3235 h3236 u3025 s3235
3024  u3021 d4640 h3238 m3238 h3239 s3229 t3042 b3239
3032  m3918 a3918 h3240 b3918 d3240 r5750 u5700 u3040
3040  h3241 u3044 b3916 u3040 b4638 m3239 s3236 a4636
3048  r5750 u5700 m3241 h3242 b4636 m3239 a4638 a3236
3056  r5750 u5700 h3243 u3100 z0000 z0000 z0000 z0000

3100  b2402 a3228 h2402 e3228 s3227 t3110 b3226 m3225
3108  h3244 u3112 b3226 h3244 b2402 m3917 a3224 y3116
3116  b0000 m3244 h3245 d3242 h3246 b3245 d3243 h3247
3124  r6000 u6000 ,7f4f721f ,6f620672
3128  ,1f4f7206 ,6f461f32 ,4f1f7f06 ,207q0000
3132  b3246 h3248 b3247 h3249 r3218 u3200 b3246 c3248
3140  s3247 h3249 r3218 u3200 c6363 s3246 c3248 s3247
3148  h3249 r3218 u3200 b3247 c3249 s3246 h3248 r3218
3156  u3200 p1600 z0000 p1600 z0000 z1600 u1500 z0000

3200  b3249 m3238 a3248 m3241 a3962 c1407 s3248 m3238
3208  a3249 m3241 a3963 h1414 r1459 u1300 r1154 u1116
3216  r2754 u2700 u0000 z0000 z0000 z0000 z0000 z0000
3224  z3916 +.70710599 +.03403389 ,00000002
3228  z0001 z0008 z3020 z3023
```

Fig. B25 - Search area program - coding

 [redacted]

DECLASSIFIED

[redacted]

UNCLASSIFIED

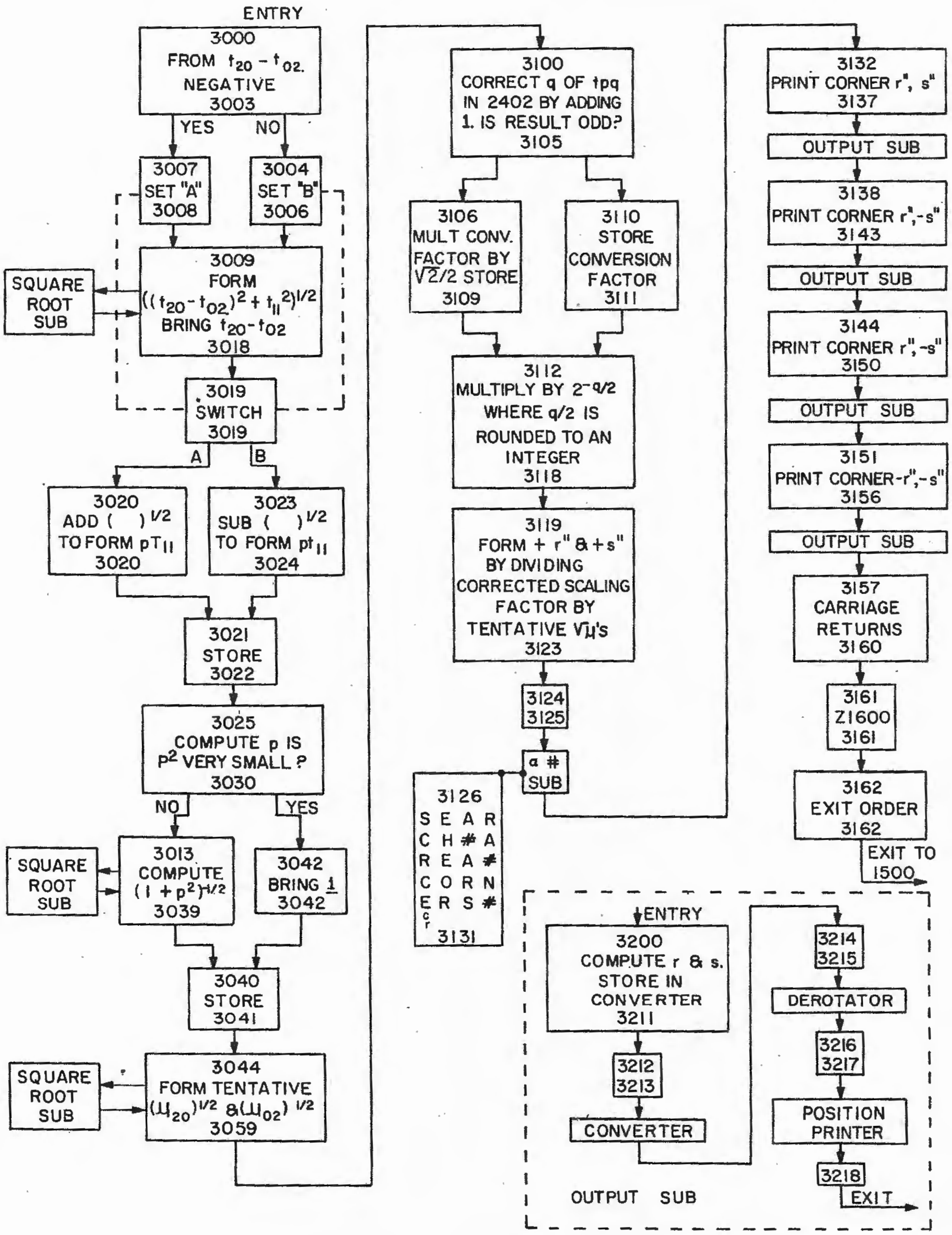


Fig. B26 - Search area program - flow chart

[redacted]

DECLASSIFIED

[redacted]

DISTRIBUTION

		Copy No.
CNO		
Attn:	Code Op-30G/640 (3801 Nebraska Ave., N.W.)	1 - 12
NRL		
Attn:	Code 2027, Documents	13 - 14
	Code 5430	15 - 30

[redacted]

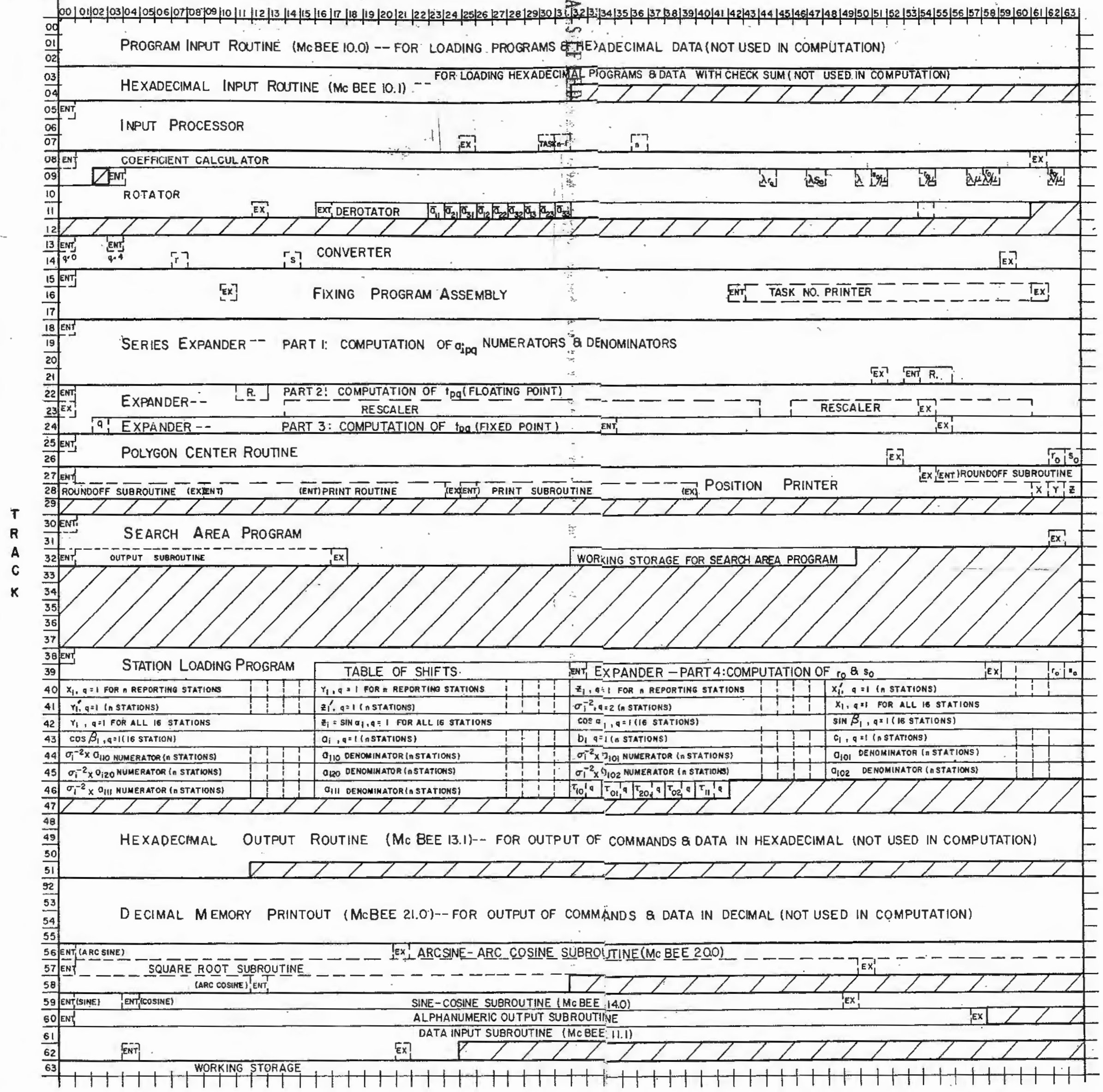


Fig. 7 - Storage allocation chart