(12) LEVEL

NSWC/DL TR 3636

COMPUTATION OF THE BIVARIATE NORMAL DISTRIBUTION OVER CONVEX POLYGONS

by
A. R. DiDONATO
M. P. JARNAGIN, Jr.
R. K. HAGEMAN
Strategic Systems Department

ADA069406

SEPTEMBER 1978

Approved for public release; distribution unlimited.

NAVAL SURFACE WEAPONS CENTER
Dahlgren, Virginia 22448 • Silver Spring, Maryland 20910

79 06 04 050

NAVAL SURFACE WEAPONS CENTER
Dahlgren, Virginia 22448

Paul L. Anderson, Capt., USN
Commander

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER (14) NSWC/DL-TR-3886	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) (6) COMPUTATION OF THE BIVARIATE NORMAL DISTRIBUTION OVER CONVEX POLYGONS	5. TYPE OF REPORT & PERIOD COVERED (9) Final repts
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) A. R. Di Donato, M. P. Jarnagin, Jr., R. K. Hageman	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Surface Weapons Center (K05) Dahlgren, Virginia 22448	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NIF
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Surface Weapons Center (K05) Dahlgren, Virginia 22448	12. REPORT DATE (11) September 1978
	13. NUMBER OF PAGES 61
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 58 p.	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Bivariate normal distribution over convex polygons

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A procedure is given for computing the bivariate normal probability over an angular region or a convex polygon. The procedure is implemented into a Fortran IV computer program which is designed to yield 3, 6, or 9 decimal digits of accuracy. Comparisons with two other published methods, for the same achievable accuracy, show our program to be much faster

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

391598

S/N 0102- LF- 014- 6601

FOREWORD

The work described in this report was done in the Science and Mathematics Research Group of the Strategic Systems Department. Requests for such work over the last fifteen years were instrumental in its initiation. We acknowledge Ann Howes' contributions to the editorial aspects of this report.

R. A. NIEMANN, Head
Strategic Systems Department

ABSTRACT

A procedure is given for computing the bivariate normal probability over an angular region or a convex polygon. The procedure is implemented into a Fortran IV computer program which is designed to yield 3, 6, or 9 decimal digits of accuracy. Comparisons with two other published methods, for the same achievable accuracy, show our program to be much faster.

CONTENTS

1. INTRODUCTION

In this report we give a numerical procedure for integrating the bivariate normal density function over a convex polygon. The Fortran IV computer program* developed from it is fast and is designed to yield the output probability to 3, 6, or 9 decimal digit accuracy. As far as we know, it is the fastest and most versatile program of its kind – most versatile in the sense that it handles, with three prespecified levels of accuracy in the output, arbitrary convex polygons** rather than just triangles and quadrilaterals. We make note at this time that the program serves as a basic subroutine for the automatic computation of the bivariate normal over an *arbitrary* polygon. A complete program for this much more general case has been written, checked out, and is operational. Its description is deferred to a later report.

Our procedure for the convex polygon case depends on a fast method, with prespecified accuracy, to evaluate the bivariate normal distribution over an angular region, Λ. In particular, we wish to evaluate

$$(1)\qquad \bar{P}(\Lambda) = \frac{(1-\rho^2)^{-1/2}}{2\pi\sigma_1\sigma_2}\iint_\Lambda \exp\left\{-\left[\left(\frac{w-\mu_1}{\sigma_1}\right)^2 - 2\rho\,\frac{(w-\mu_1)(z-\mu_2)}{\sigma_1\sigma_2} + \left(\frac{z-\mu_2}{\sigma_2}\right)^2\right]\Big/2(1-\rho)^2\right\}dw\,dz,$$

where (μ_1, μ_2) is the mean and $\begin{bmatrix}\sigma_1{}^2 & \rho\sigma_1\sigma_2\\ \rho\sigma_1\sigma_2 & \sigma_2^2\end{bmatrix}$ the covariance matrix of the normal random variable (w,z) with correlation coefficient ρ. The angular region Λ is defined as the semi-infinite part of the plane bounded by two intersecting directed straight lines. Of course by this definition there are four such regions, and therefore it is necessary to always state which of them is involved.

The well-known linear transformation

$$(2)\qquad x = \left[\frac{w-\mu_1}{\sigma_1} - \rho\left(\frac{z-\mu_2}{\sigma_2}\right)\right]\Big/\sqrt{1-\rho^2}, \qquad y = \frac{z-\mu_2}{\sigma_2},$$

reduces the integrand of (1) to one with circular symmetry, namely

$$(3)\qquad \bar{P}(\Lambda) = P(\Lambda) = \frac{1}{2\pi}\iint_\Lambda \exp[-(x^2 + y^2)/2]\,dx\,dy,$$

*The program is coded for the CDC-6700, a large-scale binary computer capable of one million operations per second. It has a 60 bit binary word length of which 48 are used to express the mantissa of a number.

**The term convex polygon will always mean a *closed* convex polygon.

where A, like Λ, is an angular region, since (2) takes straight lines into straight lines. Thus we deal only with (3) hereafter unless noted otherwise.

An extensive literature exists on methods for integrating the bivariate normal variate over various simple geometries, where the ultimate objective is to appropriately utilize these integrations to evaluate the distribution over a polygon, [2,3,4,5,6,7]. One such case is where Λ in (1) forms a right angle at (h,k) with the sides of the angle directed parallel to the w and z directions. When the mean is zero and the variances are equal to one, (1) for the angular region just described is denoted by $\Phi(h,k,\rho)$ and is called the bivariate normal integral [3] or the bivariate normal probability function [9, p. 936]. We shall make reference to Φ in Section 6. We show it is equivalent to (3) where the given right angle is transformed to an angular region A and then show that a recent method for computing Φ, [3], is slower than our procedure for obtaining the same result from (3).

The idea of integrating over an angular region seems to have originated with Gideon and Gurland (hereafter G & G), [4]* [5]. As observed by them, the idea of integrating over an angular region, as expressed by (3), is a natural and easily visualized way to obtain the probability over a polygon. In Section 5 we shall discuss and compare their computing method with ours.

In Section 3 we show how, by utilizing (3) over a set of angular regions, we obtain the probability over a convex polygon. Our approach differs here also from what G & G advocate. In Section 7, we give some numerical results. The computer program is described in Section 4 with its Fortran listing given in Appendix D. In the next section we give some analysis and also the algorithm for evaluating (3). Its implementation into a computer program is not straightforward since certain precautions are necessary as will be explained in Section 4.

2. ALGORITHM FOR $P[A(R, \theta_1, \theta_2)]$

In this section we derive the algorithm by which we evaluate (3); i.e., we obtain that part of the circular normal distribution over the angular region $A(R, \theta_1, \theta_2)$ as the shaded region shown in Figure 1. Lines ① and ② form the boundaries of this region. R denotes the distance from the origin to the vertex of $A(R, \theta_1, \theta_2)$.

It is convenient because of circular symmetry in the integrand of (3), to perform a rotation of axes such that the line L and the x axis coincide with A rigidly rotated as shown in Figure 2. Hereafter we shall always assume such a rotation, through the angle ψ, has been carried out.

The coordinate transformation

$$(4) \qquad x = R + r\cos\theta, \qquad y = r\sin\theta, \qquad |\theta| \leq \pi,$$

is used in (3) to obtain

*We are grateful to Pete Shugart at White Sands Missile Range, New Mexico for bringing their Wisconsin report [4], to our attention.

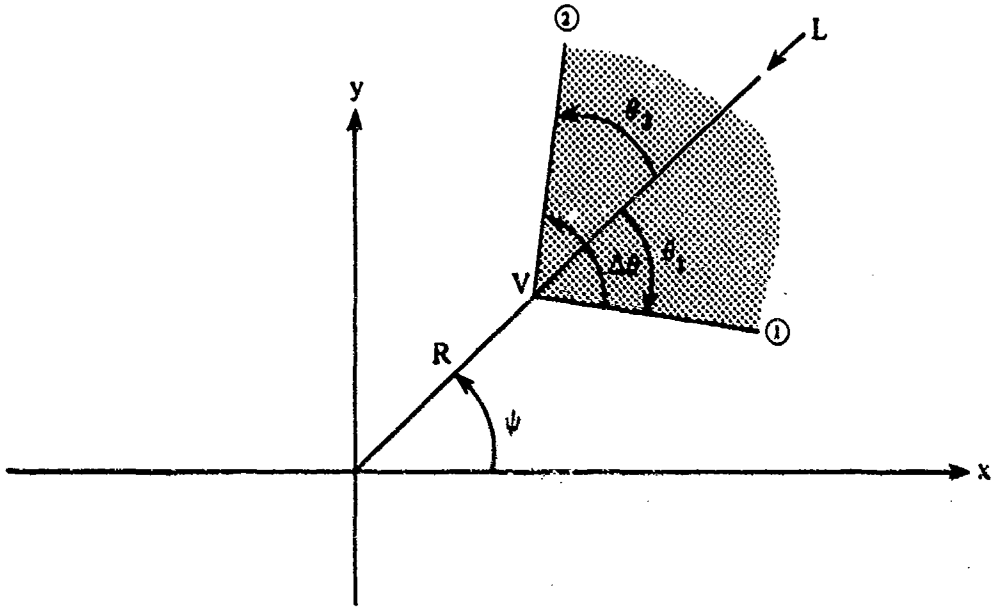


Figure 1. Angular Region, $A(R, \theta_1, \theta_2)$, (shaded region)

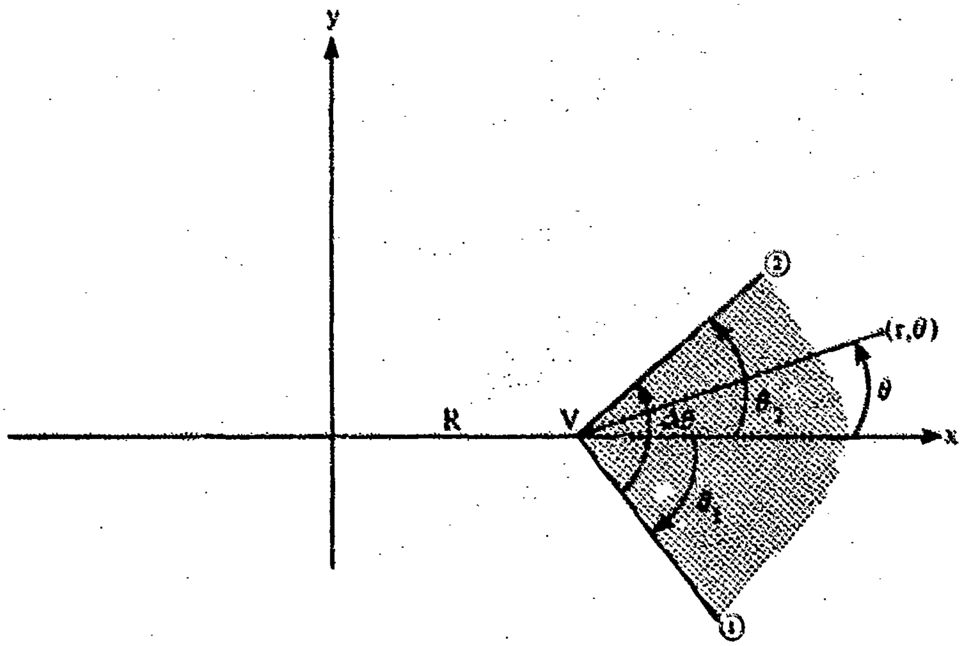


Figure 2. Angular Region $A(R, \theta_1, \theta_2)$ After Rotation

$$(5)\qquad P(A) = \frac{1}{2\pi}\int_{\theta_1}^{\theta_2}\int_0^\infty \exp\left[-\frac{1}{2}(R^2 + 2rR\cos\theta + r^2)\right] r\,dr\,d\theta$$

$$= \frac{1}{2\pi} e^{-R^2/2}\int_{\theta_1}^{\theta_2}\int_0^\infty re^{-r^2/2} e^{-pr}\,dr\,d\theta,$$

where

$$(6)\qquad p \equiv R\cos\theta.$$

An integration by parts on the integral in r yields

$$(7)\qquad \int_0^\infty re^{-r^2/2}e^{-pr}dr = 1 - p\int_0^\infty e^{-r^2/2}e^{-pr}dr$$

$$= 1 - pe^{p^2/2}\int_0^\infty e^{-\frac{1}{2}(r+p)^2}dr$$

$$= 1 - p\sqrt{2}\,[\mathrm{erfc}(p/\sqrt{2})]/z(p/\sqrt{2}),$$

where

$$(8)\qquad \mathrm{erfc}(x) = 1 - \mathrm{erf}(x) \equiv \int_x^\infty z(t)dt, \quad z(x) \equiv \frac{2}{\sqrt{\pi}}e^{-x^2}.$$

Using (7) in (5), carrying out the obvious part of the θ integration, (5) becomes

$$(9)\qquad \boxed{P(A) = e^{-R^2/2}\left\{\frac{\theta_2 - \theta_1}{2\pi} - \frac{1}{\pi}\int_{\theta_1}^{\theta_2} u[\mathrm{erfc}(u)/z(u)]\,d\theta\right\},}$$

where

$$(10)\qquad u \equiv p/\sqrt{2} = (R/\sqrt{2})\cos\theta.$$

We note for $R = 0$, (9) gives the exact result directly,

$$(11)\qquad P[A(0,\Delta\theta)] = \Delta\theta/2\pi, \quad \Delta\theta \equiv \theta_2 - \theta_1.$$

Equation (9) gives the relation for P(A) upon which our program is based. A similar relation was originally derived by Amos, [1], in an entirely different way.

The difficulty in evaluating the integral in (9) is resolved by obtaining, for a given $\delta > 0$, the minimax polynomial fit to erfc (u)/z(u) for $0 \leq u \leq C(\delta)$. Namely, a set of constants a_k and a least positive integer K are found such that

$$\left|\operatorname{erfc}(u) - z(u)\sum_{0}^{K} a_k u^k\right| \leqslant \frac{2}{\sqrt{\pi}}\,\delta,\quad 0 \leqslant u \leqslant C(\delta)\,. \tag{12}$$

The constant C is chosen, once δ is specified, such that

$$\frac{1}{2\pi}\iint_{\tilde{A}} \exp\left[-\frac{1}{2}(x^2 + y^2)\right] dx\,dy = \frac{1}{2}\operatorname{erfc}(\tilde{R}/\sqrt{2}) = \epsilon \equiv \delta/\sqrt{\pi}\,, \tag{13}$$

with

$$C = \tilde{R}/\sqrt{2}, \qquad \tilde{A} \equiv A\left(\tilde{R}, -\frac{\pi}{2}, \frac{\pi}{2}\right), \tag{14}$$

For $\delta \cong 5(-4), 5(-7), 5(-10), 2(-13)$ the a_k and C are given in Appendix A. For example for $\delta \cong 5(-7)$, $(\equiv 5 \times 10^{-7})$, we give $C = 3.5505$, $K = 9$. The way in which ϵ was chosen in (13) is explained below.

The integration in (9) can now be carried out numerically by recurrence relations. Indeed, from (12) and (9) we have

$$P(A) \sim \frac{e^{-R^2/2}}{\pi}\left[\frac{\theta_2 - \theta_1}{2} - \sum_{0}^{K} a_k \left(\frac{R}{\sqrt{2}}\right)^{k+1}\int_{\theta_1}^{\theta_2} \cos^{k+1}\theta\, d\theta\right], \quad |\theta| \leqslant \frac{\pi}{2}\,. \tag{15}$$

where

$$J_k \equiv \left(\frac{R}{\sqrt{2}}\right)^k \int_{\theta_1}^{\theta_2} \cos^k\theta\, d\theta\,, \tag{16}$$

so that

$$\boxed{J_{k+1} = \frac{1}{k+1}\left[\left(\frac{R}{\sqrt{2}}\cos\theta\right)^k \frac{R}{\sqrt{2}}\sin\theta\,\Bigg|_{\theta_1}^{\theta_2} + k\left(\frac{R}{\sqrt{2}}\right)^2 J_{k-1}\right],} \tag{17}$$

with

$$J_0 = \theta_2 - \theta_1\,, \qquad J_1 = \frac{R}{\sqrt{2}}\sin\theta_2 - \frac{R}{\sqrt{2}}\sin\theta_1\,. \tag{18}$$

Hence

$$\boxed{P(A) \sim \frac{e^{-R^2/2}}{\pi}\left[\frac{\theta_2 - \theta_1}{2} - \sum_{0}^{K} a_k J_{k+1}\right],\quad |\theta| \leqslant \frac{\pi}{2}\,,} \tag{19}$$

where it is emphasized that (15) and (19) hold only when $|\theta_i| \leqslant \pi/2$, $i = 1, 2$. This follows from (10), because $u \geqslant 0$ in (12) and $R \geqslant 0$ in (10) imply $\cos\theta \geqslant 0$. For cases outside this range, we make use of the fact that

$$(20) \qquad P[A(R,0,\theta)] = \frac{1}{2}\operatorname{erfc}\left(\frac{R}{\sqrt{2}}\sin\theta\right) - P[A(R,0,\pi-\theta)], \quad \frac{\pi}{2} \leqslant \theta \leqslant \pi,$$

where we prefer to work with the coerror function, erfc (see (8)), instead of the univariate cumulative distribution function of a normal variable. They are related by

$$(21) \qquad \frac{1}{2}\operatorname{erfc}(x/\sqrt{2}) = 1 - \frac{1}{2\sqrt{2}}\int_{-\infty}^{x} z(t/\sqrt{2})\,dt.$$

The implementation of (19) and (20) is discussed in Section 4, which deals with the computer program.

We now show that if a maximum error of $\frac{2}{\sqrt{\pi}}\delta$ is made in approximating

$$(22) \qquad f(u) = \operatorname{erfc}(u), \qquad 0 \leqslant u \leqslant C(\delta),$$

as noted in (12), then the truncation error in computing P(A), using (19), can be no larger than $\delta/\sqrt{\pi}$. Indeed, from (12)

$$(23) \qquad \left|F(u) - \sum_{0}^{K} a_k u^k\right| \leqslant \delta\, e^{u^2}, \quad u \geqslant 0, \quad F(u) \equiv \frac{f(u)}{z(u)},$$

and since $u \geqslant 0$, we have

$$(24) \quad -\frac{\delta}{\pi}e^{-R^2/2}\int_{\theta_1}^{\theta_2} ue^{u^2}\,d\theta \leqslant \frac{e^{-R^2/2}}{\pi}\int_{\theta_1}^{\theta_2}\left[uF(u) - \sum_{0}^{K} a_k u^{k+1}\right]d\theta \leqslant \frac{\delta}{\pi}e^{-R^2/2}\int_{\theta_1}^{\theta_2} ue^{u^2}\,d\theta.$$

But, with (10),

$$(25) \qquad e^{-R^2/2}\int_{\theta_1}^{\theta_2} ue^{u^2}\,d\theta = \int_{\theta_1}^{\theta_2}\left(\frac{R}{\sqrt{2}}\cos\theta\right)\exp\left[-\frac{R^2}{2}\sin^2\theta\right]d\theta$$

$$= \frac{\sqrt{\pi}}{2}\left[\operatorname{erf}\left(\frac{R}{\sqrt{2}}\sin\theta_2\right) - \operatorname{erf}\left(\frac{R}{\sqrt{2}}\sin\theta_1\right)\right] \leqslant \sqrt{\pi},$$

and (24) then implies

$$(26) \qquad \left|\frac{e^{-R^2/2}}{\pi}\int_{\theta_1}^{\theta_2}\left[uF(u) - \sum_{0}^{K} a_k u^{k+1}\right]d\theta\right| \leqslant \delta/\sqrt{\pi} \ (= \epsilon).$$

This accounts for the way ϵ was chosen in (13).

The dominant part of the computation in evaluating P(A) from (19) is the generation of the sum of terms $\{a_k J_{k+1}\}$. Two situations can occur for which this sum does not contribute to the value of P(A) to within the accuracy specified, namely when R is "small" and when R is "large." In the first case, we have

$$(27) \qquad P(A) \cong (1 - R^2/2)\left[\frac{\Delta\theta}{2\pi} - \frac{1}{\pi}\int_{\theta_1}^{\theta_2} g(u)\, d\theta\right] + O(R^3),$$

where with $uF(u) \equiv g(u)$,

$$g(u) = \frac{R}{\sqrt{2}}\cos\theta\,(1 + \frac{R^2}{2}\cos^2\theta)\frac{\sqrt{\pi}}{2}\left[1 - \frac{2}{\sqrt{\pi}}\frac{R}{\sqrt{2}}\cos\theta + O(R^3)\right]$$

$$= \frac{R}{\sqrt{2}}\cos\theta\left[\frac{\sqrt{\pi}}{2} - \frac{R}{\sqrt{2}}\cos\theta + O(R^3)\right].$$

Carrying out the θ integration in (27), we obtain

$$(28) \qquad \boxed{P(A) \cong \frac{\Delta\theta}{2\pi} - \frac{1}{2\sqrt{\pi}}\left(\frac{R}{\sqrt{2}}\sin\theta_2 - \frac{R}{\sqrt{2}}\sin\theta_1\right) + \frac{1}{4\pi}\left(\frac{R^2}{2}\sin 2\theta_2 - \frac{R^2}{2}\sin 2\theta_1\right) + O(R^3).}$$

Thus, when

$$(29) \qquad \frac{1}{2\sqrt{\pi}}\left|\frac{R}{\sqrt{2}}\sin\theta_2 - \frac{R}{\sqrt{2}}\sin\theta_1\right| \leqslant \frac{R}{\sqrt{2}}\frac{1}{\sqrt{\pi}} \leqslant \epsilon, \quad (\epsilon \equiv \delta/\sqrt{\pi}, \text{ see } (13)),$$

then

$$(30) \qquad P(A) \cong \Delta\theta/2\pi.$$

Extending the above analysis one can show that the R^3 term in (28) is given by

$$\frac{1}{2\sqrt{\pi}}\left(\frac{R}{\sqrt{2}}\right)^3 \sin^2\theta\cos\theta,$$

and upon integration yields

$$(31) \qquad E \equiv +\frac{1}{6\sqrt{\pi}}\left(\frac{R}{\sqrt{2}}\right)^3(\sin^3\theta_2 - \sin^3\theta_1).$$

Hence P(A) is approximated to within ϵ by (28), without the $O(R^3)$ term, when

$$(32) \qquad |E| \leqslant \left(\frac{R}{\sqrt{2}}\right)^3\frac{1}{3\sqrt{\pi}} \leqslant \epsilon.$$

In the other circumstance, when R is sufficiently large, a parameter $\overline{R}$ can be determined, depending on ϵ, such that if

$$(33) \qquad R \geqslant \overline{R} \quad (\text{or } R^2/2 \geqslant \overline{R}^2/2),$$

then $P[A(R, \theta_1, \theta_2] < \epsilon$ for $|\theta_1|, |\theta_2| \leq \pi/2$. So in this case that part of the computation for P(A) which uses (19) can be omitted, but one erfc function is still required for each $|\theta_i| \geq \pi/2$ ($i = 1, 2$) (see (20)).

We note from (13) and the fact that $P[A(R, -\pi/2, \theta)]$ is an increasing function of θ, that for $R \geq \overline{R}$

$$(34) \qquad P[A(R, \theta_1, \theta_2)] \leq P\left[A(\overline{R}, -\frac{\pi}{2}, \frac{\pi}{2})\right] = \frac{1}{2}\,\mathrm{erfc}\,(\overline{R}/\sqrt{2}), \ |\theta_1|, |\theta_2| \leq \pi/2, \ \theta_1 \leq \theta_2.$$

Consequently, we choose $\overline{R}$ such that

$$(35) \qquad \frac{1}{2}\mathrm{erfc}\,(\overline{R}/\sqrt{2}) = \epsilon = \delta/\sqrt{\pi}\ ,$$

and observe that C from (14) and $\overline{R}/\sqrt{2}$ are the same for a given ϵ. Geometrically it means that the region to the right of the vertical line $x = \overline{R}/\sqrt{2}$ does not contribute to P(A) to within the specified accuracy.

3. USE OF P(A) TO COMPUTE P(H)

In this section we show how, using probabilities over angular regions, the probability, P(H), over a convex polygon H is obtained. In [4] they propose using probabilities over triangles and quadrilaterals to obtain the same. Our procedure, however, is, in general more efficient. As shown below, we require only N angular regions for an N-sided convex polygon, whereas they need at least $3(N - 2)$ regions if H is decomposed into triangles. If, for N even, H is decomposed into quadrilaterals, or quadrilaterals plus one triangle for N odd, then one needs $2N - 4$ or $2N - 3$ angular regions, respectively. We remind the reader that our ultimate purpose in developing a program for computing P(H) is to use it as a subroutine to evaluate the probability over an arbitrary polygon. As stated earlier this has been done and will be discussed together with a computing program in a later report.

Let $H(N, t_1, \ldots, t_N)$ denote a convex polygon of N sides with vertices at coordinate points $t_1, \ldots, t_N$, where $t_k \equiv (x_k, y_k)$ and the points $\{t_i\}$ are given in counterclockwise order; i.e., so that the area of H is on the left as one traverses the boundary continuously.* Then

$$(36) \qquad P(H) = P(A_1) - \sum_{i=2}^{N-1} P(A_i) + P(A_N),$$

where using Figure 3 with $N = 6$, A_1 is the angular region determined by any interior angle of H with its vertex assigned as t_1, A_i, $i = 2, \ldots, N - 1$, are angular regions determined by the exterior angles of H at vertices $t_2, \ldots, t_{N-1}$, respectively, as shown in Figure 3 and A_N is the angular region obtained from the vertical angle of the interior angle of H at t_N. It is easy to argue the validity of (36) by noting, e.g. in Figure 3 ($N = 6$), that the probabilities over the disjoint shaded regions E_i, $i = 2, 3, \ldots, N - 1$, excessively diminish the result for P(H) by an amount exactly compensated for by the addition of $P(A_N)$. A formal proof of (36) is not given in this report.

*Note that (2) maps convex polygons into convex polygons.

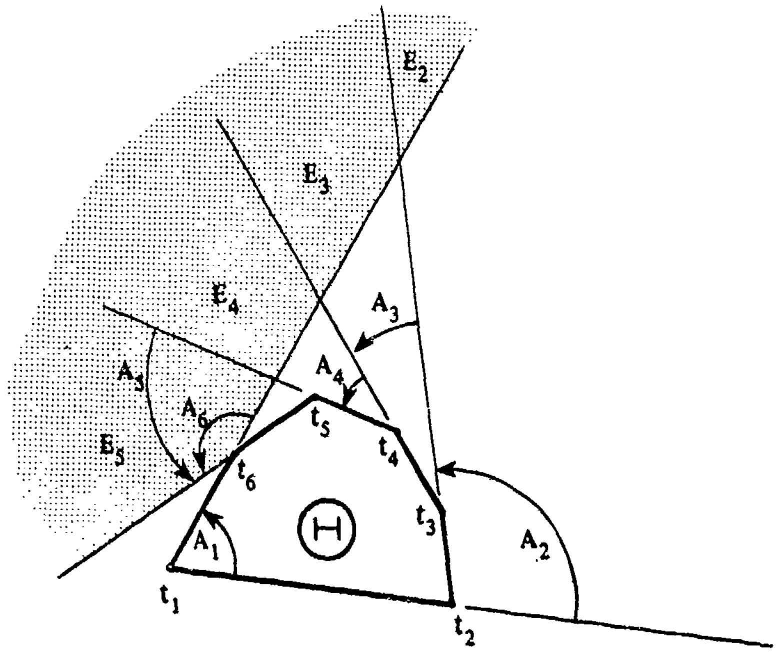


Figure 3. Convex Polygon, N = 6, Showing Angular Regions

In keeping with our efforts to maintain an efficient program, as described in the previous section, we make use of the following result. Let $\theta(i)$ denote the quantity $\theta_2 - \theta_1$, which appears in (9), for the i^{th} angular region probability, $P(A_i)$. Then, since the interior angles of an N-convex polygon add up to $(N-2)\pi$ radians, we have

$$\theta(N) = -\theta(1) + \sum_{i=2}^{N-1} \theta(i). \tag{37}$$

Thus our program generally will require only N-1 calls to the $\tan^{-1}$ routine instead of N by using (37). The accumulation of the right-hand side is denoted in the flow charts as ϕ; e.g., see boxes [2,8,11,24,33] in flow chart II, page 11.

4. COMPUTER PROGRAM FOR P(H) (AND P(A))

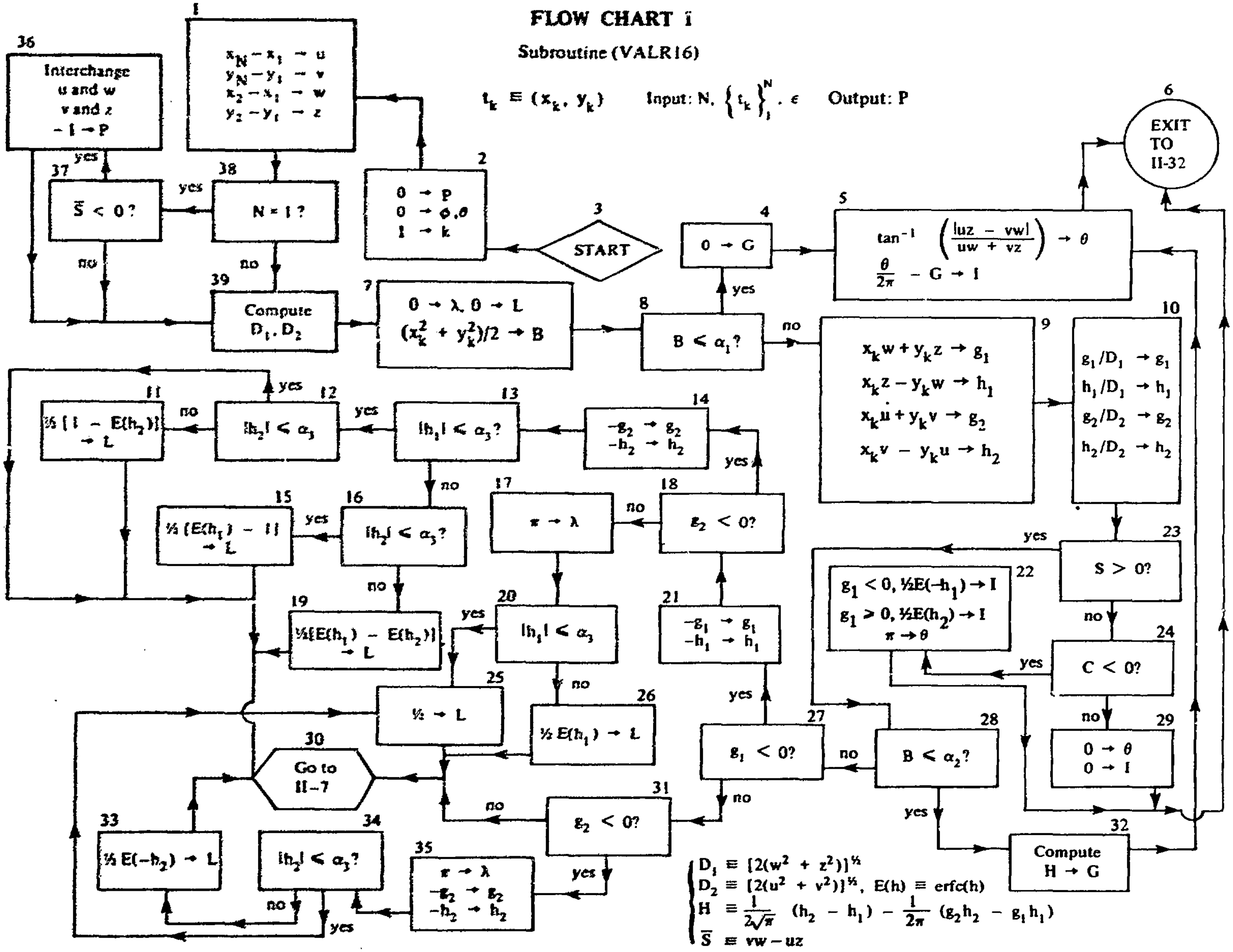
In this section we discuss the Fortran IV program for the evaluation of P(H) or P(A), the normal probability distribution over a convex polygon H of N sides, or an angular region A, respectively. Two flow charts I, II are given on the next two pages which show the flow and major steps of the program. It will be helpful to refer to these charts during the discussion. A location in the flow charts will be identified by chart number and box number, e.g. [I,10] refers to chart I box 10.

One input to the program is the sequence $\{t_k\}_1^N$, where t_k denotes the (x,y) coordinates of the k^{th} vertex of H with the vertices ordered in a counterclockwise direction. The value of N is

FLOW CHART i

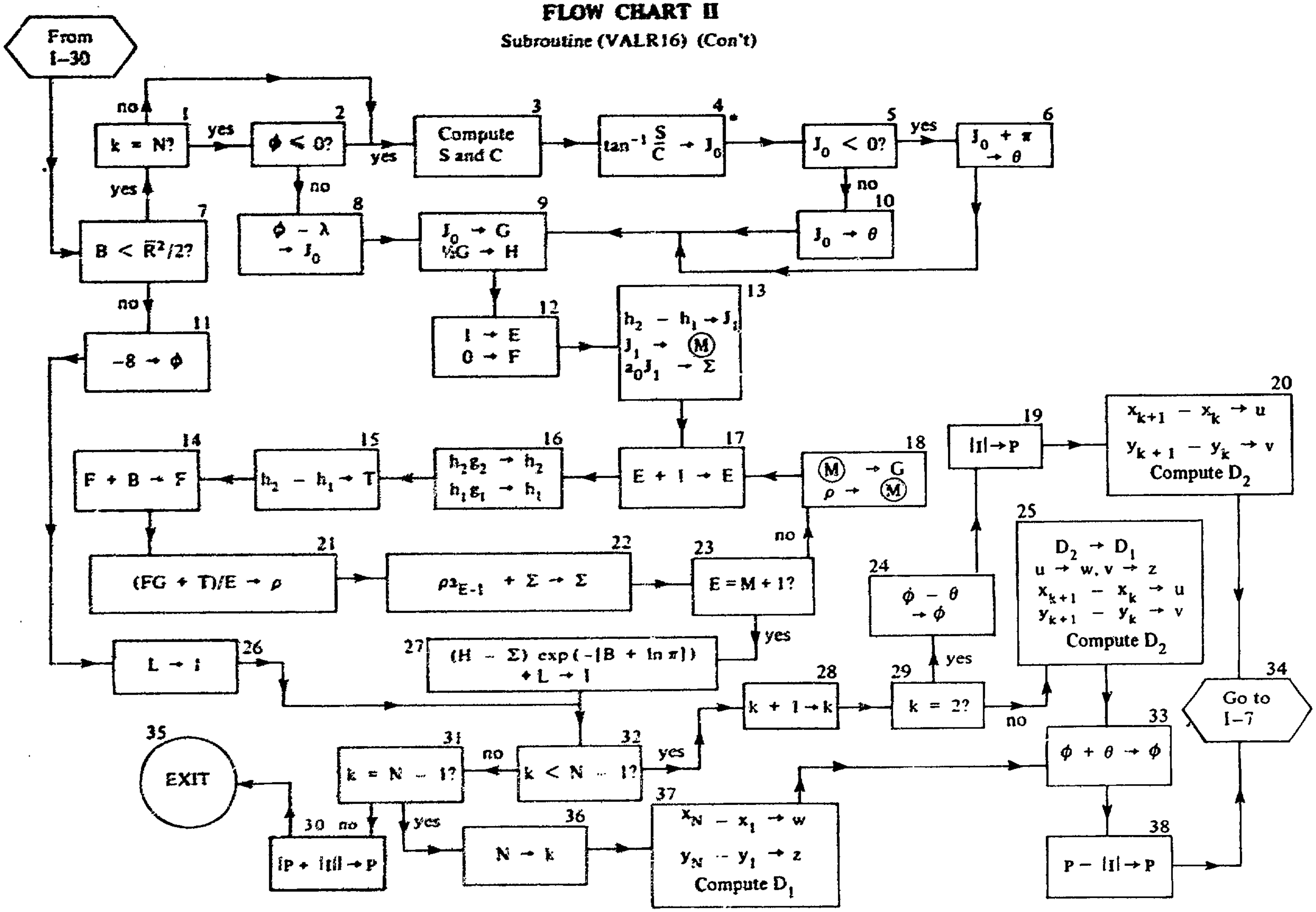
Subroutine (VALR16)

$t_k \equiv (x_k, y_k)$ Input: N, $\{t_k\}_1^N$, ϵ Output: P



FLOW CHART II

Subroutine (VALR16) (Con't)



*Use 4-quadrant $\tan^{-1}$ routine $(-\pi, \pi]$

$S \equiv g_1 h_2 - g_2 h_1$, $C \equiv g_1 g_2 + h_1 h_2$ I–7 → chart I, box 7.

specified with N set to one if P(A) is desired. In this case 3 points are required, as in the case of a triangle where N = 3, in counterclockwise order, i.e., so that the region A is to the left as one tranverses the boundary lines with the only vertex at t_1. A parameter is set specifying whether 3, 6, or 9 decimal digits are desired in the output P(H) or P(A).* The associated values of various parameters are given in Appendix A, namely a_i, $\alpha_1, \alpha_2, \alpha_3, \overline{R}/\sqrt{2}$. Also listed in that Appendix are values of these parameters for P(H) or P(A) computable to twelve decimal digits. These however are not incorporated into the program but could be with no difficulty if desired.

It is imperative for the program to operate properly that the t_k be given in counterclockwise order; i.e., with the area on the left as one travels along the boundary of H or A. Two typical examples are shown in Figure 4, where P is wanted over the shaded or hatched regions.

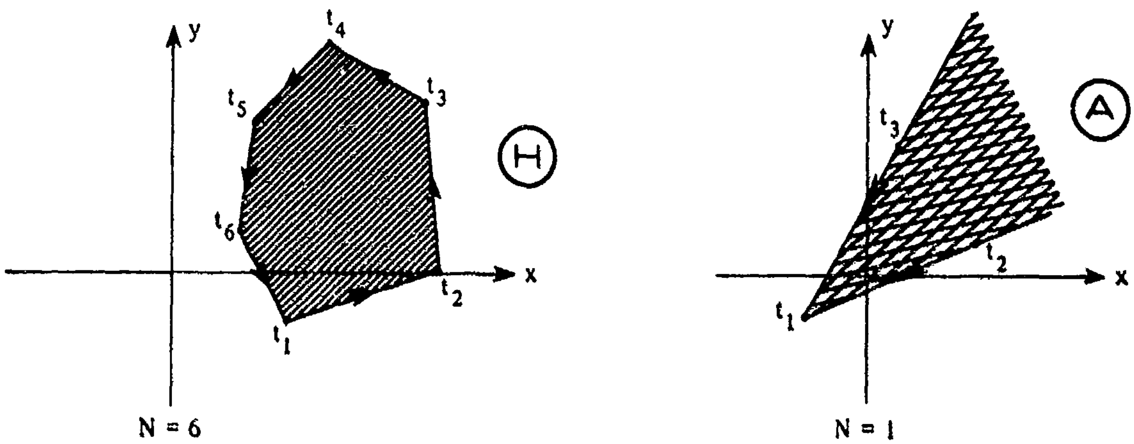


Figure 4. Typical Regions for H and A

Point t_1 for H can be taken initially as any vertex point, however when this program is used for arbitrary polygons, it will cycle the points and renumber them so that the new t_1 is the point with the property

(38) $$t_1 = \left[(x_j, y_j) \middle| , y_j \le y_k \text{ with } x_j < x_k \text{ if } y_j = y_k, k = 1, \ldots, N\right],$$

(This feature is not shown on the flow charts.).

For A, t_1 must be specified as the vertex point, as shown in Figure 4 above.

In order to evaluate P(H), N angular regions $\{A_k\}$ are treated, one at each vertex of H, and their probabilities $\{P(A_k)\}$ combined appropriately as explained in Section 3 (see (36)). For a particular $A_k = A(R, \theta_1, \theta_2)$, the inequality below

*We make note of the fact here that the specified number of correct decimal digits in computing P(H) may not be achieved in the unlikely case that the errors associated with a majority of the angular regions have the same sign and thus add to a total error of as much as $N\delta$.

(39) $$B = R^2/2 \leqslant \alpha_1 = \pi\epsilon^2$$

is tested where α_1 is taken from (29). If it holds then [I, 5] is used to evaluate P(A), which is then stored in I,

$$P(A) = (\theta_2 - \theta_1)/2\pi = \frac{1}{2\pi}\tan^{-1}\left(\frac{|uz - vw|}{uw + vz}\right),$$

where the $\tan^{-1}$ is obtained from a four quadrant subroutine which gives its output in $(-\pi,\pi]$. The quantities u, v, w, z are initially defined in [I, 1] and subsequently in [II, 20, 25, 37] depending on which angular region is involved. The angles θ_2 and θ_1 are measured in radians and are as shown in Figures 1 or 2, page 3, with $\Delta\theta$ always positive from θ_1 counterclockwise to θ_2.

If the inequality in (39) is not true, then a rotation of axes is carried out, [I, 9], as indicated in Figures 1 and 2. Quantities g_1/D_1, h_1/D_1, g_2/D_2, h_2/D_2 are computed, [I, 10], where

(40) $$\begin{cases} g_1/D_1 = \dfrac{R}{\sqrt{2}}\cos\theta_1 \rightarrow g_1, & h_1/D_1 = \dfrac{R}{\sqrt{2}}\sin\theta_1 \rightarrow h_1, \\ g_2/D_2 = \dfrac{R}{\sqrt{2}}\cos\theta_2 \rightarrow g_2, & h_2/D_2 = \dfrac{R}{\sqrt{2}}\sin\theta_2 \rightarrow h_2, \end{cases}$$

with

(41) $$D_1 \equiv [2(w^2 + z^2)]^{1/2}, \quad D_2 = [2(u^2 + v^2)]^{1/2}.$$

We have for the first of (40)

(42) $$\frac{R}{\sqrt{2}}\cos\theta_1 = \left(\frac{x_k^2 + y_k^2}{2}\right)^{1/2} \frac{x_k w + y_k z}{[(x_k w + y_k z)^2 + (x_k z - y_k w)^2]^{1/2}}$$

$$= (x_k w + y_k z)/[2(w^2 + z^2)]^{1/2} = g_1/D_1.$$

The other relations in (40) are found in the same way. The location denoted in the charts as P contains the output P(A) if N = 1, or P(H) if N $\geqslant$ 3. Location ϕ contains 0 if N = 1. If N $\geqslant$ 3 and (33) does not hold, then ϕ contains θ(N), (See (37)), at exit.

In [I, 28], the inequality

(43) $$B \leqslant \alpha_2 \equiv (9\pi\epsilon^2)^{1/3}, \quad (B = R^2/2),$$

is tested. If it holds then $P(A_k)$ is given by (28), [I, 5, 32], where α_2 is taken from (32).

In general, the program distinguishes 12 different types of angular regions which are exhaustive and are characterized by the signs of the numbers g_1, g_2, h_1, h_2 as computed in [I, 10]. Examples of each of the 12 regions are shown below in Figure 5 with the terms used to evaluate P,

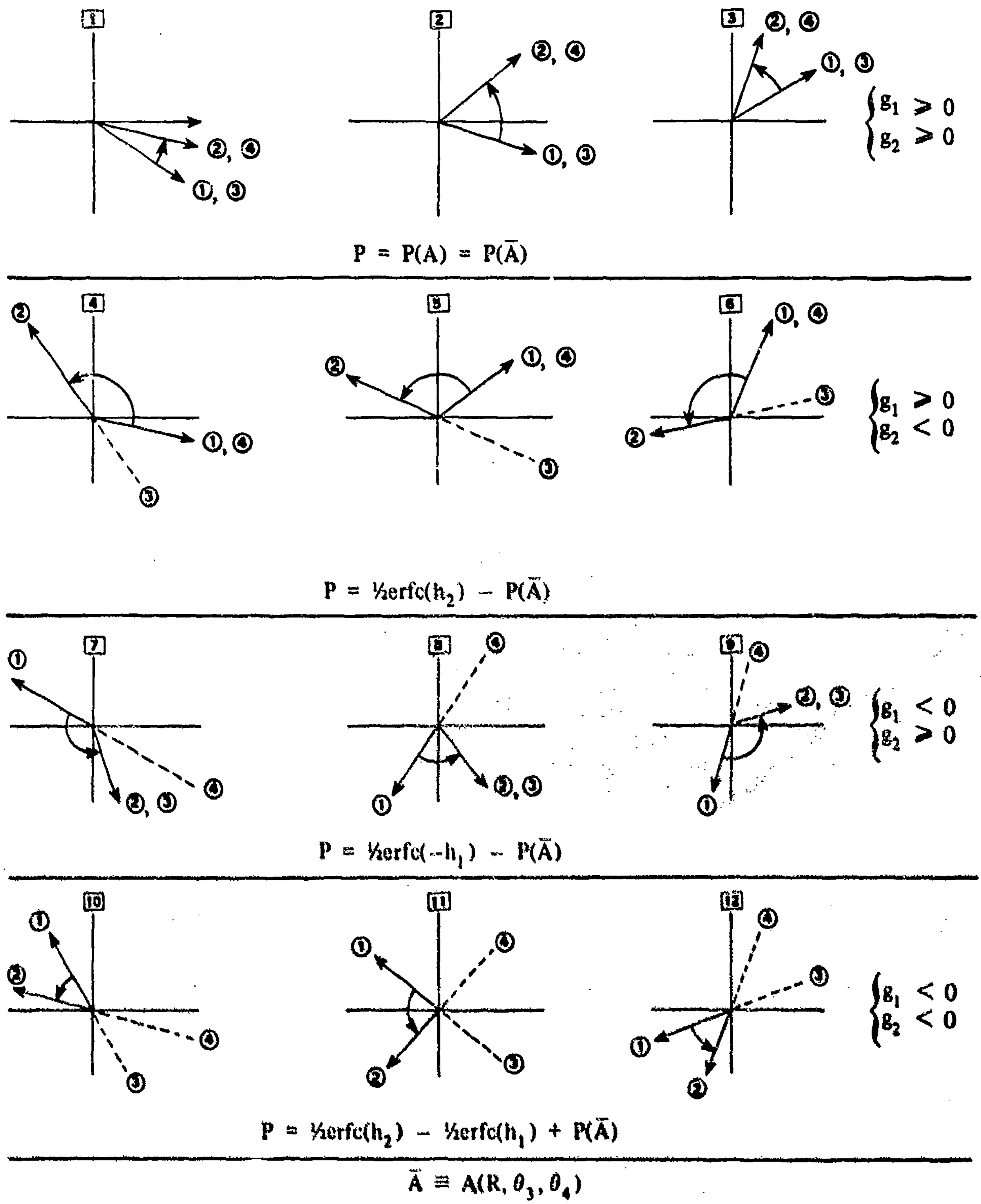


Figure 5. Various Cases for A

Note h_1, g_1, h_2, g_2 here refer to [1, 10].

(P = P(A)). It is assumed the rotation, [I, 9] has been carried out as described above, so that the vertex of A is on the positive x-axis (not at the origin). The angle between the directed lines labelled (1) and (2) is always measured from (1) to (2) in the counterclockwise direction and it is non-negative and always no larger than π ($\sin(\theta_2 - \theta_1) \geq 0$), since we are dealing with convex polygons. We allow $\pi \leq \Delta\theta \leq 2\pi$ only if N = 1. In this case we evaluate P(E-A) $\equiv \bar{P}$, where E denotes the entire plane, and find P(A) = 1 $-$ $\bar{P}$. The boxes that apply for N = 1 only, showing the details just mentioned, are [I, 36, 37, 38]. In the situations shown in Figure 5, we denote the probability over the angular region between (1) and (2) by P, and note in the expressions for P below each diagram, that if $g_i < 0$ (i = 1, 2) then erfc (h_i) is required where $|h_i|$ is the normal distance from line (i) to the origin (See (20)). The lines (3) and (4) shown in the diagrams bound the angular region denoted by $\bar{A}$. In diagrams $\boxed{1}$, $\boxed{2}$, $\boxed{3}$, A and $\bar{A}$ coincide.

If $|h|$ ($|h_1|$ or $|h_2|$) is sufficiently small, erfc (h) can be replaced by one and a call to the erfc routine avoided. This feature appears in the program through [I,12,13,16,20,34] where the inequality

(44) $$|h| \leq \alpha_3$$

is tested. We have if (44) holds

(45) $$\frac{1}{2}\left|\mathrm{erfc}(h) - 1\right| \leq \frac{1}{2}\frac{2}{\sqrt{\pi}}\ |h| \leq \frac{1}{\sqrt{\pi}}\ \alpha_3 = \epsilon/2\ , \ (\epsilon \text{ defined in (13)})$$

so that α_3 is taken as

(46) $$\alpha_3 = \sqrt{\pi}\ \epsilon/2.$$

Box [II,7] is used to check if R is sufficiently large for the computation of (19) to be by-passed. The choice for $\bar{R}/\sqrt{2}$, which has already been discussed on page 8, is made so that with $R \geq \bar{R}$,

$$P[A(R,-\frac{\pi}{2},\frac{\pi}{2})] \leq \epsilon.$$

The program for P($\bar{A}$) by (19) is displayed in [II,12–23] and [II,27], with [II,4] showing the computation for J_0 which denotes $\pm$ the angle of $\bar{A}$ where the sign agrees with the sign preceding P($\bar{A}$) in the relations given for P in the diagrams of Figure 5 (See (18), also).

The program is designed to recognize and avoid a subtle situation that can occur due to round-off error that leads to a catastrophic erroneous result. As an example suppose we are dealing with a polygon where one of the exterior angular regions, say A_k, k $\neq$ 1, N, as shown by the solid lines in Figure 6, subtends an angle θ of nearly π radians with sides of A at large perpendicular distances from the origin, so that P(A) ~ 1.

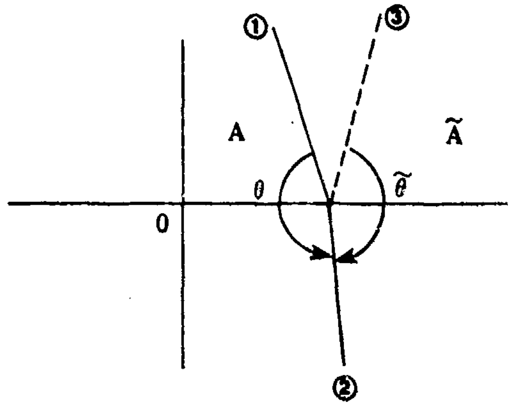


Figure 6. Shows A Singular Case Situation

Suppose, however, by rounding error line ① is actually given by the computer as line ③ so that the angular region $\tilde{A}$ subtends an angle $\tilde{\theta}$, near $(-\pi)$ radians. Thus the program in this case would yield a value $[-P(\tilde{A})]$; i.e., a small value. Moreover it would be negative since $\tilde{\theta}$ is measured from ③ to ② which is clockwise rather than counterclockwise. This singular case situation and others that can occur are handled in the program through boxes [I,22], [I,23], [I,24], and [I,29]. When $N \geqslant 3$, a singular case occurs for the k^{th} angular region of H ($\Delta\theta \notin [0, \pi]$) when

(47) $$S \equiv \frac{R}{\sqrt{2}} \sin(\theta_2 - \theta_1) \leqslant 0.$$

If this is the case, a second inequality is tested, namely,

(48) $$C \equiv \frac{R}{\sqrt{2}} \cos(\theta_2 - \theta_1) < 0.$$

If (47) and (48) are satisfied, as in Figure 6, we set $P(A_k) = ½ \operatorname{erfc}(t)$, where $t = -h_1$ if $g_1 < 0$, or $t = h_2$ if $g_1 \geqslant 0$. If (47) holds and (48) does not, then we set $P(A_k) = 0$ since $|\Delta\theta| \sim 10^{-14}$. A singular situation that cannot be resolved occurs in the unlikely case that (47) holds, (48) does not, $R \gg \hat{R}$, and g_1, g_2 are negative. When all of these conditions are true, A_k may contain the origin so that for sufficiently large R ($> 10^4$), $P(A_k)$ is not close to zero. However $\Delta\theta(\sim 10^{-14})$ should always be in $[0, \pi]$ for a convex polygon, but it is not since (47) holds. Hence we cannot find, within the single precision capabilities of the CDC6700, the value of $P(A_k)$, because the value of $\Delta\theta$ cannot be resolved.

In the next section, we discuss the Gideon-Gurland method (G & G) for evaluating P(A). In their report and published paper however they do not consider the programming aspects of their method, which must also deal with the singular case problem just mentioned.

Extensive checking of our program was carried out. Comparisons of results were made with a program of the G & G method that we developed. Also comparisons were made with two other independent programs for computing P(H) for the special case of triangles. These programs also allowed independent checking for convex polygons other than triangles, since a convex polygon can always be decomposed into a set of triangles.

Our computing program is designated as VALR16. In Appendix C a Fortran IV listing is given of a test program which generates coordinates representing the vertices of a set of triangles such that all phases of the VALR16 program are tested by evaluating the probability over these triangles. Some numerical results are also given there.

5. COMPARISON WITH GIDEON AND GURLAND METHOD

The work of Gideon and Gurland (G & G) [4], [5] gives a set of relations by which P(A) can be evaluated. Their unique procedure is very efficient and though limited to 5 decimal digit accuracy appears to be one of the best of the methods we reviewed in the literature [1, 2, 3, 6, 7]. Essentially they assume the angular region A has been rotated as shown in Figure 2 such that if $|\theta_i| \leqslant \pi/4$, $(i = 1, 2)$ then

$$(49)\quad P[A(R, 0, \theta_i)] = \frac{1}{4}\operatorname{erfc}(R/\sqrt{2})\,[b_0 + b_1 R + b_2 R^2)\theta_i + (b_3 R + b_4 R^2)\theta_i^3 + (b_5 R + b_6 R^2)\theta_i^5]^{*}.$$

The coefficients b_j were determined by least squares for each of 15 subintervals in R, [0, 1/2], [1/2, .75], . . . , [j/4, (j + 1)/4], . . . , [3.75, 4]. In order to evaluate P(A), they need to use (49) twice, with the same value of R, once for θ_1 and once for θ_2. Because the use of (49) is constrained to $|\theta| \leqslant \pi/4$, G & G require in addition to (20) the relation

$$(50)\quad P[A(R, 0, \theta)] = \frac{1}{4}\operatorname{erfc}\left(\frac{R}{\sqrt{2}}\sin\theta\right)\operatorname{erfc}\left(\frac{R}{\sqrt{2}}\cos\theta\right) - P[A(R, 0, \frac{\pi}{2} - \theta)]\,,\quad \frac{\pi}{4} < \theta \leqslant \pi/2\,.$$

We have programmed the (G & G) method and found the average computing time per angular region to be about 20% longer than ours at the 6 decimal digits accuracy level. We estimate a 25% to 30% difference if we modified our method for 5 instead of 6 decimal digit accuracy.

Although it takes less time to evaluate the righthand side of (49) twice, without erfc $(R/\sqrt{2})$, than it does to evaluate the recursive procedure given by (19), our method has significantly less calls to the various special function routines, except for the exponential. In particular since the minimax approximation for erfc (u)/z(u) holds for $u \geqslant 0$. ($|\theta| \leqslant \pi/2$), we do not need (50). Moreover, in evaluating the number of erfc functions required by G & G it is recalled that we need an erfc function when $\pi/2 < \theta < 3\pi/4$ or when $3\pi/4 \leqslant \theta < \pi$. In the second case they also need one erfc, however for the first inequality they need two. Consequently, for each A, counting the erfc function needed in (49) once and using (20) and (50) it is easy to show by enumeration of cases (for example, in $\boxed{3}$ of Figure 5, page 14, they could need 5 while we would need none) that their method takes on the average 3½ times as many erfc functions as ours. In addition, they treat θ_1 and θ_2 separately while we treat the difference $\theta_2 - \theta_1$ (except for the functions g_1, h_1, g_2, h_2 which are expressed as algebraic functions of the coordinates of A). Thus, they need two separate calls to the arctangent routine for P(A) whereas we require one, and for H a triangle they need 5 arctangents (taking advantage of (37)) while we need only 2. They also need R which requires a square root while we need an exponential. The average number of calls to special functions for a convex polygon of N sides is summarized in Table 1.

*We note a serious omission in [5] where it is not explicitly stated that (49) only holds for $|\theta| \leqslant \pi/4$.

Table 1. Average No. of Calls to Special Function Routines for N-Angular Regions

	Us	G & G
erfc	N	3.5N
$\tan^{-1}$	N-1	2N-1
square root	N+1	2N+1
exponential	N	0

We also note that in [4] they advocate treating N sided polygons by decomposing them into sets of quadrilaterals and triangles. In the case of N-convex polygons, this would mean treating 2N-3 angular regions for N odd, and 2N-4 for N even, whereas we would only require N angular regions as explained in Section 3. Also in the case of an arbitrary polygon it will be more efficient in general to decompose it into as few convex polygons as possible rather than, as G & G propose, into triangles and quadritalerals.

6. COMMENTS ON DREZNER'S METHOD

In a recent paper by Drezner, [3], a method is given for computing the bivariate normal integral, $\Phi(m, k, \rho)$; i.e.,

$$\Phi(m,k,\rho) = (2\pi\sqrt{1-\rho^2})^{-1}\int_{-\infty}^{m}\int_{-\infty}^{k} \exp[-\frac{w^2 - 2\rho wz + z^2}{2(1-\rho^2)}]\, dwdz. \tag{51}$$

By letting

$$\begin{cases} u = (m - w)/\sqrt{2(1-\rho^2)}\ , \ v = (k - z)/\sqrt{2(1-\rho^2)}\ , \\ M = m/\sqrt{2(1-\rho^2)}\ , \ K = k/\sqrt{2(1-\rho^2)}\ , \end{cases} \tag{52}$$

he obtains

$$\Phi(m,k,\rho) = \frac{\sqrt{1-\rho^2}}{\pi}\int_0^{\infty}\int_0^{\infty} e^{-u^2}e^{-v^2}\, f(u,v)dudv, \tag{53}$$

where

$$f(u,v) = \exp[M(2u - M) + K(2v - K) + 2\rho(u - M)(v - K)]. \tag{54}$$

Drezner then uses Gaussian integration, when $m < 0,\ k < 0,\ \rho < 0$, so that

$$\Phi(m,k,\rho) \cong \frac{\sqrt{1-\rho^2}}{\pi}\sum_{j=1}^{J}\sum_{i=1}^{J} A_i A_j f(u_i, v_j). \tag{55}$$

where the weights A_i and abscissae u_i (or v_j) are given in [8]. He makes the significant observation that if

$$(56) \qquad m \leqslant 0,\ k \leqslant 0,\ \rho \leqslant 0,$$

then $f(u,v) \leqslant 1$ and he can use (55) directly to evaluate Φ within a given error for relatively small values of J. For example, the maximum observed error for J = 5, is reported to be 5.5(−7). He also takes advantage of the fact that if the argument of the exponent in (54) is sufficiently less than zero, f can be replaced by zero. For J = 5, his cutoff value is stated as −12.

In cases where one or more of the three inequalities in (56) does not hold, one or two erfc functions must also be computed. In case $mk\rho>0$, then two sums such as appear in (55) are needed in addition to possibly one or two erfc functions. The necessary relations are all given in [3]. Two typographical errors are noted there. In (10) $1/\sqrt{2\pi}$ should replace $1/2\pi$ and in (12)

$$(57) \qquad \delta_{mk} = \frac{1 - \mathrm{Sgn}(m)\,\mathrm{Sgn}(k)}{4},$$

where the minus sign replaces an incorrect plus sign.

Clearly (1) with $\mu_i = 0, \sigma_i = 1, i = 1, 2$, reduces to (51) where the angular region A has a right angle at (m,k). Applying the transformation in (2) which reduces to

$$(58) \qquad x = (w - \rho z)/\sqrt{1 - \rho^2}\ ,\ y = z,$$

the 90° angular region A in the w − z plane is transformed into an angular region A in the x − y plane with vertex at (x_0, y_0), where

$$x_0 = (m - \rho k)/\sqrt{1 - \rho^2}\ ,\ y_0 = k,$$

with a subtended angle θ_0.

$$\theta_0 = \tan^{-1}[\sqrt{1 - \rho^2}/(-\rho)]\ ,$$

where θ_0 is measured counterclockwise from the negative side of the line y = k. The angular region A therefore is always below the line y = k.

In particular, given a set of values (m,k,ρ) there exists a corresponding angular region in the x − y plane specified by R, θ_1, θ_2 (See Figures 1 and 2). The connection between these sets of variables can be shown to be

$$
(59)\quad \begin{cases} R = [(m^2 - 2\rho mk + k^2)/(1 - \rho^2)]^{1/2} \\ \theta_1 = \tan^{-1}[k\sqrt{1-\rho^2}/(\rho k - m)], \quad \theta_2 = \tan^{-1}[-m\sqrt{1-\rho^2}/(\rho m - k)], \\ g_1 = (\rho k - m)/\sqrt{2(1-\rho^2)}, \quad g_2 = (\rho m - k)/\sqrt{2(1-\rho^2)} \\ h_1 = k/\sqrt{2}, \quad h_2 = -m/\sqrt{2}, \end{cases}
$$

or

$$
(60)\quad \begin{cases} m = -\sqrt{2}\,\dfrac{R}{\sqrt{2}}\sin\theta_2 = -\sqrt{2}\,h_2, \; k = \sqrt{2}\,\dfrac{R}{\sqrt{2}}\sin\theta_1 = \sqrt{2}\,h_1 \quad \text{(see (40))}, \\ \rho = -\dfrac{2}{R^2}(g_1 g_2 + h_1 h_2) = -\cos(\theta_2 - \theta_1), \\ \sqrt{1-\rho^2} = \sin(\theta_2 - \theta_1) \geq 0. \end{cases}
$$

We have programmed the Drezner procedure and compared it to our method. A Fortran IV listing is given in Appendix B. We did not expect it to be as efficient because of the large number of exponentials required. For $J = 5$, 25 exponentials are required when $mk\rho \leq 0$, and 50 are needed when $mk\rho > 0$. However neither method suffers in comparison to the other in computing additional erfc functions (or equivalently normal probability integrals in one dimension) since it can be shown both require the same number (none, one, or two) in any particular case.

Timing runs for the two programs showed that the Drezner method is 4 times slower on the average than ours for 6 decimal digit accuracy and 8 times as slow for 9 decimal digits of accuracy.

We also note that Drezner's procedure is incomplete for programming because he does not state how to treat the cases $\rho = 1$, $\rho = -1$. These values can occur through numerical rounding and must be dealt with before a working program can be obtained. This problem is resolved by noting that if $\rho = 1 - \epsilon$, $\epsilon > 0$, then

$$
(61)\qquad \lim_{\epsilon \to 0} \Phi(m,k,1-\epsilon) = \tfrac{1}{2}\operatorname{erfc}(-T/\sqrt{2}),
$$

where $T \equiv$ minimum of m and k;

if $\rho = -1 + \epsilon$, $\epsilon > 0$, then

$$
(62)\qquad \lim_{\epsilon \to 0} \Phi(m,k,-1+\epsilon) = \begin{cases} \tfrac{1}{2}[\operatorname{erfc}(-k/\sqrt{2}) - \operatorname{erfc}(m/\sqrt{2})], & \text{if } k > -m \\ 0 \text{ otherwise.} \end{cases}
$$

These formulas are easily seen to be true by noting that the line $w = m$ transforms by (58) to the line, call it L,

(63) $$x\sqrt{1-\rho^2} + y\rho = m.$$

The line L clearly has the property that, whether m is positive or negative, it is tangent to the circle (D)

(64) $$x^2 + y^2 = m^2.$$

The following additional facts are easily proved from (63) and will help the reader in following Figures 7 and 8 and similar situations: (1) The slope of line L, or dy/dx, is $+\sqrt{1-\rho^2}/(-\rho)$, and hence the slope has the opposite sign to that of ρ, so that ρ must be negative in Figure 7 and positive in Figure 8; (2) The x-intercept of L is $m/\sqrt{1-\rho^2}$, so that m is positive in Figure 7 and negative in Figure 8, m having the same sign as the x-intercept of L.

Hence if $\rho = -1 + \epsilon$, $\epsilon > 0$ and small, $k > -m$, we have the situation shown in Figure 7. Now as $\epsilon \to 0$, the point (C) approaches $+\infty$ along $y = k$, and L approaches tangency

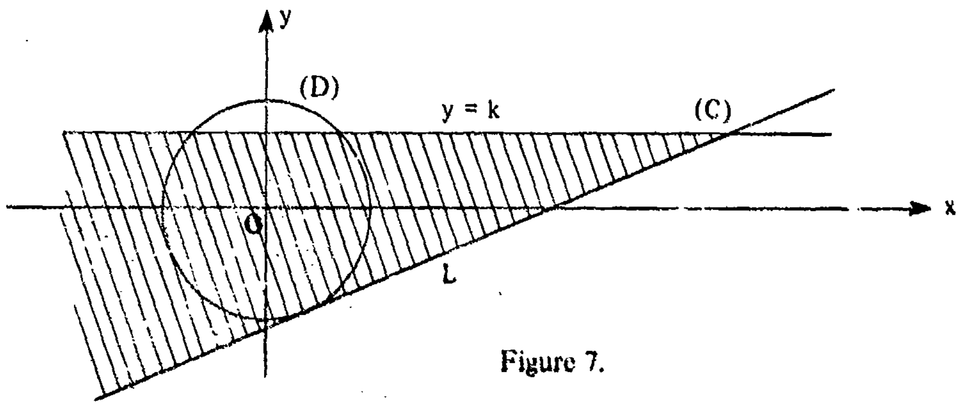


Figure 7.

to the circle (D) at $(0,-m)$ (but note that $0 < k < m$ in this case since the x-intercept of line L is positive). Consequently, in the limit as $\epsilon \to 0$, $\Phi(m,k,-1)$ is given by (62). Similar heuristic arguments, which can be made rigorous, can be given for any other situation with $\rho \to 1$ as well as $\rho \to -1$. For example, with $\rho = 1 - \epsilon$, $\epsilon > 0$, $m < k$ as in Figure 8.

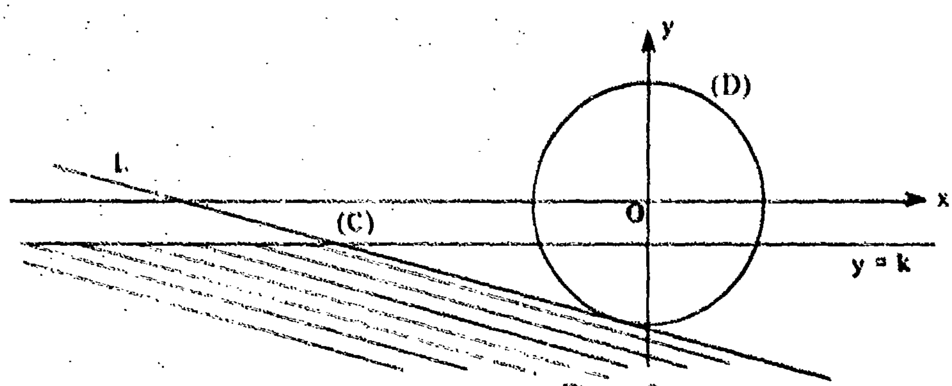


Figure 8.

In this figure, $\rho = 1 - \epsilon$, $m < k < 0$ (x-intercept negative), and we observe that as $\epsilon \to 0$, the shaded area in Figure 8 approaches the region below and including the line $y = m(< k < 0)$ as required by (61), or the limit is $(1/2)\ \mathrm{erfc}(-m/\sqrt{2})$.

7. SOME NUMERICAL RESULTS

In this section we give the numerical results for $P(H_1)$ and $P(H_2)$, using our program VALR-16, where H_1 is a 6 sided convex polygon containing the origin and H_2 is an 8 sided convex polygon not containing the origin. The x, y columns of Table 2 below contain the x, y coordinates of the vertices; the three columns that follow list the values of $P(A_k)$ for each angular region (See Figure 3) for $\epsilon_1 \cong 2.5(-4)$, $\epsilon_2 \cong 2.6(-7)$, $\epsilon_3 \cong 2.9(-10)$. The last row headed P(H) contains the value of P(H) for $\epsilon_1, \epsilon_2, \epsilon_3$. All the P values have been truncated from 14 digit CDC 6700 output.

Table 2

k	x	y	$P(A_k), \epsilon_1$	$P(A_k), \epsilon_2$	$P(A_k), \epsilon_3$
1	0.5	−2.0	.911227	.911064879	.91106477067
2	2.0	0.0	.046858	.046998988	.04699911886
3	0.5	2.0	.052500	.052666886	.05266699792
4	−0.5	1.5	.059487	.059482771	.05948276788
5	−1.5	0.0	.042640	.042515227	.04251511748
6	−1.0	−1.5	.017780	.017747368	.01774728061
	$P(H) = P(H_1) \to$		.727521	.727148375	.72714804914
1	1.5	−1.5	.851151	.850975856	.85097578896
2	2.0	−0.75	.018552	.018726585	.01872664573
3	1.75	1.75	.038192	.038305815	.03830565474
4	1.25	1.75	.064039	.064042498	.06404250011
5	0.50	1.50	.486789	.486841686	.48684172637
6	0.25	0.25	.042253	.042256194	.04225618944
7	0.25	−0.25	.026270	.026273494	.02627348809
8	0.50	−1.25	.116365	.116775266	.11677520430
	$P(H) = P(H_2) \to$		.291919	.291304849	.29130478878

(See (36))

REFERENCES

1. D. E. Amos, "On Computation of the Bivariate Normal Distribution," *Math. Comp.*, v. 23, 1969, pp. 655-659.

2. D. J. Daley, "Computation of Bi- and tri-variate Normal Integrals," *Appl. Statist.*, v. 23, 1974, pp. 435-438.

3. Z. Drezner, "Computation of the Bivariate Normal Integral," *Math. Comp.*, v. 32, 1978, pp. 277-279.

4. R. A. Gideon and J. Gurland, "A Method of Obtaining the Bivariate Normal Probability over an Arbitrary Polygon," Dept. of Statistics, University of Wisconsin, Tech. Rep..#304, Madison, Wis., May 1972.

5. R. A. Gideon and J. Gurland, "A Polynomial Type Approximation for Bivariate Normal Variates," *SIAM J. Appl. Math.* v. 34, 1978, pp. 681-684.

6. D. B. Owen, "Tables for Computing Bivariate Normal Probabilities," *Ann. Math. Statist.*, v. 27, 1956, pp. 1075-1090.

7. R. R. Sowden and D. Secrest, "Computation of the Bivariate Normal Integral," *Appl. Statist.*, v. 18, 1969, pp. 169-180.

8. N. M. Steen, G. O. Byrne, and E. M. Gelbard, "Gaussian Quadrature Formulas," *Math. Comp.*, v. 23, 1969, pp. 661-671.

9. U.S. Dept. of Commerce, National Bureau of Standards, *NBS Handbook of Mathematical Functions*, Appl. Math. Series, v. 55, U.S. Govt. Printing Office, Washington, D.C., 1964.

APPENDIX A
PROGRAM PARAMETERS. CHEBYSHEV COEFFICIENTS FOR
$\mathrm{erfc}(x)/z(x),\quad x \geqslant 0$

In this appendix we list the pertinent constants that appear in the program for three levels of accuracy (3,6,9 decimal digits), and an additional set which is designed to yield 12 correct decimal digits for the probability over an angular region.

Acc.	δ	$C(\delta) = \frac{\overline{R}}{\sqrt{2}}$	ϵ	α_1	α_2	α_3	$\frac{1}{2}E(\overline{R}/\sqrt{2})$
Ⓐ	4.50(−4)	2.46	2.54(−4)	2.02(−7)	1.22(−2)	2.25(−4)	2.52(−4)
Ⓑ	4.56(−7)	3.5505	2.57(−7)	2.08(−13)	1.23(−4)	2.28(−7)	2.57(−7)
Ⓒ	5.21(−10)	4.382	2.94(−10)	2.72(−19)	1.35(−6)	2.61(−10)	2.88(−10)
Ⓓ	1.78(−13)	5.1092	1.00(−13)	3.17(−26)	6.58(−9)	8.90(−14)	2.50(−13)

$\epsilon = \delta/\sqrt{\pi}$	See page 6.	$\alpha_2 = (9\pi\epsilon^2)^{1/3}$	See pages 7, 13.
$C(\delta)$	See page 5.	$\alpha_3 = \delta/2$	See page 15.
$\overline{R}/\sqrt{2}$	See page 8.	$\frac{1}{2}E(\overline{R}/\sqrt{2}) \equiv \frac{1}{2}\mathrm{erfc}(\overline{R}/\sqrt{2})$ $= 2.5\epsilon$ (for Ⓓ)	See page 8.
$\alpha_1 = \pi\epsilon^2$	See pages 7, 13.		

The first column of the table labeled Acc. (for accuracy) lists Ⓐ , Ⓑ , Ⓒ , Ⓓ referring to 3, 6, 9, 12 decimal digits of accuracy, respectively, for the probability over an angular region. Pages are given above where the parameters are defined in the report.

The minimax coefficients, a_k, for approximating $\mathrm{erfc}(x)$ on $C(\delta)$ (See (12), (15)) are given below for four accuracy levels as indicated in the tables below by Ⓐ , Ⓑ , Ⓒ , Ⓓ . They were computed by a double precision minimax subroutine utilizing values of $\mathrm{erfc}(x)$ accurate to 18 significant digits on $(\frac{1}{2}, C]$ and $\mathrm{erf}(x)$ accurate to 25 digits on $[0,\frac{1}{2}]$.

For Ⓐ (Average time per angular region = 2.2×10^{-4} sec)

a_0 = .885777518572895D + 00 $\quad$ a_1 = −.981151952778050D + 00
a_2 = .759305502082485D + 00 $\quad$ a_3 = −.353644980686977D + 00
a_4 = .695232092435207D − 01

For Ⓑ (Average time per angular region = 4.6 x 10^{-4} sec)

a_0 = .886226470016632D + 00	a_1 = −.999950714561036D + 00
a_2 = .885348820003892D + 00	a_3 = −.660611239043357D + 00
a_4 = .421821197160099D + 00	a_5 = −.222898055667208D + 00
a_6 = .905057384150449D − 01	a_7 = −.254906111884287D − 01
a_8 = .430895168984138D − 02	a_9 = −.323377239693247D − 03

For Ⓒ (Average time per angular region = 6.5 x 10^{-4} sec)

a_0 = .886226924931465D + 00	a_1 = −.999999899776252D + 00
a_2 = .886223733186722D + 00	a_3 = −.666626670510907D + 00
a_4 = .442851899328568D + 00	a_5 = −.265638206366025D + 00
a_6 = .145060043403012D + 00	a_7 = −.714909837799889D − 01
a_8 = .309199295521210D − 01	a_9 = −.112323532148441D − 01
a_{10} = .324944543171185D − 02	a_{11} = −.704260243309096D − 03
a_{12} = .105787574480633D − 03	a_{13} = −.971864864160461D − 05
a_{14} = .408335517232165D − 06	

For Ⓓ (Average time per angular region = 9.1 x 10^{-4} sec)

a_0 = .886226925452593D + 00	a_1 = −.999999999948597D + 00
a_2 = .886226922786746D + 00	a_3 = −.666666611866661D + 00
a_4 = .443112868048919D + 00	a_5 = −.266662729091411D + 00
a_6 = .147687136321938D + 00	a_7 = −.761365855850292D − 01
a_8 = .368032849350860D − 01	a_9 = −.167195096888183D − 01
a_{10} = .710292625734052D − 02	a_{11} = −.278170932906224D − 02
a_{12} = .981112629090333D − 03	a_{13} = −.302588640752108D − 03
a_{14} = .789960968802448D − 04	a_{15} = −.168685181767046D − 04
a_{16} = .283646635409322D − 05	a_{17} = −.358314466908290D − 06
a_{18} = .317679497040006D − 07	a_{19} = −.175440651940430D − 08
a_{20} = .452534347337305D − 10	

Average time per angular region refers to the average computing time on the CDC-6700 to obtain P(A).

APPENDIX B.

LISTING OF DREZNER PROGRAM

This appendix contains a listing of the program for computing P(A) or P(H) by Drezner's procedure. It is designed to use J = 3, 5, 8 where J is defined by equation (5) in [3]. Thus, referring to Table 1 in [3], P will be computed correctly to at least 3, 6, or 9 digits, respectively by this program.

Call line to Z. Drezner Subroutine

CALL DREZNR (x, y, N, P_k, IOP) where

x is the input array of abscissas of the vertices of the polygon
y is the input array of ordinates of the vertices of the polygon
{ Vertices must be listed in <u>counterclockwise</u> order. See pp. 9, 10.

N is the number of sides of the polygon.*

P_k is the location of the answer as computed by the Drezner method

IOP = 1 specifies the Drezner subroutine to use a table of J = 3 weights in computing P_k (See (55)).

IOP = 2 specifies the Drezner routine to use a table of J = 5 weights in computing P_k.

IOP = 3 specifies a table of J = 8 weights in computing P_k.

*N = 1 for an angular region A with 3 points given in <u>counterclockwise</u> order with first point at vertex of A, (See page 12). Note $0 \leq \Delta\theta \leq 2\pi$ for N = 1, but $0 \leq \Delta\theta \leq \pi$ for $N \geq 3$.

```
      SUBROUTINE DREZNR( X,Y,N,ANS,IOP )
      DIMENSION X(1),Y(1),U(2),V(2),G(2),H(2)
      DIMENSION AM(51),AK(51),RHO(51)
      REAL L
      DATA RT2 / 1.4142 13562 373 /
      NM1=N-1
      K=1
      ANS=0.
      NBAR=N
      IF ( N.EQ.1 )  NBAR=3
      U(2)=X(NBAR)-X(1)
      V(2)=Y(NBAR)-Y(1)
      KP1=K+1
      U(1)=X(KP1)-X(K)
      V(1)=Y(KP1)-Y(K)
      IF ( N.GT.1 )  GO TO 3141
      SGN=1.
      SN=V(2)*U(1)-U(2)*V(1)
      IF ( SN.GE.0. )  GO TO 3141
      SGN=-1.
      T1=U(1)
      U(1)=U(2)
      U(2)=T1
      T1=V(2)
      V(2)=V(1)
      V(1)=T1
 3141 CONTINUE
      BGD1=SQRT( 2.*(U(1)*U(1)+V(1)*V(1)))
      BGD2=SQRT( 2.*(U(2)*U(2)+V(2)*V(2)))
 3151 CONTINUE
      L=0.
      B=.5*(X(K)*X(K)+Y(K)*Y(K))
      G(1)=U(1)*X(K)+V(1)*Y(K)
      G(2)=U(2)*X(K)+V(2)*Y(K)
      H(1)=-Y(K)*U(1)+X(K)*V(1)
      H(2)=-Y(K)*U(2)+X(K)*V(2)
      G(1)=G(1)/BGD1
      G(2)=G(2)/BGD2
      H(1)=H(1)/BGD1
      H(2)=H(2)/BGD2
      AM(K)=-RT2*H(2)
      AK(K)=RT2*H(1)
      IF ( B.NE.0. )  GO TO 3181
      RHO(K)=-(2.*(U(2)*U(1)+V(2)*V(1)))/(BGD1*BGD2)
      GO TO 3191
 3181 CONTINUE
```

```
                    RHO(K)=-(G(1)*G(2)+H(1)*H(2))/B
3191  CONTINUE
      IF ( K.LT.NM1 )  GO TO 3631
      IF ( K.EQ.NM1 )  GO TO 3661
      CALL FLAN ( AM(K),AK(K),RHO(K),ANS1,IOP )
      ANS=ANS+ANS1
      IF ( N.NE.1 )  RETURN
      IF ( SGN.EQ.1. )  RETURN
      ANS=1.-ANS
      RETURN
3631  CONTINUE
      K=K+1
      KP1=K+1
      IF ( K.NE.2 )  GO TO 3651
      KM1=K-1
      CALL FLAN ( AM(KM1),AK(KM1),RHO(KM1),ANS1,IOP )
      ANS=ANS1
      U(2)=X(KP1)-X(K)
      V(2)=Y(KP1)-Y(K)
      BGD2=SQRT( 2.*(U(2)*U(2)+V(2)*V(2)))
      GO TO 3151
3651  CONTINUE
      U(1)=U(2)
      V(1)=V(2)
      U(2)=X(KP1)-X(K)
      V(2)=Y(KP1)-Y(K)
      BGD1=BGD2
      BGD2=SQRT( 2.*(U(2)*U(2)+V(2)*V(2)))
      GO TO 3671
3661  CONTINUE
      K=N
      U(1)=X(N)-X(1)
      V(1)=Y(N)-Y(1)
      BGD1=SQRT( 2.*(U(1)*U(1)+V(1)*V(1)))
3671  CONTINUE
      KM1=K-1
      CALL FLAN ( AM(KM1),AK(KM1),RHO(KM1),ANS1,IOP )
      ANS=ANS-ANS1
      GO TO 3151
      END
```

```
      SUBROUTINE PLAN ( H,AK,R,ANS,IOP )
      DIMENSION EPS3(11)
      DATA ( EPS3(I),I=1,3 ) / 2.E-5,2.E-7,2.E-10 /
      DATA RT2/1.4142 13562 373 /
      OM=1.-EPS3(IOP)
      ANS=0.
      IF ( R.LE.-OM )  GO TO 3171
      IF ( (H*AK*R).GT.0. )  GO TO 3155
      IF ( H.GT.0. )  GO TO 2031
      IF ( AK.GT.0. )  GO TO 2021
      IF ( R.GT.0. )  GO TO 2011
      ANS=BFHI(H,AK,R,IOP )
      GO TO 3161
 2011 CONTINUE
      IF ( AK.NE.0. )  GO TO 2061
      GO TO 2023
 2021 CONTINUE
      IF ( R.LT.0. )  GO TO 2041
 2023 CONTINUE
      ANS=EC9(H,AK,R,IOP )
      GO TO 3161
 2031 CONTINUE
      IF ( AK.EQ.0. )  GO TO 2051
 2035 CONTINUE
      IF ( AK.LT.0. )  GO TO 2061
 2041 CONTINUE
      ANS=EC7(H,AK,R,IOP )
      GO TO 3161
 2051 CONTINUE
      IF ( R.GT.0. )  GO TO 2061
      GO TO 2041
 2061 CONTINUE
      ANS=EC8(H,AK,R,IOP )
      GO TO 3161
 3155 CONTINUE
      ANS=EC11(H,AK,R,IOP )
 3161 CONTINUE
      RETURN
 3171 CONTINUE
      IF ( AK.LE.(-H+EPS3(IOP)))  GO TO 3161
      T1=-AK/RT2
      T2=H/RT2
      ANS=.5*(ERFC(0,T1)-ERFC(0,T2))
      GO TO 3161
      END
```

```
FUNCTION  EQ7 (H,AK,R,IOP )
DATA RT2/1.4142 13562 373 /
T=-H/RT2
T1=-AK/RT2
EQ7=BPHI(-H,-AK,R,IOP )+.5*(ERFC(0,T)+ERFC(0,T1))-1.
RETURN
END
```

```
FUNCTION  EQ8 (H,AK,R,IOP )
DATA RT2/1.4142 13562 373 /
T=-AK/RT2
EQ8=-BPHI(-H,AK,-R,IOP )+.5*ERFC(0,T)
RETURN
END
```

```
FUNCTION  EQ9 (H,AK,R,IOP )
DATA RT2/1.4142 13562 373 /
T=-H/RT2
EQ9=-BPHI(H,-AK,-R,IOP )+.5*ERFC(0,T)
RETURN
END
```

```
      FUNCTION  EQ11(H,AK,R,IOP )
      DIMENSION EPS3(11)
      DATA ( EPS3(I),I=1,3 ) / 2.E-5,2.E-7,2.E-10 /
      DATA RT2/1.4142 13562 373 /
   91 FORMAT ( 1H0,3E22.15 )
      OM=1.-EPS3(IOP)
      IF ( R.LT.OM )  GO TO 2001
      T=H
      IF ( AK.LE.H )  T=AK
      T1=-T/RT2
      EQ11=.5*ERFC(0,T1)
      GO TO 1991
 1991 CONTINUE
      RETURN
 2001 CONTINUE
      CST=SQRT(H*H-2.*R*H*AK+AK*AK )
      T1=R*H-AK
      C1=1.
      T2=SIGN(C1,H)
      T1=(T1*T2)    /CST
      T4=1.
      T3=H*AK
      T5=SIGN(T4,T3)
      TDEL=(1.-    T5)*.25
      T3=R*AK-H
      C1=1.
      T2=SIGN(C1,AK)
      T3=(T3*T2)    /CST
      IF ( H.GT.0. )  GO TO 2031
      IF ( T1.GT.0. )  GO TO 2023
      T4=BPHI(H,0.,T1,IOP )
      GO TO 2051
 2023 CONTINUE
      T4=EQ9(H,0.,T1,IOP )
      GO TO 2051
 2031 CONTINUE
      IF ( T1.LT.0. )  GO TO 2041
      T4=EQ8(H,0.,T1,IOP )
      GO TO 2051
 2041 CONTINUE
      T4=EQ7(H,0.,T1,IOP )
 2051 CONTINUE
      IF ( AK.GT.0. )  GO TO 3031
      IF ( T3.GT.0. )  GO TO 3023
      T6=BPHI(AK,0.,T3,IOP )
      GO TO 3351
 3023 CONTINUE
      T6=EQ9(AK,0.,T3,IOP )
```

```
      GO TO 3051
3031  CONTINUE
      IF ( T3.LT.0. )  GO TO 3041
      T6=EQ8(AK,0.,T3,IOP )
      GO TO 3051
3041  CONTINUE
      T6=EQ7(AK,0.,T3,IOP )
3051  CONTINUE
      EQ11=T4+T6-TDEL
      RETURN
      END
      FUNCTION BPHI ( H,AK,R ,IOP )
      DIMENSION A(21),X(21),LLO(6),LHI(6)
      DIMENSION EPS1(11 )
      DIMENSION EPS3(11)
      DATA ( A(I) ,I=1,8 ) /
     1 4.4602 97704 66658E-1,  3.9646 82669 98335E-1,
     2 4.3728 88798 77644E-2,  2.4840 61520 28443E-1,
     3 3.9233 10666 52399E-1,  2.1141 81930 76057E-1,
     4 3.3246 66035 13439E-2,  5.2485 33445 15628E-4 /
      DATA ( X(I),I=1,8 ) /
     1 1.9055 41497 98192E-1,  8.4825 18675 44577E-1,
     2 1.7997 76578 41573E+0,  1.0024 21519 68216E-1,
     3 4.8281  39660 46201E-1,  1.0609 49821 52572E+0,
     4 1.7797 29418 52026E+0,  2.6697 60356 38766E+0 /
      DATA ( A(I),I=9,16 ) /
     1 1.3410 91884 53360E-1, 2.6833 07544 72640E-1,
     2 2.7595 33979 88422E-1, 1.5744 82826 18790E-1,
     3 4.4814 11991 74625E-2, 5.3679 35756 (2526E-3,
     4 2.0206 36491 32437E-4, 1.1925 96926 59532E-6 /
      DATA ( X(I),I=9,16 ) /
     1 5.2978 64393 18514E-2, 2.6739 83721 67767E-1,
     2 6.1630 28841 82402 E-1, 1.0642 46312 11623E+0,
     3 1.5688 55862 27006E+0, 2.1839 21153 09586E+0,
     4 2.8631 33883 70808E+0, 3.6860 07162 72440E+0 /
      DATA ( EPS1(I),I=1,3 ) / -8.,-12.,-20. /
      DATA PI / 3.1415 92653 58979 /
      DATA ( LLO(I),I=1,3 ) / 1,4,9 /
      DATA ( LHI(I),I=1,3 ) / 3,8,16 /
      DATA RT2 / 1.4142 13562 373 /
      DATA ( EPS3(I),I=1,3 ) / 2.E-5,2.E-7,2.E-10 /
      OM=1.-EPS3(IOP)
      ILO=LLO(IOP)
      IHI=LHI(IOP)
      EPS=EPS1(IOP)
      RSQ=R*R
      IF ( RSQ.LT.1. )  GO TO 2991
      T3=1.
      CST=0.
      GO TO 3001
```

```
2991  CONTINUE
      T3=SQRT(1.-RSQ)
      CST=RT2*T3
3001  CONTINUE
      BPHI=0.
      IF ( R.LE.-OM )  GO TO 3011
      IF ( R.LT.OM )  GO TO 3331
      T=H
      IF ( AK.LE.H )  T=AK
      T1=-T/RT2
      BPHI=.5*ERFC(0,T1)
      GO TO 3371
3011  CONTINUE
      IF ( AK.LE.(-H+EPS3(IOP) ) )  GO TO 3371
      T1=-AK/RT2
      T2=H/RT2
      ANS=.5*(ERFC(0,T1)-ERFC(0,T2))
      GO TO 3371
3331  CONTINUE
      H1=H/CST
      AK1=AK/CST
      SUM=0.
      DO 3361  I=ILO,IHI
      SUM1=0.
      DO 3351  J=ILO,IHI
      T1=H1*(2.*X(I)-H1)+AK1*(2.*X(J)-AK1)
     1 +2.*R*(X(I)-H1)*(X(J)-AK1)
      IF ( T1.LT.EPS )  GO TO 3351
      SUM1=SUM1+EXP(T1)*A(J)
3351  CONTINUE
      SUM=SUM+A(I)*SUM1
3361  CONTINUE
      BPHI=(SUM*T3)/PI
3371  CONTINUE
      RETURN
      END
```

APPENDIX C.

LISTING OF TEST PROGRAM WITH SOME NUMERICAL RESULTS

The test program listed in this appendix is designed to "see" all paths of the VALR16 subroutine, the basic routine of this report. The test program treats three different sets of triangles for each ϵ. i.e., ϵ_1, ϵ_2, ϵ_3. A total of 351 triangles are treated. Our subroutine VALR16 is used to obtain the probability P(H) over each triangle and the result is compared with the result obtained by the routine based on Drezner's method. The numerical results below state the case number, (x, y) vertices, VALR16 result, Drezner result, and absolute value of the difference, corresponding to that case for which the absolute value of the difference in P(H) for the two methods was a maximum for each set and for each ϵ. Thus there are nine cases given below.

Case No.	x	y	P(H) and $\lvert\Delta P\rvert$
		ϵ_1 = 2.54(−4)*	
3	2.0000 00000 0000 1.0000 00000 0000 3.0000 00000 0000	1.0000 00000 0000 0.0000 00000 0000 1.0000 00000 0000	.01116 23895 4828 .01144 55124 4546 2.83(−4)
116	3.0000 00000 0000 0.0000 00000 0000 3.0000 00000 0000	1.5000 00000 0000 −0.0006 06881 7000 0.0000 00000 0000	.07276 76379 5214 .07312 88147 1695 3.61(−4)
76	.11048 34376 7180 −1.8895 16562 3282 −1.8895 16562 3282	.11048 34376 7180 .11048 34376 7180 −.88951 65623 2820	.07464 96837 3470 .07443 77215 8773 2.12(−4)
		ϵ_2 = 2.57(−7)*	
15	0.0000 00000 0000 1.0000 00000 0000 0.0000 00000 0000	−2.0000 00000 0000 0.0000 00000 0000 1.0000 00000 0000	.17865 07387 5631 .17864 99501 5890 7.89(−7)
90	3.0000 00000 0000 3.0000 00000 0000 6.0000 00000 0000	3.0000 00000 0000 0.0000 00000 0000 1.5000 00000 0000	.00059 65636 6379 .00059 75925 2985 9.29(−7)
79	.01109 91882 5761 −1.9889 00811 7424 .01109 91882 5761	.01109 91882 5761 .01109 91882 5761 −.98890 08117 4239	.11200 59368 5515 .11200 54478 9902 4.89(−7)

*See Appendix A.

Case No.	x	y	P(H) and $\|\Delta P\|$
$\epsilon_3 = 2.94(-10)$*			
96	4.0000 00000 0000 2.0000 00000 0000 0.0000 00000 0000	0.0000 00000 0000 2.0000 00000 0000 7.8256 90500 (−10)	.12383 33015 1674 .12383 33012 5254 2.64(−10)
65	1.9999 80000 0000 −3.0000 00000 0000 3.0000 00000 0000	4.5000 00000 0000 0.0000 00000 0000 0.0000 00000 0000	.47645 25380 3718 .47645 25375 5211 4.85(−10)
94	.00116 10499 1180 −1.9988 38950 0882 2.0011 61049 9118	.00116 10499 1180 −1.9988 38949 3056 −1.9988 38950 0882	.22803 35273 2106 .22803 35268 0156 5.19(−10)

*See Appendix A.

```
      PROGRAM DRET ( OUTPUT )
      COMMON IOP
      DIMENSICN X(201),Y(201),X1(201),Y1(201)
      DIMENSICN EPS1(4),X3(3),Y3(3)
      DIMENSICN APH21(3),APH31(3)
      DIMENSION IRAY(21)
      DIMENSICN DEL1(3),B1(3),ALPHA(3)
C      SET A FOR PJ 3V HULL DECK
      DATA ( X(I),I=1,48 ) /
     1  1.,3.,2.,  1.,2.,-1.,  1.,2.,-1.,  1.,0.,0.,
     2  1.,0.,0.,  1.,-1.,2.,  1.,0.,0.,  1.,0.,2.,
     3  1.,0.,3.,  1.,2.,2.,  1.,2.,2.,  1.,3.,0.,
     4  0.,2.,1.,  0.,1.,-1.,  0.,-1.,1.,  0.,1.,2. /
      DATA ( Y(I),I=1, 48 ) /
     1  0.,1.,1.,  0.,1.,1.,  0.,1.,-1.,  0.,2.,1.,
     2  0.,1.,-2.,  0.,1.,-2.,  0.,-1.,-2.,  0.,-1.,-1.,
     3  0.,-2., 1.,  0.,-2.,-1.,  0.,-1.,2.,  0.,-1.,2.,
     4  0.,1.,1.,  0.,1.,1.,  0.,-2.,1.,  0.,-1.,-1. /
C      SET B FOR PJ 3V HULL DECK
      DATA ( X(I),I=49,90 ) /
     1  1.,.5,2.,  1.,1.,2.,  1.,1.5,2.,  1.,.5,-1.,
     2  1.,1.,-1.,  1.,.66666,-1.,  1.,2.,1.33333,  1.,2.,1.,
     3  1.,2.,.5,  1.,-1.,-1.,  1.,-1.,1.,  1.,1.5,-1. ,
     4  1.,1.,2.,  1.,2.,1. /
      DATA ( Y(I),I=49,90 ) /
     1  0.,-1.5,0.,  0.,-1.,0.,  0.,-.5,0.,  0.,.5,0.,
     2  0.,1.,0.,  0.,1.5,0.,  0.,0.,.5,  0.,0.,1.,
     3  0.,0.,1.5,  0.,0.,-1.,  0.,0.,-1.,  0.,1.,0. ,
     4  0.,-2.,1.,  0.,.5,1. /
      DATA ( X(I),I=91,102 ) /
     1  .5,1.5,0.,  2.,0.,4.,  2.,0.,4.,  .5,1.5,0. /
      DATA ( Y(I),I=91,102 ) /
     1  .5,1.,2.,  2.,1.,0.,  2.,1.,0.,  .5,1.,2. /
      DATA ( X(I),I=103,117 ) /
     1  1.,0.,0.,  1.,0.,0.,  1.,0.,0.,  1.,0.,1.,
     2  1.,1.,0. /
      DATA ( Y(I),I=103,117 ) /
     1  0.,1.,1.,  0.,1.,-.5,  0.,.5,1.,
     2  0.,1.,-.5,  0.,.5,1. /
      DATA (  EPS1(I),I=1,3 ) /
     1 2.5362 66450E-4,              2.5714 45247E-7,2.9434 48712E-10/
      DATA ( DEL1(I),I=1,3 ) /
     1  4.49542E-4,4.55777E-7,5.217127E-10 /
      DATA ( B1(I),I=1,3 ) / 2.46,3.5505,4.382 /
      DATA ( APH31(I),I=1,3 ) /
     1  .224771E-3,.2278885E-6,.26085635E-9 /
      DATA ( APH21(I),I=1,3 ) /
     1  .12206 59E-1,.12319 198E-3,.13480 369E-5 /
```

```
   90  FORMAT ( 1H0,60X,I10 )
   91  FORMAT ( 1H-,8E16.8 )
   92  FORMAT ( 1H- )
   93  FORMAT ( 1H0,I10 )
   97  FORMAT ( 1H1 )
   95  FORMAT ( 1H ,44X,2E12.4 )
       N=3
       PRINT 97
       DO 3021  I=1,151

       X1(I)=X(I)
       Y1(I)=Y(I)
 3021  CONTINUE
       DO 3071  I3=1,3
       PRINT 97
       PRINT 90,I3
       IFNT=0
       DO 3061  I5=1,3
       PRINT 92
       TMAX=0.
       M=0
       DO 3025  I=1,151
       X(I)=X1(I)
       Y(I)=Y1(I)
 3025  CONTINUE
       APH3=APH31(I3)
       Y(92)=1.5-3.*APH3
       Y(95)=3.*APH3
       Y(98)=-3.*APH3
       Y(101)=1.5+3.*APH3
       Y(104)=.9*APH3
       Y(105)=-.9*APH3
       Y(107)=Y(104)
       Y(111)=Y(105)
       Y(113)=Y(104)
       Y(117)=Y(105)
       DO 3027  I=1,151
       Y1(I)=Y(I)
```

```
3027  CONTINUE
      IF ( I5.EQ.1 )  GO TO 3035
      IF ( I5.EQ.3 )  GO TO 3035
      DO 3031  I=1,151
      X(I)=3.*X1(I)
      Y(I)=3.*Y1(I)
3031  CONTINUE
3035  CONTINUE
      DO 3051  I=1,115,3
3037  CONTINUE
      IP2=I+2
      DO 3049  I1=1,3
      M=M+1
      IF ( IPNT.NE.0 )  PRINT 93,M
3041  CONTINUE
      IF ( I1.EQ.1 )  GO TO 3047
      T1=X(I)
      T2=Y(I)
      X(I)=X(I+1)
      Y(I)=Y(I+1)
      X(I+1)=)(IF2)
      Y(I+1)=Y(IP2)
      X(IP2)=T1
      Y(IP2)=T2
3047  CONTINUE
      IF ( I5.NE.3 )  GO TO 3045
      IF ( I1.GT.1 )  GO TO 3045
      T1=APH21(I3)
      T2=SQPT(T1)-1.E-12
      SX=X(I)-T2
      SY=Y(I)-T2
      DO 3043  I7=I,I+2
      X(I7)=X(I7)-SX
      Y(I7)=Y(I7)-SY
```

```
3043  CONTINUE
3045  CONTINUE
3048  CONTINUE
      T1=(X(I)-X(I+1))*(Y(I)-Y(I+2))
      T2=(X(I)-X(I+2))*(Y(I)-Y(I+1))
      IF ( T1.GE.T2 )  GO TO 304
      T1=X(I+2)
      X(I+2)=X(I+1)
      X(I+1)=T1
      T2=Y(I+2)
      Y(I+2)=Y(I+1)
      Y(I+1)=T2
 304  CONTINUE
      IF ( IFNT.NE.0 )
     1 PRINT 95,(X(J),Y(J),J=I,IP2 )
      IOP1=I3
      CALL VALR16( X(I),Y(I),N,ANS1,IOP1 )
      ANS=ANS1
      IOP=I3
      CALL DREZNR ( X( I) ,Y(I),3,ANS3,I3 )
      DEL=ABS(ANS-ANS3)
      IF ( DEL.LE.TMAX )  GO TO 3049
      MSAV=M
      SAVDEL=DEL
      TMAX=DEL
      X3(1)=X(I)
      X3(2)=X(I+1)
      X3(3)=X(I+2)
      Y3(1)=Y(I)
      Y3(2)=Y(I+1)
      Y3(3)=Y(I+2)
      SAVDR=ANS3
      SAVPJ=ANS
3049  CONTINUE
3051  CONTINUE
      PRINT 93,MSAV
      PRINT 96,X3(1),Y3(1)
      PRINT 96,X3(2),Y3(2)
      PRINT 96,X3(3),Y3(3)
      PRINT 94,SAVPJ,SAVDR,SAVDEL
3061  CONTINUE
3071  CONTINUE
4011  CONTINUE
  94  FORMAT ( 1H0,44X,E22.15 )
  96  FORMAT ( 1H0,6E22.15 )
9011  CONTINUE
      STOP
      END
```

APPENDIX D.

FORTRAN LISTING OF THE PROGRAM

This appendix contains the basic subroutine of this report which calculates P(A) or P(H) to 3, 6, or 9 correct decimal digits.

CALL VALR16 (x, y, N, ans, IOP)

where:

x, y are input arrays of the coordinates of the vertices.

N is the number of sides of the polygon.*

Vertices must be listed in counterclockwise order. See pp. 9, 10.

ans identifies the location where the P_k is returned.

IOP = 1, 2 or 3 for 3, 6 or 9 decimal digits of accuracy, respectively.

*N = 1 for an angular region A specified by three points given in counterclockwise order with the first point at the vertex of A, (See page 12). Note $0 \leq \Delta\theta \leq 2\pi$ for N = 1, but $0 \leq \Delta\theta \leq \pi$ for $N \geq 3$.

```
      SUBROUTINE VALR16( X,Y,N,ANS,IOP )
      DIMENSION RSQ(4)
      DIMENSION X(1),Y(1),U(2),V(2),G(2),H(2)
      DIMENSION E(5),E2(10),E3(15)
      DIMENSION APH1(3),APH2(3),APH3(3)
      REAL L
      DATA PI/3.1415 92653 58979 /
      DATA TWOPI/6.2831 85307 17958 /
      DATA ALNPI/1.1447 29885 84940 /
      DATA C1/.28209 47917 73877 /
      DATA C2/.15915 49430 91895 /
       DATA ( E(I),I=1,  5) /
     1         .885777518572895E+00 ,         -.981151952778050E+00 ,
     2         .759305502082485E+00 ,         -.353644980686977E+00 ,
     3         .695232092435207E-01 /
       DATA (E2(I),I=1, 10) /
     1         .886226470016632E+00 ,         -.999950714561036E+00 ,
     2         .885348820003892E+00 ,         -.660611239043357E+00 ,
     3         .421821197160099E+00 ,         -.222898055667208E+00 ,
     4         .905057384150449E-01 ,         -.254906111884287E-01 ,
     5         .430895168984138E-02 ,         -.323377239693247E-03 /
       DATA (E3(I),I=1, 15) /
     1         .886226924931465E+00 ,         -.999999899776252E+00 ,
     2         .886223733186722E+00 ,         -.666626670510907E+00 ,
     3         .442851899328569E+00 ,         -.265638206366025E+00 ,
     4         .145060043403014E+00 ,         -.714909837799889E-01 ,
     5         .309199295521210E-01 ,         -.112323532148441E-01 ,
     6         .324944543171185E-02 ,         -.704260243309096E-03 ,
     7         .105787574480633E-03 ,         -.971864864160461E-05 ,
     8         .408335517232165E-06 /
      DATA ( APH1(I),I=1,3 ) /
     1 2.02E-7,2.08E-13,2.72E-19 /
      DATA ( APH2(I),I=1,3 ) /
     1 1.22E-2,1.23E-4,1.35E-6 /
      DATA ( APH3(I),I=1,3 ) /
     1 2.25E-4,2.28E-7,2.61E-10 /
      DATA (  RSQ (I),I=1,3 ) /
     1 6.0516,12.60605 ,19.201924 /
      NM1=N- 1
      K=1
      PHIN=0.
      PHIK=0 .
      ANS=0.
      U(1)=X( 2 )-X(K)
      V(1)=Y( 2 )-Y(K)
      IF ( N.NE.1 )  GO TO 3131
      U(2)=X(3)-X(1)
      V(2)=Y(3)-Y(1)
```

```
      SN=V(2)*U(1)-U(2)*V(1)
      IF ( SN.GE.0. )  GO TO 3141
      ANS=-1.
      T1=U(1)
      U(1)=U(2)
      U(2)=T1
      T1=V(2)
      V(2)=V(1)
      V(1)=T1
      GO TO 3141
 3131 CONTINUE
      U(2)=X(N)-X(1)
      V(2)=Y(N)-Y(1)
 3141 CONTINUE
      BGD1=SQRT( 2.*(U(1)*U(1)+V(1)*V(1)))
      BGD2=SQRT( 2.*(U(2)*U(2)+V(2)*V(2)))
 3151 CONTINUE
      L=0.
      ALAM=0.
      B=.5*(X(K)*X(K)+Y(K)*Y(K))
      IF ( B.GT.APH1(IOP) )  GO TO 3171
      CAPG=0.
 3161 CONTINUE
      T1=ABS(V(2)*U(1)-U(2)*V(1))
      T2=U(2)*U(1)+V(2)*V(1)
      PHIK=ATAN2(T1,T2)
      ANS1=PHIK/TWOPI-CAPG
      GO TO 3621
 3171 CONTINUE
      G(1)=U(1)*X(K)+V(1)*Y(K)
      G(2)=U(2)*X(K)+V(2)*Y(K)
      H(1)=-Y(K)*U(1)+X(K)*V(1)
      H(2)=-Y(K)*U(2)+X(K)*V(2)
      G(1)=G(1)/BGD1
      G(2)=G(2)/BGD2
      H(1)=H(1)/BGD1
      H(2)=H(2)/BGD2
      SN=(2.*(V(2)*U(1)-U(2)*V(1)))/(BGD1*BGD2)
      IF ( SN.GT.0. )  GO TO 3185
      CN=G(1)*G(2)+H(1)*H(2)
      IF ( CN.GE.0. )  GO TO 3183
      PHIK=PI
      IF ( G(1).LT.0. )  GO TO 3181
      ANS1=.5*ERFC(0,H(2) )
      GO TO 3621
 3181 CONTINUE
      ANS1=.5*ERFC(0,-H(1) )
      GO TO 3621
```

```
3183  CONTINUE
      PHIK=0.
      ANS1=0.
      GO TO 3621
3185  IF ( B.LE.APH2(IOP) )  GO TO 3301
      IF ( G(1).LT.0. )  GO TO 3261
      IF ( G(2).GE.0. )  GO TO 3471
      G(2)=-G(2)
      H(2)=-H(2)
      ALAM=PI
      IF ( ABS(H(2)).LE.APH3(IOP) )  GO TO 3251
      L=.5*ERFC(0,-H(2))
      GO TO 3471
3251  CONTINUE
      L=.5
      GO TO 3471
3261  CONTINUE
      G(1)=-G(1)
      H(1)=-H(1)
      IF ( G(2).LT.0. )  GO TO 3271
      ALAM=PI
      IF ( ABS(H(1)).LE.APH3(IOP) )  GO TO 3251
      L=.5*ERFC(0,H(1))
      GO TO 3471
3271  CONTINUE
      G(2)=-G(2)
      H(2)=-H(2)
      IF ( ABS(H(1)).LE.APH3(IOP) )  GO TO 3291
      IF ( ABS(H(2)).LE.APH3(IOP) )  GO TO 3281
      L=.5*(ERFC(0,H(1))-ERFC(0,H(2)))
      GO TO 3471
3281  CONTINUE
      L=.5*(ERFC(0,H(1))-1.)
      GO TO 3471
3291  CONTINUE
      IF ( ABS(H(2)).LE.APH3(IOP) )  GO TO 3471
      L=.5*(1.-ERFC(0,H(2)))
      GO TO 3471
3301  CONTINUE
      CAPG=C1*(H(2)-H(1))-C2*(G(2)*H(2)-G(1)*H(1))
      GO TO 3161
3471  CONTINUE
      IF ( B.LT.RSQ(IOP) )  GO TO 3479
      PHIN=-8.
      GO TO 3495
3479  CONTINUE
      IF ( K.NE.N )  GO TO 3480
      IF ( PHIN  .LE.0. )  GO TO 3480
```

```
      AJ0=PHIN-ALAM
      GO TO 3481
3480  CONTINUE
      SN=G(1)*H(2)-G(2)*H(1)
      CN=G(1)*G(2)+H(1)*H(2)
      AJ0=ATAN2(SN,CN)
      PHIK=AJ0
      IF ( AJ0.LT.0. )  PHIK=PI+AJ0
3481  CONTINUE
      CAPG=AJ0
      CAPH=.5*AJ0
      M=1
      F=0.
      AJ1=H(2)-H(1)
      CIRCM=AJ1
      IF ( IOP.EQ.3 )  GO TO 3681
      IF ( IOP.EQ.2 )  GO TO 3701
      SUM=E(M)*AJ1
3482  CONTINUE
      M=M+1
      H(2)=H(2)*G(2)
      H(1)=H(1)*G(1)
      T=H(2)-H(1)
      F=F+B
      CAPV=(F*CAPG+T)/M
      SUM=SUM+E(M)*CAPV
      IF( M .GE. 5    )   GO TO 3491
      CAPG=CIRCM
      CIRCM=CAPV
      GO TO 3482
3491  CONTINUE
      ANS1=L+EXP(-(B+ALNPI))*(CAPH-SUM)
      GO TO 3621
3495  CONTINUE
      ANS1=L
3621  CONTINUE
      IF ( (K-NM1) )  3631,3661,3623
3623  CONTINUE
      ANS=ABS(ANS+ABS(ANS1) )
      RETURN
3631  CONTINUE
      K=K+1
      KP1=K+1
      IF ( K.NE.2 )  GO TO 3651
      ANS=ABS(ANS1)
      U(2)=X(KP1)-X(K)
      V(2)=Y(KP1)-Y(K)
      PHIN=PHIN-PHIK
```

```
      BGD2=SQRT( 2.*(U(2)*U(2)+V(2)*V(2)))
      GO TO 3151
3651  CONTINUE
      U(1)=U(2)
      V(1)=V(2)
      U(2)=X(KP1)-X(K)
      V(2)=Y(KP1)-Y(K)
      BGD1=BGD2
      BGD2=SQRT( 2.*(U(2)*U(2)+V(2)*V(2)))
      GO TO 3671
3661  CONTINUE
      K=N
      U(1)=X(N)-X(1)
      V(1)=Y(N)-Y(1)
      BGD1=SQRT( 2.*(U(1)*U(1)+V(1)*V(1)))
3671  CONTINUE
      PHIN=PHIN+PHIK
      ANS=ANS-ABS(ANS1)
      GO TO 3151
3681  CONTINUE
      SUM=E3(M)*AJ1
3691  CONTINUE
      M=M+1
      H(2)=H(2)*G(2)
      H(1)=H(1)*G(1)
      T=H(2)-H(1)
      F=F+B
      CAPV=(F*CAPG+T)/M
      SUM=SUM+E3(M)*CAPV
      IF ( M.GE.15 )  GO TO 3491
      CAPG=CIRCM
      CIRCM=CAPV
      GO TO 3691
3701  SUM=E2(M)*AJ1
3711  M=M+1
      H(2)=H(2)*G(2)
      H(1)=H(1)*G(1)
      T=H(2)-H(1)
      F=F+B
      CAPV=(F*CAPG+T)/M
      SUM=SUM+E2(M)*CAPV
      IF ( M.GE.10 )  GO TO 3491
      CAPG=CIRCM
      CIRCM=CAPV
      GO TO 3711
      END
```

DISTRIBUTION LIST

Chief of Naval Operations
Department of the Navy
Washington, D.C. 20350
Attn: OP-980
OP-982
OP-982E
OP-982F
OP-983
OP-987
OP-961

Commanding Officer and Director
Naval Ship Research and Development Center
Washington, D.C. 20034
Attn: Code 18
Code 154
Code 184
Code 1541
Code 1802
Code 1805
Library

Commander, Naval Facilities Engineering Command
Department of the Navy, Washington, D.C. 20390
Attn: 0322

ECOM Offic Building
U.S. Army Electronic Command
Fort Monmouth, New Jersey 07703
Attn: Technical Library

Director, Naval Research Laboratory
Washington, D.C. 20390
Attn: Code 7800
Library

Office of Naval Research
Washington, D.C. 20360
Attn: Math and Information Sciences Division
Library

Commander, Naval Air Systems Command
Department of the Navy
Washington, D.C. 20360
Attn: Code NAIR-03
Code NAIR-03D
Code NAIR-5034

Commander, Naval Electronics Systems Command
Department of the Navy
Washington, D.C. 20360
Attn: Code NELEX-03A

Commander, Naval Sea Systems Command
Department of the Navy
Washington, D.C. 20362
Attn: SEA 03A
SEA 034E
SEA 035
SEA 035B

Fleet Analysis Center
Naval Weapons Center
Seal Beach
Corona, California 91720
Attn: Library

Commander, U.S. Naval Weapons Center
China Lake, California 93555
Attn: Code 6073
Code 60704
Library

Naval Sea Systems Command
Department of the Navy
Washington, D.C. 20362

U.S. Naval Observatory
34th Street and Massachusetts Avenue, N.W.
Washington, D.C. 20390
Attn: Library

U.S. Naval Oceanographic Office
Washington, D.C. 20390
Attn: Code 0814
Library

Navy Publications and Printing Service Office
Naval District of Washington
Washington, D.C. 20390

Commanding Officer
U.S. Army Harry Diamond Laboratories
Washington, D.C. 20438

AFADSB Headquarters, U.S. Air Force
Washington, D.C. 20330

Director
Defense Research and Engineering
Washington, D.C. 20390
Attn: WSEG
Deputy Director, Tactical Warfare Programs
Deputy Director, Test and Evaluation

The Library of Congress
Washington, D.C. 20540
Attn: Exchange and Gift Division (4)

Director
Defense Intelligence Agency
Washington, D.C. 20301

Lawrence Radiation Laboratory
Technical Information Department
P.O. Box 808
Livermore, California 94550

Sandia Corporation
Livermore Branch
P.O. Box 969
Livermore, California 94550
Attn: Technical Library

Numerical Analysis Research Library
University of California
405 Hilgard Avenue
Los Angeles, California 90024

Superintendent
U.S. Naval Postgraduate School
Monterey, California 93940
Attn: Library, Tech Reports Section

Commander
Naval Undersea Research and Development Center
3202 East Foothill Boulevard
Pasadena, California 91107
Attn: Code 254
Code 2501
Code 25403
Code 25406

Director
Office of Naval Research Branch Office
1030 East Green Street
Pasadena, California 91101

Commanding Officer
Marine Air Detachment
Naval Missile Center
Point Mugu, California 93041

Office of Naval Research
Branch Office, Chicago
219 South Dearborn Street
Chicago, Illinois 60604

Superintendent
U.S. Naval Academy
Annapolis, Maryland 21402
Attn: Library, Serials Division

Commanding Officer
U.S. Army Aberdeen R&D Center
Aberdeen, Maryland 21005
Attn: Dr. F. E. Grubbs
Library

Director
U.S. Army Munitions Command
Edgewood Arsenal, Maryland 21010
Attn: Operations Research Group

Director
National Security Agency
Fort George G. Meade, Maryland 20755
Attn: Dr. M. Kupperman
Library

Director
National Bureau of Standards
Gaithersburg, Maryland 20760
Attn: Library

Director
National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland 20771
Attn: Library

Commanding Officer
Naval Weapons Evaluation Facility
Kirtland Air Force Base
Albuquerque, New Mexico 87117

Los Alamos Scientific Laboratory
P.O. Box 1663
Los Alamos, New Mexico 87544
Attn: Report Library

Commanding General
White Sands Missile Range
Las Cruces, New Mexico 88002
Attn: Technical Library, Documents Section

Air Force Armament Laboratory
Eglin Air Force Base, Florida 32542

Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439
Attn: Dr. A. H. Jaffey, Building 200

The RAND Corporation
4921 Auburn Avenue
Bethesda, Maryland 20014
Attn: Library

The RAND Corporation
1700 Main Street
Santa Monica, California 90406 (2)

University of Chicago
Chicago, Illinois 60637
Attn. Prof. W. Kruskel, Statistics Dept.

Journal of Mathematics and Mechanics
Mathematics Department, Swain Hall East
Indiana University
Bloomington, Indiana 47401

NASA Scientific and Technical Information Facility
P.O. Box 33
College Park, Maryland 20740

The Johns Hopkins University
Applied Physics Laboratory
8621 Georgia Avenue
Silver Spring, Maryland 20910
Attn: Strategic Analysis Support Group
Document Librarian

Prof. George F. Carrier
Pierce Hall, Room 311
Harvard University
Cambridge, Massachusetts 02138

Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
Attn: Computation Center

University of Michigan
Institute of Science and Technology
P.O. Box 618
Ann Arbor, Michigan 48107
Attn: Operations Research Division
Dr. E. H. Jobe

Joint Strategic Target Planning Staff
Offutt Air Force Base, Nebraska 68113 (2)

Systems Engineering Group (2)
Wright-Patterson Air Force Base, Ohio 45433

President
Naval War College
Newport, Rhode Island 02840

Booz, Allen Applied Research, Inc.
6151 West Century Boulevard
Los Angeles, California 90045

Prof. M. Albertson
Department of Civil Engineering
Colorado State University
Fort Collins, Colorado 80521

California Institute of Technology
Pasadena, California 91109
Attn: Prof. T. Y. Wu

Rutgers University
Statistics Center
New Brunswick, New Jersey 08903
Attn: Prof. M. F. Shakun

Dr. George Ioup, Physics Department
Louisiana State University
Lake Front
New Orleans, Louisiana 70122

Burroughs Corporation Research Center
Paoli, Pennsylvania 19301
Attn: R. Mirsky, Advanced Systems

Pennsylvania State University
University Park, Pennsylvania 16802
Attn: Prof. P. C. Hammer, Computer Science Department

Defense Documentation Center
Cameron Station
Alexandria, Virginia 22314 (12)

Langley Aeronautical Laboratory
National Aeronautics and Space Administration
Langley Field, Virginia 23365
Attn: Mr. J. B. Parkinson

Prof. Gary Makowski
Department of Math and Statistics
Marquette University
Milwaukee, Wisconsin 53233

University of Wyoming
Statistics Department
Box 3275, University Station
Laramie, Wyoming 82070
Attn: Dr. W. C. Guenther
Mr. J. Terragno

Mrs. Pamela M. Morse
Canada Department of Agriculture
Sir John Carling Building, Room E265
Statistical Research Service
C.E.F. Ottawa, Ontario
Canada

National Research Council
Montreal Road
Ottawa 2, Canada
Attn: Mr. E. S. Turner

Prof. H. Primas
Swiss Federal Institute of Technology
Physical Chemistry Laboratory
Universitatstrasse 22
8006 Zurich
Switzerland

V.P.I. and State University
Blacksburg, Virginia 24060
Attn: Dr. J. Arnold, Statistics Dept.
Dr. R. H. Myers, Statistics Dept.

Dr. Donald Amos
Division 5122
Sandia Laboratories
Albuquerque, New Mexico 87115

Alan B. Bligh
Code 7810
Naval Research Laboratory
Washington, D.C. 20375

Gene H. Gleissner
Code 18
Naval Ship Research and Development Center
Bethesda, Maryland 20084

Allen Miller
Code 1723
Naval Research Laboratory
Washington, D.C. 20375

Dr. Richard Nance
Department of Computer Science
562 McBryde Hall
V.P.I. and State University
Blacksburg, Virginia 24061

B. L. Ball QEC.
Naval Weapons Station
Seal Beach, California 90740

Director Tradoc System Analysis Activity
USA TRASANA
ATAA – TDS
White Sands Missile Range
New Mexico 88002

Donald Baker Moore
Exploratory Technology
P.O. Box KK
Fairfield, California 94533

I. Sugai 8-100
The Johns Hopkins University
Applied Physics Laboratory
Johns Hopkins Road
Laurel, Maryland 20810

P. Faglioni
Universita di Modena
Clinica delle Malattie Nervose e Mentali
via del Pozzo, 71 - 41100 Modena
Italia

Prof. George Marsaglia
Computer Science Department
Avery 451
Washington State University
Pullman, Washington 99164

U.S. Army Electronic Command
ECOM Office Building
Ft. Monmouth, New Jersey 07703
Attn: Technical Library

Institute of Statistics
University of Stockholm
P.O. Box 6701
S-113-85 Stockholm, Sweden
Attn: Prof. Karin Dahmström

Dr. D. Rasch
Akademie der Landwirtschaftswissen
Shaften der DDR Ferschungszenirum
I. Tierproduktion Dummerstorf-Rostock
Bibliotek
2551 Dummerstorf
Germany

Prof. J. Gurland
University of Wisconsin-Madison
Department of Statistics
1210 West Dayton St.
Madison, Wisconsin 53706

Mr. John A. Simpson
Senior Systems Engineer
Norden, Div. of United Technologies
Norwalk, Connecticut 06856

M. Miller
Schering Corporation
Bloomfield, New Jersey 07003

Dr. David Giri
Airforce Weapons Lab./ELP
Kirtland Airforce Base
New Mexico 87117

Prof. Dr. A. Seeger
Max-Planck Institute
7000 Stuttgart 80
Heisenbergstrasse 1
Germany

Jason A. C. Gallas
Institu de Fisica
Universidade Federal do Rio Grande do Sol
90000 Porto Alegre - RS - Brasil

Harvey S. Picker
Physics Department
Trinity College
Hartford, Connecticut 06106

Dr. A. Pellegatti
Laboratoire de Chimie Theorique
Universite de Provence
Place Victor-Hugo
13331 Marseille - Cedex 3
France

Energy Research & Development Admin.
Division of Military Applications
Washington, D.C. 20545
Attn: Library

Naval Electronic Laboratory Center
San Diego, California 20362
Attn: Library

H. Saunders
Building 41 - Room 319
General Electric Company
One River Road
Schenectady, New York 12345

R. F. Hausman
Lockheed Missile & Space Company
Department 6213 Building 104
P.O. Box 504
Sunnyvale, California 94088

Melvin Cohen
Computer Science Department
McGill University
805 Sherbrooke Street, West
Montreal, Quebec
Canada H3A - 2K6

Local Distribution:

D
D-1
K
K-01
K-02
K-04
K-05D(30)
K-05H(Dr. A. V. Hershey)
K-05J(Dr. M. P. Jarnagin)
K-10
K-11
K-11(Dr. B. Zondek)
K-20
K-21
K-22
K-23
K-30
K-30(Dr. M. Thomas) (2)
K-30(D. Snyder)
K-40
K-50
K-50(Dr. E. Ball)
K-50(Dr. A. Evans)
K-55
K-60
K-70
K-71
K-72
K-73
K-74
F
G
G-10
G-10(F. Clodius)

N
N-10
N-10(S. Vittoria)
N-20
N-30
N-40
X-21(2)

Math Department, WOL
Dr. J. W. Enig, R-10
Dr. A. H. VanTuyl, A-43
Technical Library, X-21 (4)