NASA TECHNICAL NOTE

NASA TN D-7924

2. u/u

NASA TN D-7924

4.

PROGRAMS FOR COMPUTING ABSCISSAS AND WEIGHTS FOR CLASSICAL AND NONCLASSICAL GAUSSIAN QUADRATURE FORMULAS

Robert N. Desmarais

Langley Research Center
Hampton, Va. 23665

3.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • OCTOBER 1975

5.

1. Report No. NASA TN D-7924	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle PROGRAMS FOR COMPUTING ABSCISSAS AND WEIGHTS FOR CLASSICAL AND NONCLASSICAL GAUSSIAN QUADRATURE FORMULAS		5. Report Date October 1975
		6. Performing Organization Code
7. Author(s) Robert N. Desmarais		8. Performing Organization Report No. L-9653
		10. Work Unit No. 505-02-21-01
9. Performing Organization Name and Address NASA Langley Research Center Hampton, Va. 23665		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Note
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

Computer programs for computing Gaussian quadrature abscissas and weights are described. For the classical case the programs use Laguerre iteration to compute abscissas as zeros of orthogonal polynomials. The polynomials are evaluated from known recursion coefficients. The nonclassical case is handled similarly except that the recursion coefficients are computed by numerical integration. A sample problem, with input and output, is presented to illustrate the use of the programs. It computes the quadrature abscissas and weights associated with the weight function $(1 - x)^{1/2} \ln (1/x)$ over the interval (0,1) for quadrature orders from 16 to 96 in increments of 8.

17. Key Words (Suggested by Author(s)) Gaussian quadrature Orthogonal polynomials Numerical integration	18. Distribution Statement Unclassified – Unlimited Subject Category 61

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 73	22. Price* $4.25

*For sale by the National Technical Information Service, Springfield, Virginia 22161

CONTENTS

PROGRAMS FOR COMPUTING ABSCISSAS AND WEIGHTS FOR CLASSICAL AND NONCLASSICAL GAUSSIAN QUADRATURE FORMULAS

Robert N. Desmarais
Langley Research Center

SUMMARY

Computer programs for computing Gaussian quadrature abscissas and weights are described. For the classical case the programs use Laguerre iteration to compute abscissas as zeros of orthogonal polynomials. The polynomials are evaluated from known recursion coefficients. The nonclassical case is handled similarly except that the recursion coefficients are computed by numerical integration. A sample problem, with input and output, is presented to illustrate the use of the programs. It computes the quadrature abscissas and weights associated with the weight function $(1 - x)^{1/2} \ln (1/x)$ over the interval (0,1) for quadrature orders from 16 to 96 in increments of 8.

INTRODUCTION

This paper describes a set of computer programs for computing the abscissas and weights of Gaussian quadrature formulas. The programs permit the calculation of both classical and nonclassical abscissas and weights. For the classical case the programs are complete in the sense that the user need only specify the order, interval, and constants appearing in the weight function, and the programs will do the rest. For the nonclassical case, the user has to set up the quadrature scheme to be used for computing the recursion coefficients of the orthogonal system associated with the Gaussian quadrature formula desired.

The programs described herein were developed to generate quadrature formulas for use in computing unsteady aerodynamic forces. For example, the kernel of the integral equation relating lift to downwash in unsteady subsonic flow can be computed, in part, using a Laguerre-Gauss quadrature; integrals of the pressure distribution can be computed using Jacobi-Gauss quadrature; and integrals of pressure distributions over a control-surface hinge can be evaluated using a nonclassical Gaussian quadrature with the weight function $\ln (1/x)$.

For all of the quadrature formulas mentioned in the preceding paragraph except the one with the logarithmic weight the existing published tables of abscissas and weights are more than adequate. Even if tables are available, however, for classical weight functions it is more economical to generate the abscissas and weights from a computer program than it is to keypunch and verify a set of tabulated values. This is not true for nonclassical weight functions. However, for these weight functions the existing tables are not adequate.

SYMBOLS

a,b	limits of integration
c_n, b_n	coefficients of recursion formula
C_ℓ	correction term for a singularity not appearing in the weight function
$f(x)$	function to be integrated
$f_L(x)$	truncated Taylor's expansion of $f(x)$ about x_s
$g(x)$	either $p_n^2(x)$ or $xp_n^2(x)$
h_n	integral defined by equation (14)
h'_n	integral defined by equation (15)
I	a definite integral to be approximated
k	index of summation in a quadrature formula
L	degree of truncated Taylor's expansion
$L(x)$	Lagrange's interpolation function
ℓ	an index of summation
$\ell_k(x)$	$= \dfrac{\pi(x)}{(x - x_k)\pi'(x_k)}$

M	number of terms in quadrature sum used to approximate h_n or h_n'
m	an index of summation or degree of orthogonal polynomial
N	order of a Gaussian quadrature formula
n	order of a Gaussian quadrature formula or degree of orthogonal polynomial
$p(x)$	orthogonal polynomials
$q(x)$	quotient when $f(x)$ is divided by $\pi(x)$
$r(x)$	remainder when $f(x)$ is divided by $\pi(x)$
S_k	kth moment
$\overline{S}_\ell$	the ℓth shifted moment of $\rho(x)$
$U(\alpha,m)$	$= \gamma + \psi(\alpha + m)$
w_k	a quadrature weight
x	variable of integration
x_k	a quadrature abscissa
x_s	abscissa of a singularity of $\rho(x)$
α,β	exponents appearing in $\rho(x)$ or $\tau(x)$
δ_{nm}	Kronecker delta (0 if $n \neq m$; 1 if $n = m$)
γ	Euler's constant, 0.57721 . . .
$\pi(x)$	$= (x - x_1)(x - x_2) \ldots (x - x_n)$ where x_k for $k = 1$ to n are quadrature abscissas

$\rho(x)$ weight function

$\sigma(x)$ a factor of $\rho(x)$ that is singular at $x = x_s$

$\tau(x)$ a classical or almost classical factor of $\rho(x)$

$\psi(Z)$ digamma function, $\Gamma'(Z)/\Gamma(Z)$

Prime denotes first derivative.

Double prime denotes second derivative.

Superscript within parentheses indicates a specific derivative.

PROBLEM DESCRIPTION

The most efficient way to evaluate integrals with integrable singularities is to use a Gaussian quadrature formula. In this procedure the singular part of the integrand is factored out and processed analytically. Also the abscissas at which the remaining factors of the integrand are to be evaluated are chosen so as to maximize the degree of the polynomial for which the procedure is exact. That is, a Gaussian quadrature formula is a numerical integration rule of the form

$$I = \int_a^b \rho(x)\, f(x)\, dx \approx \sum_{k=1}^{n} w_k\, f(x_k) \tag{1}$$

where $\rho(x)$, x_k, and w_k are subject to the three following restrictions:

1. The weight function $\rho(x)$ does not change sign in (a,b).

2. All integrals

$$S_k = \int_a^b \rho(x)\, x^k\, dx \tag{2}$$

exist whenever k is a positive integer or zero.

3. The quadrature abscissas x_k and quadrature weights w_k are computed so that equation (1) is exact whenever $f(x)$ is a polynomial of degree $(2n - 1)$ or less.

Given a weight function $\rho(x)$ that satisfies conditions 1 and 2 above, the problem is to find the abscissas and weights that satisfy condition 3. In this paper a numerical procedure for computing these abscissas and weights is described along with a set of computer subprograms written to facilitate implementation of the procedure. The procedure is based upon several well-known properties of orthogonal polynomials. (See sections 10.3 and 10.4 of ref. 1, for example.) These properties are:

1. Gaussian quadrature abscissas x_k are the zeros of polynomials which are orthogonal with respect to the weight function $\rho(x)$.

2. The associated weights w_k can be computed from the abscissas and the orthogonal polynomials.

3. Consecutive orthogonal polynomials are connected by a three-term recursion relation.

4. The recursion coefficients needed to evaluate a polynomial of any degree can be computed from integrals of products of lower degree orthogonal polynomials.

A flow chart of the procedure is presented in figure 1.

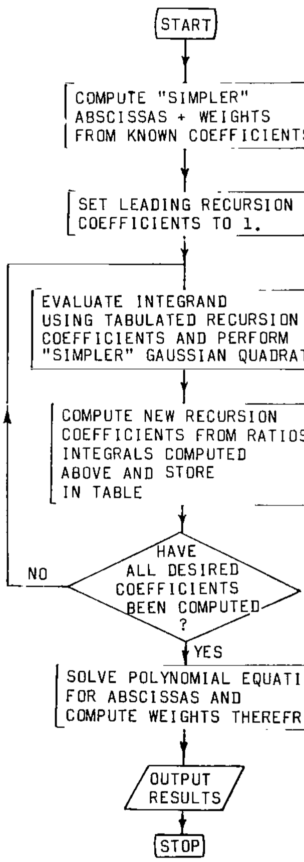

Figure 1.- Flow chart of a method for computing Gaussian abscissas and weights using numerical integration to generate recursion coefficients of orthogonal polynomials.

Inspection of the flow chart shows that the recursion coefficients are computed by numerical integration and then a polynomial root finder is used to compute the abscissas and weights.

The "simpler" Gaussian quadrature mentioned in the fourth box of the flow chart is a closely related quadrature (i.e., its weight function has some of the same singularities) for which the abscissas and weights are already known. Usually it is a classical Gaussian quadrature. It is used to remove as many as possible of the singularities of the weight function, and Taylor's theorem is used to reduce the effect of those that remain. Because all singularities of an arbitrary weight function $\rho(x)$ are either removed by incorporation into the "simpler" quadrature, or else have their effect reduced, this procedure is very accurate. Because Taylor's theorem is used as a method of last resort to reduce the effect of singularities the procedure will be called the "Taylor's theorem method" in the rest of this paper. The mathematical details of Taylor's theorem method are described in the section entitled "Numerical Evaluation of h_n and h_n'."

Relation of Orthogonal Polynomials to Gaussian Quadrature

Suppose that in equation (1) we have a set of abscissas x_k for $k = 1, 2, \ldots, n$ chosen arbitrarily. Let

$$\pi(x) = (x - x_1)(x - x_2) \ldots (x - x_n) \tag{4}$$

Then it can be seen that

$$\ell_k(x) = \frac{\pi(x)}{\pi'(x_k)(x - x_k)} \tag{5}$$

is equal to 1 if $x = x_k$ and is equal to zero if $x = x_1, x_2, \ldots,$ or x_n (excluding x_k). This means that the polynomial

$$L(x) = \sum_{k=1}^{n} f(x_k)\, \ell_k(x) \tag{6}$$

coincides with $f(x)$ at $x = x_k$. The function $L(x)$ is called the Lagrange interpolation polynomial. If $L(x)$ is substituted for $f(x)$ in equation (1) an expression for w_k is obtained

$$w_k = \int_a^b \rho(x) \frac{\pi(x)}{\pi'(x_k)(x - x_k)} dx \tag{7}$$

This expression is valid whether the x_k are Gaussian abscissas or are chosen arbitrarily.

Now suppose that $f(x)$ is a polynomial of degree $(2n - 1)$. If $f(x)$ is divided by $\pi(x)$, a quotient $q(x)$ and a remainder $r(x)$ both of degree $(n - 1)$ are obtained. Thus

$$f(x) = q(x)\,\pi(x) + r(x) \tag{8}$$

If this is inserted into equation (1) the result is

$$\int_a^b \rho(x)\, q(x)\, \pi(x)\, dx + \int_a^b \rho(x)\, r(x)\, dx \approx \sum_{k=1}^{n} w_k\, q(x_k)\, \pi(x_k) + \sum_{k=1}^{n} w_k\, r(x_k) \tag{9}$$

The first sum above is zero because $\pi(x_k) = 0$ by equation (4). The approximation is exact for $f(x)$, an arbitrary polynomial of degree $(2n - 1)$, only if the first integral is also zero. This will occur if $\pi(x)$ is orthogonal to all polynomials $q(x)$ of degree $(n - 1)$ or less.

The abscissas and weights of a Gaussian quadrature formula can be obtained by constructing the sequence of polynomials $p_n(x)$ such that

$$\int_a^b \rho(x)\, p_n(x)\, p_m(x)\, dx = \delta_{nm} h_n \tag{10}$$

and then computing the zeros x_k of $p_n(x)$ and computing

$$w_k = \int_a^b \frac{\rho(x)\, p_n(x)}{p_n'(x_k)(x - x_k)} dx \tag{11}$$

Recursion Formulas and Christoffel-Darboux Identity

A set of polynomials $p_n(x)$, $n = 0, 1, 2, 3, \ldots$, for which equation (10) is true is called an orthogonal system with respect to the weight function $\rho(x)$ over the interval (a,b). Equation (10) itself is not sufficient to define the polynomials $p_n(x)$ uniquely. If h_n is not specified p_n may have an arbitrary factor and if h_n is specified the sign of p_n is ambiguous. By specifying, in addition to equation (10), the coefficient k_n of the highest power term in $p_n(x)$, p_n can be defined unambiguously. This is called the standardization of the system of polynomials. Except for the classical polynomials the standardization adopted herein is $k_n = 1$. For this standardization the three-term recursion relation for a system of orthogonal polynomials is

$$\left.\begin{aligned} & p_0(x) = 1 \\ & p_1(x) = x - b_1 \\ & p_n(x) = (x - b_n)\, p_{n-1}(x) - c_n\, p_{n-2}(x) \qquad (n \geqq 1) \end{aligned}\right\} \tag{12}$$

where

$$\left.\begin{aligned} b_n &= \frac{h'_{n-1}}{h_{n-1}} \\ c_n &= \frac{h_{n-1}}{h_{n-2}} \end{aligned}\right\} \tag{13}$$

and where

$$h_n = \int_a^b \rho(x)\, p_n{}^2(x)\, dx \tag{14}$$

$$h'_n = \int_a^b \rho(x)\, x\, p_n{}^2(x)\, dx \tag{15}$$

The expressions for b_n and c_n are obtained by multiplying equations (12) through by $\rho(x)\, p_{n-1}(x)$ and $\rho(x)\, p_{n-2}(x)$, respectively, and integrating from a to b. The same procedure using $\rho(x)\, p_m(x)$, $m = 0, 1, \ldots, n - 3$, furnishes an inductive proof that the sequence of polynomials generated by equations (12) is orthogonal.

The integrals h_n and h'_n are computed numerically and used to generate a table of recursion coefficients b_k, c_k. If equations (12) are differentiated, recursion formulas for p'_n and $p_n^{(m)}$ are obtained as follows:

$$p'_0(x) = 0 \tag{16a}$$

$$p_1'(x) = 1 \tag{16b}$$

$$p_n'(x) = (x - b_n)\, p_{n-1}'(x) - c_n\, p_{n-2}'(x) + p_{n-1}(x) \tag{16c}$$

$$\left.\begin{aligned}
&\frac{1}{m!}\, p_0^{(m)}(x) = 0 \\
&\frac{1}{m!}\, p_1^{(m)}(x) = 0 \\
&\frac{1}{m!}\, p_n^{(m)}(x) = (x - b_n)\,\frac{p_{n-1}^{(m)}(x)}{m!} - c_n\,\frac{p_{n-2}^{(m)}(x)}{m!} + \frac{p_{n-1}^{(m-1)}(x)}{(m-1)!}
\end{aligned}\right\} \tag{17}$$

The quadrature abscissas x_k are obtained by solving the polynomial equation $p_n(x) = 0$ using Laguerre's iteration formula as described in appendix A. Laguerre's iteration formula requires values of $p_n(x)$, $p_n'(x)$, and $p_n''(x)$ and these are furnished by equations (12), (16), and (17).

After the abscissas have been computed, the weights are computed from

$$w_k = \frac{h_{n-1}}{p_n'(x_k)\, p_{n-1}(x_k)} \tag{18}$$

This is obtained from equation (11) as follows. Let

$$F_n(x,y) = \frac{p_n(x)\, p_{n-1}(y) - p_{n-1}(x)\, p_n(y)}{(x - y)h_{n-1}} \tag{19}$$

If equations (12) are used to eliminate $p_n(x)$ and $p_n(y)$, one obtains

$$F_n(x,y) = \frac{1}{h_{n-1}}\, p_{n-1}(x)\, p_{n-1}(y) + F_{n-1}(x,y) \tag{20}$$

Repeating the process (n - 1) times leads to the sum

$$F_n(x,y) = \sum_{m=0}^{n-1} \frac{1}{h_m}\, p_m(x)\, p_m(y) \tag{21}$$

Equations (19) and (21) are a form of the Christoffel-Darboux identity for orthogonal polynomials (see eq. 10.3 (10) of ref. 1 for the more usual form). If y is replaced by x_k, a zero of $p_n(x)$, the result is

$$\frac{p_n(x)}{x - x_k} = \frac{h_{n-1}}{p_{n-1}(x_k)} \sum_{m=0}^{n-1} \frac{1}{h_m} p_m(x)\, p_m(x_k) \tag{22}$$

Substituting this into equation (11) gives equation (18). Observe that all terms of the sum except the $m = 0$ term integrate to zero because of the orthogonality of the polynomials $p_m(x)$.

Numerical Evaluation of h_n and h'_n

The quantities h_n and h'_n used to compute the recursion coefficients are obtained by a combination of numerical and closed-form integration. The first step is to evaluate the first few moments of the weight function analytically. That is,

$$S_\ell = \int_a^b \rho(x)\, x^\ell \, dx \tag{23}$$

is computed for $\ell = 0, 1, 2, \ldots, L$. The integer L is the order of the singularity extraction (to be described later) and is usually 2 or 3. Then the first few pairs of recursion coefficients b_k and c_k are computed from these moments using equations (13), (14), and (15)

$$\left.\begin{aligned}
h_0 &= S_0 & h'_0 &= S_1 \\
b_1 &= \frac{h'_0}{h_0} & h_1 &= S_2 - 2b_1S_1 + b_1{}^2S_0 \\
c_2 &= \frac{h_1}{h_0} & h'_1 &= S_3 - 2b_1S_2 + b_1{}^2S_1 \\
b_2 &= \frac{h'_1}{h_1} & &\ldots
\end{aligned}\right\} \tag{24}$$

This furnishes the entries up to $k = L - 1$ in a table of recursion coefficients b_k, c_k. Each time a pair of integrals h_k, h'_k is computed a new pair of entries $b_{k+1} = h'_k/h_k$ and $c_{k+1} = h_k/h_{k-1}$ is added to this table. As a consequence, each time an h_k or h'_k integral is to be computed, all the recursion coefficients needed to evaluate the integrand are available.

Either h_n or h'_n is obtained by evaluating the integral

$$I = \int_a^b \rho(x)\ g(x)\ dx \tag{25}$$

numerically where $g(x) = p_n^2(x)$ if h_n is being evaluated and $g(x) = x\ p_n^2(x)$ if h'_n is being evaluated. Since the objective of this report is to describe a procedure for computing quadrature abscissas and weights for a weight function $\rho(x)$ that has a certain set of singular points, particular attention is paid to these singular points when equation (25) is integrated. Let $\rho(x)$ be factored into two functions

$$\rho(x) = \sigma(x)\ \tau(x) \tag{26}$$

where τ is chosen so that (1) $\tau(x)$ contains as many singularities of $\rho(x)$ as possible, and (2) $\tau(x)$ is a weight function over (a,b) whose Gaussian quadrature abscissas and weights are relatively easy to compute. Occasionally it will be necessary to let $\tau(x) = 1$.

After $\rho(x)$ has been factored the Gaussian quadrature abscissas x_k and weights w_k associated with the numerical integration

$$I = \int_a^b \tau(x)\ f(x)\ dx \approx \sum_{k=1}^{N} w_k\ f(x_k) \tag{27}$$

are computed. The quadrature order N is usually taken to be about five times the maximum order desired for the weight function $\rho(x)$.

Reduction of effect of singularities in $\sigma(x)$.- If $\sigma(x)$ has no singularities then h_n and h'_n are evaluated using equation (27) with $f(x) = \sigma(x)\ g(x)$. If $\sigma(x)$ has one or more singularities (the usual case) the strength of these singularities can be reduced by integrating the leading terms of Taylor's expansion about these singularities in closed form. Before describing the method in detail an example will be discussed showing the motivation for the method.

Consider the four integrals

$$I_1 = \int_{-1}^{1} \left(1 - x^2\right)^{-1/2} dx = \pi = 3.141\ 592\ 65$$

$$I_2 = \int_{-1}^{1} \left(1 - x^2\right)^{1/2} dx = \frac{\pi}{2} = 1.570\ 796\ 33$$

$$I_3 = \int_{-1}^{1} \left(1 - x^2\right)^{3/2} dx = \frac{3\pi}{8} = 1.178\ 097\ 25$$

and

$$I_4 = \int_{-1}^{1} \left(1 - x^2\right)^{5/2} dx = \frac{5\pi}{16} = 0.981\ 747\ 70$$

If these integrals are approximated by eight-point Legendre-Gauss quadratures (i.e., eq. (1) with $n = 8$ and $\rho(x) = 1$) then

$G_8(I_1) = 2.936\ 842\ 06$ for a -6.5-percent error

$G_8(I_2) = 1.572\ 152\ 22$ for a 0.087-percent error

$G_8(I_3) = 1.178\ 033\ 49$ for a -0.005-percent error

and

$G_8(I_4) = 0.981\ 755\ 58$ for a 0.0008-percent error

The reason for the difference in accuracy is the order of the singularities at ± 1. The integrand of I_1 is infinite at ± 1 while I_4 merely has an infinite third derivative. This is so because for each singularity x_s in the integrand of I_1 the integrand of I_4 has $\left(x - x_s\right)^3$ as a factor.

Now consider the integral I of equation (25). It can be written

$$I = \int_a^b \tau(x)\ \sigma(x)\ g(x)\ dx \tag{28}$$

It is assumed for now that $\sigma(x)$ has only one singularity at $x = x_s$. The detrimental effect of this singularity on the numerical quadrature of equation (27) is greatly reduced if $g(x)$ is replaced by something that has a power of $(x - x_s)$ as a factor. By Taylor's theorem

$$g(x) = g_L(x) + R_L(x) \tag{29}$$

where

$$g_L(x) = \sum_{\ell=0}^{L} \frac{(x - x_s)^\ell}{\ell!} g^{(\ell)}(x_s) \tag{30}$$

is the leading part of the Taylor's expansion of $g(x)$ about x_s, the singularity of $\sigma(x)$ and

$$R_L(x) = g(x) - g_L(x) \tag{31}$$

is the remainder. Note that $R_L(x)$ has $(x - x_s)^{L+1}$ as a factor. Equation (28) can be expressed

$$I = \int_a^b \rho(x)\, g_L(x)\, dx + \int_a^b \tau(x)\, \sigma(x)\, R_L(x)\, dx \tag{32}$$

The first integral can be evaluated analytically

$$I_1 = \int_a^b \rho(x)\, g_L(x)\, dx = \sum_{\ell=0}^{L} \frac{g^{(\ell)}(x_s)}{\ell!} \overline{S}_\ell \tag{33}$$

where the shifted moments $\overline{S}_\ell$ are obtained from the moments S_ℓ about the origin that were previously computed (see eq. (23))

$$\overline{S}_\ell = \sum_{m=0}^{\ell} (-1)^m \binom{\ell}{m} S_{\ell-m} x_s{}^m \tag{34}$$

The second integral in equation (32) is evaluated numerically using equation (27)

$$I_2 \approx \sum_{k=1}^{N} w_k \, \sigma(x_k) \left[g(x_k) - g_L(x_k) \right] \tag{35}$$

Equations (33) and (35) can be combined to give

$$I \approx \sum_{k=1}^{N} w_k \, \sigma(x_k) \, g(x_k) + \sum_{\ell=0}^{L} \frac{1}{\ell !} C_\ell \, g^{(\ell)}(x_s) \tag{36}$$

where

$$C_\ell = \bar{S}_\ell - \sum_{k=1}^{N} w_k \, \sigma(x_k) \, (x_k - x_s)^\ell \tag{37}$$

is independent of $g(x)$. This is potentially a very accurate way to approximate an integral, but care has to be exercised in choosing L. If L is larger than the degree of $g(x)$ the quadrature is exact; that is, there is no truncation error.[1] In this case it is merely a scheme for expressing I as a linear combination of the moments and this method is known to be ill-conditioned (i.e., uncontrolled growth of round-off error[1]). Each correction coefficient C_ℓ is the difference between the shifted moment

$$\bar{S}_\ell = \int_a^b \rho(x) \, (x - x_s)^\ell \, dx$$

and a Gaussian approximation to the shifted moment. As either ℓ or N becomes large, C_ℓ will become very small. Since C_ℓ is the difference of two quantities that are not approaching zero, eventually for some ℓ, C_ℓ will have no significant figures at all. Thus, L should be chosen so that this does not happen. When this method is used, a plot of $\log |C_\ell|$ against ℓ gives points that lie approximately on a straight line when C_ℓ has significant figures and lie above the line when C_ℓ has no significant figures. Typical values for L are 2 or 3.

---

[1]Truncation error is the error resulting from using insufficient terms in a limiting process, such as a series summation or a quadrature, while round-off error is the error resulting from performing mathematical operations with finite-length computer words.

If $p_n^2(x)$ and $x\, p_n^2(x)$ are substituted for $g(x)$ in equation (36), the following numerical integration formulas for h_n and h_n' are obtained:

$$\left.\begin{aligned} h_n &\approx \sum_{k=1}^{N} w_k\, \sigma(x_k)\, p_n^2(x_k) + \sum_{\ell=0}^{L} C_\ell\, q_\ell(x_s) \\ h_n' &\approx \sum_{k=1}^{N} w_k\, \sigma(x_k)\, x_k\, p_n^2(x_k) + x_s \sum_{\ell=0}^{L} C_\ell\, q_\ell(x_s) + \sum_{\ell=1}^{L} C_\ell\, q_{\ell-1}(x_s) \end{aligned}\right\} \tag{38}$$

where

$$q_\ell(x) = \frac{1}{\ell!}\frac{d^\ell}{dx^\ell} p_n^2(x) = \sum_{m=0}^{\ell} \frac{p_n^{(m)}(x)}{m!}\frac{p^{(\ell-m)}(x)}{(\ell - m)!} \tag{39}$$

In equations (28) to (35) it was assumed that $\sigma(x)$ had only a single singular point x_s. If $\sigma(s)$ has a second singularity then $R_L(x) = g(x) - g_L(x)$ must be expressed as a truncated Taylor's series plus remainder expanded about the second singularity.

Simpler quadratures.- An essential step in the evaluation of h_n and h_n' for a particular weight function $\rho(x)$ is the computation of the abscissas and weights associated with a simpler weight function $\tau(x)$ where $\tau(x)$ is a factor of $\rho(x)$. This so-called "simpler quadrature" that is indicated in equation (27) must have abscissas and weights that are computable without the necessity of evaluating their orthogonal polynomial recursion coefficients by nonexact numerical quadratures. Simpler quadratures can be classified as either classical or almost classical Gaussian quadratures depending upon the nature of their associated orthogonal polynomials.

The classical orthogonal polynomials are the sytems of orthogonal polynomials that can be generated from a generalized Rodrigues' formula

$$p_n(x) = \frac{1}{K_n\, \rho(x)} \cdot \frac{d^n}{dx^n}\left[\rho(x)\, Q^n(x)\right] \tag{40}$$

where K_n is a constant and $Q(x)$ is a polynomial in x that is independent of n. It can be shown (see section 10.7 of ref. 1, for example) that the only zeros of $Q(x)$ are the limits of integration in equation (1) and hence $Q(x)$ must be of degree 2, 1, or 0. The associated orthogonal polynomials are, except possibly for a linear change of scale, the classical Jacobi, generalized Laguerre, and Hermite polynomials, respectively. If equation (40) is substituted for one of the $p_n(x)$ factors in equations (14) or (15) the integral can be evaluated in closed form by integration by parts. This furnishes closed form expressions for the recursion coefficients (see section 22.7 of ref. 2, for example). The classical orthogonal polynomials also each satisfy a second-order differential equation and a first-order differential relation with respect to degree (see sections 22.6 and 22.8 of ref. 2). This makes it possible to compute $p_n'(x)$ and $p_n''(x)$ directly from the recursion formula for $p_n(x)$. Separate recursion relations such as equations (16) and (17) are not needed for classical polynomials. The fact that p_n' and p_n'' are obtained free for the classical polynomials is the motivation for using Laguerre iteration for computing quadrature abscissas instead of the QR algorithm as in references 3, 4, and 5.

The almost classical orthogonal polynomials are those for which equations (14) and (15) can be evaluated as exact classical Gaussian quadratures. A Gaussian quadrature

$$I = \int_a^b \rho(x)\ f(x)\ dx \approx \sum_{k=1}^{n} w_k\ f(x_k) \tag{41}$$

is exact (i.e., no truncation error) if $f(x)$ is a polynomial of degree $(2n - 1)$ or less. Inspection of equations (14) and (15) shows that h_n and h_n' can be evaluated by exact quadratures if $\rho(x)$ can be expressed as a classical weight function times a polynomial or if (a,b) can be partitioned into a set of abutting intervals such that within each interval $\rho(x)$ can be expressed as the product of the classical weight function for that interval times a polynomial. Sometimes it is possible to replace $\rho(x)$ by its integral definition and perform a multidimensional exact quadrature. For example, consider the weight function $\ln(1/x)$ over interval (0,1). Then

$$h_n' = \int_0^1 \left(\ln \frac{1}{x}\right) x\ p_n^{2}(x)\ dx \tag{42}$$

In this case $\rho(x)$ has the integral definition

$$\ln \frac{1}{x} = \int_x^1 \frac{dv}{v} \tag{43}$$

so

$$h_n' = \int_0^1 \int_x^1 \frac{1}{v} x\, p_n^2(x)\; dv\; dx = \int_0^1 \int_0^v \frac{1}{v} x\, p_n^2(x)\; dx\; dv \tag{44}$$

The substitution $x = uv$ gives

$$h_n' = \int_0^1 \int_0^1 uv\; p_n^2(uv)\; du\; dv \tag{45}$$

This can be evaluated as an exact Gaussian quadrature

$$h_n' = \sum_{k=1}^{n+1} w_k\, x_k \sum_{m=1}^{n+1} w_m x_m\, \rho_n^2(x_k x_m) \tag{46}$$

where x_k, x_m, w_k, w_m are the abscissas and weights associated with the $(n + 1)$-point classical Gaussian quadrature with weight function $\rho(x) = 1$ over $(0,1)$. Several of these multidimensional exact quadratures are considered in reference 6.

The purpose of the simpler quadrature is to minimize the number of singularities of $\rho(x) = \sigma(x)\tau(x)$ that have to be treated by Taylor's theorem. The simpler quadrature is the quadrature associated with the factor $\tau(x)$ as a weight function and should be either a classical or almost classical Gaussian quadrature.

PROGRAM ORGANIZATION

The FORTRAN computer program to implement the Taylor's theorem method of calculating Gaussian quadrature abscissas and weights consists of a user-written calling program and a furnished subprogram package. The user-written calling program handles input/output (I/O) and that portion of the computing task that is peculiar to the weight function being processed. The subprogram package handles the portion of the programming task that is common to all weight functions. The complexity of the user-written calling program depends upon the nature of the weight function and the I/O services desired. For a classical weight function it could be as simple as a call to subroutine CGAUSS followed by a print statement whereas for a nonclassical weight function with several singularities in $\rho(x)$ it could be very complicated.

The subprogram package consists of four subprograms that are called by the user plus eight other subprograms. The former are:

CGAUSS a subroutine to compute classical Gaussian abscissas and weights for an arbitrary interval and arbitrary weight function exponents

PNDER a subroutine to compute $\frac{1}{m!} p_n^{(m)}(x)$ for $m = 0,M$ from the recursion formulas (12), (16), and (17)

PNFUN a function to compute $p_n(x)$. This is an abridged version of PNDER that executes faster when only $p_n^{(0)}(x)$ is needed

NGAUSS a subroutine to compute abscissas and weights for a nonclassical Gaussian quadrature

For computing nonclassical Gaussian quadratures, the user-written calling program has two tasks to perform. The first and more difficult task is to compute the recursion coefficients b_n and c_n required by NGAUSS. Specifically this requires FORTRAN instructions to:

1. Implement equations (23), (24), and (34).

2. Compute the abscissas and weights required by equation (27) (a call to CGAUSS).

3. Implement equation (37).

4. Compute as many b_n, c_n as needed from equations (38), (39), and (13). Note that PNFUN and PNDER are used when implementing equations (38) and (39).

The second and simpler task, to compute the desired nonclassical Gaussian abscissas and weights, is accomplished by simply a call to NGAUSS. The user-written calling program also manages two labeled COMMON blocks BOFN and COFN that contain the recursion coefficients b_n and c_n that are given by equations (13). These coefficients are used by PNDER, PNFUN, and NGAUSS. NGAUSS also requires h_0 (see eq. (14)) and since c_1 is not used, h_0 is passed as the first word of COFN. The easiest way to describe in detail how a calling program is written is to explain on a step-by-step basis how the program was written for the sample problem.

A Sample Problem Illustrating How Nonclassical Abscissas and Weights are Computed

The sample problem is a FORTRAN program, calling CGAUSS, PNFUN, PNDER, and NGAUSS, to compute nonclassical Gaussian quadrature abscissas and weights to evaluate

$$I = \int_0^1 (1 - x)^{1/2} \ln \frac{1}{x} f(x)\, dx \approx \sum_{k=1}^{n} w_k f(x_k) \tag{47}$$

for values of n from 8 to 96 in increments of 8. The weight function $(1 - x)^{1/2} \ln (1/x)$ occurs when integrating the chordwise pressure distribution over an aircraft control surface in subsonic flow. The coordinates have been rescaled so that $x = 0$ is the location of the hinge line and $x = 1$ is the control-surface trailing edge.

One of the purposes of the user-written calling program (henceforth referred to as program SAMPLE) is to handle input/output. It is possible for SAMPLE to have no input (i.e., all parameters are built in). However, a program of this sort should be fairly general, hence, as many parameters as possible are read in as input.

In the problem statement min(n), max(n), and Δn were specified as 8, 96, and 8, respectively. These are read in as N1, N2, and N3. Similarly, the L appearing in equation (36) is read in as LMAX, the number of correction coefficients C_ℓ computed and printed, and also as LX, the number of correction terms used in equation (39). The value of N appearing in equation (27) is read in as NC. The total number of pairs of recursion coefficients to be computed is read in as NMAX.

Instead of writing the program to process the actual weight function specified in equation (47), it is written to process the more general function

$$\rho(x) = (1 - x)^{\alpha - 1} \ln \frac{1}{x} \tag{48}$$

and α is read in using the program variable name ALPHA. A summary of the program input is as follows:

IFLAG	if 0, compute b_n and c_n; if 1, read b_n and c_n
ALPHA	α
UA	$\gamma + \psi(\alpha)$
LMAX	number of correction coefficients computed
LX	number of correction coefficients used
NC	order of classical Gaussian quadrature

NMAX number of b_n, c_n pairs to be computed or read

N1,N2,N3 delimiters of loop that calls NGAUSS

Before program SAMPLE can be written, expressions for S_ℓ, the moments of $\rho(x)$, must be derived and $\rho(x)$ must be factored into $\sigma(x)$ and $\tau(x)$

$$S_\ell = \int_0^1 (1 - x)^{\alpha - 1} \ln \frac{1}{x} x^\ell \, dx \tag{49}$$

To evaluate this integral, let $x = 1 - u$ and note that

$$\ln \frac{1}{1 - u} = \int_0^u \frac{dv}{1 - v} \tag{50}$$

Then

$$S_\ell = \int_0^1 \int_0^u \frac{u^{\alpha - 1}(1 - u)^\ell}{1 - v} \, dv \, du = \int_0^1 \int_v^1 \frac{u^{\alpha - 1}(1 - u)^\ell}{1 - v} \, du \, dv$$

$$= \sum_{m=0}^{\ell} (-1)^m \binom{\ell}{m} \int_0^1 \int_v^1 \frac{u^{\alpha - 1 + m}}{1 - v} \, du \, dv = \sum_{m=0}^{\ell} (-1)^m \binom{\ell}{m} \frac{U(\alpha, m)}{\alpha + m} \tag{51}$$

where

$$U(\alpha, m) = \int_0^1 \frac{1 - v^{\alpha + m}}{1 - v} \, dv \tag{52}$$

This can be expressed as a digamma function (see eq. 6.3.22 of ref. 2)

$$U(\alpha, m) = \gamma + \psi(\alpha + 1 + m) \tag{53}$$

which can be written (eq. 6.3.6 of ref. 2)

$$U(\alpha, m) = \gamma + \psi(\alpha) + \sum_{n=0}^{m} \frac{1}{\alpha + n} \tag{54}$$

Equation (54) suggests that $U(\alpha,m)$ should be computed recursively

$$\left.\begin{aligned} U(\alpha,-1) &= \gamma + \psi(\alpha) \\ U(\alpha,m) &= U(\alpha,m-1) + \frac{1}{\alpha + m} \end{aligned}\right\} \tag{55}$$

Instead of having the program compute $\psi(\alpha)$, the quantity $U(\alpha,-1) = \gamma + \psi(\alpha)$ is read in with the variable name of UA. This is done because the digamma function takes on a simpler form when its argument is an integer or half integer. Some values of $U(\alpha,-1)$ are

$$\left.\begin{aligned} U(1/2,-1) &= -2 \ln 2 = -1.386\ 294\ 361\ 119\ 890\ 618\ 834\ 464\ 243 \\ U(1,-1) &= 0 \\ U(3/2,-1) &= 2 - 2 \ln 2 = 0.613\ 705\ 638\ 880\ 109\ 381\ 165\ 535\ 757 \\ U(2,-1) &= 1 \\ U(5/2,-1) &= \frac{8}{3} - 2 \ln 2 = 1.208\ 372\ 305\ 546\ 776\ 047\ 832\ 202\ 424 \\ U(3,-1) &= \frac{3}{2} \end{aligned}\right\} \tag{56}$$

The weight function $\rho(x)$ factors into $(1 - x)^{\alpha-1}$ and $\ln (1/x)$. Either could be used for the simpler quadrature of equation (27) because $(1 - x)^{\alpha-1}$ is classical and $\ln (1/x)$ is almost classical. The choice

$$\sigma(x) = \ln \frac{1}{x} \tag{57}$$

$$\tau(x) = (1 - x)^{\alpha-1} \tag{58}$$

leads to a simpler program.

All the information needed to write the program is now available. Figure 2 is a flow chart of the program. Figure 3 is an expanded version of the box of the figure 2 flow chart that contains the b_n and c_n calculations for $n > 3$. The program was written from the flow chart for the CDC 6000 series FORTRAN "RUN" compiler. The program listing follows along with a listing of subroutine COMPARE that was used for accuracy estimation when ALPHA = 1. Usage description of the subroutine package is given in appendix B.

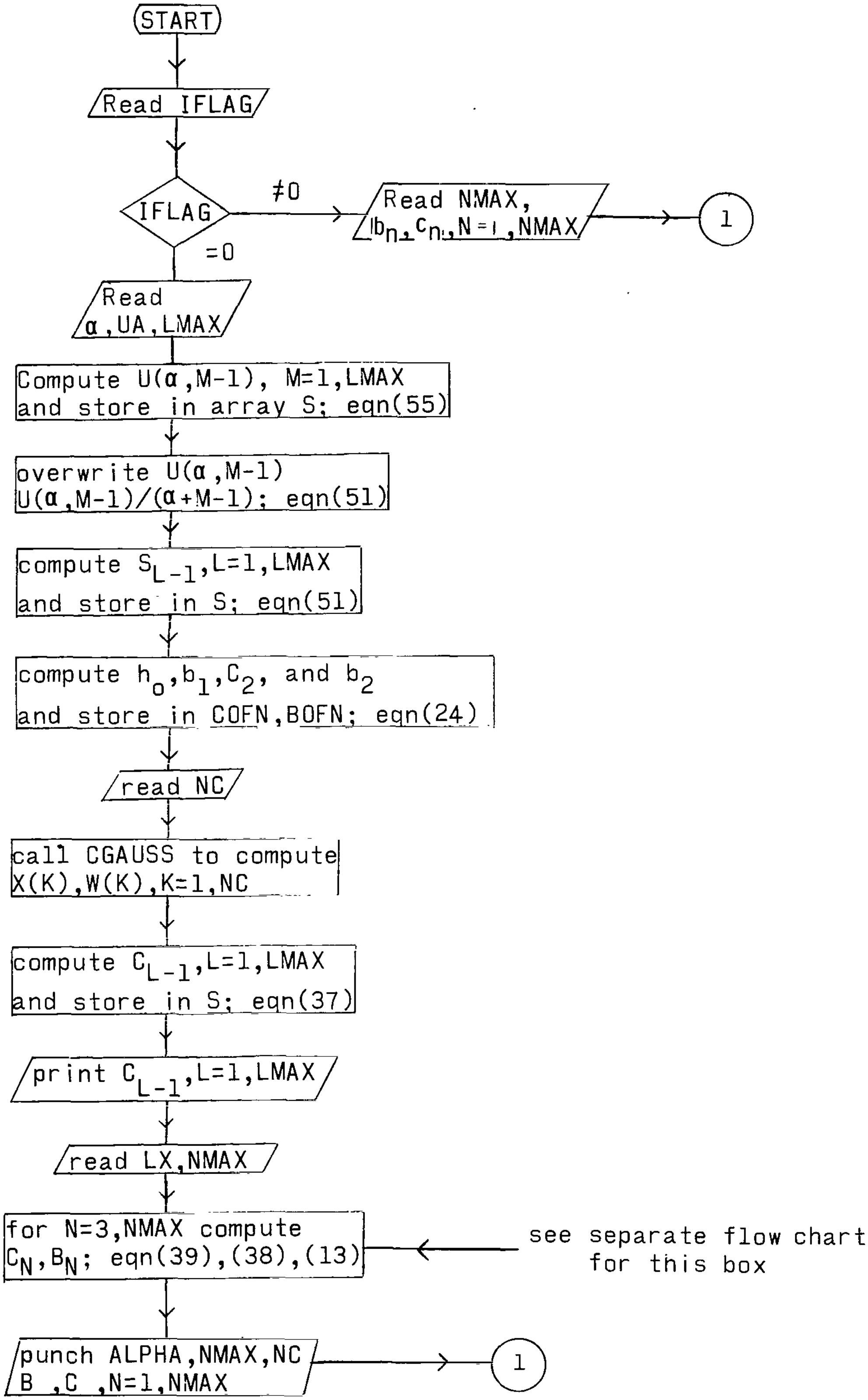

Figure 2.- Program flow chart.

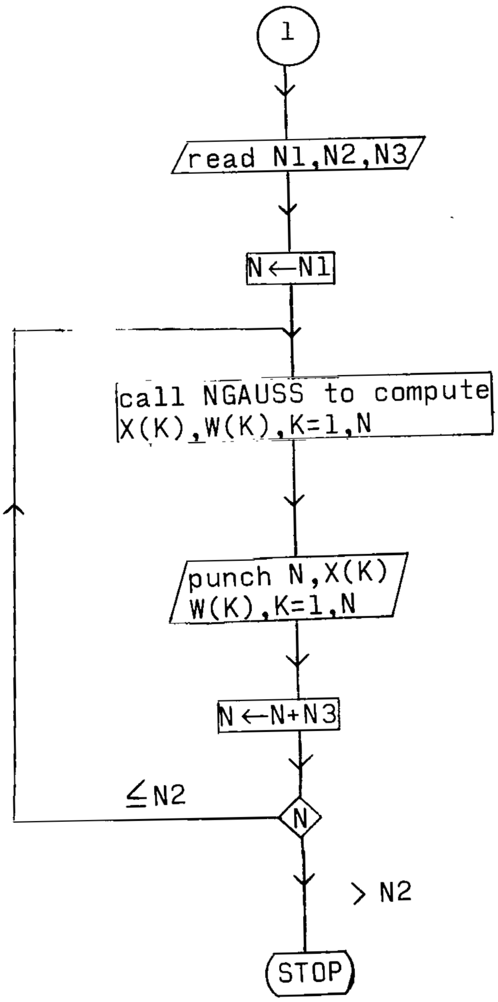

Figure 2.- Concluded.

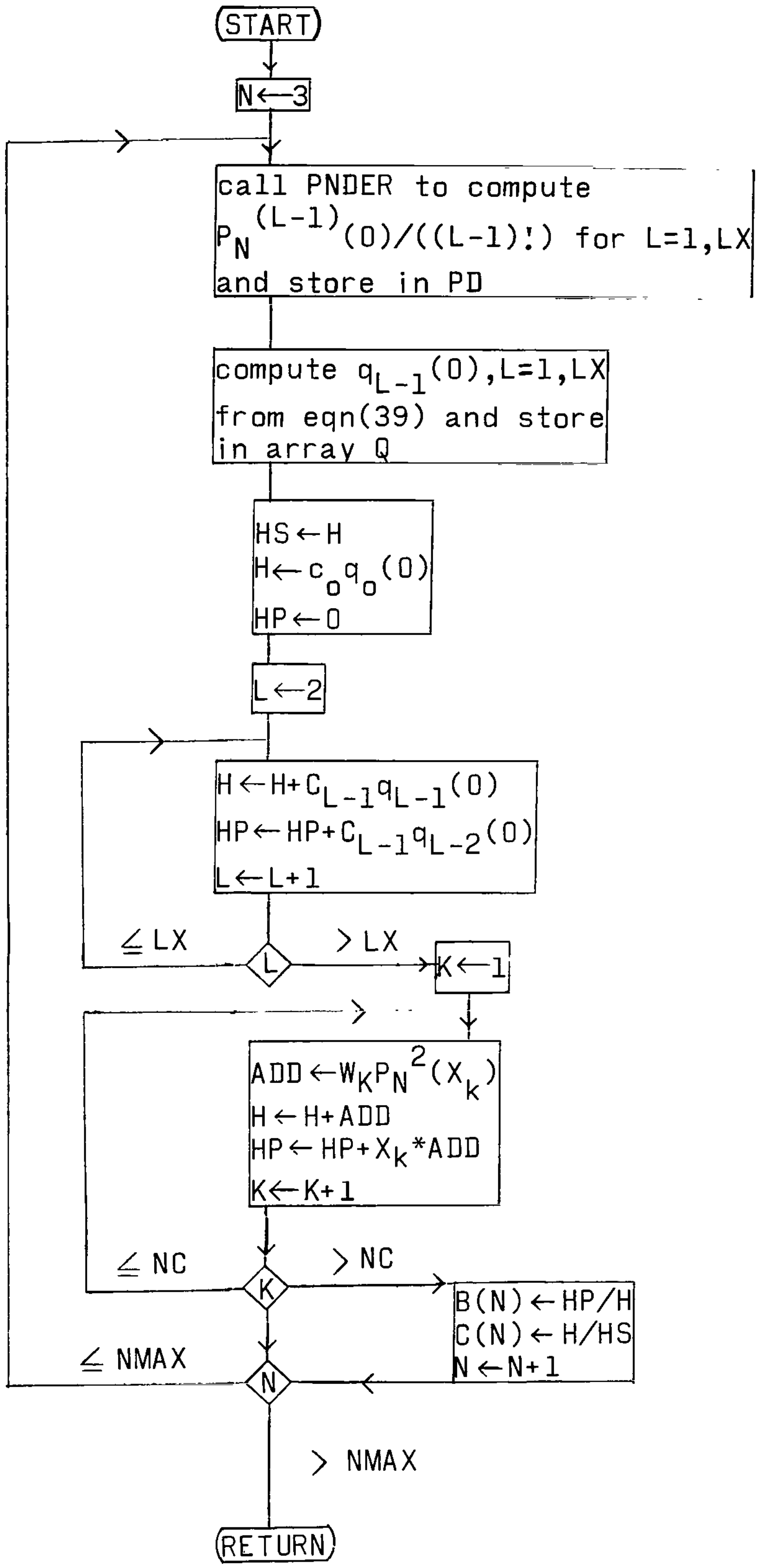

Figure 3.- Recursion-coefficient flow chart.

Sample-Problem Program Listing

```
      PROGRAM SAMPLE(INPUT=1     ,OUTPUT=1     ,PUNCH=1
     +                ,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7=PUNCH)
      COMMON//X(2000),W(2000)
      DOUBLE X,W
      COMMON/BCFN/B(100)/CCFN/C(100)
      DOUBLE B,C
      DOUBLE S(10),PD(10),PS(10),Q(10)
      DOUBLE ALPHA,UA,H,HP,HS,ADD,FAC
      DOUBLE DLOG,PNFUN
      READ 101, IFLAG
      IF(IFLAG.NE.0) GO TO 12
*   IF IFLAG.NE.0 THE RECURSION COEFFICIENTS ARE READ IN.
*
*   PART 1  -  RECURSION COEFFICIENT CALCULATION.

      READ 102, ALPHA,UA
      READ 101, LMAX
COMPUTE U(ALPHA,M-1) AND STORE IN S(M) USING EQN (55)
      S(1)=UA+1./ALPHA
      DO 1 M=2,LMAX
    1 S(M)=S(M-1)+1./(M-1+ALPHA)
      DO 2 M=1,LMAX
    2 S(M)=S(M)/(M-1+ALPHA)
      SGN=(-1.)**(LMAX-1)
      DO 4 LL=2,LMAX
COMPUTE S(L) FROM EQN (51).
      L=2+LMAX-LL
      FAC=1.
      DO 3 M=2,L
      FAC=-FAC*(1.+L-M)/(M-1.)
    3 S(L)=S(L)+FAC*S(1+L-M)
      S(L)=SGN*S(L)
    4 SGN=-SGN
COMPUTE H0,B1,C2,AND B2 FROM EQN (24).  NOTE THAT H0 IS STORED IN C(1)
      C(1)=S(1)
      B(1)=S(2)/C(1)
      H=S(3)-B(1)*(2.*S(2)-B(1)*S(1))
      C(2)=H/C(1)
      B(2)=(S(4)-B(1)*(2.*S(3)-B(1)*S(2)))/H
CALL CGAUSS TO SET UP SIMPLER GAUSSIAN QUADRATURE
      READ 101, NC
      CALL CGAUSS(NC,X,W,0.D,1.D,ALPHA-1.D,0.D)
COMPUTE C(L-1) FROM EQN (37) AND STORE IN S(L).
      DO 5 K=1,NC
      ADD=W(K)*DLOG(X(K))
      S(1)=S(1)+ADD
      DO 5 L=2,LMAX
      ADD=ADD*X(K)
    5 S(L)=S(L)+ADD
      PRINT 201, ALPHA,NC
      M=0
      DO 6 L=1,LMAX
      CLA=ALOG10(ABS(SNGL(S(L))))
      PRINT 202, M,S(L),M,CLA
    6 M=M+1
      READ 101, LX,NMAX
COMPUTE RECURSION COEFFICIENTS FOR N=3 THRU NMAX.
      DO 10 N=3,NMAX
      CALL PNDER(N-1,LX,0.D,PD,PS)
```

```
COMPUTE Q FROM EQN (39).
      DO 7 L=1,LX
      Q(L)=0.
      DO 7 M=1,L
    7 Q(L)=Q(L)+PD(M)*PD(L+1-M)
      HS=H
      H=S(1)*Q(1)
      HP=0.
COMPUTE TAYLOR'S SERIES CORRECTION PART OF H,HP INTEGRALS IN EQN (38).
      DO 8 L=2,LX
      H=H+S(L)*Q(L)
    8 HP=HP+S(L)*Q(L-1)
      DO 9 K=1,NC
COMPUTE GAUSSIAN PART OF H,HP INTEGRALS IN EQN (38).
      ADD=-DLOG(X(K))*W(K)*PNFUN(X(K),N-1)**2
      H=H+ADD
    9 HP=HP+ADD*X(K)
      B(N)=HP/H
   10 C(N)=H/HS
      PUNCH 301, ALPHA,LX,NC,NMAX
      DO 11 N=1,NMAX
   11 PUNCH 302, N,B(N),C(N)
      GO TO 14

*  PART 2  -  QUADRATURE ABSCISSA AND WEIGHT CALCULATION.

   12 READ 301, ALPHA,LX,NC,NMAX
      DO 13 N=1,NMAX
   13 READ 302, N,B(N),C(N)
*  THIS PART OF THE PROGRAM COMPUTES THE ABSCISSAS AND WEIGHTS.
*  IT IS WEIGHT FUNCTION INDEPENDENT.
   14 READ 101, N1,N2,N3
      IF(N2.LE.0) STOP 2
      IF(N3.LE.0) L3=1
      PUNCH 303, ALPHA,N1,N2,N3
      DO 15 N=N1,N2,N3
      PUNCH 304, N
      CALL NGAUSS(N,X,W)
      DO 15 M=1,N
   15 PUNCH 302, M,X(M),W(M)
      IF(ALPHA.EQ.1..AND.NMAX.GE.16) CALL COMPARE
      STOP 1

  101 FORMAT(3I4)
  102 FORMAT(2D40.30)
  201 FORMAT(//36H CORRECTION COEFFICIENTS FOR ALPHA =,D38.28/
     + 9H AND NC =,I5//)
  202 FORMAT(4H  C(,I1,3H) =,D38.28,8X12HLOG10(ABS(C(,I1,5H))) =,F8.2)
  301 FORMAT(//80H RECURSION COEFFICIENTS FOR THE WEIGHT FUNCTION LN(1./
     +X)*(1.-X)**(ALPHA-1.) OVER/22H THE INTERVAL (0.,1.).//6X7HALPHA =,
     +D36.28/9X4HLX =,I4/9X4HNC =,I5/7X6HLMAX =,I4//3X1HN,
     +22X4HB(N),34X4HC(N)/)
  302 FORMAT(I4,2D38.28)
  303 FORMAT(//80H  ABSCISSAS AND WEIGHTS FOR THE WEIGHT FUNCTION LN(1./
     +X)*(1.-X)**(ALPHA-1.) OVER/34H  THE INTERVAL (0.,1.) FOR ALPHA =,
     +D36.28,4H AND/9H  FOR N =I3,9H THRU N =I3,17H IN INCREMENTS OF,I3)
  304 FORMAT(///38X3HN =,I3//3X1HM,22X4HX(M),34X4HW(M)/)
      END              PROGRAM SAMPLE
```

```
      SUBROUTINE CGAUSS(N,X,W,A,B,ALPHA,BETA)
COMPUTES ABSCISSAS AND WEIGHTS FOR CLASSICAL GUASSIAN QUADRATURE
      DOUBLE X(1),W(1),A,B,ALPHA,BETA
      DOUBLE C0,C1,C2,DSQRT
CASE 1. (JACOBI)
    1 IF(BETA.LE.-1.) GO TO 3
      IF(ALPHA.LE.-1.) GO TO 5
      CALL JGAUSS(N,X,W,ALPHA,BETA)
      C0=.5*(B+A)
      C1=.5*(B-A)
      C2=C1**(ALPHA+BETA+1.)
      DO 2 I=1,N
      W(I)=C2*W(I)
    2 X(I)=C0+C1*X(I)
      RETURN
CASE 2. (LAGUERRE)
    3 IF(ALPHA.LE.-1.) GO TO 5
      CALL LGAUSS(N,X,W,ALPHA)
      C1=1./B
      C2=C1**(ALPHA+1.)*DEXP(-B*A)
      DO 4 I=1,N
      W(I)=C2*W(I)
    4 X(I)=A+C1*X(I)
      RETURN
CASE 3. (HERMITE)
    5 NU=N/2
      NN=NU+NU
      N1=N+1
      IF(N.GT.NN) GO TO 8
      CALL LGAUSS(NU,X,W,-.5D0)
      DO 6 MU=1,NU
      X(NU+MU)=DSQRT(X(MU))
    6 W(NU+MU)=.5*W(MU)
      DO 7 MU=1,NU
      X(MU)=-X(N1-MU)
    7 W(MU)=W(N1-MU)
      GO TO 13
    8 IF(NU.EQ.0) GO TO 12
      CALL LGAUSS(NU,X,W,.5D0)
      NU1=NU+1
      DO 9 MU=1,NU
      X(NU1+MU)=DSQRT(X(MU))
    9 W(NU1+MU)=.5*W(MU)/X(MU)
      DO 10 MU=1,NU
      X(MU)=-X(N1-MU)
   10 W(MU)=W(N1-MU)
      X(NU1)=0.
      W(NU1)=1.7724538509055160272981674833D
      C1=2.
      C2=3.
      DO 11 MU=1,NU
      W(NU1)=W(NU1)*C1/C2
      C1=C1+2.
   11 C2=C2+2.
      GO TO 13
   12 X=0.
      W=1.7724538509055160272981674833D
   13 C1=1./DSQRT(A)
      C0=.5*B/A
      C2=DEXP(A*C0*C0)
      DO 14 M=1,N
      X(M)=C1*X(M)-C0
   14 W(M)=C2*W(M)
      RETURN
      END              SUBROUTINE CGAUSS
```

```
      SUBROUTINE JGAUSS(N,X,W,ALPHA,BETA)
COMPUTES JACOBI-GAUSS ABSCISSAS + WEIGHTS FOR ALPHA,BETA.GT.-1.
      DOUBLE X(1),W(1),ALPHA,BETA
      DOUBLE XC,P,P1,CON,A,B,C,D,A1,B1,A2,B2,AB2,S,DSQRT
      CALL JWTCON(CON,N,ALPHA,BETA)
      IF(N-2)9,8,1
    1 RN=N
      AMB=ALPHA-BETA
      AB=ALPHA+BETA
      ABN=AB+RN+1.
      AN=ALPHA+RN
      BN=BETA+RN
      C31=C21=AMB
      C12=-RN
      C11=C21*RN/(AN+BN)
      C13=2.*AN*BN/(AN+BN)
      C22=AB+2.
      C23=-RN*ABN
      C32=C22+2.
      C33=C22+C23
      XC=-1.+2.*(BETA+1.)/(ABN+(RN-1.)*SQRT(AN*ABN/(BETA+2.)))
      DO 2 K=1,23
      CALL JRECUR(N,ALPHA,BETA,XC,P,P1)
      PS=P
      PS1=P1
      XS=XC
      RX=1./(1.-XS*XS)
      PD=((C11+C12*XS)*PS+C13*PS1)*RX
      P2D=((C21+C22*XS)*PD+C23*PS)*RX
      Q=PS/PD
      V=1.-Q*P2D/PD
      H=-RN*Q/(1.+SQRT((RN-1.)*(RN*V-1.)))
      XC=XC+H
      IF(K.LT.3) GO TO 2
      IF(H.LT.1.E-24.OR.XC.EQ.XS) GO TO 3
    2 CONTINUE
    3 CONTINUE
      CALL JRECUR(N,ALPHA,BETA,XC,P,P1)
      X(1)=XC
      W(1)=CON*(1.-XC*XC)/P1/P1
      AS=BS=0.
      DO 7 M=2,N
      P3D=((C31+C32*XS)*P2D+C33*PD)*RX
      Q=.5*P2D/PD
      U=Q-AS
      V=(Q*Q-P3D/PD/3.-BS)/(U*U)
      RM=N-M
      R1=RM+1.
      M1=M-1
      XC=XC-R1/U/(1.+SQRT(RM*(R1*V-1.)))
      DO 5 K=1,23
      CALL JRECUR(N,ALPHA,BETA,XC,P,P1)
      PS=P
      PS1=P1
      XS=XC
      RX=1./(1.-XS*XS)
      PD=((C11+C12*XS)*PS+C13*PS1)*RX
      P2D=((C21+C22*XS)*PD+C23*PS)*RX
      AS=BS=0.
      DO 4 I=1,M1
      DS=1./(XS-X(I))
      AS=AS+DS
    4 BS=BS+DS*DS
```

```
      Q=PS/PD
      U=1.-AS*Q
      V=(1.-Q*P2D/PD-BS*Q*Q)/(U*U)
      H=-R1*Q/U/(1.+SQRT(RM*(R1*V-1.)))
      XC=XC+H
      IF(K.LT.3) GO TO 5
      IF(H.LT.1.E-24.OR.XC.EQ.XS) GO TO 6
    5 CONTINUE
    6 CONTINUE
      CALL JRECUR(N,ALPHA,BETA,XC,P,P1)
      X(M)=XC
      W(M)=CON*(1.-XC*XC)/P1/P1
    7 CONTINUE
      RETURN
    8 A1=ALPHA+1.
      A=A1+1.
      B1=BETA+1.
      B=B1+1.
      C=A+B-1.
      S=DSQRT(A*B*C)
      D=4.*CON*(A*B+S)
      X=2.*B*B1/(B*C+S)-1.
      W=D*B/(B1*(S+B)**2)
      X(2)=1.-2.*A*A1/(A*C+S)
      W(2)=D*A/(A1*(S+A)**2)
      RETURN
    9 IF(N.LT.1)RETURN
      A1=ALPHA+1.
      B1=BETA+1.
      AB2=A1+B1
      A2=A1+1.
      B2=B1+1.
      X=(BETA-ALPHA)/AB2
      W=CON*(1.-X*X)
      W=W*2.*AB2*(1./A1+1./B1)**2/(A2*B2*(1./A2+1./B2)**2)
      RETURN
      END              SUBROUTINE JGAUSS
```

```
      SUBROUTINE JRECUR(N,ALPHA,BETA,X,P,P1)
C COMPUTES NTH AND (N-1)TH JACOBI POLYNOMIALS
      DOUBLE ALPHA,BETA,X,P,P1,P2,AM,BM,CM,DM,G,C0,C2
      AM=ALPHA
      BM=BETA
      P2=DM=1.
      C0=AM+BM
      G=AM-BM
      CM=C0+1.
      C2=CM+1.
      P1=.5*(G+C2*X)
      G=G*C0
      DO 1 M=2,N
      AM=AM+1.
      BM=BM+1.
      DM=DM+1.
      C0=C2
      C2=C2+2.
      CM=CM+1.
      P=(.5*(C0+1.)*(G+C0*C2*X)*P1-AM*BM*C2*P2)/(DM*CM*C0)
      P2=P1
    1 P1=P
      P1=P2
      RETURN
      END               SUBROUTINE JRECUR


      SUBROUTINE JWTCON(C,N,A,B)
      DOUBLE C,A,B,AM,BM,ABM,DM,DGAMF
C COMPUTES THE CONSTANT PART OF THE JACOBI-GAUSS QUADRATURE WEIGHT.
      AM=A+1.D
      BM=B+1.D
      ABM=AM+BM-1.
      DM=1.
      C=.25*2.D**ABM*DGAMF(AM)*DGAMF(BM)*AM*BM/(DGAMF(ABM)*ABM)
      DO 1 M=2,N
      AM=AM+1.
      BM=BM+1.
      ABM=ABM+1.
      DM=DM+1.
    1 C=C*AM/DM*BM/ABM
      C=C*(1./AM+1./BM)**2
      RETURN
      END               SUBROUTINE JWTCON
```

```
      SUBROUTINE LGAUSS(N,X,W,ALPHA)
C     COMPUTES LAGUERRE-GAUSS ABSCISSAS + WEIGHTS FOR ALPHA.GT.-1.
      DOUBLE X(1),W(1),ALPHA
      DOUBLE XC,P,P1,CON,A1,A2,DSQRT
      CALL LNTCON(CON,N,ALPHA)
      IF(N-2)9,3,1
    1 XC=(1.+ALPHA)/(1.+(N-1.)/SQRT(2.+ALPHA))
      RN=N
      AN=ALPHA+RN
      DO 2 K=1,23
      CALL LRECUR(N,ALPHA,XC,P,P1)
      PS=P
      PS1=P1
      XS=XC
      PD=(RN*PS-AN*PS1)/XS
      P2D=((XS-1.-ALPHA)*PD-RN*PS)/XS
      Q=PS/PD
      V=1.-Q*P2D/PD
      H=-RN*Q/(1.+SQRT((RN-1.)*(RN*V-1.)))
      XC=XC+H
      IF(K.LT.3) GO TO 2
      IF(H.LT.1.E-24.OR.XC.EQ.XS) GO TO 3
    2 CONTINUE
    3 CONTINUE
      CALL LRECUR(N,ALPHA,XC,P,P1)
      X(1)=XC
      W(1)=CON*XC/P1/P1
      A=B=0.
      DO 7 M=2,N
      P3D=((XS-2.-ALPHA)*P2D-(RN-1.)*PD)/XS
      Q=.5*P2D/PD
      U=Q-A
      V=(Q*Q-P3D/PD/3.-B)/(U*U)
      RM=N-M
      R1=RM+1.
      M1=M-1
      XC=XC-R1/U/(1.+SQRT(RM*(R1*V-1.)))
      DO 5 K=1,23
      CALL LRECUR(N,ALPHA,XC,P,P1)
      PS=P
      PS1=P1
      XS=XC
      PD=(RN*PS-AN*PS1)/XS
      P2D=((XS-1.-ALPHA)*PD-RN*PS)/XS
      A=B=0.
      DO 4 I=1,M1
      D=1./(XS-X(I))
      A=A+D
    4 B=B+D*D
      Q=PS/PD
      U=1.-A*Q
      V=(1.-Q*P2D/PD-B*Q*Q)/(U*U)
      H=-R1*Q/U/(1.+SQRT(RM*(R1*V-1.)))
      XC=XC+H
      IF(K.LT.3) GO TO 5
      IF(H.LT.1.E-24.OR.XC.EQ.XS) GO TO 6
    5 CONTINUE
    6 CONTINUE
      CALL LRECUR(N,ALPHA,XC,P,P1)
      X(M)=XC
      W(M)=CON*XC/P1/P1
    7 CONTINUE
      RETURN
```

```
    8 A2=ALPHA+2.
      X(2)=A2+DSQRT(A2)
      W(2)=CON*A2/X(2)
      A1=ALPHA+1.
      X(1)=A2*A1/X(2)
      W(1)=CON*X(2)/A1
      RETURN
    9 IF(N.LT.1)RETURN
      X=1.+ALPHA
      W=2.*CON*(X+1.)/X
      RETURN
      END               SUBROUTINE LGAUSS


      SUBROUTINE LRECUR(N,ALPHA,X,P,P1)
      DOUBLE ALPHA,X,P,P1,P2,RM,AM
C COMPUTES NTH AND (N-1)TH LAGUERRE POLYNOMIALS
      RM=1.
      AM=ALPHA
      P2=1.
      P1=RM+AM-X
      DO 1 M=2,N
      RM=RM+1.
      AM=AM+1.
      P=((RM+AM-X)*P1-AM*P2)/RM
      P2=P1
    1 P1=P
      P1=P2
      RETURN
      END               SUBROUTINE LRECUR


      SUBROUTINE LWTCON(C,N,A)
      DOUBLE C,A,AM,DM,DGAMF
C COMPUTES THE CONSTANT PART OF THE LAGUERRE-GAUSS QUADRATURE WEIGHT.
      AM=A+1.D
      C=DGAMF(AM)
      DM=1.
      DO 1 M=2,N
      C=C*AM/DM
      AM=AM+1.
    1 DM=DM+1.
      C=C/(AM*DM)
      RETURN
      END               SUBROUTINE LWTCON
```

```
      DOUBLE FUNCTION DGAMF(X)
      DOUBLE B(7),C(7),DLOG,DEXP,X,F,Z
C     COMPUTES GAMMA FUNCTION FOR REAL ARGUMENT.  DGAMF AND X MUST BE TYPED DOUBLE IN
C     CALLING PROGRAM.  X MUST BE LESS THAN 144 AND NOT A NEGATIVE INTEGER.
      DATA F / 2.50662 82746 31000 50241 57652 84810 D /
      DATA B/1.D,1.D,53.D,195.D,22999.D,29944523.D,109535241009.D/
      DATA C/12.D,30.D,210.D,371.D,22737.D,19733142.D,48264275462.D/
C     CHECK FOR X AN INTEGER.  IF SO GO TO 3.
      N=IDINT(X)
      Z=X-N
      IF(Z.EQ.0.) GO TO 3
C     CHECK FOR X A HALF INTEGER.  IF SO GO TO 5.
      N=IDINT(X-.5)
      Z=X-.5-N
      IF(Z.EQ.0.) GO TO 5
      N=145-X
      IF(N.LT.1)STOP 3777
      Z=X+N
      DGAMF=0
      DO 1 I=1,7
    1 DGAMF=B(8-I)/C(8-I)/(Z+DGAMF)
      DGAMF=F*DEXP(DGAMF-Z+(Z-.5)*DLOG(Z))
      DO 2 I=1,N
    2 DGAMF=DGAMF/(N-I+X)
      RETURN
    3 DGAMF=1.
      Z=0.
      DO 4 I=2,N
      Z=Z+1.
    4 DGAMF=DGAMF*Z
      RETURN
    5 DGAMF=1.772453850905516027298167483D
      IF(N)6,8,9
    6 N=-N
      Z=.5
      DO 7 I=1,N
      Z=Z-1.
    7 DGAMF=DGAMF/Z
    8 RETURN
    9 Z=-.5
      DO 10 I=1,N
      Z=Z+1.
   10 DGAMF=DGAMF*Z
      RETURN
      END              DOUBLE FUNCTION DGAMF
```

```
      SUBROUTINE NGAUSS(N,X,W)
C     COMPUTES ABSCISSAS + WEIGHTS FOR A NON-CLASSICAL GAUSSIAN QUADRATURE
C     CONSTANTS B AND C MUST BE FURNISHED BY USER
      COMMON/BOFN/B(100)/COFN/C(100)
      DOUBLE B,C
      DOUBLE X(1),W(1)
      DOUBLE XC,PN,PN1,PNP,PN2P,HN
      RN=N
      RN1=RN-1.
      R=-B
      RP=0.
      HN=C
      DO 2 M=2,N
      HN=HN*C(M)
      RP=RP-R*B(M)-C(M)
    2 R=R-B(M)
      XC=-(R+SQRT(RN1*(RN1*R*R-2.*RN*RP)))/RN
      DO 3 K=1,23
      CALL PNRECUR(N,XC,PN,PN1,PNP,PN2P)
      XS=XC
      Q=PN/PNP
      V=1.-Q*PN2P/PNP
      DX=-RN*Q/(1.+SQRT(RN1*(RN*V-1.)))
      XC=XC+DX
      IF(K.LT.3) GO TO 3
      IF(DX.LT.1.E-24.OR.XC.EQ.XS) GO TO 4
    3 CONTINUE
    4 CONTINUE
      CALL PNRECUR(N,XC,PN,PN1,PNP,PN2P)
      XS=XC
      X(1)=XC
      W(1)=HN/(PN1*PNP)
      AS=BS=0.
      DO 8 M=2,N
      RM=N-M
      R1=RM+1.
      M1=M-1
      XC=XC-PNP/(.5*PN2P-AS*PNP)
      DO 6 K=1,23
      CALL PNRECUR(N,XC,PN,PN1,PNP,PN2P)
      XS=XC
      AS=BS=0.
      DO 5 I=1,M1
      D=1./(XS-X(I))
      AS=AS+D
    5 BS=BS+D*D
      Q=PN/PNP
      U=1.-AS*Q
      V=(1.-Q*PN2P/PNP-BS*Q*Q)/(U*U)
      DX=-R1*Q/U/(1.+SQRT(RM*(R1*V-1.)))
      XC=XC+DX
      IF(K.LT.3) GO TO 6
      IF(DX.LT.1.E-24.OR.XC.EQ.XS) GO TO 7
    6 CONTINUE
    7 CONTINUE
      CALL PNRECUR(N,XC,PN,PN1,PNP,PN2P)
      XS=XC
      X(M)=XC
      W(M)=HN/(PN1*PNP)
    8 CONTINUE
      RETURN
      END              SUBROUTINE NGAUSS
```

```
      DOUBLE FUNCTION PNFUN(X,N)
      DOUBLE X,P1,PS
      COMMON/BOFN/B(100)/COFN/C(100)
      DOUBLE B,C
      IF(N-1)4,3,1
    1 P1=1.
      PNFUN=X-B(1)
      DO 2 M=2,N
      PS=PNFUN
      PNFUN=(X-B(M))*PNFUN-C(M)*P1
    2 P1=PS
      RETURN
    3 PNFUN=X-B(1)
      RETURN
    4 PNFUN=1.
      RETURN
      END              DOUBLE FUNCTION PNFUN


      SUBROUTINE PNRECUR(N,X,PN,PN1,PNP,PN2P)
      COMMON/BOFN/B(100)/COFN/C(100)
      DOUBLE B,C
      DOUBLE X,PN,PN1,PNP,PN2P,XB,PNP1,PN2P1,PS,PSP,PS2P
      PNP1=PN2P1=PN2P=0.
      PN1=PNP=1.
      PN=X-B
      IF(N.LT.2) RETURN
      DO 1 M=2,N
      PS=PN
      PSP=PNP
      PS2P=PN2P
      XB=X-B(M)
      PN2P=XB*PN2P-C(M)*PN2P1+2.*PNP
      PN2P1=PS2P
      PNP=XB*PNP-C(M)*PNP1+PN
      PNP1=PSP
      PN=XB*PN-C(M)*PN1
    1 PN1=PS
      RETURN
      END              SUBROUTINE PNRECUR


      SUBROUTINE PNDER(N,M,X,PD,PS)
      DOUBLE X,PD(1),PS(1),XB,S
      COMMON/BOFN/B(100)/COFN/C(100)
      DOUBLE B,C
*   COMPUTES (D/DX)**K*P(X)/(K-FACTORIAL) FOR K=0 THRU M-1.  RESULTS FOR
*   POLYNOMIAL OF DEGREE N ARE STORED IN PD.  RESULTS FOR POLYNOMIAL OF
*   DEGREE N-1 ARE STORED IN PS.  M MUST BE AT LEAST 2.   PD,PS MUST BE
*   DIMENSIONED M OR LARGER IN CALLING PROGRAM.  N MUST BE AT LEAST 2.
      PD(1)=X-B(1)
      DO 1 K=2,M
      PS(K)=0.
    1 PD(K)=0.
      PD(2)=PS(1)=1.
      DO 2 L=2,N
      XB=X-B(L)
      S=PD(1)
      PD(1)=XB*S-C(L)*PS(1)
      PS(1)=S
      DO 2 K=2,M
      S=PD(K)
      PD(K)=XB*S-C(L)*PS(K)+PS(K-1)
    2 PS(K)=S
      RETURN
      END              SUBROUTINE PNDER
```

```
      SUBROUTINE COMPARE
C COMPARES 2ND THRU 16TH RECURSION COEFFICIENTS WITH THOSE TABULATED IN REF. 5.
      COMMON/BOFN/B(100)/COFN/C(100)
      DOUBLE B,C
      DOUBLE BS(16),CS(16),ERROR
      READ 10, BS(1),CS(1)
      IF(EOF,5)4,1
    1 BMAX=CMAX=0.
      MB=MC=0
      DO 3 M=2,16
      READ 10, BS(M),CS(M)
      TEST=DABS(BS(M)-B(M))
      IF(TEST.LE.BMAX) GO TO 2
      BMAX=TEST
      MB=M
    2 TEST=DABS(CS(M)-C(M))
      IF(TEST.LE.CMAX) GO TO 3
      CMAX=TEST
      MC=M
    3 CONTINUE
      ERROR=B(MB)-BS(MB)
      PRINT 11, MB,B(MB),MB,BS(MB),ERROR
      ERROR=C(MC)-CS(MC)
      PRINT 12, MC,C(MC),MC,CS(MC),ERROR
    4 RETURN
   10 FORMAT(2D40.30)
   11 FORMAT(//4H  B(,I2,3H) =,D36.28/
     +          4H BS(,I2,3H) =,D36.28/
     +               9H   ERROR =,D36.28/)
   12 FORMAT(//4H  C(,I2,3H) =,D36.28/
     +          4H CS(,I2,3H) =,D36.28/
     +               9H   ERROR =,D36.28/)
      END            SUBROUTINE COMPARE
```

Sample-Problem Program Input

For the case ALPHA = 1 the program input consists of six data cards containing the following parameters with formats included in parentheses:

IFLAG(9I4)

ALPHA, UA(2D40.30)

LMAX(I4)

NC(I4)

LX,NMAX(2I4)

N1,N2,N3(3I4)

The input to the sample case is listed below as card images.

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
--------------------------------------------------------------------------------

   0
    1.5                                  .613705638880109381165535757
   6
1400
   3  96
   8  96   6

--------------------------------------------------------------------------------
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
```

Sample-Problem Program Output

The sample program output consists of a line printer listing of the correction coefficients C_ℓ (see eq. (37)), the recursion coefficients b_n and c_n (including h_0 stored in c_1) punched on cards, and the requested quadrature abscissas and weights punched on cards. Tables I, II, and III were copied from the sample-case output files. The quantities output in tables I, II, and III are presented using a 29-significant-figure format because this is the double precision word length of the computer used. However, the numbers in tables II and III are only accurate to about 18 significant figures for the sample case.

TABLE I.- CORRECTION COEFFICIENTS AND THEIR LOGARITHMS

```
CORRECTION COEFFICIENTS FOR ALPHA =     1.5000000000000000000000000000D+00
AND NC = 1400

 C(0) =     3.2178903515575434569121109401D-07          LOG10(ABS(C(0))) =   -6.49
 C(1) =    -6.5232304204402282805693427485D-14          LOG10(ABS(C(1))) =  -13.19
 C(2) =     4.3818710617832521634049440467D-20          LOG10(ABS(C(2))) =  -19.36
 C(3) =    -5.9711526938260432205944689451D-26          LOG10(ABS(C(3))) =  -25.22
 C(4) =     3.8869914451771161117338755670D-28          LOG10(ABS(C(4))) =  -27.41
 C(5) =     2.3640622065185838856285580445D-28          LOG10(ABS(C(5))) =  -27.63
```

TABLE II. - RECURSION COEFFICIENTS

RECURSION COEFFICIENTS FOR THE WEIGHT FUNCTION LN(1./X)*(1.-X)**(ALPHA-1.) OVER THE INTERVAL (0.,1.).

ALPHA = 1.5000000000000000000000000000D+00
LX = 3
NC = 1400
LMAX = 96

N	B(N)	C(N)
1	2.1255452108712293503099269304D-01	8.5358153703118403188813494907D-01
2	4.3334311136112682898924144053D-01	3.8026122530947720344701403922D-02
3	4.7051645047184396971938340374D-01	5.3551511959857113571457154721D-02
4	4.8336562327826096605258877946D-01	5.7910733387526000979077283180D-02
5	4.8931347348281783264233531732D-01	5.9718866657356819770940030996D-02
6	4.9255292321316419525029341257D-01	6.0637358642310478703429982289D-02
7	4.9451218089515113286052437470D-01	6.1166488109233090920787700382D-02
8	4.9578753649295292627049727074D-01	6.1498705488386840725921590225D-02
9	4.9656424171033999765794949129D-01	6.1720783414353921696078515232D-02
10	4.9729283539227180280595770293D-01	6.1876494618355942833607046355D-02
11	4.9775890966420902445615265462D-01	6.1989853939519003416118653936D-02
12	4.9811406844518439188740146099D-01	6.2074925438185495395598995425D-02
13	4.9839093449774137555188619974D-01	6.2140387899955068114083510535D-02
14	4.9861095491703230400137180014D-01	6.2191831598819920083393798731D-02
15	4.9878870157596104738035897477D-01	6.2232988738673364503403854157D-02
16	4.9893435703835665349951249392D-01	6.2266427904292420487961180683D-02
17	4.9905521127970136206190808913D-01	6.2293963609323191074354856724D-02
18	4.9915659491364374213025110730D-01	6.2316907071732131119014913345D-02
19	4.9924247902902940536748188328D-01	6.2336224843649369097506219061D-02
20	4.9931587115813006223515655340D-01	6.2352642046586715325929462781D-02
21	4.9937908290611996726949434675D-01	6.2366711261444330010048711556D-02
22	4.9943391468100770776609994280D-01	6.2378359535058023306583447830D-02
23	4.9948178563177697275323013998D-01	6.2389421100397452473130013097D-02
24	4.9952382661374846604322333060D-01	6.2398660564633160938707310711D-02
25	4.9956094773664592773637420379D-01	6.2406789610964784964546322105D-02
26	4.9959388813948912862806667185D-01	6.2413979207490811371440096218D-02
27	4.9962325314387665001341925282D-01	6.2420368652937784509594453649D-02
28	4.9964954231558850647949756348D-01	6.2426072362171671990433527527D-02
29	4.9967317089242228291445441105D-01	6.2431185014511386487032300355D-02
30	4.9969448631078128828150018703D-01	6.2435785501142332023352614556D-02
31	4.9971378107312016688023638800D-01	6.2439939981369433965305364895D-02
32	4.9973130285315106863838839998D-01	6.2443704270399768880826160157D-02
33	4.9974726249661505719556927683D-01	6.2447125720650221247104433206D-02
34	4.9976184040403966881353463179D-01	6.2450244715717955251850813787D-02
35	4.9977519165896237520455947859D-01	6.2453095865530231851122173726D-02
36	4.9978745017578623809247367204D-01	6.2455708969069476589170541310D-02
37	4.9979873207576102038358586626D-01	6.2458109794924200081428006766D-02
38	4.9980913845102778453171690164D-01	6.2460320718017853786865967781D-02
39	4.9981875764030341895381787674D-01	6.2462361242018872174393224176D-02
40	4.9982766711239804364277732473D-01	6.2464248430297786549727385839D-02
41	4.9983593503295452967834581311D-01	6.2465997263278350296749015105D-02
42	4.9984362157387542322235967200D-01	6.2467620936205389179586284301D-02
43	4.9985078001263161386615647914D-01	6.2469131108417064246001055687D-02

TABLE II. - Concluded

44	4.9985745765912649440695888280D-01	6.2470538112941174826986275798D-02
45	4.9986369664035615961618443146D-01	6.2471851133471039655681982421D-02
46	4.9986953456726770113894459856D-01	6.2473078354395885217520063416D-02
47	4.9987500510360541833560514757D-01	6.2474227088473805976695140007D-02
48	4.9988013845287128906152058211D-01	6.2475303885874933038164908901D-02
49	4.9988496177660096364889219225D-01	6.2476314627637627912453640901D-02
50	4.9988949955480937459630065650D-01	6.2477264606032674627760522176D-02
51	4.9989377389756762436760088574D-01	6.2478158593890033177907882608D-02
52	4.9989780431514004933122011676D-01	6.2479000904587013768041022911D-02
53	4.9990161045286346571182452249D-01	6.2479795444108153235272064972D-02
54	4.9990520729593200508122399448D-01	6.2480545756351947810776343738D-02
55	4.9990861034841548840553704788D-01	6.2481255062667228932082962229D-02
56	4.9991183329015138621996804015D-01	6.2481926296443960929633370305D-02
57	4.9991488361458189677451066481D-01	6.2482562133452959412735231694D-02
58	4.9991778775013614017160834529D-01	6.2483165018521217510789879578D-02
59	4.9992054116736502055348725288D-01	6.2483737189039994039281325288D-02
60	4.9992315847370859874479459109D-01	6.2484280695728211430606752231D-02
61	4.9992564849750128792591479755D-01	6.2484797421011339596871558407D-02
62	4.9992801936258949313706989586D-01	6.2485289095323633629854392620D-02
63	4.9993027855474188206437096827D-01	6.2485757311597590857303531371D-02
64	4.9993243298086812750682539419D-01	6.2486203538167368319637184413D-02
65	4.9993448902192265883757203616D-01	6.2486629130281490928846440129D-02
66	4.9993645258025156134779999864D-01	6.2487035340393530871064961243D-02
67	4.9993832912203987942450177238D-01	6.2487423327376768306121299418D-02
68	4.9994012371543038488052572249D-01	6.2487794164789508205972015934D-02
69	4.9994184106481106792404620989D-01	6.2488148848301196371600414718D-02
70	4.9994348554170524633719116127D-01	6.2488483302375308590197020309D-02
71	4.9994506121264367548885725547D-01	6.2488813386292814357689469465D-02
72	4.9994657186435103360829337449D-01	6.2489124899589535459406566061D-02
73	4.9994802102653853293536859682D-01	6.2489423586971675010737015001D-02
74	4.9994941199255922679815498369D-01	6.2489710142765970704323770844D-02
75	4.9995074783815205363745809521D-01	6.2489985214954147572034956038D-02
76	4.9995203143847411377059290821D-01	6.2490249408835459468102340962D-02
77	4.9995326548359754876470888735D-01	6.2490503290355987186372953530D-02
78	4.9995445249262720831959206865D-01	6.2490747389138896737653432846D-02
79	4.9995559482657763928842986435D-01	6.2490982201245962326382198166D-02
80	4.9995669470013247009268444018D-01	6.2491208191597247203738794111D-02
81	4.9995775419239569635993582524D-01	6.2491425796772845587549509790D-02
82	4.9995877525673244800474015250D-01	6.2491635426117963677218332355D-02
83	4.9995975972978631824115936616D-01	6.2491837464670309082649627920D-02
84	4.9996070933975107583290323049D-01	6.2492032274426724270804490919D-02
85	4.9996162571396640390292255472D-01	6.2492220196064205308459926635D-02
86	4.9996251038590007455485235232D-01	6.2492401550428861611101446381D-02
87	4.9996336480157255992653682416D-01	6.2492576639904969188863816238D-02
88	4.9996419032547439465695133241D-01	6.2492745749675026206587295040D-02
89	4.9996498824602155338209297712D-01	6.2492909148880615817917665857D-02
90	4.9996575978058961279938753414D-01	6.2493067091692900096020483473D-02
91	4.9996650608016346416335518412D-01	6.2493219818300695624445557102D-02
92	4.9996722823363577056102970431D-01	6.2493367555823303026159040347D-02
93	4.9996792727178417320090732342D-01	6.2493510519154568157788744879D-02
94	4.9996860417095439789935994231D-01	6.2493648911744032076108192467D-02
95	4.9996925985647385819938789818D-01	6.2493782926320471630003254383D-02
96	4.9996989520581806115068681387D-01	6.2493912745562635149963302458D-02

TABLE III. - GAUSSIAN QUADRATURE ABSCISSAS AND WEIGHTS

ABSCISSAS AND WEIGHTS FOR THE WEIGHT FUNCTION LN(1./X)*(1.-X)**(ALPHA-1.) OVER THE INTERVAL (0.,1.) FOR ALPHA = 1.5000000000000000000000000000D+00 AND FOR N = 8 THRU N = 96 IN INCREMENTS OF 8

N = 8

M	X(M)	W(M)
1	1.2680038934452116127845578102D-02	1.5718203172778601498140828672D-01
2	7.5901195770415456632420393470D-02	2.2174934527998940036478348976D-01
3	1.8866231663356723694610260401D-01	2.0128072278334840587012852334D-01
4	3.3879530732848935656743047651D-01	1.4376445523233242235254655243D-01
5	5.0893989499144713786895339134D-01	8.2138641547402390020161251237D-02
6	6.7895356659561174978806411200D-01	3.5772063912403149839414969480D-02
7	8.2848285140070974074362424352D-01	1.0361694925489808927536033154D-02
8	9.3953064862774799214139589312D-01	1.3325816224324395321558434828D-03

N = 16

M	X(M)	W(M)
1	3.7931154497795031347003693635D-03	5.9316720162846239790811208232D-02
2	2.2402355507272990123433501393D-02	9.9709207346790417457858225237D-02
3	5.6702376934222478739846813151D-02	1.1677898484684400385996117835D-01
4	1.0577762953498042215155698503D-01	1.1895920937779104557207604350D-01
5	1.6810526904723649206913878011D-01	1.1111112944660206290583566440D-01
6	2.4166468612963175272006932714D-01	9.6954292191577192955701154378D-02
7	3.2402391153231886719359215814D-01	7.9556314928561400911800125768D-02
8	4.1243063815621911363176771648D-01	6.1408754479649004952985110691D-02
9	5.0391082517919737086010831642D-01	4.4405947067203581160790759907D-02
10	5.9537342872155576957932296134D-01	2.9824686325330419186821307812D-02
11	6.8371842052804175344129441880D-01	1.8340297594238718785739815230D-02
12	7.6594472039958407639274637265D-01	1.0089478340615662010911100941D-02
13	8.3925450071199931281222389916D-01	4.7769154760752030585457790630D-03
14	9.0115040384073492835336033946D-01	1.8145329715141382892179200999D-03
15	9.4952265954422595248513739521D-01	4.7734751643063648319020641689D-04
16	9.8272588230703660036913811661D-01	5.7718959114304505889350584866D-05

N = 24

M	X(M)	W(M)
1	1.8109129041480912311338347957D-03	3.1843708366600495720973447686D-02
2	1.0592086167781420931426696979D-02	5.6666264550731142074309748353D-02
3	2.6840154862650659852024292807D-02	7.1164347313575677487923472433D-02
4	5.0387348332726127978694944465D-02	7.8911485480967482681690682830D-02
5	8.0913308049570469234312712105D-02	8.1659621063806899231725401411D-02
6	1.1797244348537204006142260212D-01	8.0589546407418295401727290021D-02
7	1.6100745427106726268779596411D-01	7.6635064038328576368790185995D-02
8	2.0936095615166528372647136150D-01	7.0592178162104779126951710676D-02
9	2.6228735825978483108116517206D-01	6.3157979513461648921556418730D-02
10	3.1896553979399683788109339393D-01	5.4942415964288336261926701622D-02
11	3.7851241984859998066273127386D-01	4.6469305184875872793587298035D-02
12	4.3999734151559705387315099866D-01	3.8173866989293556008659246112D-02
13	5.0245711161400927984076253368D-01	3.0400372783239707596441095936D-02
14	5.6491149540549387615207894449D-01	2.3401768859427295483165987277D-02
15	6.2637894218397016704676163383D-01	1.7342180929209863346186734146D-02
16	6.8589230504959468064413531035D-01	1.2302619466507537123303030260D-02
17	7.4251431315965702922665295391D-01	8.2898043543545227636619216740D-03
18	7.9535255572475649807068909948D-01	5.2477380193496300871425885860D-03
19	8.4357374327925045719898480313D-01	3.0714457236965285032794076887D-03
20	8.8541702333153426328560706196D-01	1.6221552546921810047879487697D-03

21	9.2320614631125637314653813709D-01	7.4309906206605649180060968831D-04
22	9.5336031492097733163153739915D-01	2.7508607157359943526767139387D-04
23	9.7640368320836294914295631498D-01	7.1004664131143827409931088107D-05
24	9.9197463190542210508612542422D-01	8.4788074832041458664182470294D-06

N = 32

M	X(M)	W(M)
1	1.0602177045663330035532332155D-03	2.0126261287995058945659571570D-02
2	6.1602969803603404095699585820D-03	3.6889234899458827361728991218D-02
3	1.5598811573879618595402425941D-02	4.7879559230246330271052716801D-02
4	2.9332508913241529202534471329D-02	5.5115006086964382801648084823D-02
5	4.7250733017406964458985445308D-02	5.9523189767906228052687788273D-02
6	6.9237694783201794575186912900D-02	6.1678375706589944825667661038D-02
7	9.5077021933040017853244056565D-02	6.2001520918050557898940110182D-02
8	1.2455483705332485804178011652D-01	6.0834457700744099221916264735D-02
9	1.5741250988072321080195562124D-01	5.8471972646425379630330091277D-02
10	1.9335942543008825554885799405D-01	5.5176621700323007935486880885D-02
11	2.3207588890182335178034764368D-01	5.1185539522380835241171961756D-02
12	2.7321620858844281431300057466D-01	4.6713274809313071239988748542D-02
13	3.1641196346428647921569763923D-01	4.1952622033378019315875104636D-02
14	3.6127544497267180200531896234D-01	3.7074502393770133631689269508D-02
15	4.0740325799524068500134866154D-01	3.2227495824703135170076552064D-02
16	4.5438002022602779866543642270D-01	2.7537384138219062025902078631D-02
17	5.0178223547830532586207483724D-01	2.3106924728368956576905546101D-02
18	5.4918213402095548008467976899D-01	1.9015985756570365293123974019D-02
19	5.9615162146782225153725917786D-01	1.5322113806477334451554664398D-02
20	6.4226619681298972929035542029D-01	1.2061561756551013320056909878D-02
21	6.8710883979699438074462713302D-01	9.2507718694719421051164933403D-03
22	7.3027382783282958030803864779D-01	6.8882835309372753011439507587D-03
23	7.7137044817241348814272237459D-01	4.9570148989754285782454310544D-03
24	8.1002657182654000851081261826D-01	3.4268519782147256718316590226D-03
25	8.4589205695114579310410714675D-01	2.2574667801821795023448052526D-03
26	8.7864195099137758477040970807D-01	1.4012779684294874155304335064D-03
27	9.0797946291195546317216650524D-01	8.06462522265097909903499661753D-04
28	9.3363867965686088290757019619D-01	4.1992533394397186905937326643D-04
29	9.5538700589915445861288607334D-01	1.9013510585851467465287176078D-04
30	9.7302731975958878783699967680D-01	6.9739297765897619586104556124D-05
31	9.8639990427650833430173519931D-01	1.7877883692605131382829340583D-05
32	9.9538486992850268659755531807D-01	2.1251466252817087430297664468D-06

N = 40

M	X(M)	W(M)
1	6.9648670435564147785420902660D-04	1.3983238548631146876681630059D-02
2	4.0273424236905518774639150632D-03	2.6112753246873274965844856982D-02
3	1.0188034692230884321429570888D-02	3.4561983541501604404472748083D-02
4	1.9166323991554972526929466051D-02	4.0636640122706989933701855635D-02
5	3.0922656523929278795903760713D-02	4.4921677864137992853769256711D-02
6	4.5397130750097582071980853631D-02	4.7763670543044737350598673575D-02
7	6.2511727979594879652983620330D-02	4.9403799576561301808954594275D-02
8	8.2171572509482831689963028828D-02	5.0028553301343386081064934578D-02
9	1.0426590385010619017397492676D-01	4.9792767043584246703832996506D-02
10	1.2866896860571371737422088430D-01	4.8831211771049531355149691461D-02
11	1.5524091130103578635001849652D-01	4.7264757264037887076857708446D-02
12	1.8382869792796551699060886983D-01	4.5203717398694385265212208359D-02
13	2.1426708678307009971236367079D-01	4.2749638441321539737681912236D-02
14	2.4637965197247697122901860596D-01	3.9996194924725446734103679030D-02
15	2.7997986009018099705727004671D-01	3.7029568599747925000319548853D-02
16	3.1487219776642535000761549222D-01	3.3928535102264271214359215254D-02
17	3.5085334603969535583498775874D-01	3.0764399135667118576443329260D-02
18	3.8771339635665948761470184073D-01	2.7600869662160752980224183190D-02

TABLE III. - Continued

19	4.2523710221010425506701554251D-01	2.4493936019718656857236761100D-02
20	4.6320515985832985963822634229D-01	2.1491785923068461895835763253D-02
21	5.0139551115987300714850666143D-01	1.8634792612637955099609623974D-02
22	5.3958466126398627462838614247D-01	1.5955588560228129777267468438D-02
23	5.7754900369566701322902178184D-01	1.3479235732400626427203068425D-02
24	6.1506614524916561472535003735D-01	1.1223496633811769357316322278D-02
25	6.5191622304621333649944371882D-01	9.1992057130620524328460073709D-03
26	6.8788320611830736929876736035D-01	7.4107369160991529610876166230D-03
27	7.2275617393181939562618065235D-01	5.8565600355189819910096155139D-03
28	7.5633056438718388424697156104D-01	4.5298759171988980993351053928D-03
29	7.8840938398649111100182415409D-01	3.4193184803520129647647731045D-03
30	8.1880437307564265462390501512D-01	2.5097098409917540908397717581D-03
31	8.4733711932691863076545592614D-01	1.7828535748257371643420857777D-03
32	8.7384011293633670470733337219D-01	1.2183502949714074973249815627D-03
33	8.9815773737334674920556963296D-01	7.9441923766935925003784035213D-04
34	9.2014718995772990344826624219D-01	4.8870943134996236877755536916D-04
35	9.3967932711300061978477539986D-01	2.7908425597977628544546016045D-04
36	9.5663943007542956044828098613D-01	1.4436376336469563338008730618D-04
37	9.7092788899414331778332724165D-01	6.5010004874083094197353393218D-05
38	9.8246081102885567940345379781D-01	2.3741777442017629346230180828D-05
39	9.9117060027858802752073971019D-01	6.0666249976636518438706900986D-06
40	9.9700698088176468250417467434D-01	7.1959256733843981456595479722D-07

N = 48

M	X(M)	W(M)
1	4.9277059499139973926965495315D-04	1.0336044471578385859947260388D-02
2	2.8387829373924688047228504402D-03	1.9557190632688007612400610740D-02
3	7.1747567702533032056032037343D-03	2.6228321628951305065068236277D-02
4	1.3497961984472121803878575729D-02	3.1265801208397929572956685444D-02
5	2.1790547021715745884490970503D-02	3.5075651628137972944077612107D-02
6	3.2024010444108138256308135593D-02	3.7893081063924451805390704171D-02
7	4.4160525785844888404311923605D-02	3.9876758174095759273728002370D-02
8	5.8153609790582355631935846567D-02	4.1145147562252592933842073584D-02
9	7.3948575132903717067255306642D-02	4.1793331331284513686461739862D-02
10	9.1482913369021750681516678289D-02	4.1901728142400573447074768627D-02
11	1.1058664314406989053510520465D-01	4.1540953221011812198094989342D-02
12	1.3148266458366336589542671413D-01	4.0774653696659124916707668740D-02
13	1.5378712065300027469425182510D-01	3.9661203796331356487914885596D-02
14	1.7750977454338895031931229454D-01	3.8254723703428961798662403665D-02
15	2.0255440518217145495442529124D-01	3.6605682178096404765730577771D-02
16	2.2881922180290639559093084663D-01	3.4761237115288603153442507028D-02
17	2.5619729751946731039266210024D-01	3.2765409815575690269754376045D-02
18	2.8457702121679219397424260350D-01	3.0659154891600815172381083622D-02
19	3.1384256664861912899521335848D-01	2.8480367228945078723129863003D-02
20	3.4387437733657928766571094225D-01	2.6263854494639139866201578649D-02
21	3.7454966564969525624433775314D-01	2.4041295227459313975451680737D-02
22	4.0574292428212064231935574514D-01	2.1841196804749157167123306127D-02
23	4.3732644822398385149615057203D-01	1.9688863546514495867081661783D-02
24	4.6917086522569662517144669852D-01	1.7606382282408193519295522353D-02
25	5.0114567268389170101104684478D-01	1.5612630498995262379456206551D-02
26	5.3311977882328144480188534573D-01	1.3723310470345092382950473304D-02
27	5.6496204601054846322818963012D-01	1.1951011405929981031644134582D-02
28	5.9654183401210792820380482982D-01	1.0305300529879196544945596424D-02
29	6.2772954099597931987851495653D-01	8.7928430719619592855965047490D-03
30	6.5839714007818036277359720734D-01	7.4175503617978332861130403080D-03
31	6.8841870922534033851597609849D-01	6.1807545462601256569414911040D-03
32	7.1767095234710067622966136815D-01	5.0814078776117902918419438023D-03
33	7.4603370944389867767498653634D-01	4.1163040344651500965792216683D-03
34	7.7339045371755030676346892937D-01	3.2803185310916151983216833797D-03
35	7.9962877360334683481296462759D-01	2.5666649374652684006255717206D-03
36	8.2464083774290519992005242751D-01	1.9671633689209701715339975814D-03
37	8.4832384098661546541503820067D-01	1.4725175076723289773264132638D-03
38	8.7058042959328595682857467253D-01	1.0725962864447689250697543936D-03

TABLE III. - Continued

39	8.9131910388309179335789686200D-01	7.5671629513371889464404590921D-04
40	9.1045459670006324670133464025D-01	5.1392096269101015719733442111D-04
41	9.2790822615736894084271171898D-01	3.3325261620194936835866929574D-04
42	9.4360822128748185510684524132D-01	2.0401362489613760533472245661D-04
43	9.5749001944536082956731889484D-01	1.1601299584916099535036591615D-04
44	9.6949653477030287598987537779D-01	5.9794997209176431171493226983D-05
45	9.7957839831269285407785704371D-01	2.6846640352276358218066222079D-05
46	9.8769417558035418029948576293D-01	9.7811504717473016763710438952D-06
47	9.9381059659631979371583532761D-01	2.4948914562025326405150199301D-06
48	9.9790312991934381185557293689D-01	2.9558166166953267753253897001D-07

N = 56

M	X(M)	W(M)
1	3.6718163050639638127470486921D-04	7.9818624930081974797180994142D-03
2	2.1089747834105012621252429599D-03	1.5253024604093871302974932393D-02
3	5.3258996102202951125322421536D-03	2.0653065746915682788202194993D-02
4	1.0018436502206925405615961675D-02	2.4861221760266959860891055393D-02
5	1.6177771916174533443450250572D-02	2.8176989094123812769449779359D-02
6	2.3788963099518635587880366840D-02	3.0771805786042607238878078264D-02
7	3.2831481421622457725289757265D-02	3.2759220802238005486722285450D-02
8	4.3280292781865010065039135798D-02	3.4222082441666528401080646298D-02
9	5.5105446664418247274402835909D-02	3.5225232856848359283448069330D-02
10	6.8272605159393337380891207402D-02	3.5822176146839832647515315142D-02
11	8.2743145876530314759427036747D-02	3.6058870465671223229509864291D-02
12	9.8474322781839824534919372057D-02	3.5976006285134634097856754550D-02
13	1.1541942636948654183931360081D-01	3.5610427324561916895809926146D-02
14	1.3352794748688599500072384526D-01	3.4996037922100601528323126943D-02
15	1.5274574714048832361313085843D-01	3.4164389119317535027277446844D-02
16	1.7301523351177666215104177277D-01	3.3145056964368901945404565159D-02
17	1.9427554676860600027586603800D-01	3.1965983153292754086656904182D-02
18	2.1646275185768025329758772755D-01	3.0653123046865871948412491294D-02
19	2.3951003920414353532634996445D-01	2.9231530986244954744210888248D-02
20	2.6334793306756353346040571595D-01	2.7724403386140332575203552673D-02
21	2.8790450717851417228769762367D-01	2.6153593987482290316908085578D-02
22	3.1310560718823962845218085127D-01	2.4539511596553238255359436842D-02
23	3.3887507939453395825129076236D-01	2.2901107863891615838751542232D-02
24	3.6513500515322776306963721497D-01	2.1255860707319089521360588229D-02
25	3.9180594034224714892082672910D-01	1.9619757577578240914437640300D-02
26	4.1830715921132426005699786509D-01	1.8007281724430110260856143967D-02
27	4.4605690192325908637540946837D-01	1.6431403830946736465312597119D-02
28	4.7347262507092569059294585991D-01	1.4903580768803816914799214798D-02
29	5.0097125443713864901262193044D-01	1.3433762737352365982021113261D-02
30	5.2846943925146024419254434107D-01	1.2030409650106364948734370019D-02
31	5.5588380718862005614968355185D-01	1.0700517300741583647395629020D-02
32	5.8313121934713586568219619491D-01	9.4496535606718289141022754820D-03
33	6.1012902444374638756217232818D-01	8.2820046205558747889606825896D-03
34	6.3679531145922378826511626273D-01	7.2004310808567639816335608972D-03
35	6.6304915997389490108228042936D-01	6.2065335163232368104114794053D-03
36	6.8881088743665661484614297774D-01	5.3007269821197617699866931666D-03
37	7.1400229261933142251286358516D-01	4.4823237925698881358613910519D-03
38	7.3854689451879264580953324255D-01	3.7496237851732409830125220433D-03
39	7.6237016598232168585342050797D-01	3.1000111813460617443858247713D-03
40	7.8539976134707352790145067030D-01	2.5300570702425291819755086701D-03
41	8.0756573740225905341721196987D-01	2.0356264723242477604921866316D-03
42	8.2880076700264957408143730049D-01	1.6119888845127100790059091909D-03
43	8.4904034468423321579784461920D-01	1.2539311683404530758905031558D-03
44	8.6822298365730207130017351296D-01	9.5587161611913663440941432142D-04
45	8.8629040357899651492514383783D-01	7.1197401738999891367999170199D-04
46	9.0318770853661987198228926832D-01	5.1626054842511774935900358641D-04
47	9.1886355470549437864269297502D-01	3.6272232088586563691711277206D-04
48	9.3327030718234742317163057666D-01	2.4542645145130708708409859523D-04
49	9.4636418554129716626286343003D-01	1.5861855178209541812058557925D-04
50	9.5810539772570538065908163726D-01	9.6819587001324999757919546827D-05
51	9.6845826200927426354415313862D-01	5.4916110304870900196031547180D-05

52	9.7739131704805934545029230781D-01	2.8242950646792558380316106367D-05
53	9.8487742095386950174432306973D-01	1.2657508903548685810926913705D-05
54	9.9089384407022076523299448780D-01	4.6049046696826821575925290668D-06
55	9.9542238176031397000020848068D-01	1.1733100156257221845854224210D-06
56	9.9844973252025318779540987387D-01	1.3890760403027087650248542992D-07

N = 64

M	X(M)	W(M)
1	2.8427347563685099514317991792D-04	6.3680269801223392775234038021D-03
2	1.6287556462713752488023980591D-03	1.2264501972331148902638341518D-02
3	4.1102383542474003770450668919D-03	1.6729351808704335590544572586D-02
4	7.7302749278709817956903246671D-03	2.0286392766133271294321056523D-02
5	1.2484256428434123755081502879D-02	2.3166113759569423444873714392D-02
6	1.8353674565750194842733499242D-02	2.5499586037494242221519258846D-02
7	2.5356744046159734737540600748D-02	2.7372672995617940421686295231D-02
8	3.3448682895836108942418000454D-02	2.8847096907635342032257151806D-02
9	4.2621876557205412251253532498D-02	2.9970319549788975282471768345D-02
10	5.2855993839187635167433449724D-02	3.0780767739873163796426983803D-02
11	6.4128081237164263709971173624D-02	3.1310834994632127217226675623D-02
12	7.6412647658932249491153507403D-02	3.1588710877302840868756456448D-02
13	8.9681745605557735829951486207D-02	3.1639544949933624807029989039D-02
14	1.0390505206613825023055913763D-01	3.1486210718587667740689540099D-02
15	1.1904995096319303828037537964D-01	3.1149817892805717816529100019D-02
16	1.3503161820663549807761368837D-01	3.0650060390874628290637131083D-02
17	1.5196310996185934056436919412D-01	3.0005453976257448441568497871D-02
18	1.6965545446091945595870100901D-01	2.9233498020694754967805993696D-02
19	1.8811774750895926410888199041D-01	2.8350784215863048378088289439D-02
20	2.0730725172026826457811029560D-01	2.7373067775420072015924655287D-02
21	2.2717949943693061816880720090D-01	2.6315311986494737086669310682D-02
22	2.4768839922475540830332018448D-01	2.5191713873159452681855255922D-02
23	2.6878634579813867004631151009D-01	2.4015716634765313679142065816D-02
24	2.9042433319279646281754636735D-01	2.2800013064866454640858087092D-02
25	3.1255207097981790700556456168D-01	2.1556543122855777077650050943D-02
26	3.3511810329419870533806087363D-01	2.0296488081761294336846856262D-02
27	3.5806993043459206073916897141D-01	1.9030263122221995609897078069D-02
28	3.8135413277755310303990565008D-01	1.7767509825302786373767523615D-02
29	4.0491649673843653339225489690D-01	1.6517089695793335170354274341D-02
30	4.2870214250190486368880862969D-01	1.5287079595873497738976435208D-02
31	4.5265565323741279176217035549D-01	1.4084769767782640903097639715D-02
32	4.7672120550883493806750091301D-01	1.2916664960283250720440991626D-02
33	5.0084270058244228858604427566D-01	1.1788489037974972413446082626D-02
34	5.2496389633359198124384849456D-01	1.0705193338258714405719353780D-02
35	5.4902853944968972240069412867D-01	9.6709689431926876008930961987D-03
36	5.7298049762514914971851779507D-01	8.6892629491583471097338646450D-03
37	5.9676389144315768295510708202D-01	7.7627987436388525167111512677D-03
38	6.2032322563902413111036021011D-01	6.8936002336581543178636202278D-03
39	6.4350351944069726481145201810D-01	6.0830199131795692080683984940D-03
40	6.6655043568367960428915382742D-01	5.3317706060225286320413478696D-03
41	6.8911040839999326002254000477D-01	4.6399606758922396681902618219D-03
42	7.1123076858406352422119147332D-01	4.0071324553896692035276272875D-03
43	7.3285986784235242606086765358D-01	3.4323036109804294905483118203D-03
44	7.5394719963827549704647974145D-01	2.9140111305557012910729245519D-03
45	7.7444351784936457237690013769D-01	2.4503575941953412492468324874D-03
46	7.9430095235975919985737625392D-01	2.0390593668742175824342668942D-03
47	8.1347312141792781189090731633D-01	1.6774963340048483482056638966D-03
48	8.3191524049700133300296440105D-01	1.3627627867729866354259786644D-03
49	8.4958422740324673751940167814D-01	1.0917190541017830858226859798D-03
50	8.6643880338700684390811863131D-01	8.6104347168446817312042609261D-04
51	8.8243959001989510778194763816D-01	6.6728427576538599152388969552D-04
52	8.9754920161219722855991155493D-01	5.0691101013093742228277275815D-04
53	9.1173233295539169267631846411D-01	3.7636503899456057333850521103D-04
54	9.2495584218666868267702586315D-01	2.7210876601264080392574142605D-04
55	9.3718882858601286826558590641D-01	1.9067317042918253240308642305D-04

TABLE III. - Continued

56	9.4840270513227315063259461897D-01	1.2870328518165432275299205182D-04
57	9.5857126566805248963109614192D-01	8.3001258560903909691317682604D-05
58	9.6767074655911223014751790065D-01	5.0566660543212161743235764150D-05
59	9.7567988281291376764389296672D-01	2.8633717027763838848102337627D-05
60	9.8257995882705524193725355781D-01	1.4705179730360677976518554062D-05
61	9.8835485462183122062756832904D-01	6.5825662042194158946572826153D-06
62	9.9299109127975123230562762198D-01	2.3925331716976655959230842891D-06
63	9.9647789593339015528639720766D-01	6.0917685225068114400988086614D-07
64	9.9880747507024059514086574818D-01	7.2086241102130857761448684142D-08

N = 72

M	X(M)	W(M)
1	2.2655850171884081496123660218D-04	5.2102522119183557753518556876D-03
2	1.2959458364226821806065779856D-03	1.0098887237372019563507139938D-02
3	3.2683218458818679092154780337D-03	1.3856703362512563938741001330D-02
4	6.1456228827417311544475245843D-03	1.6900032916353330138555451352D-02
5	9.9253594042296251433688508107D-03	1.9411934652264001929361407329D-02
6	1.4602435009026586243928955047D-02	2.1496120519357276421000026105D-02
7	2.0169609522324743538946168663D-02	2.3220073636978336532479446485D-02
8	2.6617701869298206209211967250D-02	2.4631835210977127451405967279D-02
9	3.3935703931305475077599715915D-02	2.5767898670706446540078308084D-02
10	4.2110856750896782928314047376D-02	2.6657400851632269068008606843D-02
11	5.1128709056365432681865406558D-02	2.7324542955175773169256477226D-02
12	6.0973167178729240586361491518D-02	2.7790077559968008889350452638D-02
13	7.1626540940083093224454811878D-02	2.8072264933707703213338295517D-02
14	8.3069588003756280739339228937D-02	2.8187509794662545629724497887D-02
15	9.5281558115272233538408428878D-02	2.8150796522406917976224401101D-02
16	1.0824023808426950259009935053D-01	2.7975992347217883187703833112D-02
17	1.2192199802311601014892507457D-01	2.7676061333457338967669259916D-02
18	1.3630183915505071352294529066D-01	2.7263216527127410665250251507D-02
19	1.5135344337623321202098149869D-01	2.6749028338337665157857456431D-02
20	1.6704922467160825359985043557D-01	2.6144501431487269319018233736D-02
21	1.8336038242698407287364543632D-01	2.5460128669461290035433503387D-02
22	2.0025695663930034527644597111D-01	2.4705928197778115908549181773D-02
23	2.1770788499778795144914824301D-01	2.3891468090700948051797585442D-02
24	2.3568106178692621181327559024D-01	2.3025881831260063847831249678D-02
25	2.5414339854545127384602686601D-01	2.2117877086186585323722054739D-02
26	2.7306088640264574616401573693D-01	2.1175739654199682027629041917D-02
27	2.9239866000273835342159978600D-01	2.0207334040183228913566676531D-02
28	3.1212106291978320268979959145D-01	1.9220101791133138864825621661D-02
29	3.3219171445840372562130394523D-01	1.8221058490462402745274414671D-02
30	3.5257357772995031402261972522D-01	1.7216790123510025238000678041D-02
31	3.7322902888870239509928123874D-01	1.6213449383778135785881204228D-02
32	3.9411992740857998078477219812D-01	1.5216752375873143220008757387D-02
33	4.1520768727729689373178793162D-01	1.4231976079772183235163203930D-02
34	4.3645334898190029252526993454D-01	1.3263956866418648795232232643D-02
35	4.5781765215713396701543856415D-01	1.2317090292802955506655109631D-02
36	4.7926110876598770716613008538D-01	1.1395332352696725595302456131D-02
37	5.0074407668011527957570136891D-01	1.0502202314928851430507536999D-02
38	5.2222683352649122939076869117D-01	9.6407872429190477179957970778D-03
39	5.4366965066571024153079159507D-01	8.8137482559196110641826178448D-03
40	5.6503286716669516676985208692D-01	8.0233285631572251169981696393D-03
41	5.8627696364225747723459689737D-01	7.2713632761325855184212767116D-03
42	6.0736263580993579091862788373D-01	6.5592909812103699142953682242D-03
43	6.2825086764281494135566080830D-01	5.8881670339239326627628295350D-03
44	6.4890300397559191924145024262D-01	5.2586785178303006743438643605D-03
45	6.6928082243199885347937908620D-01	4.6711607940549969284462670836D-03
46	6.8934660454081053970218174121D-01	4.1256155526904326648762809013D-03
47	7.0906320590904989126966355339D-01	3.6217302638245671095564900031D-03
48	7.2839412532264886351340094537D-01	3.1588989140777765198607273929D-03
49	7.4730357264673131114046706439D-01	2.7362439040382605852917748199D-03
50	7.6575653539983137582689267604D-01	2.3526389728500713770793590706D-03
51	7.8371884387876510133273746610D-01	2.0067330083759130603145313187D-03
52	8.0115723471351763402974552436D-01	1.6969745947912249203033186252D-03

TABLE III.- Continued

53	8.1803941273435533872544677915D-01	1.4216371441349119567602280516D-03
54	8.3433411103653205300518514054D-01	1.1788444542175392872897831175D-03
55	8.5001114913121838532313461850D-01	9.6659653234411082706462027496D-04
56	8.6504148907489171127469252811D-01	7.8279552252068905757761367779D-04
57	8.7939728947317266143626484486D-01	6.2527157315667872327720837570D-04
58	8.9305195725910747197925853541D-01	4.9180848272094034123769766894D-04
59	9.0598019715014693040357995563D-01	3.8016896233144571177703681403D-04
60	9.1815805869261198913391329290D-01	2.8811935682393303904461043342D-04
61	9.2956298080735048725964298852D-01	2.1345366942115382843814459391D-04
62	9.4017383375577142499269555006D-01	1.5401673967394791149027300675D-04
63	9.4997095845192583380066645497D-01	1.0772642982841827222183048021D-04
64	9.5893620305480580061940778907D-01	7.2594681146457110740843058712D-05
65	9.6705295678810947182042983536D-01	4.6747308923003742693360992684D-05
66	9.7430618095928957035140595752D-01	2.8442412952591119403814074009D-05
67	9.8068243720643171831505297935D-01	1.6087288946650930723499318319D-05
68	9.8616991316015099135135821417D-01	8.2537358353542231154721164368D-06
69	9.9075844624417392689325801880D-01	3.6916639444441992724056435618D-06
70	9.9443954860745247407965804784D-01	1.3409196580877728429915421779D-06
71	9.9720644935270384037975891707D-01	3.4125330916977041264142650681D-07
72	9.9905430377490994272965701071D-01	4.0368730820068759324397652485D-08

N = 80

M	X(M)	W(M)
1	1.8498809894197773646772427537D-04	4.3495652060892343014475096782D-03
2	1.0557962177290793124767936428D-03	8.4757819524148594533819464475D-03
3	2.6611938108109087813816376926D-03	1.1686107756646427448735658670D-02
4	5.0029991549698706356455267660D-03	1.4319415810984323155559555466D-02
5	8.0798563440542840323498081246D-03	1.6524695163452386195890479941D-02
6	1.1888581399629384913698490654D-02	1.8386174933222475581675240038D-02
7	1.6424519228036557001311618489D-02	1.9958423920032790397304994210D-02
8	2.1681697083040767514272606157D-02	2.1280059409091643759581300139D-02
9	2.7652908238198887016313932457D-02	2.2380142845062903499859657754D-02
10	3.4329765801103830754825055489D-02	2.3281660192677814079064044731D-02
11	4.1702742161584196864668788291D-02	2.4003485102954722871089757825D-02
12	4.9761201109041754402420643406D-02	2.4561610766374167343009472707D-02
13	5.8493426174637774837178072754D-02	2.4969943866648340588248182623D-02
14	6.7886647140183628848499200807D-02	2.5240837639861840899619900918D-02
15	7.7927065937073728157327412733D-02	2.5385460176898727683558638994D-02
16	8.8599881913334753433115739966D-02	2.5414054616388640264752275496D-02
17	9.9889318990436948053348127115D-02	2.5336126102843414876859166763D-02
18	1.1177865147660453784063344790D-01	2.5160577792390980662988617890D-02
19	1.2425023220585098749205049711D-01	2.4895810604695707692596037023D-02
20	1.3728552100831936097365830026D-01	2.4549796690565366166873134071D-02
21	1.5086511427458133700267245352D-01	2.4130133545243995920595296317D-02
22	1.6496877554675867280188440258D-01	2.3644083691175261586275147355D-02
23	1.7957546714797733473659561804D-01	2.3098603498158584507482231528D-02
24	1.9466338284612263315822858022D-01	2.2500363772843930048755141579D-02
25	2.1020998153608080338148209060D-01	2.1855764090846290155454920336D-02
26	2.2619202191566750011767601888D-01	2.1170942373011024728608608221D-02
27	2.4258559812335891258573719813D-01	2.0451780863931145453435981777D-02
28	2.5936617630027537390877351891D-01	1.9703909416959735745612832162D-02
29	2.7650863203424817479213508600D-01	1.8932706799583452966907993681D-02
30	2.9399728363997554224587396769D-01	1.8143300588264645479554223751D-02
31	3.1177595622606484414345518176D-01	1.7340566110306310363432676111D-02
32	3.2984797149703190265488400526D-01	1.6529124803198548820345351378D-02
33	3.4817623823600307592888321800D-01	1.5713342293024925616018582816D-02
34	3.6673326841185203666474665593D-01	1.4897326438335337980509911744D-02
35	3.8549122385277515288940958255D-01	1.4084925541139942698528676274D-02
36	4.0442195842680804998928067647D-01	1.3279726889923041454320802892D-02
37	4.2349706066849459109297337446D-01	1.2485055769014920090265270814D-02
38	4.4268789678981182296434081377D-01	1.1703975042943099514596342675D-02
39	4.6196565401251465207808860642D-01	1.0939285402492704381464279769D-02
40	4.8130138415827908934293841658D-01	1.0193526340363792160733668352D-02
41	5.0066604743238272374956121867D-01	9.4689779079291240390444601435D-03

42	5.2003055633615768593758272516D-01	8.7676632902138173291541952582D-03
43	5.3936581964307838686479809818D-01	8.0913522234872969898401161291D-03
44	5.5864278637309862725613809572D-01	7.4415652685058327203027298264D-03
45	5.7783248969972605168715893957D-01	6.8195789422592725360505299250D-03
46	5.9690609072431283823387936277D-01	6.2264317018936986907133086440D-03
47	6.1583492205214695608206047458D-01	5.6629307661741734351795179670D-03
48	6.3459053110514565361530505860D-01	5.1296597523181678505316041905D-03
49	6.5314472310627969639754489306D-01	4.6269870991922366308458667322D-03
50	6.7146960367129108854578155745D-01	4.1550752416606595908542719508D-03
51	6.8953762094380654156730213133D-01	3.7138904952570903721428252534D-03
52	7.0732160721059184232980070652D-01	3.3032136052810923881745754217D-03
53	7.2479481993443659431001773768D-01	2.9226509098712222339706546484D-03
54	7.4193098214300267341126369128D-01	2.5716460625516846522989989829D-03
55	7.5870432211291122503325328849D-01	2.2494922561719302579868644749D-03
56	7.7508961228938023938948897613D-01	1.9553448870428810545307796002D-03
57	7.9106220738285572451995992038D-01	1.6882345954073751318485745712D-03
58	8.0659808158530233040824635026D-01	1.4470806161554304560812179293D-03
59	8.2167386485013205556095651707D-01	1.2307043718978194762293099180D-03
60	8.3626687818115057588755010664D-01	1.0378432391357789567286771119D-03
61	8.5035516787738815537695231580D-01	8.6716441730238688202860853249D-04
62	8.6391753868225479502211712556D-01	7.1727882989420774262763462435D-04
63	8.7693358578711674379720349617D-01	5.8675498675202008208748200342D-04
64	8.8938372564113457848261254863D-01	4.7413273677816345979654885545D-04
65	9.0124922552103512795471776279D-01	3.7793684098606629978958339705D-04
66	9.1251223181641873333826564765D-01	2.9669029675492720446110065816D-04
67	9.2315579698824717693095308379D-01	2.2892734549860485106464772084D-04
68	9.3316390516035226469087009567D-01	1.7320609764093873193801555065D-04
69	9.4252149630622570735959258800D-01	1.2812071080746853647057023144D-04
70	9.5121448899615783961540046331D-01	9.2313059482389740805206809221D-05
71	9.5922980167334199648225136648D-01	6.4483837025163102209636056665D-05
72	9.6655537243268288192976126331D-01	4.3403033878115688727721273100D-05
73	9.7318017728482575922135355330D-01	2.7919739008319797779609449593D-05
74	9.7909424690597360513067054888D-01	1.6971215096790982551117658189D-05
75	9.8428868191818565846215391226D-01	9.5912016975164703772072582108D-06
76	9.8875566687171003170821822608D-01	4.9174045191215538136744229715D-06
77	9.9248848353464427253516193226D-01	2.1981331135797202620802226351D-06
78	9.9548152593642830905738242297D-01	7.9805356977422215041992716515D-07
79	9.9773033030948315729212989180D-01	2.0302729006679389391470103951D-07
80	9.9923174155558030574063633410D-01	2.4011624368637797581823673787D-08

N = 88

M	X(M)	W(M)
1	1.5386947173044912778752606109D-04	3.6911473549880792362037274279D-03
2	8.7681682982773634163330273298D-04	7.2256389433741682345414654386D-03
3	2.2089715613637089192968609162D-03	1.0003151079453681406178104153D-02
4	4.1520237725789245733464574894D-03	1.2304878917888695879921975316D-02
5	6.7052511865070461444122554249D-03	1.4254484407287116908362247700D-02
6	9.8665991113316941440436346559D-03	1.5921792554348331601979694403D-02
7	1.3632963217942200439135257677D-02	1.7351921624037577260534238062D-02
8	1.9000309493432079512899622439D-02	1.8576647951071457990330138965D-02
9	2.2963738332947167248171963730D-02	1.9619765323589073512335855381D-02
10	2.8517524577601116713277313002D-02	2.0499941714897159628580349286D-02
11	3.4655145811292147018343185650D-02	2.1232378580091013938328636585D-02
12	4.1369304502597078606387228915D-02	2.1829837783522277034285074107D-02
13	4.8651946815788834452674254822D-02	2.2303308776005428248387411571D-02
14	5.6494279635946695821348649886D-02	2.2662458837813561044297058957D-02
15	6.4886786705399197688531727167D-02	2.2915946226848730084196098849D-02
16	7.3819244416974861295883328152D-02	2.3071643281758359967165824893D-02
17	8.3280737607368420943971871803D-02	2.3136798446688111849182343790D-02
18	9.3259675572121622058570226967D-02	2.3118155722932533503194810920D-02
19	1.0374380844730691910312439033D-01	2.3022043750488625993101365032D-02
20	1.1472024405335421640362495861D-01	2.2854442791597067694170288386D-02
21	1.2617546526317696044010951965D-01	2.2621035361372920504701484532D-02
22	1.3809534793380004739145340207D-01	2.2327244582544251208483293777D-02

TABLE III. - Continued

23	1.5046517942444481101810361996D-01	2.1978263214042463542096076443D-02
24	1.6326967771224412153724938885D-01	2.1579075525230770732951835064D-02
25	1.7649301110801930815419103402D-01	2.1134473640419958078920723557D-02
26	1.9011881856792401274554048855D-01	2.0649069586848385283804627147D-02
27	2.0413023059163293332247375764D-01	2.0127303994762987460063998896D-02
28	2.1850989069370917807472712116D-01	1.9573452188390571469189983979D-02
29	2.3323997743153572128185635660D-01	1.8991628249714575304962104563D-02
30	2.4830222697054351465548842387D-01	1.8385787518194551643254063073D-02
31	2.6367795616526873254708377290D-01	1.7759727898537535403300289039D-02
32	2.7934808613292203182523520836D-01	1.7117090278055505229736315018D-02
33	2.9529316629457910466016528903D-01	1.6461358299808925072796104796D-02
34	3.1149339885774943892988370280D-01	1.5795857693874892799414793808D-02
35	3.2792866371290837805033785296D-01	1.5123755333935929194884406137D-02
36	3.4457854371555582447707391933D-01	1.4448058157928811767783235628D-02
37	3.6142235032447018864895856060D-01	1.3771612068206940935875006526D-02
38	3.7843914956604089764256253538D-01	1.3097100907415539888364373074D-02
39	3.9560778829387346010442166337D-01	1.2427045590193454262973044779D-02
40	4.1290692071225731618777486766D-01	1.1763803457239587394088999631D-02
41	4.3031503513156037222240007396D-01	1.1109567906709727483200251767D-02
42	4.4781048092315887367885191437D-01	1.0466368347949126979292508793D-02
43	4.6537149564112209101350488186D-01	9.8360705139131760919476158941D-03
44	4.8297623222775441938942180801D-01	9.2203771610448712021859415238D-03
45	5.0060278661814747447030232580D-01	8.6208291786766550296450060067D-03
46	5.1822922466456916500923947661D-01	8.0388071240583079808126874835D-03
47	5.3583361008958636647859714793D-01	7.4755331937653996749828860655D-03
48	5.5339403169142831388081356496D-01	6.9320736374219531489292008279D-03
49	5.7088863081327088325323989847D-01	6.4093416153029495290293588275D-03
50	5.8829562869400364111411134531D-01	5.9081004974085675384144885465D-03
51	6.0559335371640376008662565900D-01	5.4289675979759055528689204844D-03
52	6.2276026851894280749982404733D-01	4.9724183360780735020262825779D-03
53	6.3977499693759088040415465102D-01	4.5387908099251560647288628839D-03
54	6.5661635074416699887427195054D-01	4.1282907697027982374688331797D-03
55	6.7326335614801431226998790017D-01	3.7409969712429762025776152895D-03
56	6.8969528002805278097427953943D-01	3.3768668905026178895844749175D-03
57	7.0589165586257984500046502090D-01	3.0357427767168811523785954150D-03
58	7.2183230932455046201133669975D-01	2.7173580201852344658299035912D-03
59	7.3749738351047106964478852741D-01	2.4213438089320582837167274927D-03
60	7.5286736377148677174854192159D-01	2.1472360469528158944794734967D-03
61	7.6792310211572662523653226182D-01	1.8944825054065930553545607415D-03
62	7.8264584115149756163032341987D-01	1.6624501769415029504299007058D-03
63	7.9701723754148245216178921885D-01	1.4504328023372620143484884845D-03
64	8.1101938493870134935842542911D-01	1.2576585378157788869819149790D-03
65	8.2463483637563524799835625346D-01	1.0832977307027764315486794906D-03
66	8.3784662607859806425487332356D-01	9.2647077061836899063978310227D-04
67	8.5063829068012924871365616782D-01	7.8625598302929350685598304500D-04
68	8.6299388780298595026714837008D-01	6.6169753180727366472742611212D-04
69	8.7489802599002955770019487679D-01	5.5181329740382075122655994410D-04
70	8.8633586395517874872921326534D-01	4.5560269736853646003265812357D-04
71	8.9729314913143050236930487858D-01	3.7205441620238151754869521472D-04
72	9.0775622549285367412217412296D-01	3.0015401194588070154648103782D-04
73	9.1771205262839562531161201353D-01	2.3889136745106301295639127155D-04
74	9.2714822046328982142132627690D-01	1.8726795497102024313566876452D-04
75	9.3605297268921758388096446514D-01	1.4430388351794898911645945010D-04
76	9.4441520564042930942423240757D-01	1.0904469938475781698798165286D-04
77	9.5222449800453235460365489335D-01	8.0567911291801652592030403126D-05
78	9.5947111594551479453691311650D-01	5.7989212803768912524074888730D-05
79	9.6614602686896091821683608113D-01	4.0468375956605854432867243867D-05
80	9.7224091073779260040955616481D-01	2.7214791434728808247523434969D-05
81	9.7774817051766335117479700554D-01	1.7492632138468439427390214141D-05
82	9.8256094176223811542471647960D-01	1.0625618574252083428786072478D-05
83	9.8697310138431482274125648334D-01	6.0013661787403346035057876211D-06
84	9.9067927576299825985891842036D-01	3.0752964460422603462826027524D-06
85	9.9377484869550176008023617207D-01	1.3740955571454329939642567781D-06
86	9.9625597122679293320686625911D-01	4.9870610596869076113371845838D-07
87	9.9811958425077738806582356445D-01	1.2683947324542366351962361446D-07
88	9.9936355464180150916658208600D-01	1.4998454144904645130381295820D-08

N = 96

M	X(M)	W(M)
1	1.3001326573902262460187732581D-04	3.1754639116380449338247073348D-03
2	7.3984886307611210608921226854D-04	6.2408161333104232146142016571D-03
3	1.8630789775310311176195553364D-03	8.6700034371066253236022825104D-03
4	3.5012332327374147178611741108D-03	1.0700111837963271730385059529D-02
5	5.6539549265194872469117506451D-03	1.2435449022036785093864752745D-02
6	8.3198848600564395960471034534D-03	1.3934871489030130124790042264D-02
7	1.1496890314501423043858750915D-02	1.5236343154146111465219076700D-02
8	1.5182161218539072300909218662D-02	1.6366522771575420882185600789D-02
9	1.9372260789426751248874261192D-02	1.7345288766532785494284394533D-02
10	2.4063156498831138600773641158D-02	1.8188153747385221984442098658D-02
11	2.9250241348908048922266938686D-02	1.8907669009579183714891737896D-02
12	3.4928349983723880236503803785D-02	1.9514295481527215542823163791D-02
13	4.1091771922928654650246882624D-02	2.0016971024960805819382659339D-02
14	4.7734263169573683729756052637D-02	2.0423494558127404043658127825D-02
15	5.4849056920478785767896563066D-02	2.0740794361221065037856409859D-02
16	6.2428873923618447725051068525D-02	2.0975120265927480743096064372D-02
17	7.0465932064043656994195168538D-02	2.1132184174811252400824528156D-02
18	7.8951957461845301754490558524D-02	2.1217264528215511108780301474D-02
19	8.7878193704318413513607441581D-02	2.1235285016796557800400447671D-02
20	9.7235412794740088162398312024D-02	2.1190874519053079667096450607D-02
21	1.0701392577362627139151549417D-01	2.1088413109276095615710857289D-02
22	1.1720359375011015223137916275D-01	2.0932067572452276764767611995D-02
23	1.2779383926826228903965480011D-01	2.0725818910412673262523078593D-02
24	1.3877365302394038167787654971D-01	2.0473483666330210190445280873D-02
25	1.5013163094096555304699045304D-01	2.0178730432511125803067190290D-02
26	1.6185593661032470014706936325D-01	1.9845092575846383130067815598D-02
27	1.7393436409219842047925792983D-01	1.9475977975134759166703918083D-02
28	1.8635432607757502543765021113D-01	1.9074676387526899460943745702D-02
29	1.9910287240380160055539345866D-01	1.8644364929216944617786641805D-02
30	2.1216670391648085537779412431D-01	1.8188112055650436973089227941D-02
31	2.2553218666853252253612321293D-01	1.7708880350175331605083128813D-02
32	2.3918536644591750975104282996D-01	1.7209528371064338056276991833D-02
33	2.5311198360841158404069326495D-01	1.6692811760769621137731013227D-02
34	2.6729748823286868040968882163D-01	1.6161383784943801897832979102D-02
35	2.8172705554559895700037459701D-01	1.5617795439840106473796004414D-02
36	2.9638560162977867197051651614D-01	1.5064495243464568381406064198D-02
37	3.1125779939318905650579075597D-01	1.4503828807007533424624065515D-02
38	3.2632809478103526703258445391D-01	1.3938038267661654627833543805D-02
39	3.4158072321811319065654255508D-01	1.3369261651203839518814070264D-02
40	3.5699972626416281025694651734D-01	1.2799532222116820462338251574D-02
41	3.7256896846586535133387418550D-01	1.2230777870118872683550885262D-02
42	3.8827215438860230919485976488D-01	1.1664820574421653202869647547D-02
43	4.0409284581079357196972172878D-01	1.1103375980583922647329405956D-02
44	4.2001447906336594268608789779D-01	1.0548053119267201828675219878D-02
45	4.3602038249666981298155411433D-01	1.0000354291365792172244106681D-02
46	4.5209379405695845752008066039D-01	9.4616751397492823242939496343D-03
47	4.6821787895436971022062643294D-01	8.9333049241181245592180470523D-03
48	4.8437574740420227559137280028D-01	8.4164270121501033319171660929D-03
49	5.0055047242315749408984220195D-01	7.9121195971417462988721058128D-03
50	5.1672510766212108861785493370D-01	7.4213566496709914066421516974D-03
51	5.3288270525698749330159688781D-01	6.9450091083829818315955241306D-03
52	5.4900633367898115008290866699D-01	6.4838463127951361520373139569D-03
53	5.6507909556590409279089040272D-01	6.0385376790026060215794587082D-03
54	5.8108414551573673568272518304D-01	5.6096546173181327773435423322D-03
55	5.9700470782403861450041738615D-01	5.1976726891826512945225496527D-03
56	6.1282409414663750703606290054D-01	4.8029739991197152299683160236D-03
57	6.2852572106915853382415289480D-01	4.4258498160656809042117919214D-03
58	6.4409312756502917844229020468D-01	4.0665034170786081186919812792D-03
59	6.5950999232370135812810562155D-01	3.7250531452038831881817672427D-03
60	6.7476015093095741686537295208D-01	3.4015356721470120137370524706D-03
61	6.8982761288331290861078110548D-01	3.0959094553684580559806433020D-03
62	7.0469657841869499396808619076D-01	2.8080583782673883470608789219D-03
63	7.1935145514576089456623835062D-01	2.5377955612571143095593173871D-03
64	7.3377687445442583765072708854D-01	2.2848673307518873104663812590D-03

TABLE III. - Concluded

65	7.4795770769039397597199172253D-01	2.0489573323800555296594712951D-03
66	7.6187908207672857581933235210D-01	1.8296907741103343680259339060D-03
67	7.7552639636575901313386975808D-01	1.6266387844243241479337345177D-03
68	7.8888533620490148152106147120D-01	1.4393228701878992839882958318D-03
69	8.0194138920026746843247637358D-01	1.2672194584653386448520543758D-03
70	8.1468235966224866490374524482D-01	1.1097645061818342394894223765D-03
71	8.2709338301759870818681863096D-01	9.6635816127114781897410651329D-04
72	8.3916193987288068985078560805D-01	8.3636945874456516352496768039D-04
73	8.5087536971451438537315806649D-01	7.1914103498381703542979953382D-04
74	8.6222138423103839894772881187D-01	6.1399384349316824027779705518D-04
75	8.7318808324359965376097083135D-01	5.2023185534325771949490928609D-04
76	8.8376395223109578418021691631D-01	4.3714672760028280044135288168D-04
77	8.9393790443682507707116430055D-01	3.6402242315746890250033264545D-04
78	9.0369926254394405883586922667D-01	3.0013976557007931880999502056D-04
79	9.1303778490749558555473994671D-01	2.4478091273903409025472570710D-04
80	9.2194367333125231562592229755D-01	1.9723373358996006199043362085D-04
81	9.3040758337812548527174716330D-01	1.5679607225252002304622072328D-04
82	9.3842063420342413726351909784D-01	1.2277988465739431804402436880D-04
83	9.4597441790082905651380798829D-01	9.4515232933431184404680019666D-05
84	9.5306100835159547517783187491D-01	7.1354123503249928384798055187D-05
85	9.5967296956827316529561852321D-01	5.2674175339871772719169687766D-05
86	9.6580336352524530027065467981D-01	3.7882105457556257239585128154D-05
87	9.7144575746988818977834832614D-01	2.6417019364619837229383400345D-05
88	9.7659423071072477215345030806D-01	1.7753494902188306425039665981D-05
89	9.8124338088403717723162597186D-01	1.1404448628067619044380227202D-05
90	9.8538832971194291626277726174D-01	6.9237746765969256083832597733D-06
91	9.8902472829463826621090685160D-01	3.9087468307758968064029164347D-06
92	9.9214876206689517723359218630D-01	2.0021753793685646541242223176D-06
93	9.9475715585029245174916586945D-01	8.9431119629162926314546929628D-07
94	9.9684718071566351283118749315D-01	3.2449036984192846793419039428D-07
95	9.9841667183016281876126484065D-01	8.2513624737479446416953221217D-08
96	9.9946414212191552148756135524D-01	9.7557462101442911416568490272D-09

ERROR ANALYSIS

This section is concerned with methods for estimating the accuracy of the computed quadrature abscissas and weights. The related subject of estimating the accuracy of integrals computed by a Gaussian quadrature will not be considered at all because it is adequately covered in the literature. (See refs. 7 and 8, for example.)

Calculation of quadrature abscissas and weights, as in the sample problem, consists of several computational tasks, any of which can introduce errors into the results. For the sample problem these tasks are:

1. Calculation of classical Gaussian abscissas and weights.

2. Reduction of the effect of the nonclassical singularity by using Taylor's theorem.

3. Calculation of h_n and h'_n by numerical integration using the abscissas and weights computed in step 1 above.

4. Evaluation of $b_{n+1} = h'_n/h_n$ and $c_{n+1} = h_n/h_{n-1}$.

5. Calculation of the desired nonclassical Gaussian abscissas and weights from the recursion coefficients of step 4.

The classical Gaussian calculation of step 1 is very stable. Comparison of abscissas and weights computed by subroutine CGAUSS with tabulated values in reference 7 indicates that the error introduced by CGAUSS is negligible. Subroutine NGAUSS uses essentially the same algorithms so it also introduces a negligible amount of error. This means that the abscissas and weights computed by NGAUSS will be accurate provided the routine is furnished accurate recursion coefficients. Thus, the only source of detectable error are steps 2 and 3 above, the singularity reduction and the simpler Gaussian quadrature.

In a computation of this sort it is useful to distinguish between two kinds of error: round-off error and truncation error. Round-off error results from performing sequential calculations with finite-length computer words, whereas truncation error results from terminating a convergent mathematical process after a finite number of steps. For example, if an integral is approximated by a finite sum as in equation (1) the truncation error is due to using too low a quadrature order, whereas the round-off error increases as the order N increases. In a Gaussian quadrature, where all weights are positive, the number of decimal digits lost due to round-off is

$$\frac{1}{2} \log_{10} N \tag{59}$$

if individual computer words are rounded, and is

$$\log_{10} N \tag{60}$$

if individual computer words are truncated. In expressions (59) and (60), N is the number of terms in the quadrature sum, which is the product of the quadrature orders if a multidimensional quadrature is performed.

For the sample problem double precision arithmetic is used, thus computer words are truncated and expression (60) is appropriate. The maximum N is 2000 so that less than four digits are lost due to round-off. Since this loss is negligible it can be inferred that essentially all the error in the computed recursion coefficients is due to truncation error in the Gaussian quadrature used to compute h_n and h_n' (eqs. (38)).

This error can be estimated by using the program itself and varying N. For sufficiently large N, if the leading digits in b_n and c_n agree for two different values of N these digits can be assumed to be correct. A more reliable way to estimate accuracy is to compare with more accurate recursion coefficients generated by a different method. The magnitude of the truncation error of a Gaussian quadrature depends upon the order and the function being integrated, but is independent of the weight function. This means that for the sample problem any value of α can be used to estimate the error. In particular, when $\alpha = 1$, the weight function is $\ln (1/x)$ and for this function values of b_n and c_n for $n = 1$ to 16 accurate to 30 digits are tabulated in reference 7. Comparison with these values shows that for $N = 1400$ the computed abscissas and weights b_n and c_n are accurate to about 17 significant digits and the accuracy decreases very slowly as the subscript of the recursion coefficient increases. Subprogram COMPARE was used for this comparison.

Langley Research Center
National Aeronautics and Space Administration
Hampton, Va. 23665
July 1, 1975

APPENDIX A

LAGUERRE ITERATION

Laguerre's iteration formula is used to compute the zeros of orthogonal polynomials from their recursion coefficients. The formula is

$$x^{(k+1)} = x^{(k)} - \frac{np}{p' \pm \sqrt{(n-1)^2 p'^2 - n(n-1)pp''}} \tag{A1}$$

where p, p', and p'' are evaluated at $x = x^{(k)}$, n is the degree of the polynomial $p(x)$, $x^{(k)}$ is the kth approximation to the zero being sought, and the sign on the radical is taken to maximize the absolute value of the denominator. Some properties of Laguerre iteration are (ref. 9):

1. Convergence to a simple zero is cubic.

2. The iteration is invariant under the bilinear transformation $\bar{x} = (ax + b)/(cx + d)$.

3. If all zeros are real and distinct the real axis is divided up into n abutting intervals, each containing a zero, such that any starter within the interval will converge to the root in that interval.

4. If all zeros are real, convergence is monotonic.

Property 1 above means that convergence is rapid. Property 2 makes it possible to compute the Laguerre iterate of $-\infty$. If the sign on the radical is taken to be the sign of p' at the zero, rather than the sign of p' at the current iterate, then the interval of convergence for the mth zero x_m extends from $x_{m-1} + 0$ to $x_{m+1} - 0$.

To describe completely a Laguerre iteration procedure used in a computer program it is necessary to state how the iteration is started, what deflation procedure is used if any, and how the iteration is terminated. Let $x_m^{(k)}$ be the kth approximation to the mth zero counting from the left and let $x_m^{(0)}$ be the starting value.

Starting Value for x_1

The starting value $x_1^{(0)}$ is the Laguerre iterate of $-\infty$ except for the classical Jacobi-Gauss and Laguerre-Gauss cases where it is the Laguerre iterate of -1 and 0, respectively.

APPENDIX A

Deflation

When equation (A1) is used for $m > 1$, $p(x)$ is replaced by

$$q(x) = \frac{p(x)}{\prod_{i=1}^{m-1} (x - x_i)} \tag{A2}$$

and n is replaced by $(n + 1 - m)$. Note that $q(x)$ is not a polynomial but is a rational function with a zero/pole pair at each computed zero. Thus deflation has no effect on the accuracy of subsequent zeros. This deflation procedure is absolutely reliable as a means of preventing convergence to a previously computed zero, provided that no two zeros are closer than the resolving ability of the computer floating point number system. To apply equation (A1) substituting q for p as given by equation (A2) note that

$$\frac{q'}{q} = \frac{p'}{p} - \sum_{i=1}^{m-1} \frac{1}{x - x_i}$$

and

$$\frac{q''}{q} - \left(\frac{q'}{q}\right)^2 = \frac{p''}{p} - \left(\frac{p'}{p}\right)^2 + \sum_{i=1}^{m-1} \frac{1}{(x - x_i)^2}$$

Starting Value for x_m

The starting value $x_m^{(0)}$ for $m > 1$ is the Newton iterate of $q(x)/(x - x_{m-1})$. This number can be computed from q'/q and q''/q, both of which are available because they were used to compute the converged value of x_{m-1}.

Iteration Termination

The starting value is always left of the zero and convergence is monotonic. Iteration is terminated when a negative correction is computed because this indicates the iteration has reached its noise level.

Other Programing Considerations

The zeros x_m are computed in DOUBLE PRECISION in the program. To do this it suffices to compute only $p(x)$ in DOUBLE PRECISION. The other terms $p'(x)$, $p''(x)$, $\sum (x - x_i)^{-1}$, and $\sum (x - x_i)^{-2}$ can all be computed in SINGLE PRECISION.

APPENDIX B

SUBPROGRAM DESCRIPTION

This appendix contains usage descriptions of the four subprograms mentioned in the second paragraph of the section entitled "Program Organization." It also contains usage descriptions of three other subprograms, JGAUSS, LGAUSS, and DGAMF that could be called independently. All other subprograms are described only by their listings and associated comments in the section entitled "Sample-Problem Program Listing."

Subroutine CGAUSS

Language: FORTRAN

Purpose: To compute the abscissas and weights for a classical Gaussian quadrature.

Use: CALL CGAUSS(N,X,W,A,B,ALPHA,BETA)

N	Input parameter containing the order of the quadrature
X	One-dimensional output array containing the abscissas
W	One-dimensional output containing the weights
A,B	Input parameters containing the integration interval delimiters or weight-function scale factors
ALPHA,BETA	Input parameters containing weight-function exponents

Restrictions: X and W must be dimensioned N or larger in the calling program. In the Jacobi case B must be greater than A. In the Laguerre case B must be positive. In the Hermite case A must be positive. All arguments except N must be typed DOUBLE PRECISION.

Discussion: The subroutine is used to compute abscissas and weights for any of the three quadrature formulas below:

$$\int_a^b (b - x)^{\alpha}(x - a)^{\beta} f(x)\, dx \approx \sum_{m=1}^{n} w_m f(x_m) \qquad \text{(Jacobi)}$$

APPENDIX B

$$\int_a^{\infty} (x - a)^{\alpha} e^{-bx} f(x)\, dx \approx \sum_{m=1}^{n} w_m f(x_m) \qquad \text{(Laguerre)}$$

$$\int_{-\infty}^{\infty} e^{-(ax^2+bx)} f(x)\, dx \approx \sum_{m=1}^{n} w_m f(x_m) \qquad \text{(Hermite)}$$

The Jacobi case is selected by calling CGAUSS with α, β greater than -1. The Laguerre case is selected by setting β to a real number less than -1. The Hermite case is selected by setting α to a real number less than -1.

Gaussian quadrature formulas are exact whenever the integrand function $f(x)$ is a polynomial of degree $(2n - 1)$ or less. The weights w_m are all positive so that numerical stability is assured. The three classical Gaussian quadratures listed above have the additional desirable property that the abscissas and weights are particularly easy to compute.

Examples: Suppose the computation of x_m and w_m for the quadrature formula was desired, then

$$\int_0^1 \sqrt{x}\, f(x)\, dx \approx \sum_{m=1}^{8} w_m f(x_m)$$

The call would be

CALL CGAUSS(8,X,W,0.D,1.D,0.D,.5D)

The table below lists the last four parameters for some of the better known Gaussian quadrature formulas

Name	A	B	ALPHA	BETA
Legendre	-1.	1.	0.	0.
Shifted Legendre	0.	1.	0.	0.
Chebyshev (1st)	-1.	1.	-.5	-.5
Chebyshev (2d)	-1.	1.	.5	.5
Jacobi	-1.	1.	α	β
Shifted Jacobi	0.	1.	p-q	q-1.
Laguerre	0.	1.	0.	-2.

APPENDIX B

Name	A	B	ALPHA	BETA
Hermite	1.	0.	-2.	-2.
Scaled Hermite	.5	0.	-2.	-2.
Gegenbauer	-1.	1.	-.5	-.5
Generalized Laguerre	0.	1.	α	-2.

Method: In the Jacobi case the abscissas and weights x_m, w_m in

$$\int_a^b (b - x)^\alpha (x - a)^\beta f(x)\, dx \approx \sum_{m=1}^{n} w_m f(x_m)$$

are computed from the $\overline{x}_m, \overline{w}_m$ associated with

$$\int_{-1}^{1} (1 - \overline{x})^\alpha (1 + \overline{x})^\beta \overline{f}(\overline{x})\, d\overline{x} \approx \sum_{m=1}^{n} \overline{w}_m \overline{f}(\overline{x}_m)$$

using the relations

$$x_m = \frac{1}{2}(b + a) + \frac{1}{2}(b - a)\overline{x}_m$$

and

$$w_m = \left(\frac{b - a}{2}\right)^{\alpha+\beta+1} \overline{w}_m$$

The standard-interval abscissas and weights $\overline{x}_m, \overline{w}_m$ are computed by a call to subroutine JGAUSS (see JGAUSS write-up for description of method used).

In the Laguerre case the x_m, w_m associated with

$$\int_a^\infty (x - a)^\alpha e^{-bx} f(x)\, dx \approx \sum_{m=1}^{n} w_m f(x_m)$$

are computed from the $\bar{x}_m, \bar{w}_m$ associated with

$$\int_0^\infty \bar{x}^\alpha e^{-\bar{x}} \bar{f}(\bar{x})\, d\bar{x} \approx \sum_{m=1}^{n} w_m f(x_m)$$

using the relations

$$x_m = \frac{\bar{x}_m}{b} + a$$

and

$$w_m = b^{-\alpha-1} e^{-ba} \bar{w}_m$$

The standard-interval and scale abscissas and weights $\bar{x}_m, \bar{w}_m$ are computed by a call to subroutine LGAUSS.

In the Hermite case the x_m, w_m associated with

$$\int_{-\infty}^{\infty} e^{-\left(ax^2+bx\right)} f(x)\, dx \approx \sum_{m=1}^{n} w_m f(x_m)$$

are computed from the $\bar{x}_m, \bar{w}_m$ associated with

$$\int_{-\infty}^{\infty} e^{-\bar{x}^2} \bar{f}(\bar{x})\, d\bar{x} \approx \sum_{m=1}^{n} \bar{w}_m \bar{f}(\bar{x}_m)$$

using the relations

$$x_m = \frac{\bar{x}_m}{\sqrt{a}} - \frac{b}{2a}$$

and

$$w_m = e^{b^2/4a} \bar{w}_m$$

APPENDIX B

If n is even, let $n = 2\nu$ and let ξ_μ, ω_μ be associated with

$$\int_0^\infty \xi^{-1/2} e^{-\xi}\, g(\xi)\, d\xi \approx \sum_{\mu=1}^{\nu} \omega_\mu\, g(\xi_\mu) \quad .$$

Then

$$\left.\begin{aligned} \bar{x}_{\nu+\mu} &= \sqrt{\xi_\mu} \\ x_{\nu+1-\mu} &= -\sqrt{\xi_\mu} \\ \bar{w}_{\nu+1-\mu} &= \bar{w}_{\nu+\mu} = \frac{1}{2}\,\omega_\mu \end{aligned}\right\} \qquad (\mu = 1(1)\nu)$$

The ξ_μ, ω_μ are computed by a call to LGAUSS with ALPHA = 0.5.

If n is odd, let $n = 2\nu + 1$ and let ξ_μ, ω_μ be associated with

$$\int_0^\infty \xi^{1/2} e^{-\xi}\, g(\xi)\, d\xi \approx \sum_{\mu=1}^{\nu} \omega_\mu\, g(\xi_\mu)$$

Then

$$\bar{x}_{\nu+1+\mu} = \sqrt{\xi_\mu}$$

$$\bar{x}_{\nu+1-\mu} = -\sqrt{\xi_\mu}$$

$$\bar{w}_{\nu+1+\mu} = \bar{w}_{\nu+1-\mu} = \frac{\omega_\mu}{2\xi_\mu}$$

$$\bar{x}_{\nu+1} = 0$$

$$\bar{w}_{\nu+1} = \frac{(2^\nu \nu!)^2 \sqrt{\pi}}{(2\nu + 1)!}$$

The ξ_μ, ω_μ are computed by a call to LGAUSS with ALPHA = 0.5.

APPENDIX B

Accuracy: The program computes x_m and w_m to about 92-bit accuracy.

Subroutines used: JGAUSS, JRECUR, JWTCON, DGAMF, DSQRT, DEXP, DLOG, LGAUSS, LRECUR, LWTCON

APPENDIX B

Subroutine PNDER

Language: FORTRAN

Purpose: To compute derivatives of orthogonal polynomials defined by three-term recursion relations.

Use: CALL PNDER(N,M,X,PD,PS)

N Degree of the orthogonal polynomial

M Number of derivatives to be computed starting with the zeroth and going through the (M - 1)th

X Argument for which $p_n(x)$ and its derivative are computed

PD One-dimensional array into which PNDER stores

$$\frac{1}{k!}\frac{d^k}{dx^k} p_n(x) \qquad \text{for} \qquad k = 0,M-1$$

PS One-dimensional array into which PNDER stores

$$\frac{1}{k!}\frac{d^k}{dx^k} p_{n-1}(x) \qquad \text{for} \qquad k = 0,M-1$$

In addition to the parameters N, M, and X the calling program must also pass the recursion coefficients b_k, $k = 1,N$ in labeled COMMON block BOFN and must pass h_0 and c_k, $k = 2,N$ in labled COMMON block COFN.

Restrictions: PD and PS must be dimensioned M or larger in the calling program. X, PD, and PS and the contents of BOFN and COFN must be typed DOUBLE PRECISION.

Method: p_n and its derivatives are computed from the recursion formulas

$$\left.\begin{aligned} p_0(x) &= 0 \\ p_0^{(k)}(x) &= 0 \end{aligned}\right\} \qquad (k = 1,M)$$

$$\left.\begin{aligned} p_1(x) &= x - b_1 \\ p_1'(x) &= 1 \\ p_1^{(k)} &= 0 \end{aligned}\right\} \qquad (k = 2, M)$$

$$p_n(x) = (x - b_n)\, p_{n-1}(x) - c_n\, p_{n-2}(x) \qquad (n = 2, N)$$

$$\frac{1}{k!}\, p_n^{(k)} = (x - b_n)\frac{1}{k!}\, p_{n-1}^{(k)} - c_n \frac{1}{k!}\, p_{n-2}^{(k)} + \frac{1}{(k-1)!}\, p_{n-1}^{(k-1)} \qquad (n = 2, N; \quad k = 1, M-1)$$

Subroutines used: None.

APPENDIX B

DOUBLE PRECISION Function PNFUN

Language: FORTRAN

Purpose: To evaluate an orthogonal polynomial from its three-term recursion relation

Use: F = PNFUN(X,N)

X Argument from which $p_n(x)$ is computed

N Degree of $p_n(x)$

Restrictions: X and PNFUN should be declared DOUBLE PRECISION in the calling program.

Method: The same recursion formula as in PNDER is used.

Subroutines used: None.

APPENDIX B

Subroutine NGAUSS

Language: FORTRAN

Purpose: To compute the abscissas and weights of a Gaussian quadrature formula from the recursion coefficients of the associated system of orthogonal polynomials.

Use: CALL NGAUSS(N,X,W)

N   Number of abscissas and weights to be computed (i.e., the order of the quadrature formula)

X   One-dimensional array into which NGAUSS stores the computed abscissas

W   One-dimensional array into which NGAUSS stores the computed quadrature weights

In addition to the parameter N, the calling program must also pass the recursion coefficients b_k, $k = 1,N$ to NGAUSS via a labeled COMMON block called BOFN and must pass h_0 and the recursion coefficients c_k, $k = 2,N$ via a labeled COMMON block called COFN. The coefficients b_k, c_k are used to define the orthogonal polynomials p_k recursively

$$p_0(x) = 1$$

$$p_1(x) = x - b_1$$

$$p_k(x) = (x - b_k)\, p_{k-1}(x) - c_k\, p_{k-2}(x) \qquad (k = 2,N)$$

and

$$h_0 = \int_a^b \rho(x)\, p_0^2(x)\, dx$$

is needed to compute the quadrature weights.

Restrictions: The arrays X and W must be typed DOUBLE PRECISION and dimensioned N or larger in the calling program. The contents of labeled COMMON block BOFN and COFN must be typed DOUBLE PRECISION.

Method: The abscissas are computed as zeros of the polynomial defined by the recursion formula using Laguerre iteration with partial deflation. The weights are computed from

$$w_k = \frac{h_0 \prod_{k=2}^{N} c_k}{p_{N-1}(x_k)\, p_N'(x_k)}$$

Subroutine used: PNRECUR

APPENDIX B

Subroutine JGAUSS

Language: FORTRAN

Purpose: To compute the abscissas and weights for a Jacobi-Gauss quadrature over the interval (-1,1).

Use: CALL JGAUSS(N,X,W,ALPHA,BETA)

N	Input parameter containing the order of the quadrature
X	One-dimensional output array containing the abscissas
W	One-dimensional output array containing the weights
ALPHA	Input parameter containing the exponent of $(1 - x)$ in the weight function
BETA	Input parameter containing the exponent of $(1 + x)$ in the weight function

Restrictions: X and W must be dimensioned N or larger in the calling program. ALPHA and BETA must each be greater than -1. All arguments exact N must be typed DOUBLE PRECISION.

Example: Suppose one wanted to compute x_m, w_m for the quadrature formula

$$\int_{-1}^{1} (1 - x)^{1/2}(1 + x)^{3/2} f(x)\, dx \approx \sum_{m=1}^{8} w_m f(x_m)$$

The call would be
CALL JGAUSS(8,X,W,.5D,1.5D)

Method: The abscissas x_m are computed by applying Laguerre iteration to the Jacobi polynomial $P_n^{(\alpha,\beta)}(x)$ to compute its zeros. The function $P_n^{(\alpha,\beta)}$ and its first two derivatives are computed from equations 22.7.1, 22.8.1, and 22.6.1, respectively, of reference (a) of this subroutine. The quadrature weights are computed from

$$w_m = \frac{\left(1 - x_m^2\right)c_n}{\left[P_{n-1}^{(\alpha,\beta)}(x_m)\right]^2}$$

where

$$c_n = \frac{2^{\alpha+\beta-1}\Gamma(n + \alpha)\Gamma(n + \beta)(2n + \alpha + \beta)}{n!\Gamma(n + \alpha + \beta - 1)(n + \alpha)(n + \beta)}$$

Reference: (a) Abramowitz, Milton; and Stegun, Irene A., eds.: Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables. NBS Appl. Math. Ser. 55, U.S. Dep. Com., June 1964.

Subroutines used: JWTCON, JRECUR, DGAMF, DSQRT, SQRT

APPENDIX B

Subrountine LGAUSS

Language: FORTRAN

Purpose: To compute the abscissas and weights for a Laguerre-Gauss quadrature over the interval $(0,\infty)$.

Use: CALL LGAUSS(N,X,W,ALPHA)

N	Input parameter containing the order of the quadrature
X	One-dimensional output array containing the abscissas
W	One-dimensional output array containing the weights
ALPHA	Input parameter containing the exponent of x in the weight function

Restrictions: X and W must be dimensioned N or larger in the calling program. ALPHA must be greater than -1. All parameters except N must be typed DOUBLE PRECISION.

Example: Suppose one wanted to compute x_m, w_m for the quadrature formula

$$\int_0^\infty x^{1/2} e^{-x} f(x)\, dx \approx \sum_{m=1}^{25} w_m f(x_m)$$

The call would be
CALL LGAUSS(12,X,W,.5D)

Method: The abscissas x_m are computed by applying Laguerre iteration to the Laguerre polynomial $L_n^{(\alpha)}(x)$ to compute its zeros. The function $L_n^{(\alpha)}(x)$ and its first two derivatives are computed from equations 22.7.12, 22.8.6, and 22.6.15, respectively, of reference (a) of this subroutine. The quadrature weights are computed from

$$w_m = \frac{x c_n}{\left[L_{n-1}^{(\alpha)}(x)\right]^2}$$

where

$$c_n = \frac{\Gamma(n + \alpha)}{n!}$$

Reference: (a) Abramowitz, Milton; and Stegun, Irene A., eds.: Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables. NBS Appl. Math. Ser. 55, U.S. Dep. Com., June 1964.

Subroutines used: LWTCON, LRECUR, DGAMF, DSQRT, SQRT

APPENDIX B

DOUBLE PRECISION Function DGAMF

Language: FORTRAN

Purpose: To compute a DOUBLE PRECISION gamma function for a real argument.

Use: GAMX = DGAMF(X)

X Argument at which $\Gamma(X)$ is to be computed

Restrictions: Both X and DGAMF must be typed DOUBLE PRECISION and X must be less than 145.0 and not a negative integer.

Method: The gamma function satisfies the functional equation

$$\Gamma(X + 1) = X\Gamma(X)$$

If X is an integer, $\Gamma(X)$ is computed from $\Gamma(0) = 1$ by repeated application of the functional equation. If X is a half integer, $\Gamma(X)$ is computed from $\Gamma(1/2) = \sqrt{\pi}$. Otherwise $\Gamma(X)$ is computed from $\Gamma(Z)$ where

(Z - X) is an integer

$144.0 \leqq Z < 145.0$

$\Gamma(Z)$ is given by equation 6.1.48 in reference (a) of this subroutine.

Reference: (a) Abramowitz, Milton; and Stegun, Irene A., eds.: Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables. NBS Appl. Math. Ser. 55, U.S. Dep. Com., June 1964.

Subroutines used: DEXP, DLOG

REFERENCES

1. Staff of Bateman Manuscript Project: Higher Transcendental Functions. Volume II. McGraw-Hill Book Co., Inc., 1953.
2. Abramowitz, Milton; and Stegun, Irene A., eds.: Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables. NBS Appl. Math. Ser. 55, U.S. Dep. Com., June 1964.
3. Gautschi, Walter: Construction of Gauss-Christoffel Quadrature Formulas. Math. Comput., vol. 22, no. 102, Apr. 1968, pp. 251-270.
4. Golub, Gene H.; and Welsch, John H.: Calculation of Gauss Quadrature Rules. Math. Comput., vol. 23, no. 106, Apr. 1969, pp. 221-230.
5. Byrd, Paul F.; and Galant, David C.: Gauss Quadrature Rules Involving Some Non-classical Weight Functions. NASA TN D-5785, 1970.
6. Danloy, Bernard: Numerical Construction of Gaussian Quadrature Formulas for $\int_0^1 (-\mathrm{Log}\ x) \cdot x^{\alpha} \cdot f(x) \cdot dx$ and $\int_0^{\infty} E_m(x) \cdot f(x) \cdot dx$. Math. Comput., vol. 27, no. 124, Oct. 1973, pp. 861-869.
7. Stroud, A. H.; and Secrest, Don: Gaussian Quadrature Formulas. Prentice-Hall, Inc., c.1966.
8. Krylov, Vladimir Ivanovich (Arthur H. Stroud, transl.): Approximate Calculation of Integrals. Macmillan Co., c.1962.
9. Parlett, Beresford: Laguerre's Method Applied to the Matrix Eigenvalue Problem. Math. Comput., vol. 18, no. 87, July 1964, pp. 464-485.

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546