acq #5708 RK

[redacted]

UNCLASSIFIED

CONTROL SYSTEMS LABORATORY

To Be Regraded to UNCLASSIFIED on Sept. 1962
(date)

Fast Collision Course Vectoring (U)

Report R-94

July, 1958

Copy 142 of 150 Copies

Contract DA-36-039-SC-56695

D/A Sub-Task 3-99-06-111

UNIVERSITY OF ILLINOIS · URBANA · ILLINOIS

[redacted]

UNCLASSIFIED

"The research reported in this document was made possible by support extended to the University of Illinois, Control Systems Laboratory, jointly by the Department of the Army (Signal Corps and Ordnance Corps), Department of the Navy (Office of Naval Research), and the Department of the Air Force (Office of Scientific Research, Air Research and Development Command), under Signal Corps Contract DA-36-039-SC-56695, D/A Sub-Task 3-99-06-111."

C O N F I D E N T I A L

FAST COLLISION COURSE VECTORING

by

Jerry C. McCall

Paul R. Peabody

Report R-94

July, 1958

CONTROL SYSTEMS LABORATORY
UNIVERSITY OF ILLINOIS
URBANA, ILLINOIS
Contract DA-36-039-SC-56695
D/A Sub-Task 3-99-06-111

Numbered Pages: 60

C O N F I D E N T I A L

Table of Contents

C O N F I D E N T I A L

Table of Contents

ABSTRACT

Most vectoring systems require the calculation of a "collision course", which is the course a friend must fly in order to simply collide with a bogey. This paper discusses a method of determining this course on a digital computer while the intercept is in progress. The course is generally expressed as an angle relative to north. This is treated as a two dimensional problem in the horizontal plane.

In practice this calculation must be made in a very short time because successive commands cannot be separated very much in time and between each of these commands the computer has many other duties to perform such as threat evaluation, weapons assignment, computation of vectors for other interceptors, etc. In addition to this time limitation for the collision course calculation only a small portion of the computer memory can be alloted to contain the instructions. Even as bigger and better computers are developed it is felt that these restrictions will remain because the complexity of the task will likewise increase.

To satisfy these restrictions one must find suitable approximations for the calculation. The input data is crude and the output data link can only transmit crude data; consequently, the accuracy of the approximations need only be consistent with this situation. In computer calculations one can very profitably exchange accuracy for reduced computing time. Within reason we are also willing to give up memory space for a reduction in computing time.

There are two natural formulations of the collision course equation; one may be called the algebraic approach and the other the trigonometrical. Both formulations are fully discussed in Appendix A of this report. It is found that the trigonometric formulation is much better suited to the use of fast approximate methods, and the appropriate equations may be summarized briefly as follows:

Let $\bar{R} : (R_x, R_y)$ be the range vector and its components from friend to bogey; $\bar{V}_f : (u_f, v_f)$ and $\bar{V}_b : (u_b, v_b)$ the velocity vectors and components of friend and bogey respectively, as shown in Figure 1.* $\bar{V}_c : (u_c, v_c)$ is the command velocity vector and components; i.e., the velocity vector which the friend must follow to collide with the bogey. The speed of the friend is assumed constant; that is, $|\bar{V}_c| = |\bar{V}_f|$.

The collision vector $\bar{V}_c$ lies at an angle θ_c from north. Two other angles are defined as shown in Figure 1. The angle from north to $\bar{R}$ is called θ, and the angle from $\bar{R}$ to $\bar{V}_c$ is called ϕ. The positive senses of θ_c, θ and ϕ are shown in Figure 1. Then the set of formulae defining θ_c are as follows.

$$\theta_c = \theta - \phi$$

$$\theta = 90^\circ - \arctan R_y/R_x$$

$$\phi = \arcsin \frac{R_x v_b - R_y u_b}{S_f R}$$

$$S_f = \sqrt{u_f^2 + v_f^2}$$

$$R = \sqrt{R_x^2 + R_y^2}$$

Obviously three special operations are required to solve these equations:

1. Arctangent operation.
2. Arcsine operation.
3. Given x and y, find $\sqrt{x^2 + y^2}$.

In addition to being essential here these operations occur frequently in problems involving plane geometry, and are therefore of considerable interest in themselves. Fast methods for performing each have been devised and these methods are discussed in Appendixes B, C, and D of this report. Multiplications and divisions are kept to a minimum

*All Figures have been placed at the end of this report.

and none of these methods involve iteration; therefore, in general they will be much faster than standard techniques no matter what computer is used. Furthermore, they are reasonably consistent among themselves in accuracy, computing time and memory space required.

The accuracy of the operations is summarized as follows:

1. The maximum error of the arctangent approximation is about 0.12 degrees.

2. The maximum error in the arcsine approximation is indicated in Table I.

Table I

The maximum error in ϕ, where ϕ = arcsine W

is	$.17^\circ$	for $|\phi| \leq 45^\circ$
	$.30^\circ$	$\leq 65^\circ$
	$.80^\circ$	$\leq 84^\circ$
	1.75°	$\leq 86^\circ$

The error is very small for small angles. For $|\phi| \geq 86^\circ$ the arcsine approximation is no longer used, and ϕ is fixed at $\pm 87.5^\circ$, depending on the sign of $\sin\phi$.

The arcsine approximation was devised with intention of making the error small for small angles, even at the sacrifice of accuracy for large angles, because cases for which ϕ is small correspond to physical cases of much greater interest. In addition it is shown in Appendix E that errors in calculating θ_c due to errors in input data may be large when ϕ is near $\pm$ 90 degrees no matter what method of calculation is used, and errors in the arcsine calculation are therefore not important in these extreme cases.

The accuracy of the collision course calculation is largely dependent on the accuracy of the routine which calculates the square root of sums of squares. A number of fast routines of varying accuracy were investigated, with an increase in accuracy being paid for by an increase in the number of computer operations required. The main properties of these various routines are summarized in Appendix D.

Since this report was motivated by the simulation work done on ILLIAC the computer operations will be discussed as they exist in ILLIAC. Figures will be given to relate ILLIAC to other machines. The ILLIAC program for calculating collision courses consists of a main routine and a subroutine. The main routine includes the arcsine and arctan and the subroutine is the square root of sum of squares routine. This breakdown was used because a number of square root of sum of squares routines are offered of various lengths and accuracies. Table 2 lists the maximum relative error of the square root of sums of squares subroutine, the total time of the collision course program, the total words in the program, and two figures of merit for each of the square root subroutines considered. The figures of merit consist of the maximum error $E(\phi)$ in θ_c, assuming exact input data, for $|\phi| \leq 30^\circ$ and for $|\phi| \leq 60^\circ$, for each of the square root of sums of squares subroutines.

Table 2

Square Root Subroutine	Max Rel. Error	Max Time	Illiac Words	Max $E(\phi)$ $\lvert\phi\rvert \leq 30^\circ$	Max $E(\phi)$ $\lvert\phi\rvert \leq 60^\circ$
1. "Revised Snyder"	.04 %	23.7ms	71	$.31^\circ$	$.48^\circ$
2. "8-line linear"	.1 %	17.1	96	$.35^\circ$	$.57^\circ$
3. "4-line linear"	.4 %	16.7	82	$.5^\circ$	1.08°
4. "Snyder"	1.0 %	16.9	60	$.9^\circ$	2.27°
5. "2-line linear"	1.4 %	15.9	69	1.28°	3.68°
6. "Nordsieck linear"	3.0 %	14.7	66	2.57°	7.61°
7. "$\frac{15}{16}L + 1/2\ S$"	6.25 %	13.8	62	5.18°	15.60°
8. "L + S/2 or L + S/4"	12.5 %	13.5	60	10.19°	30.96°

In order to interpret the time and number of words in Table 1 in terms of other computers, it must be noted that ILLIAC stores two instructions in one word, and requires about 700 microseconds to multiply, 800 to divide, 90 to add, 55 to transfer or store, and 16 to shift a number one binary place.

A brief mathematical analysis of the error is contained in Appendix E, as well as the results of testing the method for a large number of cases of input data. The program codes for the collision course calculation and the various square roots of sums of squares routines are listed and explained in Appendix G. The square root of sums of squares routines and the arcsine routine have been placed in the ILLIAC subroutine library.

Use has been made, especially in writing square root of sums of squares routines, of what is called in this paper "programmed multiplication", and a brief discussion of the method is contained in Appendix F.

APPENDIX A

DERIVATION OF FORMULA FOR COLLISION COURSE

The following notation is introduced, as indicated in Figure 1:

$\bar{R} : (R_x, R_y)$ -- range vector and components from friend to bogey.

$\bar{V}_b : (u_b, v_b)$ -- velocity vector and components of bogey.

$\bar{V}_f : (u_f, v_f)$ -- velocity vector and components of friend.

$\bar{V}_c : (u_c, v_c)$ -- collision course velocity vector and components of friend.

λ -- angle from range vector to bogey velocity vector.

ϕ -- angle from range vector to collision course velocity vector.

θ -- angle from North to range vector.

θ_c -- angle from North of collision course velocity vector.

The angles are shown as positive angles in Figure 1. The coordinate system is supposed fixed, with the x and y axes along the East and North directions respectively. Only the two-dimensional case will be considered.

The collision course velocity vector $\bar{V}_c$, and therefore θ_c, is determined by requiring

$\bar{R} + \bar{V}_b \tau = \bar{V}_c \tau$ for some τ = time to go, and

$|\bar{V}_c| = |\bar{V}_f|$.

Writing these two equations in terms of rectangular components of the vectors involved, and solving for u_c and v_c we get the algebraic solution:

$$u_c = \frac{\pm R_x \sqrt{(u_f^2 + v_f^2)(R_x^2 + R_y^2) - (u_b R_y - v_b R_x)^2} + R_y(u_b R_y - v_b R_x)}{R^2}$$

$$v_c = \frac{\pm R_y \sqrt{(u_f^2 + v_f^2)(R_x^2 + R_y^2) - (u_b R_y - v_b R_x)^2} - R_x(u_b R_y - v_b R_x)}{R^2}$$

Where R is the distance from friend to bogey, $R = \sqrt{R_x^2 + R_y^2}$.

Then the collision course, θ_c, is given by

$\theta_c = 90^\circ - \arctan v_c/u_c$

The ambiguity resulting from the $\pm$ sign before the radical will be discussed later.

Counting operations, it is seen that twelve multiplications, a square root routine, and an arctangent routine are required. The time for ILLIAC to perform these operations using standard subroutines is prohibitively long; therefore, the following trigonometrical solution is derived.

The method discussed in this paper consists of solving the FBC triangle by the law of sines and orienting the triangle in space by calculating the angle from North to the base FB. Thus,

$$\sin\phi = \frac{S_b}{S_f} \sin(\pi - \lambda) = \frac{S_b}{S_f} \sin\lambda$$

where $S_b = \sqrt{u_b^2 + v_b^2}$ and $S_f = \sqrt{u_f^2 + v_f^2}$ are the speeds of the bogey and the friend respectively. With R defined as above,

$$|\bar{R} \times \bar{V}_b| = RS_b \sin\lambda = R_x v_b - R_y u_b \text{ and}$$

$$\sin\phi = \frac{R_x v_b - R_y u_b}{S_f R}$$

Collecting these formulae leads to the set listed previously:

$$\theta_c = \theta - \phi$$

$$\theta = 90^\circ - \arctan R_y/R_x$$

$$\phi = \arcsin \frac{R_x v_b - R_y u_b}{S_f R}$$

$$S_f = \sqrt{u_f^2 + v_f^2} \text{ and } R = \sqrt{R_x^2 + R_y^2}$$

The number of multiplications and divisions have been reduced to four at the expense of adding an arcsin routine and taking two square roots rather than one. This method is significantly faster, however, for two reasons:

1. The arcsine curve can be approximated by a rational function which gives good accuracy over the range of interest.

2. Linear approximations to the square roots of sums of squares give excellent results and, since they are non-iterative, are very fast. No way of applying this kind of an approximation to the square root of an arbitrary number has as yet been found.

The main merit of the trigonometrical solution lies in the time saving effected. There are however a number of other points that bear mentioning. First, it may be desirable elsewhere in the vectoring or control system to know the range, R, a quantity specifically computed by this method. Second, it is possible that information concerning the friend's speed may be transmitted by the friend to the system. This information would presumably be more accurate than crude radar data, and would increase the accuracy of the calculations or permit the use of a shorter square root of sums of squares routine for the same overall accuracy. In addition, since S_f is used specifically in the calculations, it would effect a savings of the time necessary to calculate S_f by the square root of sums of squares routine, and, by avoiding the error of the approximation used, again increase accuracy or save time. Third, it may be very desirable to have some measure of what is called in this paper the angle ϕ. The method outlined above requires the actual computation of $\sin\phi$ and ϕ, either of which can be used easily. Fourth, tests for range becoming zero and for $\sin\phi$ exceeding a certain numerical value are essential parts of this method, and these tests may be used directly or modified for other parts of the program. Finally, since two square root of sum of squares routines are performed, it is convenient to treat this routine as a closed subroutine, available elsewhere in the program. Since it is often necessary to compute the magnitude of a two-dimensional vector, the availability of such a routine may be very valuable.

Returning to the question of the ambiguity in the sign of the radical for the algebraic solution:

$$\sqrt{(u_f^2 + v_f^2)(R_x^2 + R_y^2) - (u_b R_y - v_b R_x)^2} = S_f R \sqrt{1 - \sin^2 \phi}$$

$$= S_f R \cos\phi$$

In finding an angle from its sine, there is also the ambiguity of the quadrant of the angle in question; that is, for $|w|<1$, there are two angles, ϕ_1, and ϕ_2, where $0 \leq |\phi_1| < 90^\circ$ and $90^\circ < |\phi_2| \leq 180^\circ$, whose sin is w. The above equation shows that these two ambiguities are equivalent. The common ambiguity can be resolved by considering two cases.

Case 1. $S_b < S_f$. Then $\frac{S_b}{S_f} |\sin\lambda| = |\sin\phi| < 1$. There are two apparent solutions, ϕ_1 and ϕ_2, as above. But referring to the equation,

$$\bar{R} = (\bar{V}_c - \bar{V}_b)\Upsilon$$

and taking the scalar product with $(\bar{V}_c - \bar{V}_b)$, denoting $|\bar{V}_c - \bar{V}_b|$ by V,

$$\bar{R} \cdot \bar{V}_c - \bar{R} \cdot \bar{V}_b = \bar{V}^2 \Upsilon, \text{ or,}$$

$$\bar{V}^2 \Upsilon = RS_f \cos\phi - RS_b \cos\lambda$$

Suppose that $\cos\phi \leq 0$, but $\Upsilon > 0$. Then $|\cos\phi| < \frac{S_b}{S_f} |\cos\lambda|$.

But $\frac{S_b^2}{S_f^2} \sin^2\lambda = \sin^2\phi = 1 - \cos^2\phi > 1 - \frac{S_b^2}{S_f^2} \cos^2\lambda$, or $S_b^2/S_f^2 > 1$, a contradiction. Thus the apparent solution ϕ_2, for which $\cos\phi$ is negative, corresponds to a course for which a collision would have taken place in past time, and the only admissable solution is ϕ_1, where $0 \leq |\phi_1| < 90^\circ$, and $\cos\phi_1 > 0$.

Case 2. $S_b \geq S_f$. There will be a solution if and only if $|\sin\lambda| \leq S_f/S_b$; that is, if the radicand in question is non-negative. If there is no solution, there is no ambiguity. If there is only one solution, i.e., if $\sin\lambda = S_f/S_b$, then $\sin\phi = 1$, $\cos\phi = 0$, the radicand vanishes, $\phi = 90^\circ$, and there is no ambiguity. For $\sin\lambda < S_f/S_b$ there are two solutions, and both either correspond to collisions in the future time, if $|\lambda| \geq 90^\circ$, or to collisions in past time, if $|\lambda| < 90^\circ$. The last case may be rejected.

In order to make the interception in the shorter time, the particular ϕ which makes τ smaller must be chosen. It is evident that the geometry involved in the FBC triangle is not affected by the choice of coordinate axes; accordingly, let the x-axis be chosen along the range vector $\bar{R}$, as in Figure 2. Then $R_x = \bar{R}$, $R_y = 0$, $u_c = S_f\cos\phi$, $u_v = S_f\sin\phi$, $u_b = S_b\cos\lambda$, and $v_b = S_b\sin\lambda$. The square of the closing velocity V is:

$$V^2 = (S_f\cos\phi - S_b\sin\lambda)^2 + (S_f\sin\phi - S_b\sin\lambda)^2.$$

P[illegible] $S_f\sin\phi = S_b\sin\lambda$, and this equation reduces to $V^2 = (S_f\cos\phi - S_b\sin\lambda)^2$. Finally the time to go τ is given by

$$\tau = \frac{R}{S_f\cos\phi - S_b\cos\lambda},$$

and it is clear that τ will be smaller for the ϕ which $|\phi| < 90^\circ$.

In both cases therefore the range of ϕ may be restricted to $-90^\circ \leq \phi \leq 90^\circ$, corresponding to always using the positive sign of the radical.

The physical significance of this reasoning is shown in Figures 3 and 4. If the velocity vector $\bar{V}_b$ of the bogey, the range vector $\bar{R}$, and a speed circle of the friend (i.e., a circle with center at friend's position and radius proportional to the speed of the friend) are drawn, then the possible solutions for $\bar{V}_c$ are found by connecting the friend's position with the points of intersection of the friend's speed circle with a line drawn parallel to the range vector and through the endpoint of the bogey velocity vector. Figure 3 corresponds to the case when it is greater. A number of cases are indicated on the figures.

APPENDIX B

ARCTAN APPROXIMATION

The routine for calculating $\theta = 90^{\circ} - \arctan R_y/R_x$ is due to J. N. Snyder. The method consists of replacing the function $\Psi = \arctan w$ for $-1 \leq w < 1$ by a rational function of the form

$$\psi^* = \frac{w}{8}\left(a - \frac{b}{\frac{c}{8} - \left|\frac{w}{8}\right|}\right)$$

Taking a = 670, b = 211, and c = 5.0722 gives an approximation accurate to less than 7' of arc. This particular choice of a and b cause ψ^* to be computed in units of semisnyds, where 1 semisnyd is 1/256 of a revolution.

In addition to the approximate formula above, it is necessary to determine the quadrant of the angle. If θ is written as

$$\theta = \theta' + \theta_f$$

where θ' is an angle between -45° and 45° computed by the approximation above, and θ_f, called the fiduciary angle, corresponds to one of the four cardinal directions, North, East, South, or West, then the relationships between R_x, R_y, θ_f, and θ are as shown in Table 1B.

Table 1B

R_x	R_y	$\lvert R_x\rvert - \lvert R_y\rvert$	w	θ_f (Direction and Semisnyds)	θ(semisnyds)
≥ 0	≥ 0	<0	R_x/R_y	North 0	$0 \leq \theta \leq 32$
≥ 0	≥ 0	≥ 0	$-R_y/R_x$	East 64	$32 \leq \theta \leq 64$
≥ 0	<0	≥ 0	$-R_y/R_x$	East 64	$64 < \theta \leq 96$
≥ 0	<0	<0	R_x/R_y	South 128	$96 < \theta \leq 128$
<0	<0	<0	R_x/R_y	South 128	$128 < \theta < 160$
<0	<0	≥ 0	$-R_y/R_x$	West 192	$160 \leq \theta < 192$
<0	≥ 0	≥ 0	$-R_y/R_x$	West 192	$192 \leq \theta \leq 224$
<0	≥ 0	<0	R_x/R_y	North 0	$-32 < \theta < 0$

This table can be written concisely in terms of two tests:

1. If $|R_x| \geq |R_y|$, set $w = \frac{R_y}{R_x}$. If $R_x > 0$ set $\theta_f = 64$ semisnyds, and if $R_x < 0$, set $\theta_f = 192$ semisnyds.

2. If $|R_x| < |R_y|$, set $w = \frac{R_x}{R_y}$. If $R_y < 0$, set $\theta_f = 128$ semisnyds, and if $R_y \geq 0$, set $\theta_f = 0$.

If $|R_x| \geq |R_y|$, a third test is necessary to insure that R_x and therefore R_y are not zero. Since such a situation corresponds to a completed collision, this test may be useful elsewhere in the vectoring system.

The error curve for the approximation formula is plotted against θ for $0 \leq \theta \leq 45^\circ$ in Figure 5.

APPENDIX C

ARCSIN APPROXIMATION

The method of calculating arcsin w for $-.99756 < w < .99756$ consists of replacing $\emptyset$ = arcsin w by a rational function of the form

$$\emptyset = w\left(a - \frac{b|w|}{c - |w|}\right)$$

The constants a, b, and c, scaled so that $\emptyset$ is calculated in semisnyds, are a = 40.8530688, b = 2.5, and c = 1.1275.

The error curve in degrees vs. $\emptyset$ is given in Figure 6. The error is zero at zero degrees, is numerically less than .17 degrees up to about 40 degrees, less than .3 degrees up to about 67 degrees, and less than .8 degrees up to about 84 degrees.

The particular coefficients a, b, and c were chosen after a rather extensive search. It was found that neither a nor b was critical, and may be varied simultaneously without affecting the general nature of the error curve. Since the arcsin curve is very nearly linear for small w, the value of a scaled to semisnyds is nominally $128/\pi = 40.7436654$. However, it is convenient to compute $b\,|w|$ as a "programmed multiplication" (see Appendix F) and the value of a was increased slightly to compensate for a decrease in b necessary to achieve a simple operation.

Coding of this approximation as a closed subroutine has been placed in the ILLIAC library.

APPENDIX D

SQUARE ROOT OF SUMS OF SQUARES

Evaluating the function $\sqrt{x^2 + y^2}$ by an approximate method is equivalent to representing the right circular cone $z = \sqrt{x^2 + y^2}$ by some surface $z^* = f(x,y)$. It is usually desirable that the relative error, defined as $\epsilon = (z^* - z)/z$, be constant along a generating element y/x = constant of the cone; this condition is met if $f(x,y)$ is also a conic, or equivalently, if $f(x,y)$ is homogeneous of order one, $f(kx, ky) = k f(x,y)$. Only approximations of this kind will be considered, and

$$\epsilon = \frac{f(x,y)}{\sqrt{x^2 + y^2}} - 1 = f\left(\frac{x}{\sqrt{x^2 + y^2}}, \frac{y}{\sqrt{x^2 + y^2}}\right) - 1.$$

Investigation of various approximations is clarified by introducing the variable $\alpha = \arctan y/x$ in the above equation. This equation then becomes $\epsilon = f(\cos\alpha, \sin\alpha) - 1$. Nominally an approximation valid for a full revolution in α is required; however z is obviously independent of the signs of x and y, and the range of α over which the approximation must hold may be reduced to $0 \leq \alpha \leq 90^\circ$ by the simple requirement that $f(x,y)$ be of the form $f(|x|, |y|)$. This range can be further halved by requiring that the approximation be symmetric about $\alpha = 45$ degrees. This may be easily effected by denoting L and S to be the larger and the smaller respectively of $|x|$ and $|y|$, and requiring f to be of the form $f(L,S)$.

The simplest function of x and y which is homogeneous of order one is the linear function, $z^* = aL + bS$. Then $z^*/z = a \cos \alpha + b \sin\alpha = A \cos(\alpha - \beta)$, where $A = \sqrt{a^2 + b^2}$, $\beta = \arctan b/a$, and $\epsilon = A \cos(\alpha - \beta) - 1$. It is necessary to investigate approximations only for the interval $0 \leq \alpha \leq 45^\circ$.

Consider now a linear approximation which is to hold over the interval in α, (α_1, α_2), where $0 \leq \alpha_1 \leq \alpha_2 \leq 45^\circ$. The "best" approximation in this interval is defined as the one for which the maximum numerical error is minimum. The "best" approximation is thus obtained by choosing the angle to be the mid-angle of the interval in question, $\beta = (\alpha_1 + \alpha_2)/2$, and then choosing A so that A exceeds 1 by the same amount that A $\cos(\alpha_1 - \beta)$ and A $\cos(\alpha_2 - \beta)$ are less than 1. This in turn determines the constants a and b for this interval.

The accuracy of the approximation can be successively improved by breaking up the interval $(0,45^\circ)$ into more and more sub-intervals and using a different linear approximation in each sub-interval. The end-angles of these sub-intervals may be denoted by

$$0 = \alpha_1 < \alpha_2 < \ldots < \alpha_n < \alpha_{n+1} = 45^\circ.$$

In order to determine which set of coefficients a_k, b_k to use, i.e., to determine which sub-interval (α_k, α_{k+1}) is pertinent, S must be compared successively with L $\tan\alpha_k$.

A number of multiplications are thus indicated, a set of the form L $\tan\alpha_k$, and two of the form $a_k L$ and $b_k S$. These multiplications can however be performed as "programmed multiplications" as described in Appendix F with a considerable savings in time. In order to gain this time savings, it is necessary that the constants a_k, b_k, and $\tan\alpha_k$ have simple, convenient binary representations. This requirement forces the uses of, not the "best" coefficients for a sub-interval, and the "best" set of sub-intervals, that is, the set of equal ones, but ones which are easier to handle.

Finally, it is most efficient to use a series of binary tests to determine which of the sub-intervals (α_k, α_{k+1}) is pertinent; thus the number of sub-intervals is restricted to a power of 2, n = 1, 2, 4, 8, etc.

These considerations lead to the set of approximations whose main properties are listed in Table 1D. It can be seen that a great increase in accuracy can be obtained by using a large number of sub-intervals, and that the increase in accuracy is paid for in terms of number of words in the program, rather than in time. Moreover, the choice of endpoints is greatly affected by the number of sub-intervals to be used. For n = 2, it is better to compromise between the best pair of sub-intervals, the equal ones, and the "easiest" midangle, whose tangent is 1/2; thus the midangle α_2 for which $\tan\alpha_2 = 3/8$ is used. For n = 4 or 8, however, the number of programmed multiplications that must be written into the routine forces the use of the "easiest" midangles, for which the tangents are successively 1/4, 1/2, 3/4, for n = 4, and 1/8, 1/4, 3/8, 1/2, 5/8, 3/4, and 7/8 for n = 8.

The error curves for all but one of the approximations listed are given in Figures 7 through 13 as indicated in Table 1D.

A very simple non-linear approximation due to J. N. Snyder is accurate to 1%, although in this case a regular multiplication and division are required. The method consists of adding a correction factor to the crude linear approximation $|x| + |y|$. This correction factor is essentially a secant curve in α, but in order to preserve homogeniety it is necessary to multiply the correction factor by a quadratic factor in x and y. The resulting formula is

$$z^* = |x| + |y| - \frac{|x| \; |y|}{7/8(|x| + |y|)}.$$

The routine for this approximation requires only seven words, but takes 2.436 milliseconds on ILLIAC. The error curve is given by Figure 14, and the ILLIAC code is included in Appendix G, as are all the important linear approximations previously listed.

An improvement in the accuracy of this formula can be made by using L and S as defined before and adapting the formula so that it is symmetric about arctan $y/x = 22.5$ degrees. The formula is

$$z^* = .48\left(L + (\sqrt{2} - 1)S\right) + \frac{.52(L^2 + S^2)}{L + (\sqrt{2} - 1)S}$$

This approximation has a maximum error of .04%. Using "programmed multiplications" does not appreciably shorten the time for this routine, and 18 words and 5.82 milliseconds are required. The coding is also given in Appendix G, and the error curve in Figure 15.

Table 1D   Linear Approximations to Square Root of Sums of Squares

Name	Max Relative error (%)	Time required (mc.)	ILLIAC Words Required	Formula: k	ak	bk	Interval	Figure for Error Curve	Remarks
L+S/4 L+S/2	12.5	.68	7	1	1	$\begin{cases}1/4\\1/2\end{cases}$	$0 \leq S \leq L$	7	Well known Simple Approximations
$\frac{15}{16}$L+S/2	6.25	.86	9	1	$\frac{15}{16}$	1/2	$0 \leq S \leq L$	8	
"Best 1-Line"	4.0	---	-	-	$\frac{123}{128}$	$\frac{51}{128}$	$0 \leq S \leq L$	9	"Best" for n = 1, but too complicated. Better to use "Nordsieck".
"Nordsieck"	3.0	1.32	13	1	1	1/4	$0 \leq S \leq L/2$	10	Due to Prof. A.T. Nordsieck
				2	7/8	1/2	$L/2 < S \leq L$		
"1.4% Rule"	1.4	1.92	16	1	$\frac{127}{128}$	3/16	$0 \leq S \leq 3/8L$	11	
				2	$\frac{221}{256}$	17/32	$3/8L < S \leq L$		

Table 1D - continued

"Best 2-line"	1.0	---	---	1	507/512	101/512	$0 \leq S \leq 53/128L$	--	"Best for n=2 But too complicated. "Snyder" is better.
				2	215/256	287/512	$\frac{153}{128}L < S \leq L$		
"4-line"	.4	2.33	29	1	255/256	1/8	$0 \leq S \leq L/4$		
				2	965/1024	89/256	$L/4 < S \leq L/2$	12	
				3	109/128	135/256	$L/2 < S \leq 3/4L$		
				4	97/128	21/32	$3/4L < S \leq L$		
"8-line"	.1	2.54	43	1	1023/1024	1/16	$0 \leq S \leq L/8$		
				2	2015/2048	47/256	$L/8 < S \leq L/4$		
				3	489/512	153/512	$L/4 < S \leq 3/8L$		
				4	235/256	51/128	$3/8L < S \leq L/2$	13	
				5	7/8	497/1024	$L/2 < S \leq 5/8L$		
				6	211/256	145/256	$5/8L < S \leq 3/4L$		
				7	25/32	5/8	$3/4L < S \leq 7/8L$		
				8	93/128	11/16	$7/8L < S \leq L$		

APPENDIX E

ERRORS IN THE COLLISION COURSE

Ignoring negligible roundoff errors of multiplication and division, errors in Θ_c arise from two sources; inaccuracy of radar input information, and the approximations for the arctan, arcsin, and square root of sums of squares used in calculating Θ_c. The latter error will be of the form

$$\delta\Theta_c = \mathcal{E}_1(\Theta) + \mathcal{E}_2(\phi) + \delta(\phi;S_f,R)$$

where $\mathcal{E}_1(\Theta)$ is the error in the arctan approximation, $\mathcal{E}_2(\phi)$ the error in the arcsin approximation, and $\delta(\phi;S_f,R)$ represents the propagation of the errors in the approximation of S_f and R by whatever square root of sums of squares method is used. Let $\mathcal{E}_3(z)$ be the error in $z = \sqrt{x^2 + y^2}$ and E_3 be the maximum value of $|\epsilon_3(z)|$ for a particular method. Then if the errors are small, the last term is approximately equal to $(\epsilon_3(S_f)+\epsilon_3(R))\tan\phi$. Noting that $|\epsilon_1(\Theta)| \leq .12$ degrees, a convenient upper bound for the error in Θ_c arising from approximations in the calculation of Θ_c is

$$E(\phi) = .12 + |\epsilon_2(\phi)| + 2E_3\tan\phi$$

For small ϕ this is a realistic upper bound, since situations in which the errors in the arctan and in both square root approximations are near their maximums can be found. For ϕ near $\pm$ 90° however, $E(\phi)$ overestimates the possible error badly, since ϕ is restricted to $|\phi| \leq 90°$.

Plots of $E(\phi)$ against ϕ for various values of E_3 corresponding to different square roots of sums of squares approximations investigated are shown in Figure 16. The figures of merit in Table 1 in the body of this report were obtained by determining the maximum of $E(\phi)$ for ϕ less than 30 degrees and ϕ less than 60 degrees for each relevant value of E_3.

If E_3 is less than .004, the controlling term in E is evidently $\varepsilon_2(\phi)$. In fact, using the approximation for which E_3 = .004 gains very little in accuracy over the approximation for which E_3 = .001 in view of the much greater time of calculation required. If E_3 is greater than .004, the term $2E_3 \tan\phi$ dominates. It is felt that using one of the three approximations for which E_3 is .001, .004, or .01, according to the accuracy required in computing θ_c, represents a reasonable balance between the effects of the approximations for the arcsin and the square root of sums of squares.

In many vectoring systems, θ_c is desired only to an accuracy consistent with the quantization of the data link with the friend. Using the rule for which E_3 is .001 insures eight bit accuracy for $|\phi| \leq 67.5$ degrees and seven bit accuracy for $|\phi| \leq 84$ degrees.

In the process of testing the vectoring program described in Appendix G, $\delta\theta_c$ was determined for a large number of cases for each of the square root of sums of squares routines. Results for the cases E_3 = 0, .001, and .01 are summarized in Figures 17, 18, and 19 respectively, which show the distribution of errors for various values of ϕ for each case. Since random cases were not considered, these results may not be interpreted as a description of probable error; however, they indicate that errors for ϕ near to $\pm 90^\circ$ will be much less than indicated by the formula for $E(\phi)$. Calculation of θ_c was made to only 1/4 semisnyd, and was left unrounded, with a resulting bias in the mean error of about $.17^{\circ}$.

The other source of error in θ_c, failure of the radar data to describe the exact physical situation, is obviously independent of whatever calculating scheme is used. The problem of determining θ_c is equivalent to solving a triangle for an angle, given an angle, the included side, and the fact that the other two sides are in constant ratio, and then orienting this triangle in space by determining the angle of inclination of the included side. Using the prefix Δ to denote errors arising from radar error in a quantity, a formal expression for $\Delta\theta_c$ is

$$\Delta\theta_c = \Delta\theta - \Delta\phi \cong \Delta\theta - \left(\frac{\Delta S_b}{S_b} - \frac{\Delta S_f}{S_f}\right)\tan\phi - \frac{S_b}{S_f}\frac{\cos\lambda}{\cos\phi}\Delta\lambda$$

The quantity $\Delta\theta$ represents the irresolution of the angle of the line between friend and bogey due to irresolution of the positions of friend and bogey, and $\Delta\theta$ obviously increases as R decreases.

Now consider the quantity $\Delta\phi$, and let $k = S_b/S_f$ be the speed ratio and $\Delta k/k$ the relative error in the speed ratio of bogey to friend. Neglecting the dependence in the error in λ on R, it appears that $\Delta\phi$ may become critically large for ϕ near $\pm 90^{\circ}$. Two factors are involved. First, errors in the estimation of the speed ratio becomes critical when ϕ is large, and error in the estimation of the bogey's course is critical when ϕ is near $\pm 90^{\circ}$ but λ is not.

A plot of level curves of $\phi = \arcsin(k \sin\lambda)$ in the λ, k plane is given in Figure 20. Closely spaced curves, either horizontally or vertically, indicate regions where small errors in k or λ respectively cause relatively large errors in ϕ. In Figure 21 and 22 plots of ϕ versus k for $\lambda = 15^{\circ}$, 30°, 60°, and 75° and of ϕ versus λ for k = .5, .8, 1.0, 1.2, and 1.5 are shown. In each case errors in ϕ due to small errors in k or λ respectively will be larger when the slopes of these curves are large. In particular, these figures bear out the conjecture that errors in θ_c due to inaccurate radar data may become large for ϕ near $\pm 90^{\circ}$. These considerations lend weight to the argument that it is desirable to calculate θ_c accurately for small ϕ at the expense of accuracy for ϕ near $\pm 90^{\circ}$ if necessary.

APPENDIX F

PROGRAMMED MULTIPLICATIONS

Since ILLIAC takes a much longer time to multiply than to add and shift, a considerable savings in time can be obtained at the expense of additional instructions by replacing multiplication of a constant and a variable by a series of shifts and additions of the variable. Such an operation is called in this paper a "programmed multiplication". The method consists of writing the constant in its binary form to the number of places required for accuracy and forming the additions as indicated. Since it is as easy to subtract as to add, a sequence of 1's in the binary expansion of the constant can be replaced by one addition and one subtraction, with suitable shifts. The method is useful only if two conditions hold:

1. The binary expansion of the constant may be terminated after a few bits without loss of accuracy.

2. The binary expansion of the constant contains many zeroes or sequences of 1's.

To illustrate the method, considering the problem of multiplying x by a = .562513. The binary expansion of a, accurate to twenty bits, is .10010000000000001101. Rounding this to .1001, ax can be written as ax = (x/8 + x)1/2. The ILLIAC code for performing this operation requires 2 order pairs and .288 milliseconds, as opposed to one word and about .750 milliseconds if ax were formed by a regular multiplication.

As another illustration, consider the problem of forming a x + b y where a = .9457258 and b = .5543742. Expanding a and b in binary form and rounding to ten bits, a = .1111001000 and b = .1000111000. Replacing the sequences of 1's by subtractions, a may be written as 1 - 1/16 + 1/128, and b as 1/2 + 1/16 - 1/128. Then ax + by =

$$\left(\left((x-y)1/8 + (y-x)\right)1/8 + y\right) 1/2 + x$$

APPENDIX G

ILLIAC CODE FOR VECTORING ROUTINE

The programming and coding of a routine to evaluate Θ_c by the method described in this paper depends upon the characteristics of the particular computer to be used. However, certain features of the following codes, which were formulated for use on ILLIAC for a particular problem, may suggest ideas for other computers and other problems.

It is assumed in the following that eight numbers representing the rectangular coordinates of the position and velocity of bogey and friend are available in specified registers, that Θ_c is to be calculated and retained in the accumulator, and that R is to be calculated and stored in a specified register.

The notation is the same as previously used, with the addition of (x_b, y_b) for the bogey position and (x_f, y_f) for the friend position, with $R_x = x_b - x_f$ and $R_y = y_b - y_f$.

Velocity coordinates u_b, v_b, u_f, and v_f are six bit signed numbers stored in the first seven bits (sign and six bits) of four 40 bit registers labelled symbolically (3), (4), (5), and (6). The bits in each 40 bit register are numbered 0 through 39. Position coordinates are positive 10 bit numbers stored in positions 11-20 of the same four registers, with u_b and x_b in register (3), v_b and y_b in (4), and u_f and x_f in (5), and v_f and y_f in (6). Positions 8-10 of each register hold zeroes, while the last 20 bits are disregarded. Scaling is such that the last (sixth) bit of velocity coordinates represents 1/64 mile/second and the last (tenth) bit of position coordinates represents 1/2 mile.

Six or seven words of temporary storage are required, depending on which square root of sums of squares routine is being used. Three of these words are ILLIAC standard temporary storage, with addresses 0, 1, and 2, and the other four are labelled symbolically as (7), (8), (9), and (10). The input data in storage registers (3), (5), and (6) is destroyed. R is placed in storage (9), scaled so that the 13th bit represents 1/2 mile; i.e., R/2048 is stored. The collision course, Θ_c appears in the accumulator in bits 10-19, where the last bit (in the 19th Position) r resents 1/1024 of a revolution or, equivalently, 1/4 semisnyd. The first ten bits in the accumulator will not in general be zero, and θ_c must be extracted by a 46 order or the equivalent.

The vectoring routine is written as a closed subroutine. Entry may be made from the left hand address of instruction pair n+1 by the sequence

	any order	
n	50	n
n + 1	22	(vl)

and control is returned to the right hand instruction of n+1. The code and notes follow.

Sequence	Order	pair	Remarks
	00	1	Waste
n	K5	0	Plant Links
	42	n+42	
n+1	50	n+1	Transfer to square root of sums of square routine to compute $S_f = \sqrt{u_f^2 + v_f^2}$ and S_f in (7)
	26	(SQ)	
n+2	40	(7)	
	L5	(3)	Unpacking. Form $R_x = x_b - x_f$, shift 7 places to eliminate velocity information and store in loc (5). Do the same for R_y and store in loc (6).
n+3	L0	(5)	
	00	7	
n+4	40	(5)	
	L5	(4)	
n+5	L0	(6)	
	00	7	
n+6	40	(6)	
	L7	(6)	Arctan approximation. Test $(R_y)-(R_x)$ and transfer to n+13 if $(R_y) \geq (R_x)$
n+7	L2	(5)	

	32	n+13	
n+8			
	L5	(5)	If $(R_y)<(R_x)$ test sign of R_x and transfer to
	32	n+12	n+12 if $R_x \geq 0$.
n+9			
	L5	n+22	If $R_x < 0$ set $\Theta_F = 192 \times 2^{-17}$ semisnyds store
	46	0	Θ_F in loc 0.
n+10			
	L1	(6)	
	10	3	
n+11			
	66	(5)	Set $w = \frac{R_y}{8R_x}$
	22	n+18	
n+12			
	L5	n+19	Set $\Theta_F = 64 \times 2^{-17}$ semisnyds if $R_x \geq 0$
	26	n+10	If $(R_y) \geq (R_x)$ test for $(R_y) = 0$
n+13			
	L3	(6)	If $(R_y) = 0$ transfer to end of vectoring routine
	32	n+42	If $R_y = 10$, test sign of R_y
n+14			
	L1	(6)	
	36	n+16	
n+15			
	23	n+16	If $R_y > 0$ set Θ_F to 0
	L5	n+20	If $R_y < 0$ set Θ_F to 128×2^{-17} semisnyds
n+16			
	46	0	Store Θ_F in loc 0
	L5	(5)	
n+17			
	10	3	
	66	(6)	set $w = R_x/8R_y$
n+18			
	L1	n+46	
	S6	256	
n+19			
	40	1	
	S5	512	
n+20			

	40	2	
n+21	L5	n+47	
	66	1	
n+22	S5	768	
	L4	n+48	
n+23	40	1	
	50	1	
n+24	7J	2	Form $\Theta' = w(a + \frac{b}{-c/8+\lvert w \rvert})\,2^{-17}$
	L4	F	$\Theta = \Theta_F + \Theta'$
n+25	40	(8)	Store Θ in loc 8
	50	n+25	Transfer to square root of sum of square routine to compute $R = \sqrt{R_x^2 + R_y^2}$ and store R in (9)
n+26	26	(SQ)	
	40	(9)	
n+27	50	(9)	Compute and store $S_f R$ in (7)
	7J	(7)	
n+28	40	(7)	
	50	(5)	Compute $u_b R_y - v_b R_x$
n+29	7J	(4)	
	40	0	
n+30	50	(6)	
	7J	(3)	
n+31	L0	0	
	10	3	Compute $-\frac{\sin\phi}{8} = \frac{(u_b R_y - v_b R_x)1/8}{S_f R}$
n+32	66	(7)	
	S7	0	Arcsin approx.

n+33	L0	n+49	Test $\left\|\frac{\sin\phi}{8}\right\| - \frac{\sin 86^{\circ}}{8}$ + transfer to n+43 if positive
	36	n+43	
n+34	L0	n+50	
	40	F	
n+35	S5	1024	
	10	6	
n+36	40	(3)	
	L3	(3)	
n+37	10	2	
	L2	(3)	
n+38	10	1	
	66	0	
n+39	S5	249	
	L4	n+51	Compute $-\phi\, 2^{-17}$ (semisnyds) $= -2^{-9}\sin\phi$
n+40	40	0	$\left(c_1 + \frac{(1/2+1/8)2^{-9}(\sin\phi)}{b_1/8 - \frac{(\sin\phi)}{8}}\right)$
	50	0	
n+41	7J	(3)	
	L4	(8)	
n+42	L4	n+35	$\theta_c = \theta - \phi$ (mod 256)in semisnyds x 2^{-17} end of routine
	22	0	
n+43	S5	0	
	36	n+45	test sign of $\sin\phi$
n+44	L1	n+39	
	22	n+42	If $\sin\phi<0$ set $\phi = 62.25$ semisnyds
n+45	L5	n+39	

```
        22   n+42    If sinØ≥0 set Ø = -62.25 semisnydes
n+46
       .6134025      c/8
n+47
       211 x 2^-17   b x 2^-17
n+48
       470 x 2^-17   a x 2^-17
n+49
       .124524337262 sin 86°/8
n+50
       .16413162739  b_1/8 - sin 86°/8
n+51
       .1595823      c_1
```

Codes for the various square root of sums of squares routines follow. In each case $z = \sqrt{x^2 + y^2}$, where x and y are contained in locations (5) and (6), is computed and remains in the accumulator. Entry into any of these subroutines is effected as shown in words n+1, n+2 or n+25, n+26 in the program above. The routines are numbered SQ1 to SQ8 and headed by a descriptive nickname and the accuracy in per cent of the routine along with the time to perform the routine.

```
SQ1:  L + S/4   (12%-.7ms)
      K5
n
      42  n+6   plant link

      L7  (5)
n+1
      L2  (6)   |x|-|y|

      36  n+5   Transfer if |x|≥|y| and set L =|x|, S =|y|
n+2             |x|<|y|.  Set L = |y| and S = |x|
      L7  (5)

      10  2
n+3
      L6  (6)   |y|+ |x|/4

      22  n+6   Transfer to end.
n+4
      waste
```

n+5	L7	(6) ↲	
	10	2	
n+6	L6	(5)	$\|x\|+\|y\|/4$
	22	(x)	return to main routine

The program code for SQ1A:L + S/2 is identical except for replacing the left shift orders 10 2 in words n+3 and n+5 by left shift 10 1, i.e., shifting S one binary place rather than 2.

SQ2: 15/16L + S/2 (6.25% - .84ms.)

	K5		
n	42	n+8	Plant link
	L7	(5)	
n+1	L2	(6)	$\|x\|-\|y\|$
	36	n+6	If $\|x\|\geq\|y\|$ transfer to n+6 and set L $=\|x\|$, S$=\|y\|$
n+2	L3	(6)	If $\|x\|<\|y\|$ set L = $\|y\|$ and S = $\|x\|$
	10	3	
n+3	L6	(5)	
	10	1	
n+4	L6	(6)	$\frac{15}{16}\|y\|+\|x\|/2$
	22	n+8	Transfer to end
n+5	waste		
	L3	(5) ↲	
n+6	10	3	
	L6	(6)	
n+7	10	1	
	L6	(5)	$\frac{15}{16}\|x\|+\|y\|/2$

```
n+8
      22   (x)     Return to main routine
      SQ3: Nordsieck linear approximation  (3%- 1.32ms.)
      K5
n
      42   n+10    Plant link
      L7   (5)
n+1
      L2   (6)     |x|-|y|
      36   n+4 ─┐  If |x|≥|y| transfer to n+4 and set L = |x|, S = |y|
n+2             │
      50   (6)  │  If |x|<|y| set L = |y| and S = |x|
      L7   (5)  │
n+3             │
      26   n+5 ┐│
      50   (5)←┘│
n+4             │
      L7   (6)  │
      40   1  ←─┘  Store S in loc 1
n+5
      S7
      40   0       Store L in loc 0
n+6
      10   1
      L0   1       L/2 - S
n+7
      36   n+11    Transfer to n+11 if L/2≥S
      L1   0
n+8
      10   2
      L4   1
n+9
      10   1       If L/2<S form -1/8L + S/2
      L4   0       L/2<S form 7/8L + S/2; L/2≥S form L + S/4
n+10
      22   (x)     Return to main routine
```

```
        L5    1
n+11
        10    2       Form S/4
        26    n+10    Transfer to form L + S/4 and then return to  main
n+12                  routine
        waste
```

SQ4: 2-line Linear Approximation (1.4%- 1.92 ms)

```
        K5
n
        42    n+13
        L7    (5)     As in SQ3; puts L in loc 0 and S in loc 1.
n+1
        L2    (6)
        36    n+4
n+2
        50    (6)
        L7    (5)
n+3
        26    n+5
        50    (5)
n+4
        L7    (6)
        40    1
n+5
        S7
        40    0
n+6
        10    2
        L0    0
n+7
        10    1
        L4    1       Form -3/8L + S
n+8
        36    n+14    Transfer to n+14 if 3/8L≤S
        10    3
n+9
```

	40	2	Store $-3/64L + S/8$ in loc 2
	10	1	Form $-3/128L + S/16$
n+10	L4	2	
	40	2	
n+11	L7	0	
	10	4	
n+12	L6	0	Form $17/16L$
	L4	2	
n+13	22	(x)	If $3/8L>S$, form $\frac{127}{128}L + \frac{3}{16}S$ and return to main routine.
	10	1	If $3/8L \leq S$, form $\frac{221}{256}L + \frac{17}{32}S$
n+14	40	2	$\frac{3}{16}L + S/2$ in loc 2
	10	4	
n+15	22	n+10	Form $\frac{3}{256}L$ and $S/32$ and transfer to n+10.

SQ5: Snyder (1%-2.44ms)

	K5		
n	42	n+6	Plant link.
	L7	(5)	
n+1	L6	(6)	
	40	0	Store $\lvert x\rvert + \lvert y\rvert$ in loc 0
n+2	10	3	
	F0	0	
n+3	40	1	Store $-7/8\left[\lvert x\rvert + \lvert y\rvert\right] -2^{-39}$ in loc 1
	50	(5)	
n+4	7J	(6)	Form xy
	66	1	
n+5			

S3 Form $-\left[\frac{xy}{-7/8\left[|x| + |y|\right] - 2^{-39}}\right]$

n+6 L4 0

22 (x) Form $|x| + |y| - \left[\frac{xy}{-7/8\left[|x| + |y|\right] - 2^{-39}}\right]$ and return to main routine.

Note: An F0 order rather than an L0 order is used in pair n+3 in order to prevent a possible division of 0 by 0, which would otherwise occur whenever x = y = 0. Also, the problem must be scaled so that $|x| + |y| < 1$.

SQ6: Revised Snyder (.03% -5.83 ms)

K5

n 42 n+14

L7 (5)

n+1 L2 (6)

32 n+4 Transfer to n+4 if $|x| \geq |y|$

n+2 50 (5) If $|x| < |y|$, form $ax + y + 2^{-39}$ and store in loc 0

7J n+15

n+3 F6 (6)

26 n+6

n+4 50 (6) If $|x| \geq |y|$, form $ay + x + 2^{-39}$ and store in loc 0

7J n+15

n+5 F6 (5)

40 0

n+6 50 0

7J n+16 Store $b(aS + L + 2^{-39})$ in loc 1

n+7 40 1

50 (5)

n+8

Location	Order	Address	Remarks
	7J	(5)	
	40	2	Store x^2 in loc 2
n+9	50	(6)	
	7J	(6)	
n+10	L4	2	
	40	2	Store $x^2 + y^2$ in loc 2
n+11	50	2	
	7J	n+17	$c(x^2 + y^2)$
n+12	66	0	
	S5		
n+13	L4	1	$b(aS + L + 2^{-39}) + \frac{c(x^2 + y^2)}{aS + L + 2^{-39}}$
	50	0	
n+14	22	(x)	Return to main routine
n+15	.414213562373 a		
n+16	.480405024679 b		
n+17	.519986631700 c		

Note: The problem must be scaled so that $|x| + |y| < 1$

SQ7: 4-line Linear Approximation (.4%-2.33 ms)

Location	Order	Address	Remarks
	K5		
n	42	n+14	
	L7	(5)	
n+1	L2	(6)	
	40	0	Store $|x| - |y|$ in loc 0
n+2	32	n+4	
	50	(6)	
n+3	L7	(5)	

n+4	22	n+5	
	50	(5)	
n+5	L7	(6)	
	40	2	Store S in loc 2
n+6	S7		
	40	(9)	Store L in loc (9)
n+7	10	1	
	L2	2	
n+8	40	1	Store L/2-S in loc 1
	36	n+20	Transfer to n+20 if L/2≥S
n+9	L6	0	If L/2<S, form 3/2L-2S
	36	n+15	Transfer to n+15 if 3/4L≥S
n+10	L7	(9)	
	10	2	If 1/2L≤S<3/4L, form 97/128L + 21/32S and return to main routine
n+11	L6	2	
	10	2	
n+12	L6	2	
	10	2	
n+13	L0	1	
	10	1	
n+14	L6	(9)	
	22	(x)	out
n+15	L3	2	If 3/4L≤S≤L, form
	10	1	
	L2	(9)	

n+16	10	2	$\frac{109}{128}L + \frac{135}{256}S$ and then return to main routine
	L0	1	
n+17	10	3	
	L0	1	
n+18	L6	2	
	10	2	
n+19	26	n+14	
	L2	2	If $L/2 \geq 2S$ form $L/2 - 2S$
n+20	36	n+26	Transfer to n+26 if $L/4 \geq S$
	L3	(9)	If $L/4 < S \leq L/2$ form $\frac{963}{1024}L + \frac{89}{256}S$ and return to main routine.
n+21	10	2	
	L6	(9)	
n+22	L6	2	
	10	3	
n+23	L6	2	
	10	1	
n+24	L2	0	
	10	2	
n+25	22	n+18	
	L3	(9)	If $0 \leq S \leq L/4$ form $\frac{255}{256}L + \frac{S}{8}$ and return to main routine.
n+26	10	5	
	L6	2	
n+27	10	3	
	26	n+14	
n+28	Waste		

Note: Note This routine uses an extra word of temporary storage labelled symbolically as (9). This temporary storage is also used in the main vectoring routine. In the following routine, two additional words of temporary storage are used, labelled symbolically as (9) and (10).

SQ8: 8-line Linear Approximation (.1% -2.54 ms)

```
        K5
n       42   n+40    As in SQ7
        L7   (5)
n+1     L2   (6)
        40   0       Store |x| - |y| in loc 0
n+2     32   n+4
        50   (6)
n+3     L7   (5)
        22   n+5
n+4     50   (5)
        L7   (6)
n+5     40   (10)    Store S in loc (10)
        S7
n+6     40   (9)     Store L in loc (9)
        10   1
n+7     L2   (10)
        40   1       Store L/2 - S in loc 1
n+8     32   n+23    If L/2≥S transfer to n+23
        L6   0       If L/2<S form 3/4L - S and store in loc 2.
n+9     10   1
        40   2
```

n+10	36	n+17	If $3/4L \geq S$ transfer to n+17
	L6	0	If $3/4L < S$ form $7/4L - 2S$
n+11	36	n+16	If $7/8L \geq S > 3/4S$ transfer to n+16
	L1	2	
n+12	L6	(10)	If $7/8L < S \leq L$ form $\frac{93}{128} L + \frac{11}{16} S$ and return to main routine.
	10	2	
n+13	L6	(10)	
	10	2	
n+14	L0	1	
	10	1	
n+15	26	n+40	
	L7	(9)	If $3/4L < S \leq 7/8L$ form $\frac{25}{32} L + 5/8S$ and return to main routine
n+16	26	n+13	
	L4	1	If $L/2 < S \leq 3/4L$ form $5/4L - 2S$
n+17	36	n+21	If $5/8L \geq S > L/2$ transfer to n+21
	L3	0	If $5/8L < S \leq 3/4L$ form $\frac{211}{256} L + \frac{145}{256} S$ and return to main routine.
n+18	10	4	
	L0	2	
n+19	10	1	
	L6	(9)	
n+20	26	n+14	
	L7	(10)	If $L/2 < S \leq 5/8L$ form $7/8L + \frac{497}{1024} S$ and return to main routine
n+21	10	4	
	L2	(10)	
n+22			

	10	3	
n+23	26	n+20	
	L2	(10)	If L/2≤S form L/4 - S and store in loc 2
n+24	10	1	
	40	2	
n+25	36	n+35	If L/4≥S transfer to n+35
	L4	1	If L/4<S form 3/4L - 2S
n+26	36	n+31	If L/4<S≤3/8L form $\frac{235}{256}$ L + $\frac{51}{128}$ S and return to main routine
	L1	1	
n+27	10	1	
	L2	0	
n+28	10	3	
	L6	(10)	
n+29	10	1	
	L0	2	
n+30	10	2	
	26	n+40	
n+31	L7	(9)	If 3/8L<S≤L/2 form $\frac{489}{512}$ L + $\frac{153}{512}$ S and return to main routine.
	L6	(10)	
n+32	10	3	
	L6	0	
n+33	10	2	
	L2	(10)	
n+34	10	1	
	22	n+28	

n+35	I2	(10)	If $L/4 \geq S$ form $L/4 - 2S$
	36	n+41	If $L/8 \geq S$ transfer to n+41
n+36	L3	(9)	If $L/8 < S \leq L/4$ form $\frac{2015}{2048} L + \frac{47}{256} S$
	10	3	
n+37	I2	(10)	
	10	4	
n+38	L0	2	
	10	1	
n+39	L6	(10)	
	10	3	
n+40	L6	(9)	
	22	(x)	out
n+41	L3	(9)	If $0 \leq S \leq L/8$ form $\frac{1023}{1024} L + S/16$
	10	6	
n+42	I2	(10)	
	22	n+38	

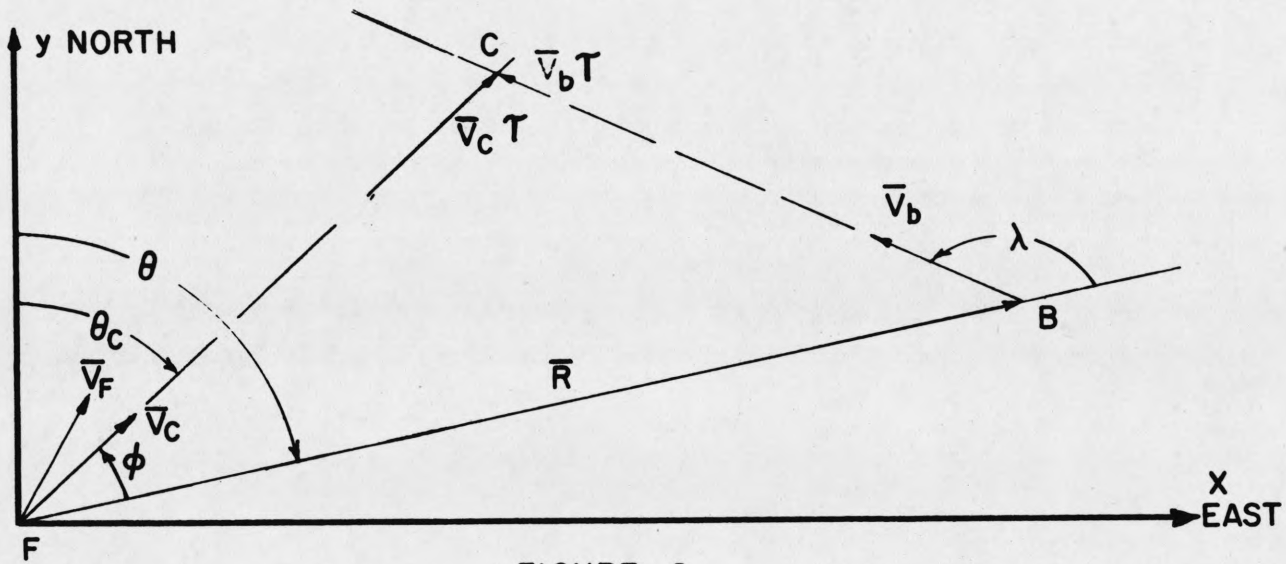


FIGURE 1

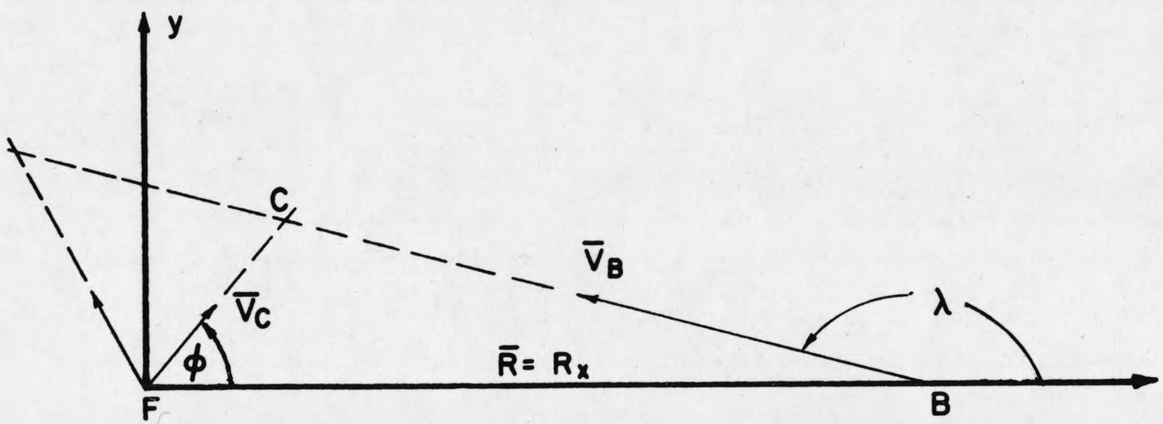


FIGURE 2

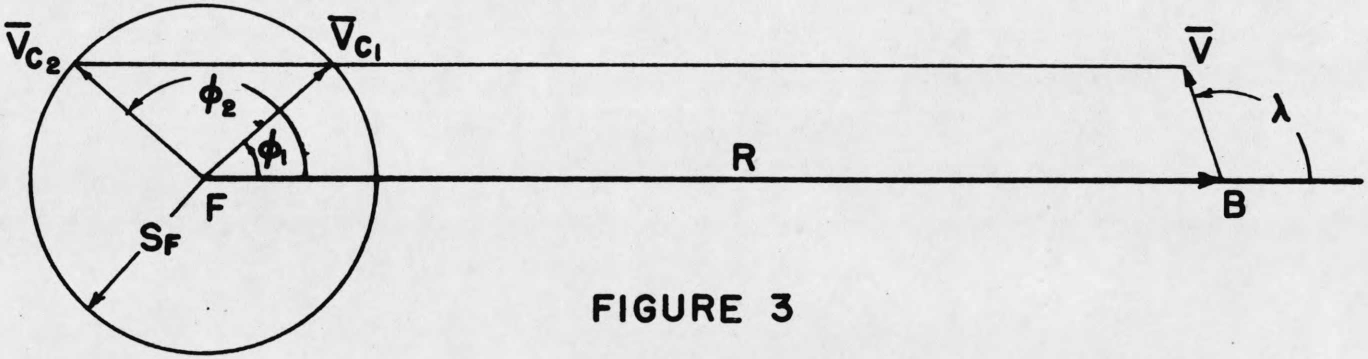


FIGURE 3

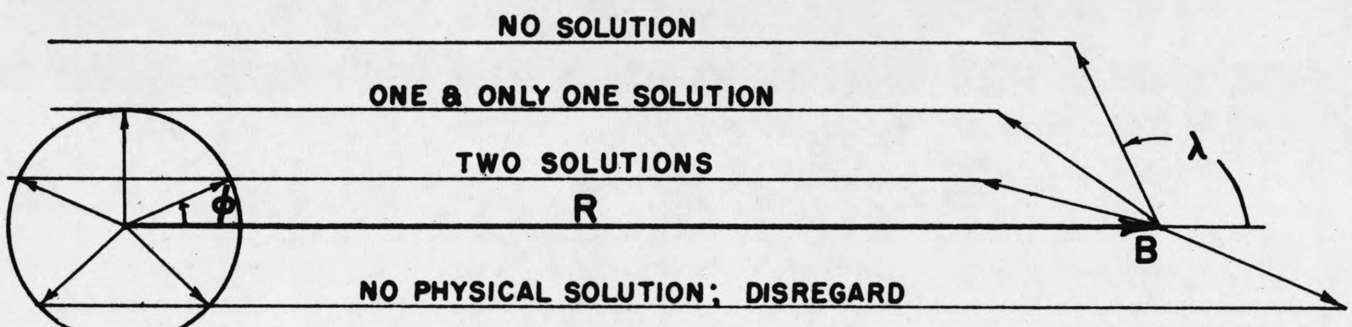


FIGURE 4

CONFIDENTIAL

CONFIDENTIAL

ERROR IN ARCTAN APPROXIMATION

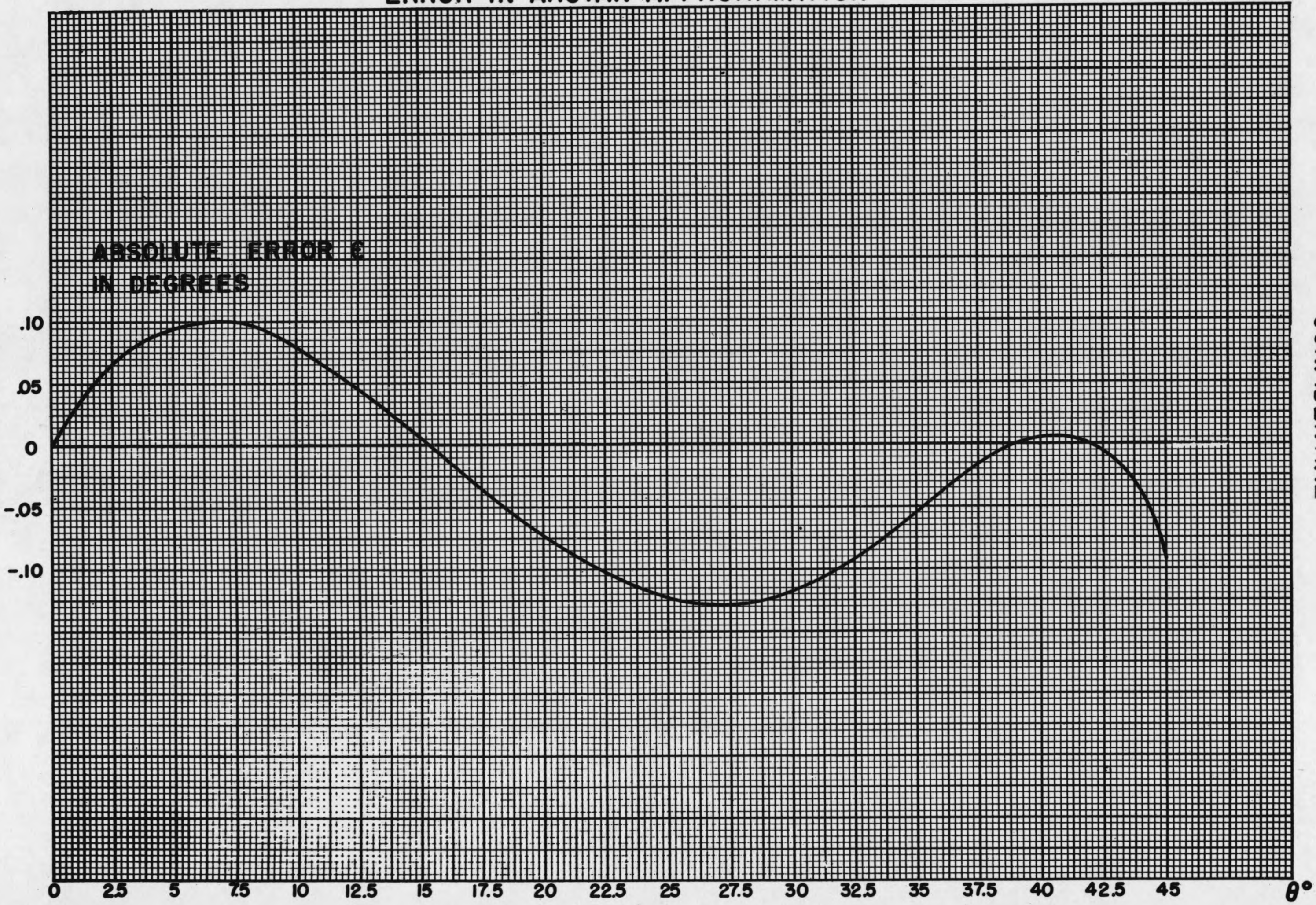


FIGURE 5

CONFIDENTIAL

120-43

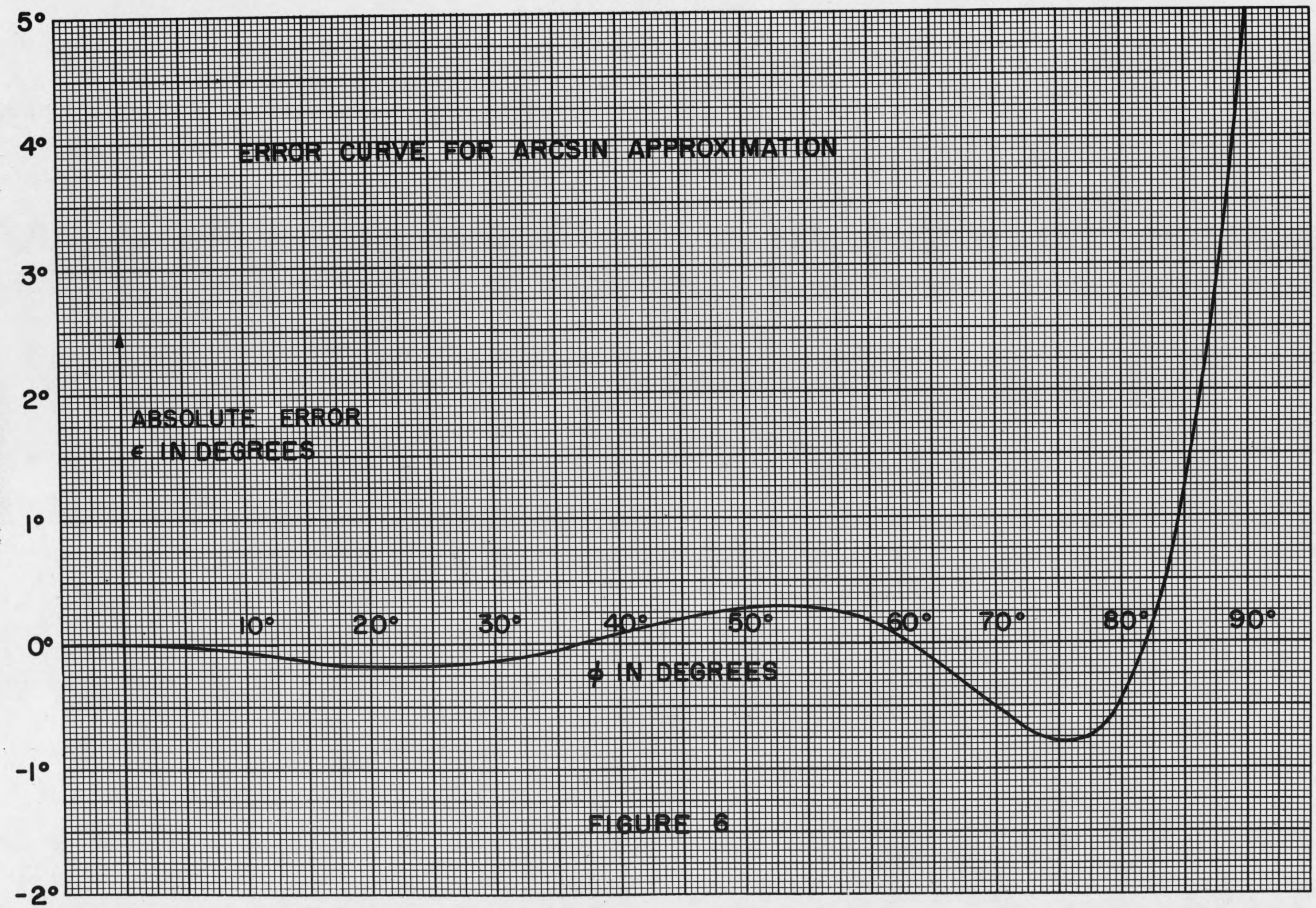


FIGURE 6

CONFIDENTIAL

CONFIDENTIAL

120-44

ERROR IN $\sqrt{x^2 + y^2}$ APPROXIMATION

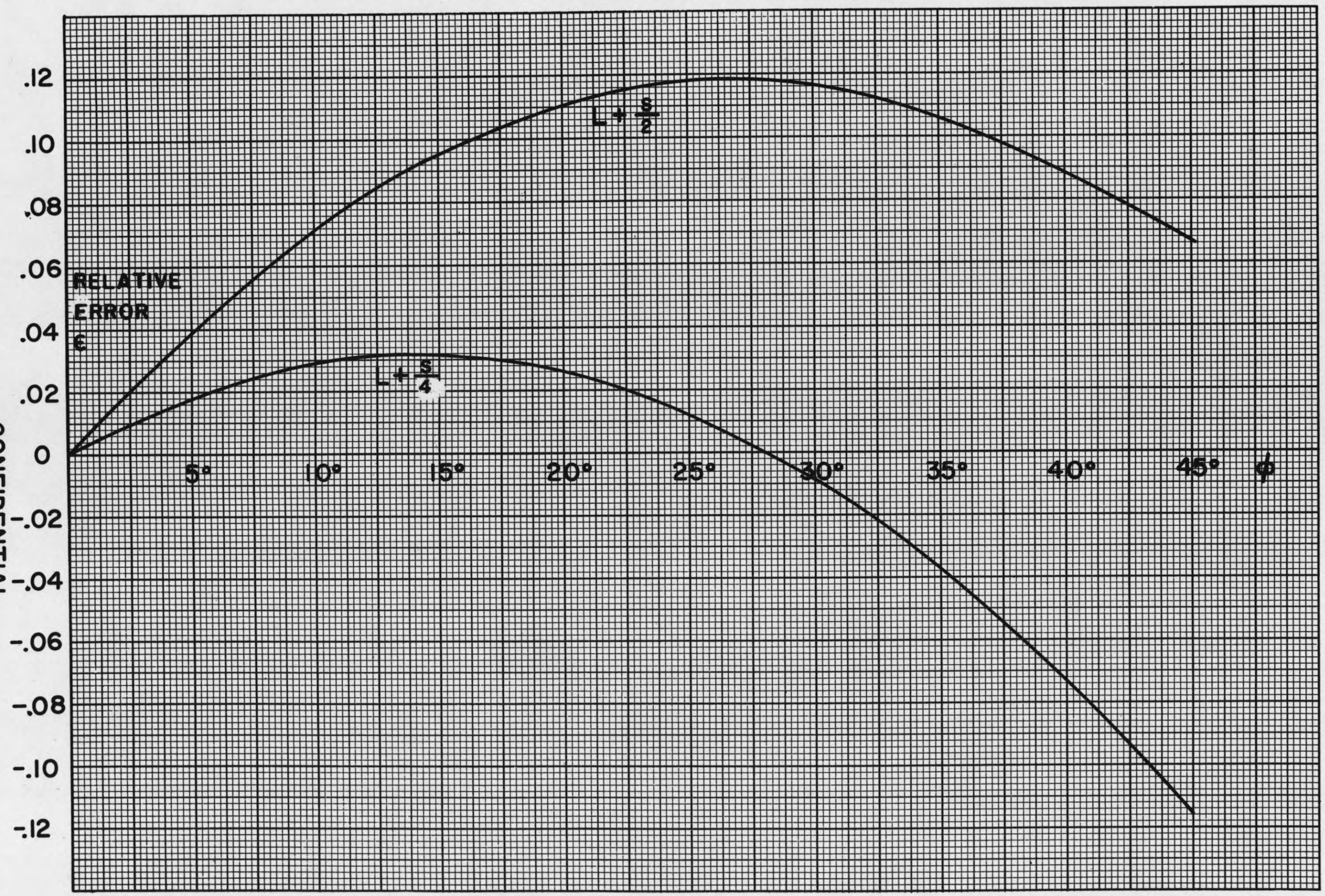


FIGURE 7

CONFIDENTIAL

CONFIDENTIAL

120-45

ERROR IN $\sqrt{x^2+y^2}$ APPROXIMATION

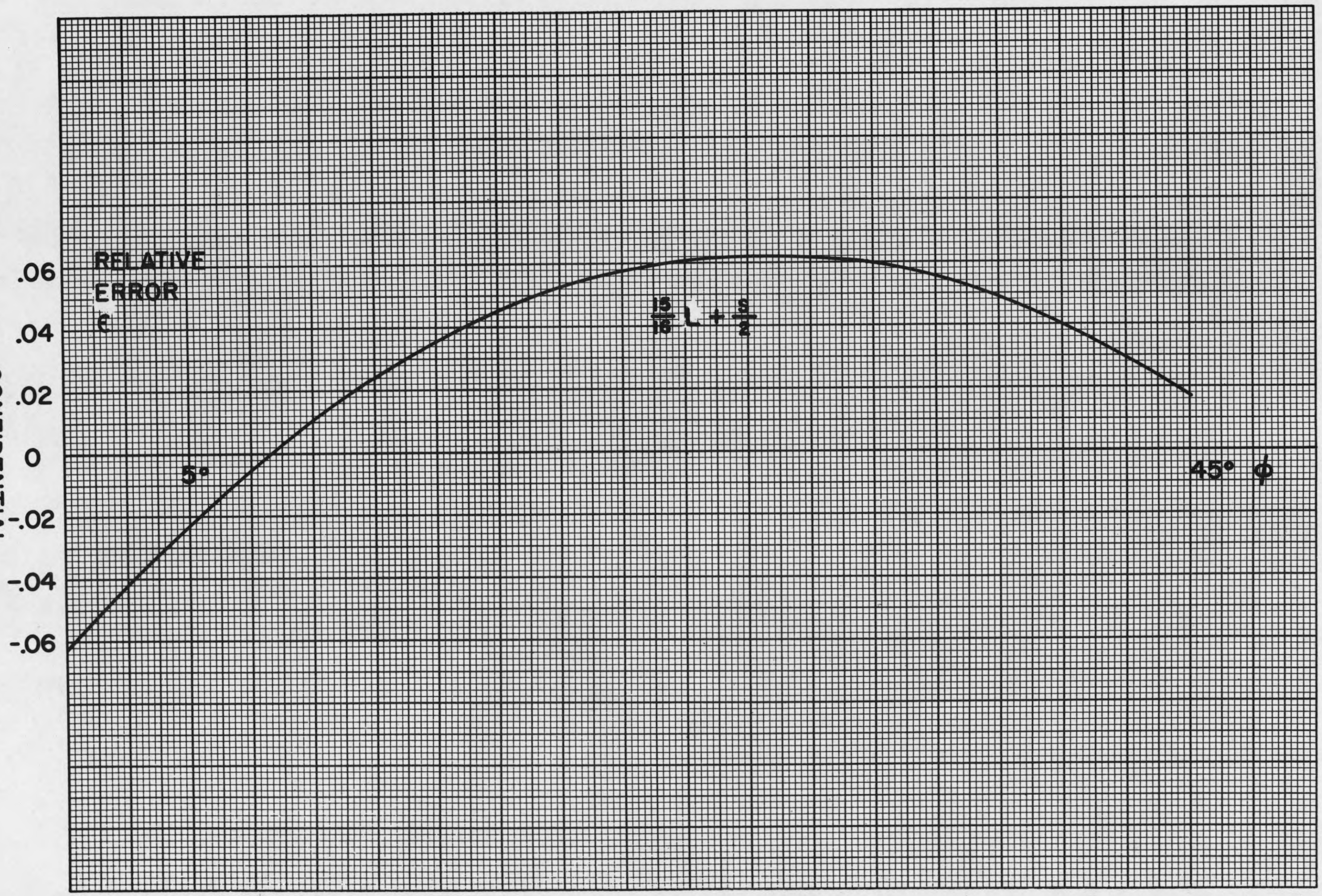
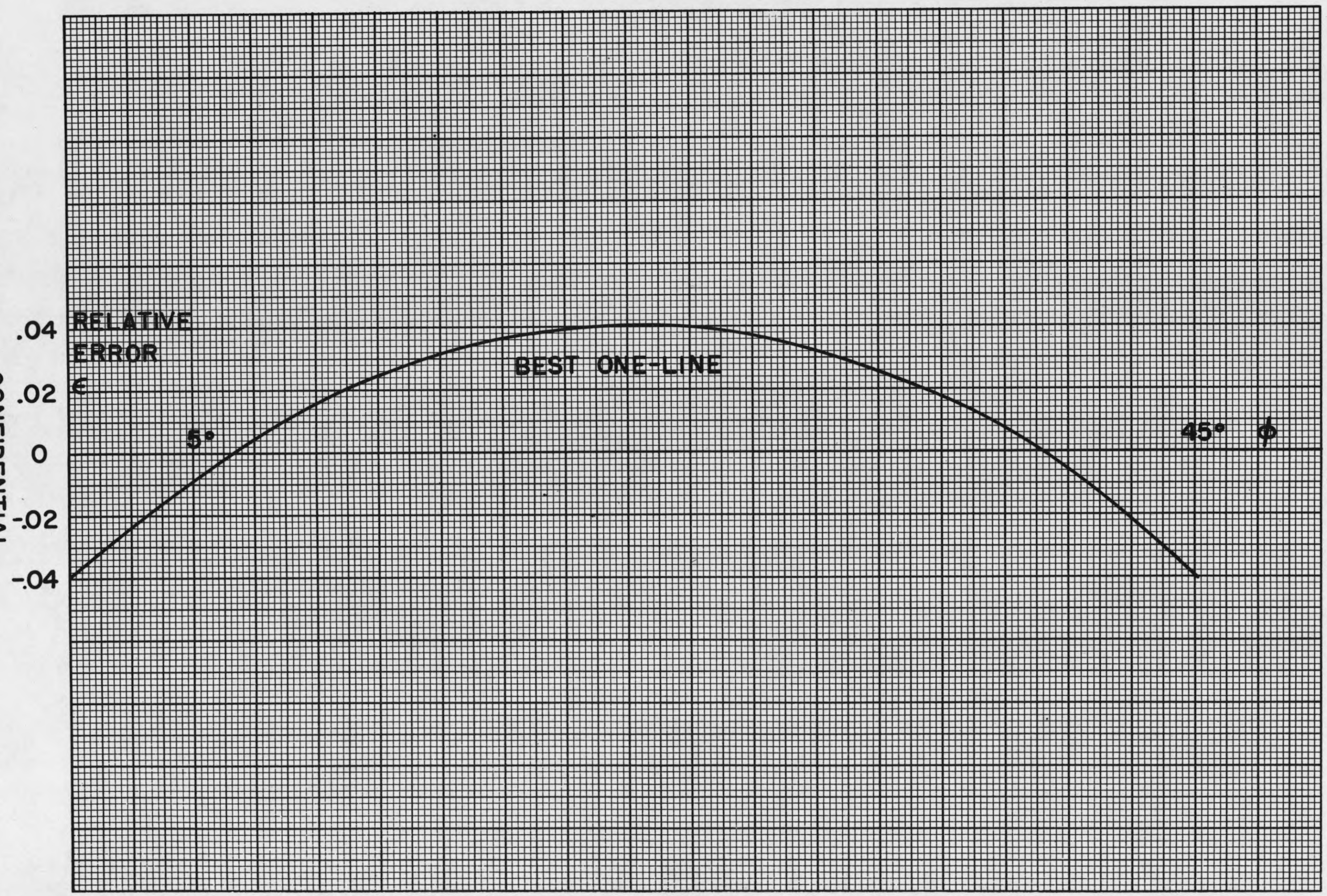


FIGURE 8

120-46

ERROR IN $\sqrt{x^2+y^2}$ APPROXIMATION



120-47

FIGURE 9

ERROR IN $\sqrt{x^2+y^2}$ APPROXIMATION

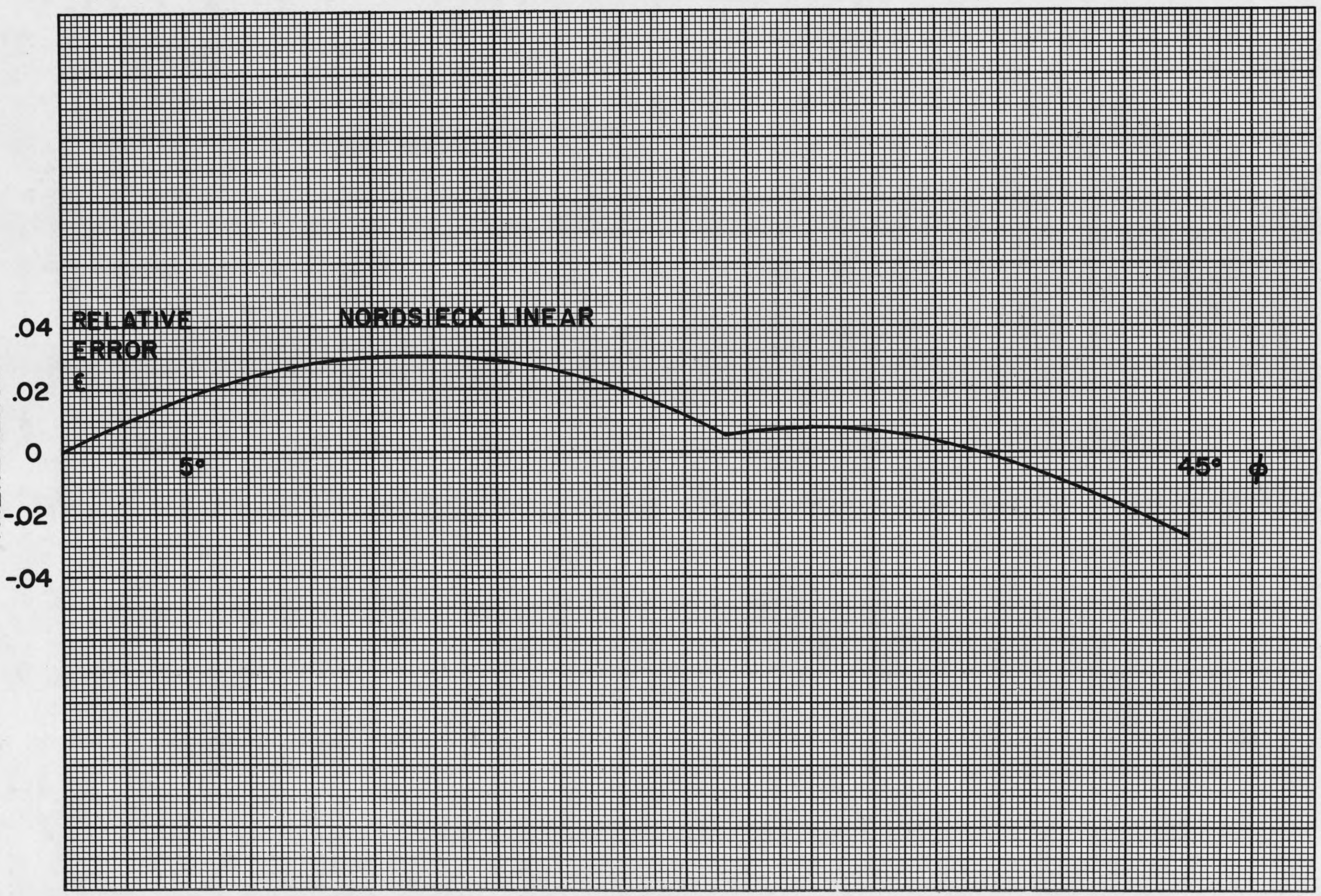


FIGURE 10

120-48

CONFIDENTIAL

ERROR IN $\sqrt{x^2 + y^2}$ APPROXIMATION

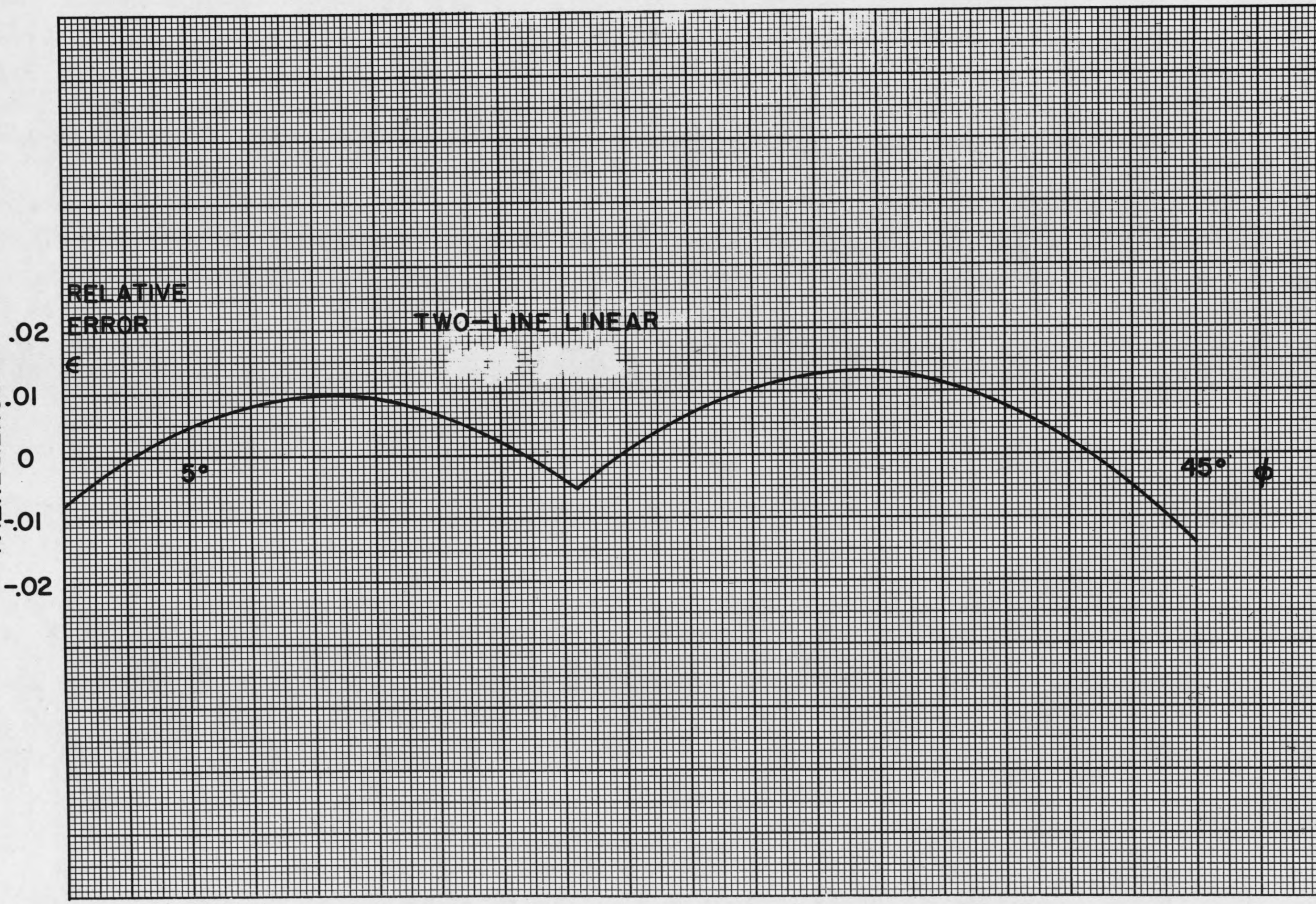


FIGURE II

CONFIDENTIAL

120-49

ERROR IN $\sqrt{x^2+y^2}$ APPROXIMATION

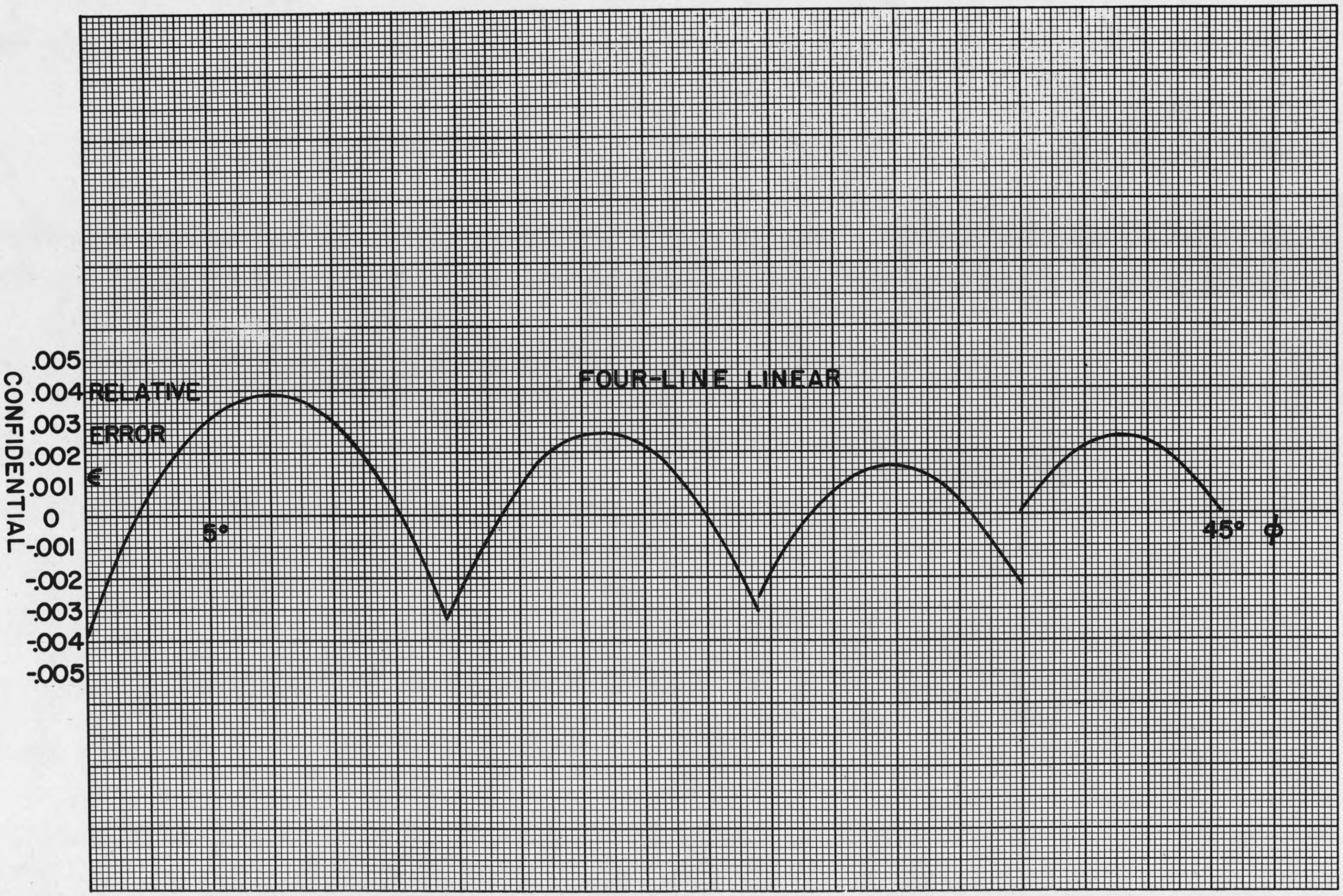


FIGURE 12

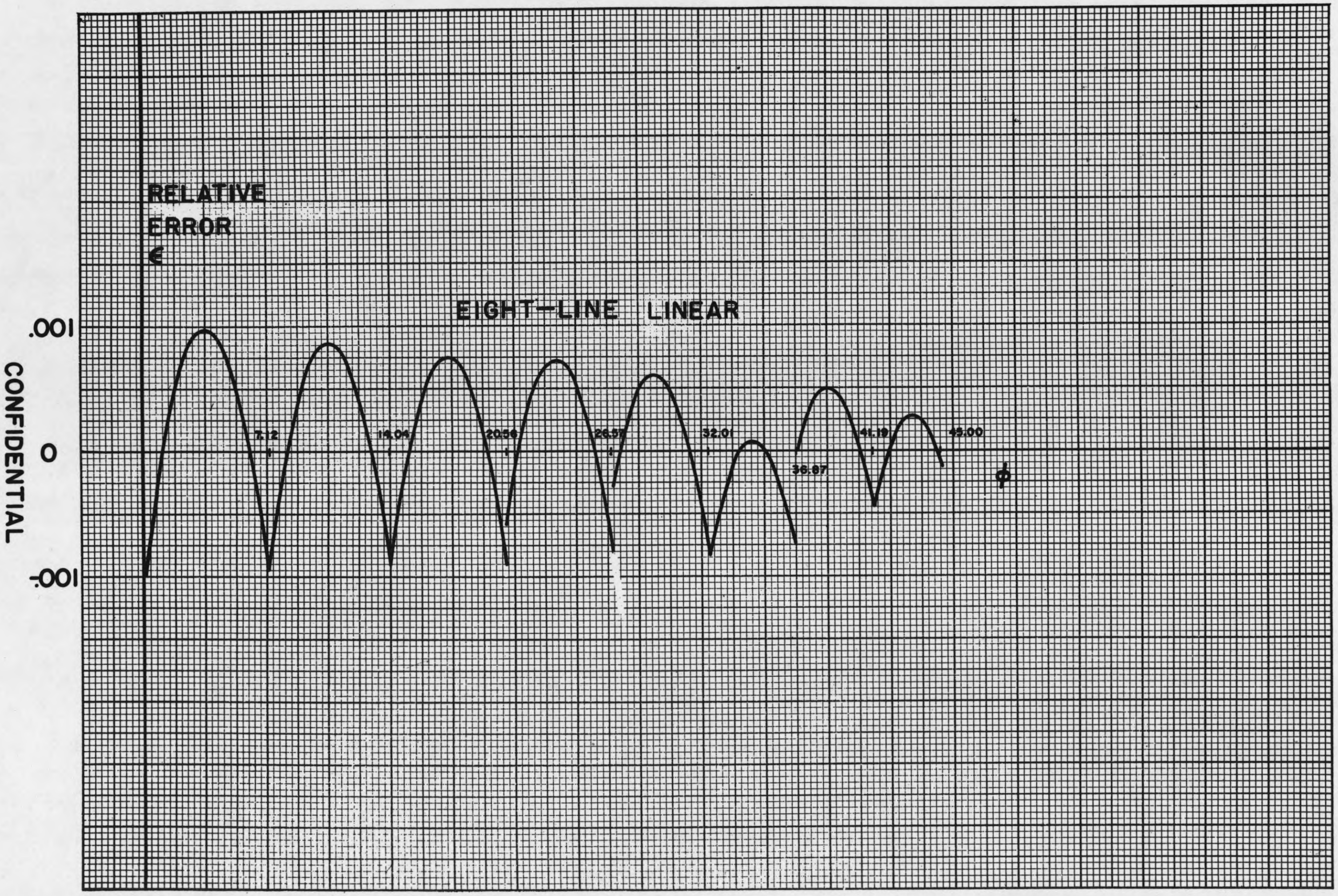
CONFIDENTIAL

CONFIDENTIAL

120-50

ERROR IN $\sqrt{x^2+y^2}$ APPROXIMATION

CONFIDENTIAL

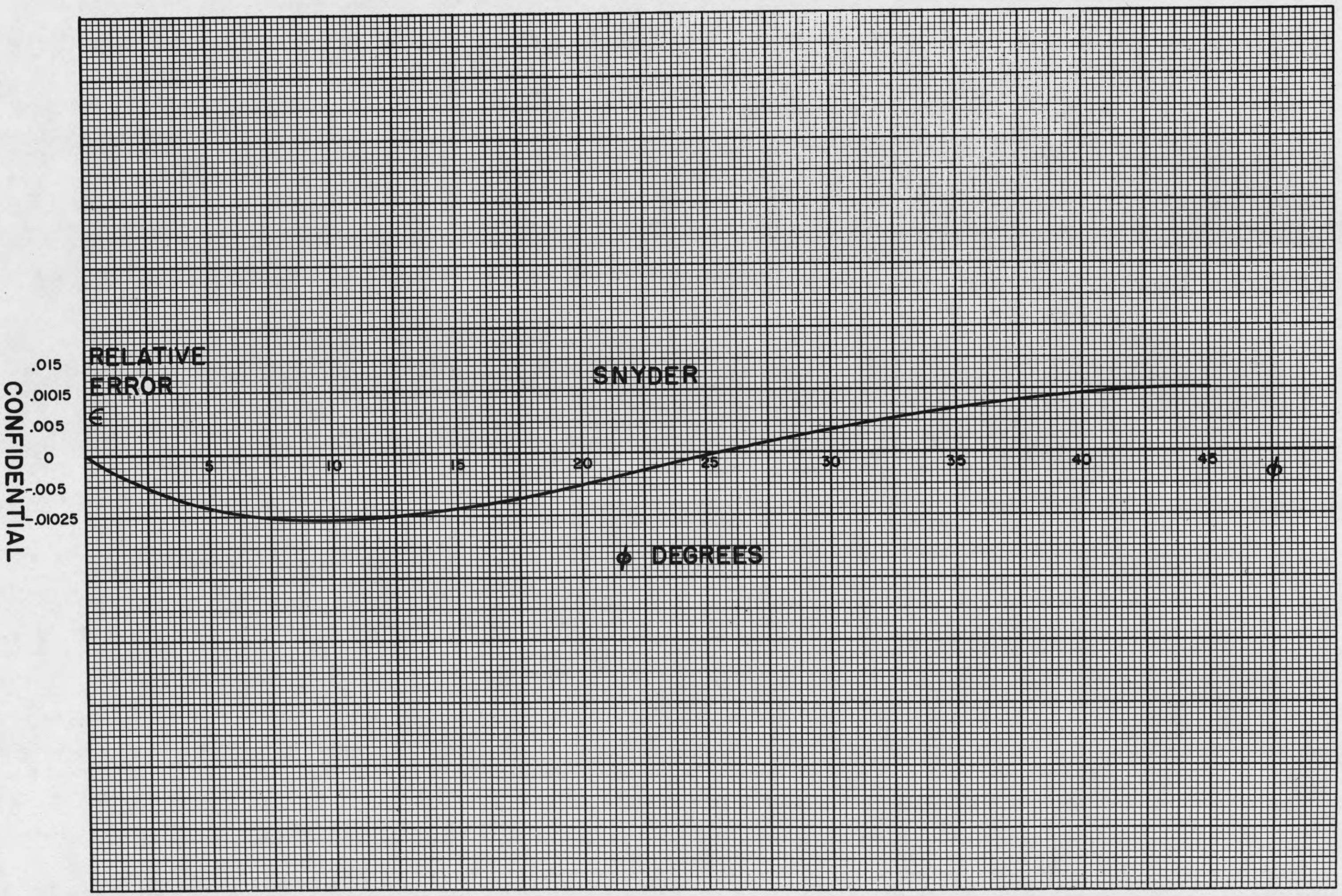


CONFIDENTIAL

120-51

FIGURE 13

ERROR IN $\sqrt{x^2+y^2}$ APPROXIMATION



CONFIDENTIAL

CONFIDENTIAL

FIGURE 14

120-52

ERROR IN $\sqrt{x^2+y^2}$ APPROXIMATION

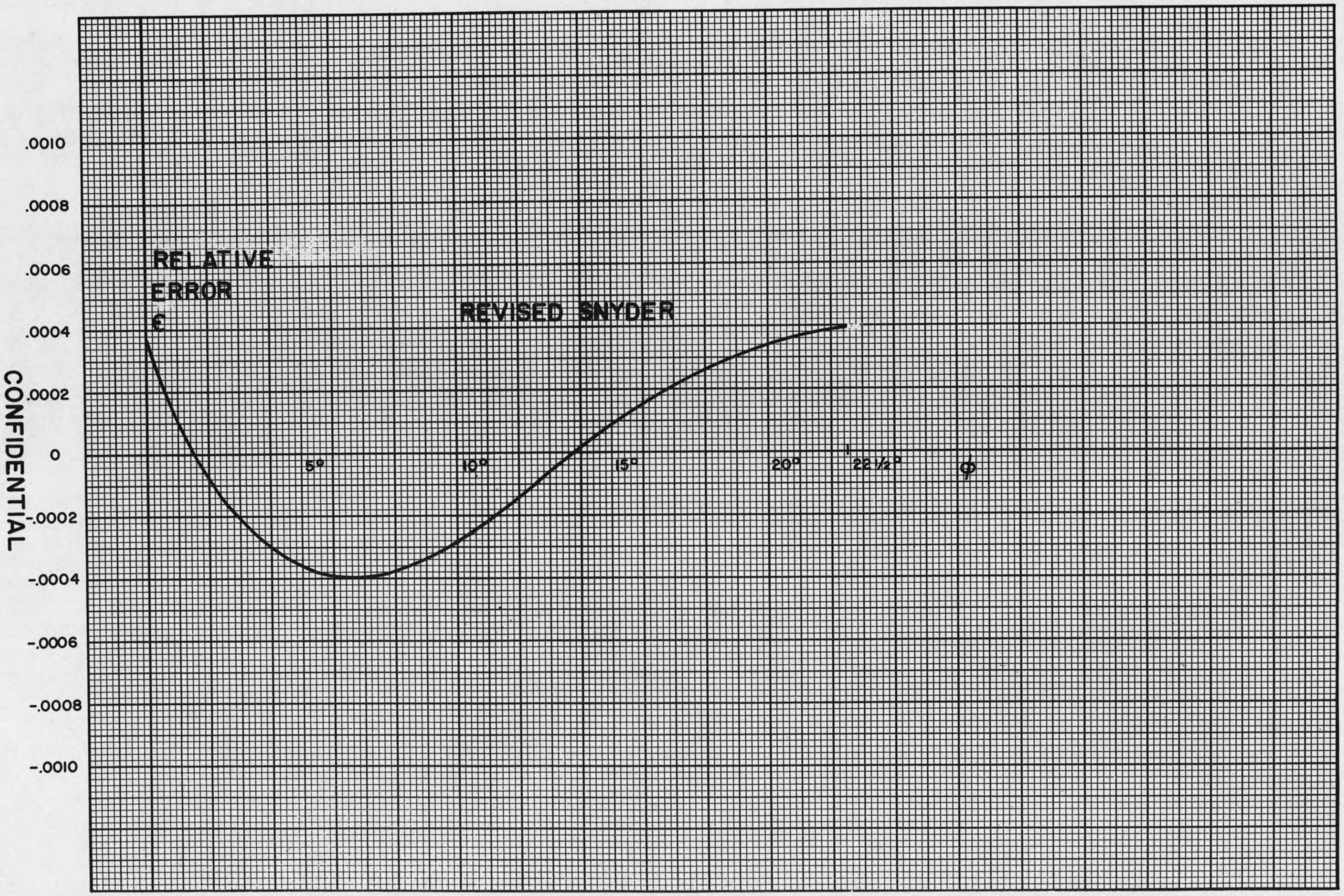


FIGURE 15

CONFIDENTIAL

CONFIDENTIAL

120-53

MAXIMUM ERRORS IN θ_L

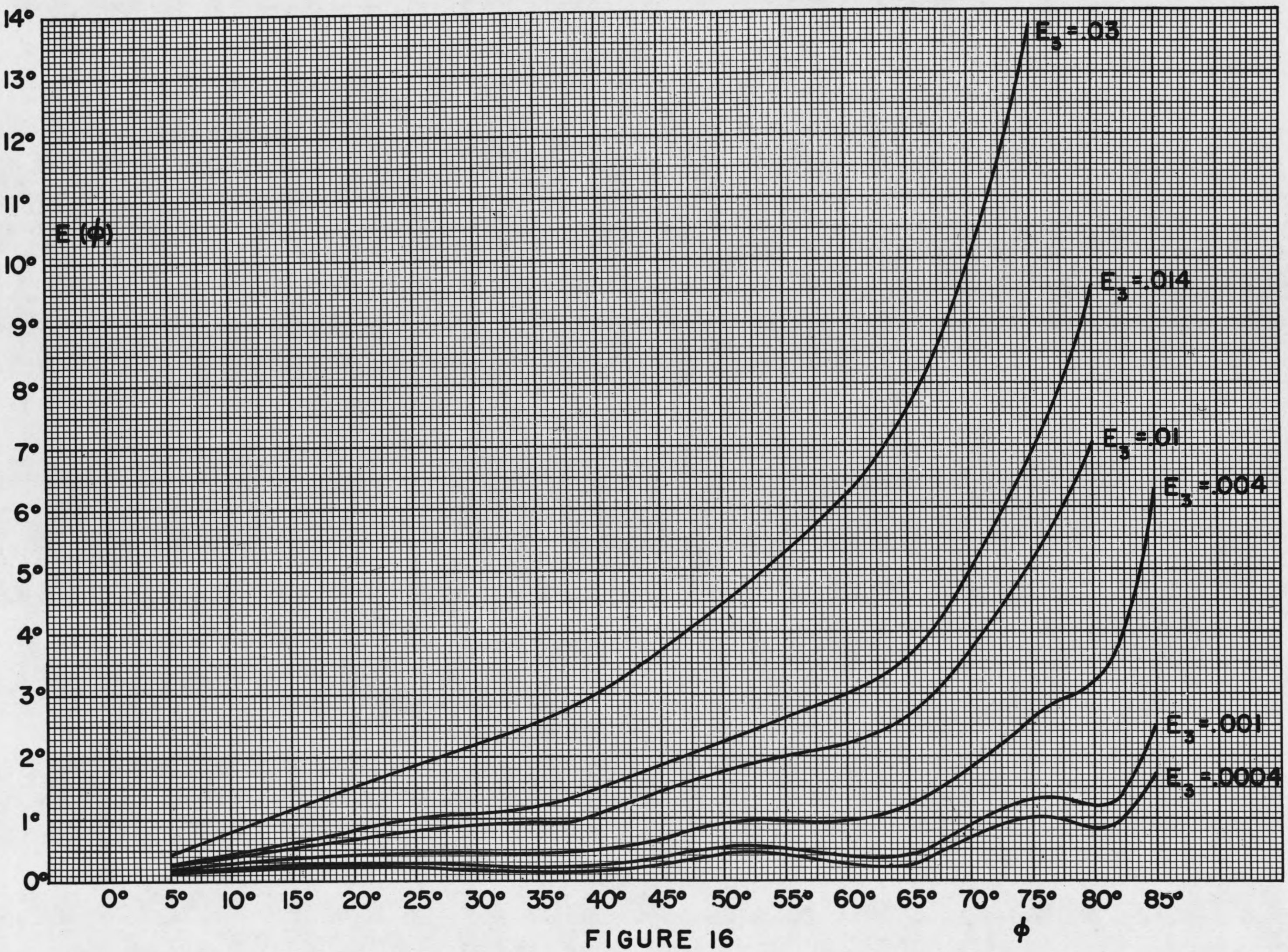


FIGURE 16

CONFIDENTIAL

CONFIDENTIAL

120–54

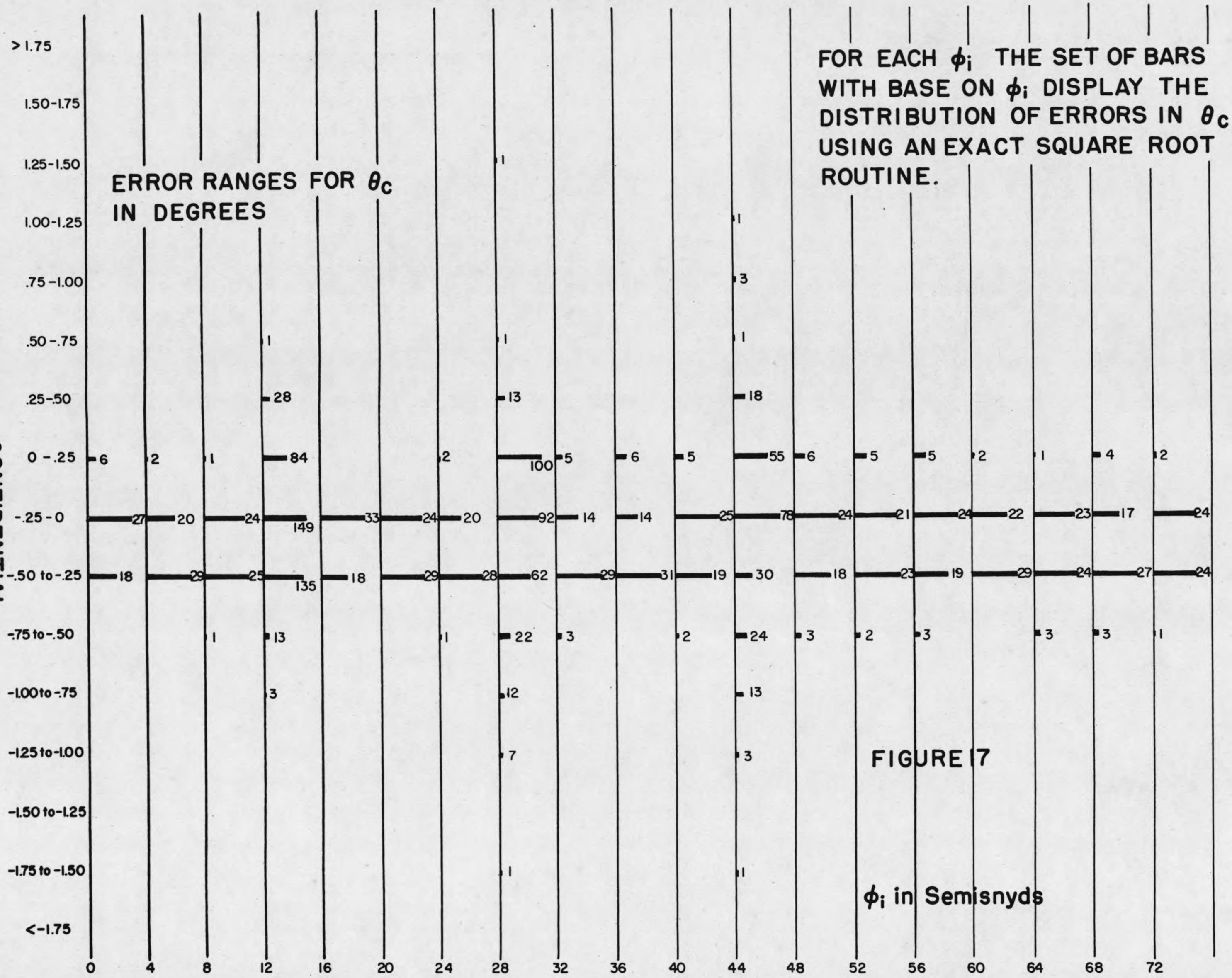


FIGURE 17

120-55

CONFIDENTIAL

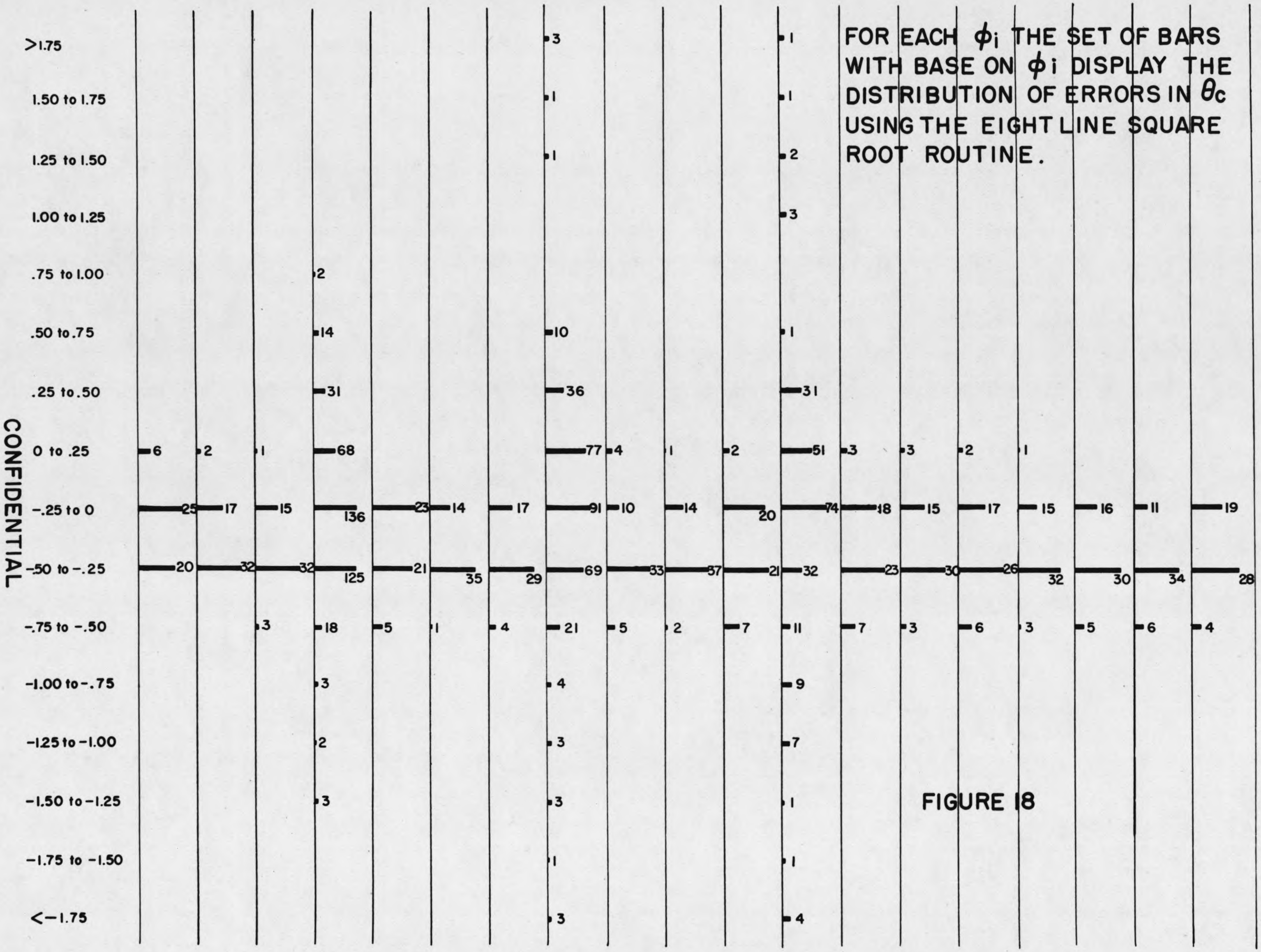


FIGURE 18

CONFIDENTIAL

120-56

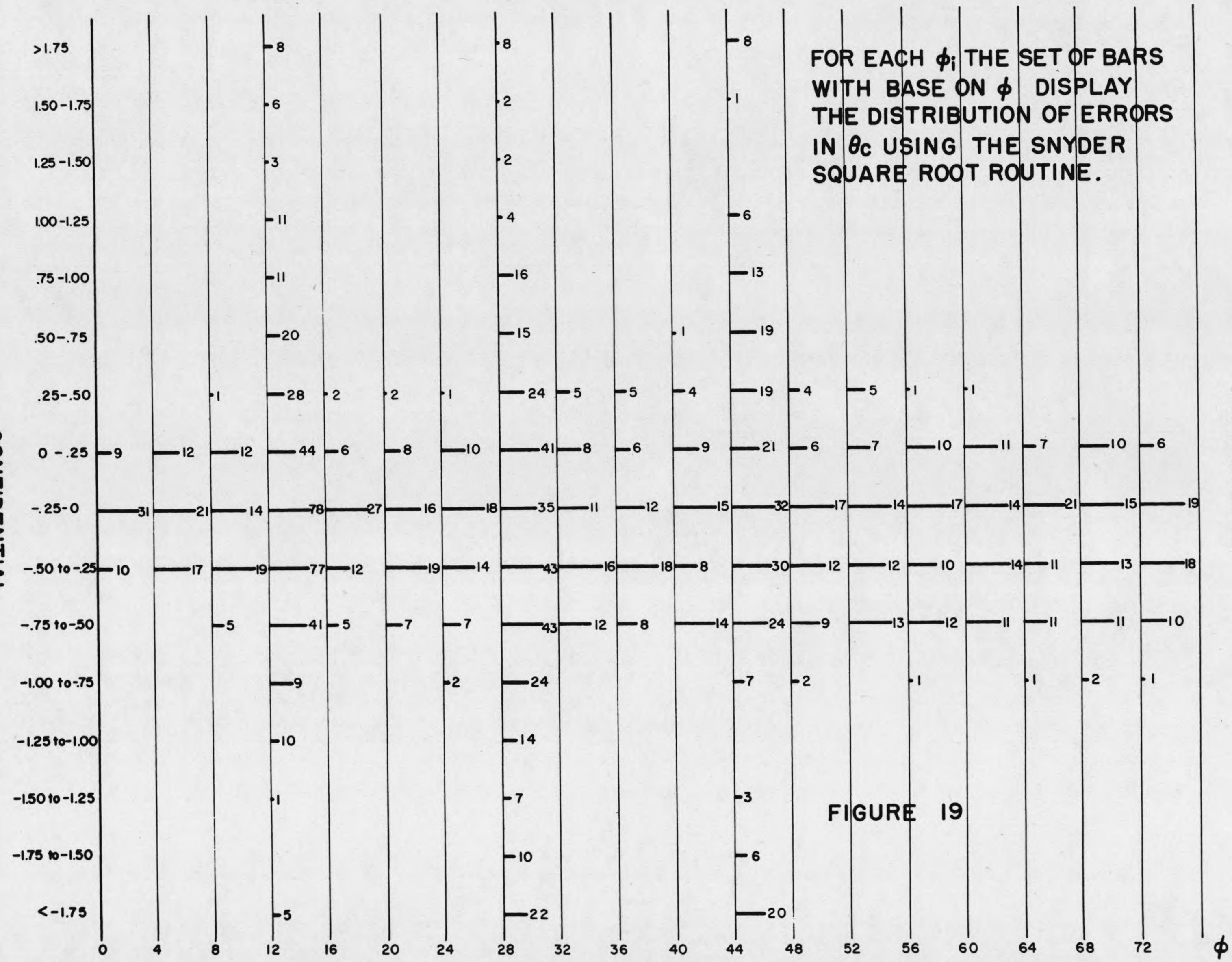


FIGURE 19

120-57

EFFECT OF ERRORS IN K OR λ ON φ

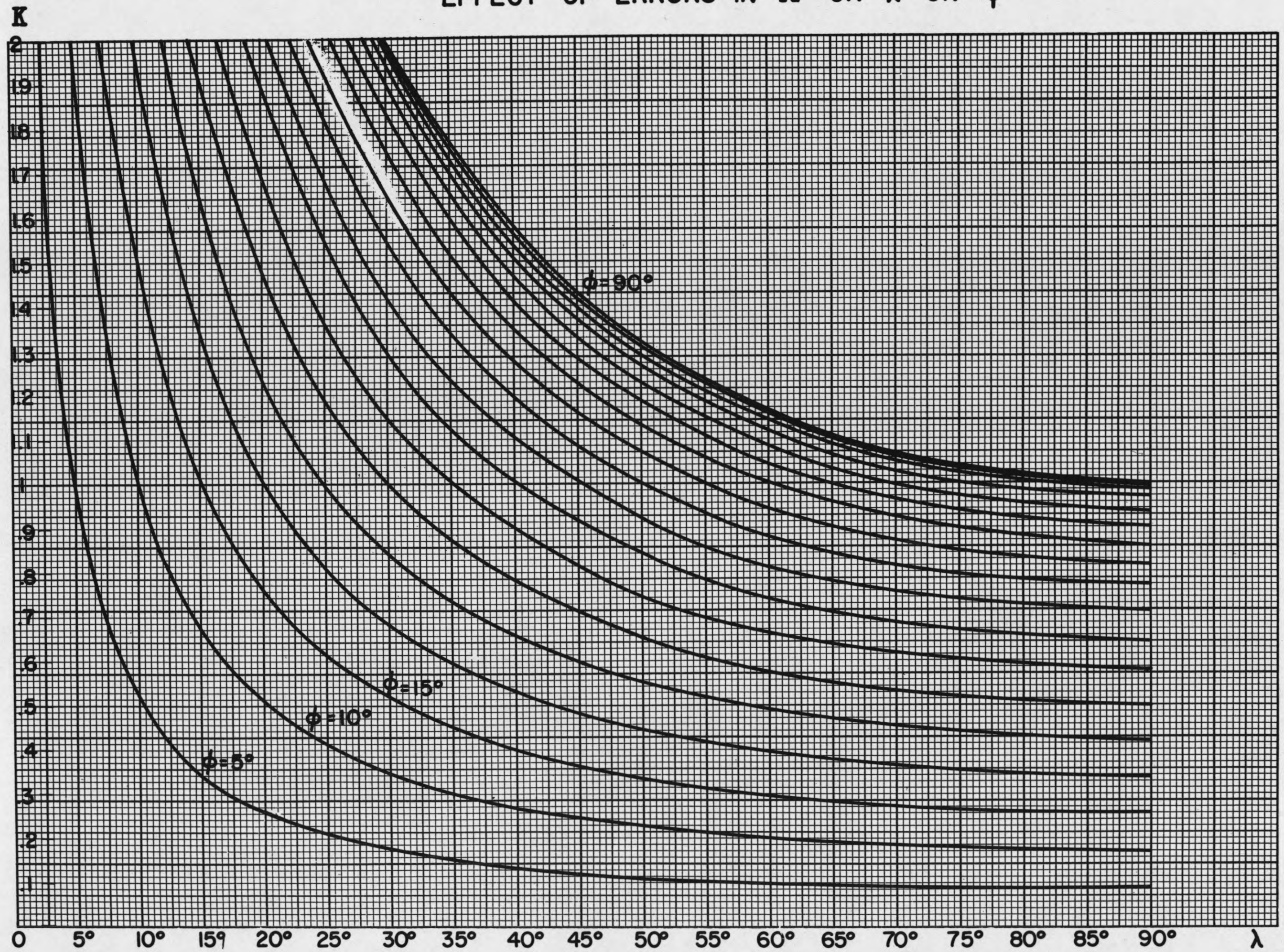


FIGURE 20

CONFIDENTIAL

CONFIDENTIAL

120-58

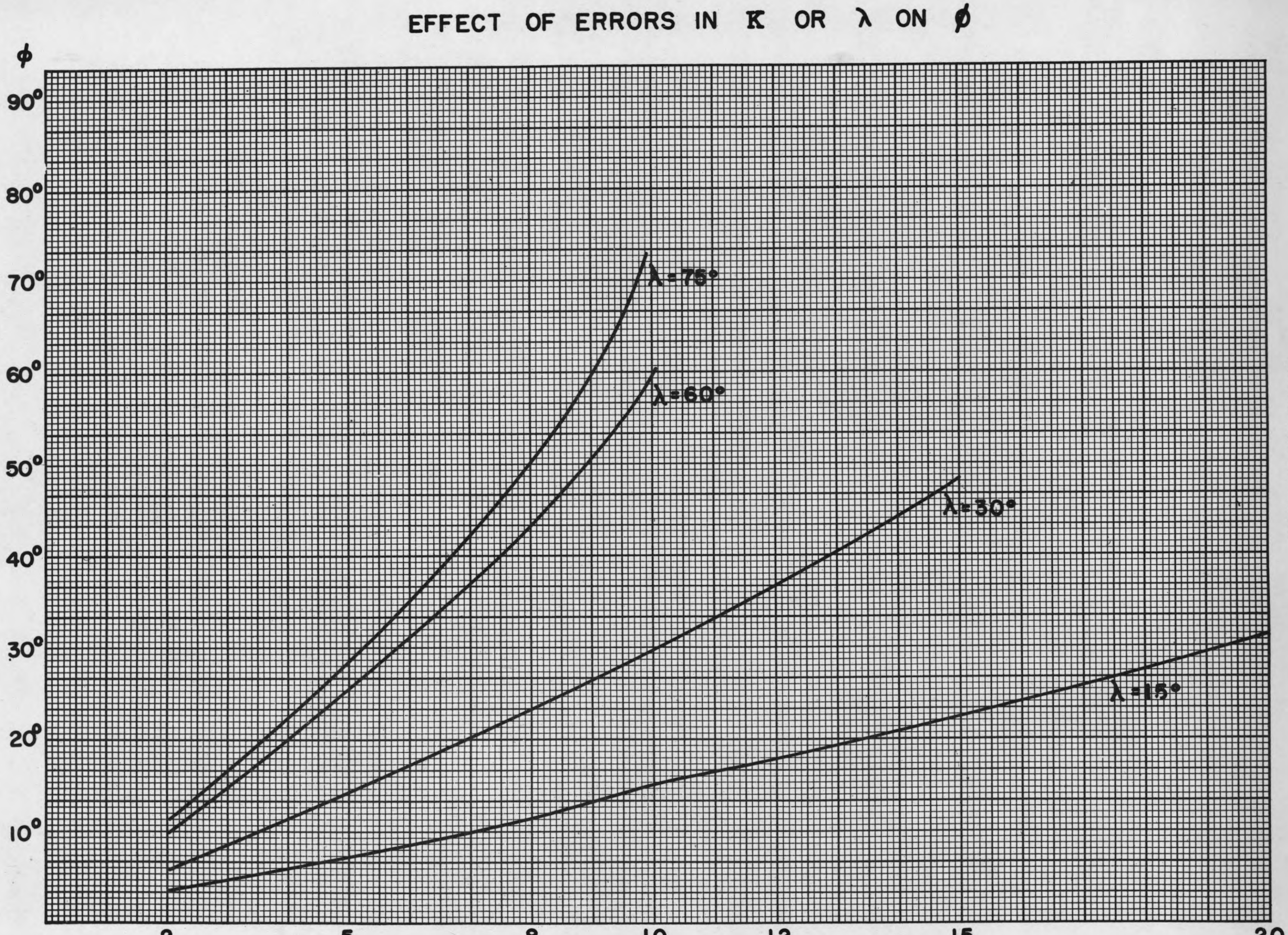


FIGURE 21

CONFIDENTIAL

CONFIDENTIAL

EFFECT OF ERRORS IN K OR λ ON φ

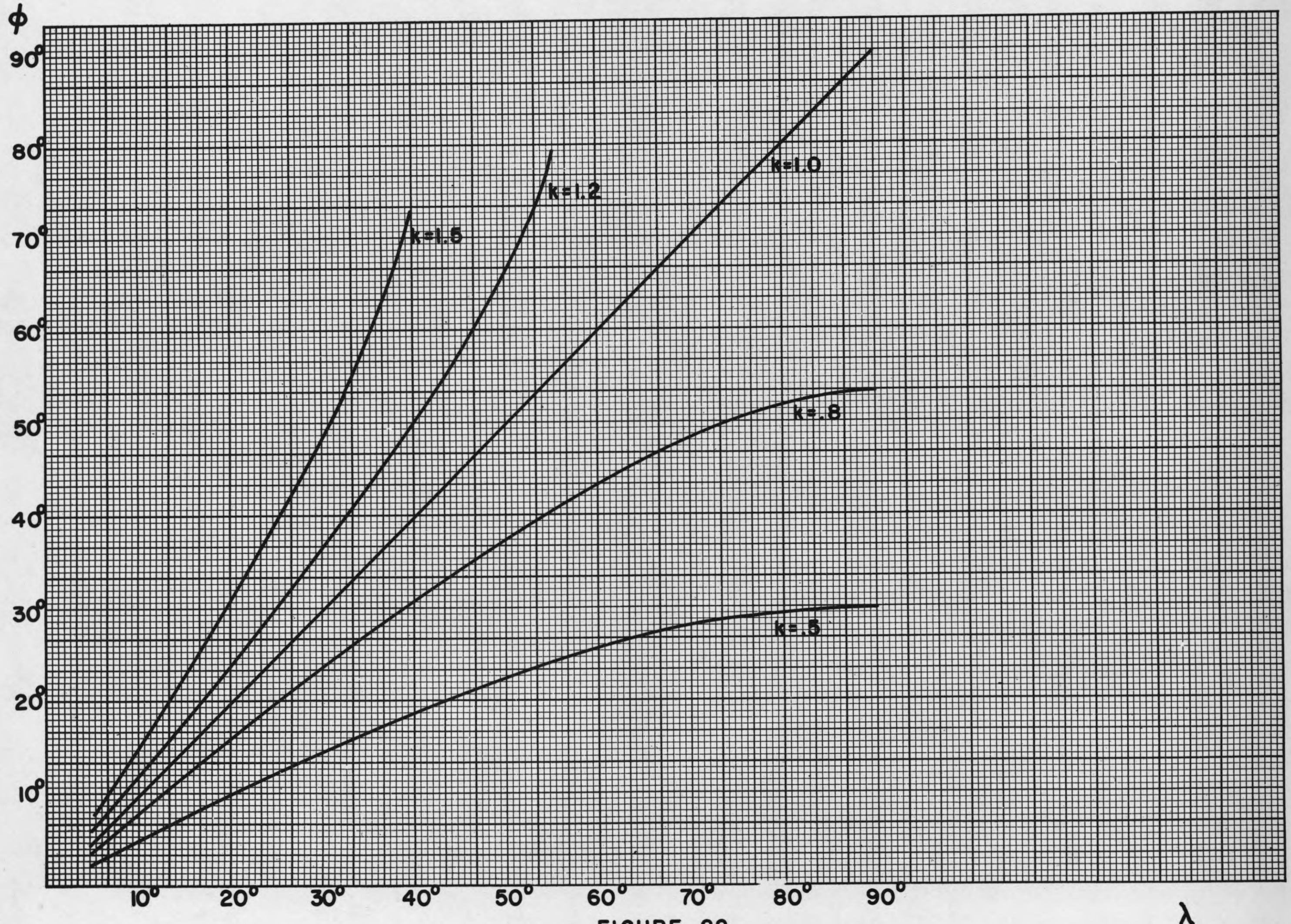


FIGURE 22

120-60

AUTHORIZED DISTRIBUTION LIST

Number of Copies	Agency
	Director of Research and Development Headquarters United States Air Force Washington 25, D. C.
1	Attn: AFDRD-AC/2
1	AFDRD-CC/2
1	Commander Headquarters, Air Research and Development Command P. O. Box 1395 Baltimore 3, Maryland Attn: RDDDR-5
1	Headquarters, Air Force Office of Scientific Research Air Research and Development Command United State Air Force Washington 25, D. C. Attn: SROP
1	Commander Air Force Cambridge Research Center Laurence G. Hanscom Field Bedford, Massachusetts Attn: CRR
	Commander Wright Air Development Center Wright-Patterson Air Force Base, Ohio
2	Attn: WCOSI-3
1	WCLG (1) WCLGR-4
1	WCLJ (1) WCLJY
1	WCLR (1) WCLGN-5
1	WCLOT-2 (1) WCLRW-2
2	WCLRW
1	Commander United States Air Force Security Service San Antonio, Texas Attn: CLR
1	Commanding Officer Rome Air Development Center Griffiss Air Force Base, New York

Number of Copies	Agency
1	Director Air University Library Maxwell Air Force Base, Alabama Attn: CR-4803a
1	Commander Air Force Armament Center Eglin Air Force Base, Florida Attn: Deputy for Operations
	Chief, Bureau of Ships Department of the Navy Washington 25, D. C.
2	Attn: Code 280
1	Code 565C
1	Code 810
1	Code 810B
1	Code 812
1	Code 820
1	Code 825
1	Code 830
1	Code 835
	Chief, Bureau of Aeronautics Department of the Navy Washington 25, D. C.
1	Attn: EL-402
1	TD-4
	Department of the Navy Office of Naval Research Washington 25, D. C.
1	Attn: Code 900
1	Code 430
2	Code 437
1	Commanding Officer Office of Naval Research Chicago Branch John Crerar Library Building 10th Floor, 86 East Randolph Street Chicago 1, Illinois
1	Bureau of Ordnance Department of the Navy Washington 25, D. C. Attn: Re4C

Number of Copies	Agency
1	Director Office of Naval Research Branch Office 1000 Geary Street San Francisco, California
2 (Progress Reports only)	U. S. Navy Inspector of Ordnance Applied Physics Laboratory The Johns Hopkins University 8621 Georgia Avenue Silver Springs, Maryland
	Commanding Officer and Director U. S. Naval Electronics Laboratory San Diego 52, California
1	Attn: Library
1	Code 2800, C. S. Manning
	Director Naval Research Laboratory Washington 25, D. C.
3	Attn: Code 5140
	Chief of Naval Operations Navy Department Washington 25, D. C.
1	Attn: OP-51
1	OP-371-C
1	OP-551
1	OP-345
1	Commanding Officer Naval Air Development Center Johnsville, Pennsylvania Attn: Code AAEL
1	Commander Naval Ordnance Laboratory White Oaks Silver Springs 19, Maryland Attn: Technical Library
1	Head, Combat Direction Systems Branch (OP-345) Department of the Navy Room 4C-518 Pentagon Washington 25, D. C.

Number of Copies	Agency
	Department of the Army Office of the Chief Signal Officer Washington 25, D. C.
1	Attn: SIGRD
1	SIGRD-9-b
1	Chief of Research and Development Office of the Chief of Staff Department of the Army Washington 25, D. C.
1	Assistant Chief of Staff, Development and Test Headquarters, Continental Army Command Fort Monroe, Virginia
1	President, U. S. Army Airborne and Electronics Board Continental Army Command Fort Bragg, North Carolina
	Office of the Chief of Ordnance Department of Ordnance Washington 25, D. C.
1	Attn: ORDTR
1	ORDTB
1	President, U. S. Army Defense Board Continental Army Command Fort Bliss, Texas
2	Commanding General Redstone Arsenal Huntsville, Alabama Attn: Technical Library
1	Commanding Officer Office of Ordnance Research 2127 Myrtle Drive Duke Station Durham, North Carolina
3	Director Ballistics Research Laboratory Aberdeen Proving Ground, Maryland Attn: Dr. L. A. Delsasso
1	Commanding Officer Frankford Arsenal Philadelphia 37, Pennsylvania

Number of Copies	Agency
1	9560 S. C. Electronics Research Unit P. O. Box 205 Mountain View, California
20	Transportation Officer Fort Monmouth Little Silver, New Jersey Marked for: SCEL Accountable Property Officer Building 2700 Camp Wood Area Inspect at Destination File No. 0060-PH-54-91(5308)
1	Director National Bureau of Standards Washington 25, D. C. Attn: Dr. S. N. Alexander
1	Librarian Instrumentation Laboratory Massachusetts Institute of Technology Cambridge 39, Massachusetts
1	Director Jet Propulsion Laboratory California Institute of Technology Pasadena, California
1	Chicago Midway Labs 6040 South Greenwood Avenue Chicago 37, Illinois Attn: Librarian
1	Hughes Research and Development Library Hughes Aircraft Company Culver City, California Attn: Miss Mary Jo Case
1	Mr. Robert R. Everett Division Head Lincoln Laboratory Massachusetts Institute of Technology Lexington 73, Massachusetts
1	Technical Documents Service Willow Run Laboratories University of Michigan Willow Run Airport Ypsilanti, Michigan

Number of Copies	Agency
1	The Rand Corporation 1700 Main Street Santa Monica, California Attn: Library
1	Joseph P. Desmond, Librarian Cornell Aeronautical Laboratory Buffalo, New York
1	Massachusetts Institute of Technology Lincoln Laboratory P. O. Box 73 Lexington 73, Massachusetts
1	W. L. Maxson Corporation 460 West 34th Street New York 1, New York
1	Stanford University Electronics Research Laboratory Stanford, California
1	Radio Corporation of America RCA Laboratories Divison David Sarnoff Research Center Princeton, New Jersey Attn: Mr. A. W. Vance
1	The Johns Hopkins University Operations Research Office 6935 Arlington Road Bethesda, Maryland Washington 14, D. C. Attn: Document Control Office
1	Light Military Electronic Equipment Department General Electric Company French Road Utica, New York For: Contract AF 33(600)-16934
1	Goodyear Aircraft Corporation Akron 15, Ohio For: Project MX 778 Contract W33-038 ac-14153
1	Mr. A. A. Lundstrom Bell Telephone Laboratories Whippany, New Jersey

Number of Copies	Agency
10	Armed Services Technical Information Agency Arlington Hall Station Arlington 12, Virginia
1	Litton Industries 336 North Foothill Road Beverly Hills, California Via: Inspector of Naval Material Los Angeles, California
1	Remington Rand Univac Division of Sperry Rand Corporation Via: Insmat, BuShips Insp. Officer 1902 West Minnehaha Avenue St. Paul 4, Minnesota
1	Technical Library Code 142 David Taylor Model Basin Washington 7, D. C.
1	Commanding Officer and Director David Taylor Model Basin Washington 7, D. C. Attn: Code 800
1	Naval Ordnance Proving Ground Computation Center Dahlgren, Virginia Attn: R. A. Niemann
1	System Development Corporation 2500 Colorado Avenue Santa Monica, California
5 (Progress Reports only)	National Advisory Committee for Aeronautics 1512 H Street Northwest Washington 25, D. C.
1	General Electric Company Defense Electronics Center Cornell University Ithaca, New York Attn: Mr. Leonard Biegelow