ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

LOW-ORDER GAUSS-CHRISTOFFEL QUADRATURE

by

I. K. Abu-Shumays

Applied Mathematics Division

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

February 1972

This document is
PUBLICLY RELEASABLE

Larry E. Williams

Authorizing Official

Date: 09/14/2006

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

leg

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Table of Contents

	<u>Page</u>
ABSTRACT	5
I. INTRODUCTION	5
II. THE ALGEBRAIC APPROACH AND THE COMPUTATION	9
III. APPLICATIONS	13
ACKNOWLEDGEMENTS	18
APPENDIX I	19
APPENDIX II	22
REFERENCES	24

List of Tables

<u>No.</u>	<u>Title</u>	<u>Page</u>
I.	Errors in Gauss-Legendre Quadrature	15
II.	Errors in Gauss-Christoffel Quadrature for Eq. (16)	17

LOW-ORDER GAUSS-CHRISTOFFEL QUADRATURE

I. K. Abu-Shumays

ABSTRACT

This report describes a Fortran IV program capable of generating, simply and cheaply, problem-oriented low-order ($n=1, \dots, 16$) Gaussian quadrature formulas. The program is applicable whenever the weight function is sufficiently smooth and has known moments. The approach is algebraic and utilizes available matrix inversion subroutines. Two examples are given; these include an application for the numerical integration of functions with logarithmic singularities.

I. INTRODUCTION

We are concerned here with Gauss-Christoffel quadrature formulas

$$\int_a^b w(x)f(x)dx \approx \sum_{i=1}^N w_i f(x_i) \quad (1)$$

where the weight function is non-negative, $w(x) \geq 0$, on the finite open interval $a < x < b$ and has known finite moments

$$\int_a^b w(x)x^\ell dx \equiv c_{\ell+1}, \quad \ell = 0, \dots, 2N-1. \quad (2)$$

The coordinates x_i and weights w_i are determined from Eqs. (1) and (2) by the requirement that Eq. (1) be exact for $f(x) = x^\ell$, $\ell = 1, \dots, 2N-1$.

The approximation (1) provides a "near optimum" [1,5,13,15] utilization of a fixed number of sample values of the integrand $f(x_i)$ and is applicable in the presence of integrable singularities of the weight function. Composites of integrals of the form (1) have been encountered in connection with variational problems in reactor mathematics (work in progress, see also [2] and [3]) with the weight function $w(x)$ possessing logarithmic singularities, whereas $f(x)$ is sufficiently smooth. In this and other applications, it is desirable to economize on computer storage by adopting sufficiently accurate (6-14 significant figures), low-order ($N=1, \dots, 10$) quadrature or composites of such quadrature. The need to minimize the number of points of evaluation becomes more important when multiple integrals are involved.

The approximation (1) of a definite integral by a finite sum has been widely investigated and extensive tables of coordinates x_i and weights w_i corresponding to special intervals (a,b) and weight functions $w(x)$ are available [1,14,15 and references therein]. For example, Stroud and Secrest [15] supply coordinates and weights evaluated to 30 significant figures for a variety of intervals, weight functions, and orders of quadrature N . They report that their high-order Gauss-Legendre formulas ($a = -1$, $b = 1$, $w(x) \equiv 1$) were computed in quadruple precision on a Control Data Corporation 1604 computer. The calculation for $N = 256$ took about one hour, while that for $n = 512$ took about four and one-half hours. Many authors [1] stress the difficulties inherent in generating high-order, high-precision weights and abscissae for a given weight function and interval; such difficulties inhibit one

from adapting the quadrature scheme to physical problems at hand unless the required tables happen to be available. The tables available in the literature [14,15] are extremely valuable for the commonly-used weight functions and intervals, such as for Gauss-Legendre quadrature. Yet it is highly desirable to cheaply produce problem-oriented quadrature. To illustrate one of the limitations of the tables, consider the following integral with weight function $w(x) = \log(1/x)$:

$$\int_0^a \log(1/x) f(x) dx \equiv a \log(1/a) \int_0^1 f(at) dt + a \int_0^1 \log(1/t) f(at) dt, \quad a \neq 1. \quad (3)$$

If the interval $[0,a]$ is used often in a certain application, then one has the option of developing Gauss-Christoffel coordinates and weights for the left-hand side of (3) or using the already available tables for the integrals on the right-hand side. Because there are two integrals on the right-hand side, it follows that by using the available tables one has to double the number of points (N points for each integral) of evaluation of $f(x)$ on $(0,a)$ in order to achieve the same accuracy as would be the case when using new tables for the left-hand side of (3).

The references [1,5,8,13,14,15] describe the various approaches that are used to evaluate coordinates and weights for the quadrature formula (1). The algebraic approach followed here has not been advocated for numerical work inasmuch as the relevant matrices may be ill-conditioned, especially for high-order quadrature.

Since we are interested only in low-order quadrature ($N=1, \dots, 16$), and in moderate accuracy (6-14 significant figures) difficulties due to ill-conditioning of the matrices involved either do not arise or are easily overcome by using matrix inversion algorithms especially designed for ill-conditioned non-singular matrices [4,10]. To achieve the maximum economy, the program given here adapts the matrix inversion subroutines already available in the Applied Mathematics Division ANL Subroutine Library [6,7]. These subroutines have been found adequate in our applications. As indicated above, severe ill-conditioning is not anticipated in most applications of the program, but if it does arise, then the program of E. H. Bareiss [4] is recommended for matrix inversion. This program is available, though not yet implemented in the ANL Subroutine Library.

If the weight function is given in tabular form and/or if the integrations (2) for the moments cannot be carried out analytically then the program developed by W. Gautschi [8,9,11] is recommended, since it is especially suited for these cases.

II. THE ALGEBRAIC APPROACH AND THE COMPUTATION

It is well known [5,15] that for $w(x) \geq 0$, the coordinates of the Gauss-Christoffel quadrature of each order N are the roots of the orthogonal polynomials

$$p_N(x) \equiv x^N + a_{N,N-1} x^{N-1} + \dots + a_{N,1} x + a_{N,0} \quad (4)$$

where the coefficients $a_{N,i}$ are related to the moments c_i of Eq. (2) by the matrix equation

$$\begin{bmatrix} c_1 & c_2 & \dots & c_N \\ c_2 & c_3 & \dots & c_{N+1} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ c_N & c_{N+1} & \dots & c_{2N-1} \end{bmatrix} \begin{bmatrix} a_{N,0} \\ a_{N,1} \\ \cdot \\ \cdot \\ a_{N,N-1} \end{bmatrix} = - \begin{bmatrix} c_{N+1} \\ c_{N+2} \\ \cdot \\ \cdot \\ c_{2N} \end{bmatrix} \quad (5)$$

Furthermore, the roots $x_{N,i}$, $i = 1, \dots, N$ of the orthogonal polynomials are all interior to the interval $[a, b]$ and possess the interlacing property

$$a < x_{N,1} < x_{N-1,1}; \quad x_{N-1,N-1} < x_{N,N} < b; \quad (6)$$

$$x_{N-1,i-1} < x_{N,i} < x_{N-1,i}, \quad i = 2, \dots, N-1.$$

The quadrature coordinates and weights for the special case $N = 1$, consistent with (1) and (2), are

$$x_{1,1} = c_2/c_1, \quad w_{1,1} = c_1. \quad (7)$$

For $N = 2$, the coefficients of the quadratic polynomial (4) as deduced from (5) are

$$a_{2,0} = \frac{c_2 c_4 - c_3^2}{c_1 c_3 - c_2^2}, \quad a_{2,1} = \frac{c_2 c_3 - c_1 c_4}{c_1 c_3 - c_2^2}. \quad (8)$$

Hence, the roots or quadrature coordinates for this case are

$$x_{2,1} = -([a_{2,1}^2 - 4a_{2,0}]^{1/2} + a_{2,1})/2, \quad (9)$$

$$x_{2,2} = ([a_{2,1}^2 - 4a_{2,0}]^{1/2} - a_{2,1})/2.$$

The corresponding quadrature weights (see Eq. (12) below for $N=2$) are

$$w_{2,1} = \frac{c_2 - c_1 x_{2,2}}{x_{2,1} - x_{2,2}}, \quad w_{2,2} = \frac{c_1 x_{2,1} - c_2}{x_{2,1} - x_{2,2}} \quad (10)$$

Equations (7)-(10) are simple analytic expressions used by the program to compute the Gauss-Christoffel quadrature coordinates and weights for $N = 1$ and 2 in terms of the first four moments (2). The program developed here is in double precision and is restricted to $N \leq 16$. For $N \geq 3$, the ANL Library Subroutine SYMINV [7] is used to compute the expansion coefficients $a_{N,i}$ of Eq. (4). The program checks the value of the determinant of the matrix in Eq. (5) and if this determinant is smaller than 10^{-20} or greater than 10^{20} , the program rescales the matrix equation by multiplying the matrix and the right-hand side of (5) by an appropriate factor. Numerical experiments indicate that such rescaling is sufficient to improve the accuracy of the solution of Eq. (5) by the subroutine

SYMINV, as well as to remove possible ill-conditioning for small N . Severe ill-conditioning is not anticipated for small N in general but, as mentioned above, if it does occur, we recommend replacing the SYMINV subroutine by the code of E. H. Bareiss [4]. If this code is not available, it is recommended that (5) be equilibrated and the solution be obtained by iterative improvement [16, p. 255].

Once the coefficients of the orthogonal polynomials (4) are determined, the program computes the roots (the coordinates of the quadrature) by Newton-Raphson iteration, starting (for $N \geq 3$) with initial values

$$\begin{aligned} x_{N,1}^0 &= (a + x_{N-1,1})/2, & x_{N,N}^0 &= (x_{N-1,N-1} + b)/2 \\ x_{N,i}^0 &= (x_{N-1,i-1} + x_{N-1,i})/2, & i &= 2, \dots, N-1. \end{aligned} \tag{11}$$

The iterations are continued until successive iterations for all the roots agree to within a tolerance EPS specified by the user. An arbitrary upper limit of 100 iterations is adopted (see the examples below). If the user desires the last two successive iterations to agree to d significant digits, he may set $\text{EPS} = 0.5 \times 10^{-d}$. To insure convergence to distinct roots (and this is essential here), the bounds given by the interlacing property (6) are checked at each stage of the iteration and, if necessary, estimates of the roots are adjusted. If, for example, at the k th iteration $x_{N,i}^k \geq x_{N-1,i}$, the program replaces the i th root ($i \neq 1, N$) $x_{N,i}^k$ by $(x_{N,i}^{k-1} + x_{N-1,i-1})/2$. If $x_{N,i}^k \leq x_{N-1,i}$, then this root is replaced by $(x_{N,i}^{k-1} + x_{N-1,i})/2$. A similar procedure is followed for $x_{N,1}^k$ and $x_{N,N}^k$.

The quadrature weights are related to the coordinates and to the first N moments by

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ x_{N,1} & x_{N,2} & \dots & x_{N,N} \\ x_{N,1}^2 & x_{N,2}^2 & \dots & x_{N,N}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1}^{N-1} & x_{N,2}^{N-1} & \dots & x_{N,N}^{N-1} \end{bmatrix} \begin{bmatrix} w_{N,1} \\ w_{N,2} \\ w_{N,3} \\ \vdots \\ w_{N,N} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_N \end{bmatrix} \quad (12)$$

The program computes the quadrature weights from Eq. (12) by using the ANL Library Subroutine MATINV [6]. Since the matrix in Eq. (12) is a Vandermonde matrix which is nonsingular for distinct coordinates $x_{N,i}$, this equation has a unique solution. Nevertheless, the program does allow for rescaling of Eq. (12), as was mentioned above in connection with the subroutine SYMINV.

III. APPLICATIONS

All the computations described above are carried out on the IBM S/360 in double precision arithmetic, though some of the results may be accurate only to single precision due to roundoff errors. The program is listed in Appendix I as SUBROUTINE GAUSSQ, and can be called by the statement

```
CALL GAUSSQ(N,XM,BL,BR,W,X,EPS,NPRINT)
```

where

- N = order of the quadrature desired ($N \leq 16$; $N \leq 7$ recommended);
- XM = user-supplied vector whose first $2N$ elements are the moments (Eq. (2)) of the chosen weight function (dimension at least $2N$);
- BL = the left boundary for integration, $BL = a$;
- BR = the right boundary for integration, $BR = b$;
- W = a vector of dimension at least $N(N+1)/2$ where the quadrature weights computed by the program are stored (see further explanation below);
- X = a vector of dimension at least $N(N+1)/2$ where the quadrature coordinates computed by the program are stored (see below);
- EPS = tolerance used to control the Newton-Raphson iteration. If $EPS = 10^{-d}$, the last two successive iterations used to compute each coordinate will agree to at least d decimal places;
- NPRINT = 0, no printout will result from the program; otherwise the output will include the number of iterations at each stage, the determinants and the scaling factor corresponding to the matrix Eqs. (5) and (12), and the errors to be expected in re-evaluating the first $2K$ moments ($K = 1, \dots, N$).

The program is designed to compute the quadrature coordinates and weights of all orders $K = 1, 2, \dots, N$ and store the results successively in the vectors X and W respectively. The K coordinates and the corresponding K weights for any fixed order of quadrature K are then the elements

$$X(K(K-1)/2 + I), W(K(K-1)/2 + I), I = 1, 2, \dots, K. \quad (13)$$

The advantage of the above is that the user can compare the quadratures of different orders before he selects the order most suitable for his application. If, on the other hand, the user supplies a negative value of N , then the program "changes the sign of N to positive" and returns only the quadrature coordinates and weights of order N , each stored as the first N elements

$$X(I), W(I), \quad I = 1, \dots, N. \quad (14)$$

The following examples illustrate the use of the program and indicate the degree of accuracy to be expected.

A. Example 1. Gauss-Legendre Quadrature

The Gauss-Legendre quadrature is the familiar special case with interval of integration $[BL, BR] = [-1, 1]$, weight function $w(x) \equiv 1$, and moments

$$c_{2k} = XM(2K) = 0, \quad c_{2k-1} = XM(2K-1) = 2/(2K-1). \quad (15)$$

Appendix IIA lists a program to call the subroutine GAUSSQ and compute the Gauss-Legendre coordinates and weights of orders $N = 1, 2, \dots, 16$ with

tolerances $EPS = .5 \times 10^{-6}$ and $EPS = .5 \times 10^{-12}$. The total time for compilation of the program and subroutine and their execution was CPU time = 8.44 sec. The computed values were compared with Table One of Ref. [15]. For $N = 1$ and $N = 2$, the results are accurate to double precision. Table I displays the errors involved for $N \geq 3$.

TABLE I. Errors in Gauss-Legendre Quadrature

Order N	Tolerance $EPS = .5 D-6^a$			$EPS = .5 D-12$		
	Maximum Error in			Maximum Error in		
	Coordinates	Weights	Moments ^b	Coordinates	Weights	Moments
3	.1D-12	.1D-11	.1D-12	.5D-15	.5D-15	.8D-16
4	.5D-15	.5D-15	.2D-15	.5D-15	.5D-15	.3D-15
5	.5D-15	.4D-14	.3D-15	.5D-15	.1D-14	.3D-15
6	.2D-14	.4D-14	.1D-14	.5D-15	.5D-15	.3D-15
7	.4D-14	.2D-13	.2D-14	.4D-14	.4D-14	.3D-15
8	.1D-13	.2D-13	.4D-14	.1D-13	.2D-13	.3D-15
9	.4D-13	.9D-13	.5D-14	.4D-13	.5D-13	.9D-15
10	.4D-12	.3D-12	.5D-14	.4D-12	.3D-12	.1D-14
11	.2D-11	.2D-11	.4D-14	.2D-11	.2D-11	.4D-14
12	.4D-11	.2D-11	.6D-14	.4D-11	.2D-11	.6D-14
13	.2D-11	.1D-11	.4D-14	.2D-11	.1D-11	.3D-14
14	.1D-9	.8D-10	.2D-13	.1D-9	.8D-10	.6D-14
15	.2D-8	.1D-8	.1D-13	.2D-8	.1D-8	.1D-13
16	.6D-8	.3D-8	.4D-13	.6D-8	.3D-8	.3D-13

^aD-6 stands for multiplicative factor 10^{-6} . The same holds for the other floating numbers listed.

^bThe moments are recomputed on the basis of the corresponding abscissas and weights and the results are compared with their exact values.

The numbers of Newton-Raphson iterations needed to compute the coordinates were 4 iterations or less for tolerance $\text{EPS} = .5 \times 10^{-6}$ and 5 iterations or less for tolerance $\text{EPS} = .5 \times 10^{-12}$. Table I indicates that the improvement achieved in the second computation of coordinates and weights (with $\text{EPS} = .5 \times 10^{-12}$) is slight but noticeable for small $N \leq 10$ and is negligible for $N > 10$. This result is expected because (a) the evaluations of the coordinates are improved for small N , and (b) the errors in the evaluation of the expansion coefficients dominate for larger values of N (this is inherent in the method used for matrix computations [4,6,7]). We note from the columns of Table I labeled moments that the accuracy to be expected by applying the above quadrature to integrals of polynomials of degrees $2N$ surpasses the accuracy of the coordinates and weights (in fact, the accuracy is much improved for polynomials of degree N , as is expected).

B. Example 2. A Weight Function with a Logarithmic Singularity

Appendix IIB lists a program to compute the Gauss-Christoffel quadrature for the integral

$$\int_0^1 \log(1/x) f(x) dx, \quad w(x) = \log(1/x). \quad (16)$$

The moments in this case are

$$c_k = XM(k) = K^{-2}. \quad (17)$$

The computed values were compared with Table 3 of [14] and Table NINE of [15]. Table II displays the errors involved for $N \geq 3$.

Table II. Errors in Gauss-Christoffel Quadrature for Eq. (16)

Order N	EPS = .5 D-6			EPS = .5 D-12		
	Maximum Error in			Maximum Error in		
	Coordinates	Weights	Moments	Coordinates	Weights	Moments
3	.1D-14	.5D-15	.1D-16	.5D-15	.5D-15	.4D-16
4	.4D-12	.1D-11	.1D-13	.2D-14	.4D-14	.9K-16
5	.1D-12	.8D-13	.6D-16	.1D-12	.8D-13	.3D-16
6	.5D-11	.6D-11	.6D-15	.5D-11	.6D-11	.6D-16
7	.3D-10	.4D-10	.2D-14	.3D-10	.4D-10	.2D-14
8	.2D-8	.1D-8	.1D-14	.2D-8	.1D-8	.2D-14
9	.4D-7	.3D-7	.2D-14	.4D-7	.3D-7	.1D-13
10	.2D-5	.1D-5	.4D-13	.2D-5	.1D-5	.2D-13

The comments made regarding the results of the program for Gauss-Legendre quadrature apply here also. In this case, however, the error due to roundoff for the matrix computation (using SYMINV) dominates for $N \geq 7$. The coordinates and weights for $N = 10$ are accurate to 6 decimal places and the accuracy improves for smaller N . The results here for $N \leq 5$ are superior to those given in Ref. [11], using ANL Library Subroutine GAUSS, where the coordinates are accurate to within 4.47×10^{-5} and the weights to within 2.65×10^{-5} . (No systematic comparison has been carried out with the program GAUSS, which is most suitable and highly recommended when the integrals for the moments cannot be performed analytically.)

Finally, as in the previous example, the accuracy of the quadrature when integrating polynomials surpasses that of the coordinates and weights; to be specific, for $N = 10$, $\text{EPS} = .5 \times 10^{-12}$, the accuracy in approximating the integrals of x^n by quadrature is to within $.5 \times 10^{-16}$ for $n \leq 10$ and to within $.3 \times 10^{-13}$ for $n \leq 20$. Equally good results were also obtained for weight functions $w(x) = x^i \log^j(1/x)$, $i, j \leq 3$, and are omitted for brevity.

ACKNOWLEDGEMENTS

The author is grateful for valuable discussions with E. H. Bareiss, W. J. Cody, and W. Gautschi. Special thanks are due to W. J. Cody for his help in documenting the computer programs.

APPENDIX I

FORTRAN IV DOUBLE PRECISION GAUSS-CHRISTOFFEL QUADRATURE SUBROUTINE

```

      SUBROUTINE GAUSSQ(N,XM,BL,BR,W,X,EPS,NPRINT)
C
C   THIS SUBROUTINE CONSTRUCTS GAUSSIAN WEIGHTS AND COORDINATES
C   FOR NUMERICAL INTEGRATION OF F(X) BETWEEN X=BL AND X=BR WITH
C   WEIGHT FUNCTION WT(X) WHEN THE MOMENTS XM(I) OF X**(I+1) ARE GIVEN
C
C   SUBROUTINE RETURNS VECTORS X(K), W(K) FOR K= 1,...,M, M= 1,...,N
C   *** IF N<0 THE SUBROUTINE SETS N=-N AND RETURNS ONLY X(K), W(K)
C   FOR K=1,...,N ***
C   EPS INDICATES THE DESIRED DEGREE OF PRECISION IN X(I)
C   NPRINT=0    NO PRINT OUT, OTHERWISE OUT PUT WILL INCLUDE
C   NUMBER OF ITERATIONS AT EACH STAGE, DETERMINANTS, ERRORS TO BE
C   EXPECTED IN EVALUATING THE FIRST K MOMENTS (K =1,2N), ETC.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(1),W(1),XM(1),Y(17),C(17,17),C1(17,17),C2(17,17)
C
      NQUAD =N
      IF(N.GT.0)GO TO 5
      N =-N
      NQUAD =0
5    DO 10 K=1,N
      C2(1,K)=1.0D0
      DO 10 L=1,N
10   C(K,L)=XM(K+L-1)
      X(1) = XM(2)/XM(1)
      W(1) = XM(1)
      SQ =XM(1)*XM(3)-XM(2)**2
      Y(1) = (XM(2)*XM(4) -XM(3)**2)/SQ
      Y(2) = (XM(2)*XM(3) -XM(1)*XM(4))/SQ
      SQ =DSQRT(Y(2)**2 -4*Y(1))
      X(2) = -(SQ +Y(2))/2
      X(3) =  (SQ -Y(2))/2
      SQ = X(2)- X(3)
      W(2) =(XM(2)-XM(1)*X(3))/SQ
      W(3) =(-XM(2)+XM(1)*X(2))/SQ
      IF(N.LE.2)GO TO 100
      N2 = 2
      N3 = 3
      IF(NPRINT.EQ.0) GO TO 15
      WRITE(6,2000)
15   SCALE1 = 2
      SCALE2 = 2
      DO 50 N1=3,N
      II=0

```

```

DX0 =1.0/N1
N1M1 =N1-1
X(N3+1) = (X(N2) +BL)/2
X(N3+N1) = (X(N3) +BR)/2
DO 20 K=2,N1M1
20 X(N3+ K) =(X(N2+K-1)+ X(N2+ K-2))/2
21 DO 22 I=1,N1
Y(I)=-XM(N1+I)*SCALE1
DO 22 J=1,N1
22 C1(I,J) = C(I,J)*SCALE1
CALL SYMINV(C1,N1,Y,1,DET,17)
DET =DABS(DET)
IF((DET.LT.1.0D+20).AND.(DET.GT.1.0D-20))GO TO 23
IF(DET.GT.1.0D+20)SCALE1 =SCALE1/2
IF(DET.LT.1.0D-20)SCALE1 =SCALE1*4
GO TO 21
23 IF(NPRINT.EQ.0) GO TO 24
NO = 1000000
WRITE(6,3000)N1,NO,DET,SCALE1
24 DO 25 K=1,N1M1
25 C1(3,K) =K*Y(K+1)
C1(3,N1)= N1
N2 = N3+1
N3 = N3+N1
30 DO 34 K=N2,N3
ERR=0.0D0
FX = (X(K) +Y(N1))*X(K) + Y(N1M1)
F1X = X(K)*C1(3,N1) +C1(3,N1M1)
DO 32 I=2,N1M1
FX = X(K)*FX +Y(N1-I)
32 F1X =X(K)*F1X +C1(3,N1-I)
DX =-FX/F1X
X(K) =X(K) +DX
IF((K.EQ.N2).AND.(X(K).LT.BL)) X(K) = (X(K)-DX+X(K-N1+1))/2
IF((K.EQ.N2).AND.(X(K).GT.X(K-N1+1)))X(K) = (X(K)-DX+BL)/2
IF((K.EQ.N3).AND.(X(K).GT.BR)) X(K) = (X(K)-DX+X(K-N1))/2
IF((K.EQ.N3).AND.(X(K).LT.X(K-N1)))X(K) = (X(K)-DX+BR)/2
IF((K.EQ.N2).OR.(K.EQ.N3))GO TO 33
IF((K.GT.N2).AND.(X(K).LT.X(K-N1)))X(K) = (X(K)-DX+X(K-N1+1))/2
IF((K.LT.N3).AND.(X(K).GT.X(K-N1+1)))X(K) = (X(K)-DX+X(K-N1))/2
33 DX = DABS(DX)
IF(DX0.LT.DX) DX = DX0
34 IF(ERR.LT.DX) ERR = DX
II = II+1
IF((ERR.GT.EPS).AND.(II.LT.100))GO TO 30
DO 40 L = 1,N1
C2(2,L) = X(N2+L-1)
DO 40 K = 3,N1
40 C2(K,L) = C2(K-1,L)*C2(2,L)
42 DO 44 I = 1,N1

```

```

      Y(I) = SCALE2*XM(I)
      DO 44 J = 1,N1
44    C1(I,J) = C2(I,J)*SCALE2
      CALL MATINV(C1,N1,Y,1,DET,17)
      DET =DABS(DET)
      IF((DET.LT.1.0D+20).AND.(DET.GT.1.0D-20))GO TO 46
      IF(DET.GT.1.0D+20)SCALE2 =SCALE2/2
      IF(DET.LT.1.0D-20)SCALE2 =SCALE2*2
      GO TO 42
46    IF(NPRINT.EQ.0) GO TO 47
      WRITE(6,3000)N1,I1,DET,SCALE2
47    DO 48 K = 1,N1
48    W(N2+K-1) = Y(K)
      IF(NPRINT.EQ.0) GO TO 50
      DO 49 K = 1,N1
      C1(K,1)= XM(K)
      C1(K,2)= XM(K+N1)
      DO 49 L = 1,N1
      C1(K,2) = C1(K,2) - C2(K,L)*C2(2,L)*C2(N1,L)*Y(L)
49    C1(K,1) = C1(K,1) - C2(K,L)*Y(L)
      WRITE(6,4000)(C1(L,1),L=1,N1), (C1(K,2),K=1,N1)
      50 CONTINUE
100   IF(NQUAD.GT.0)GO TO 1000
      N2 =N*(N-1)/2
      DO 200 I=1,N
      X(I) =X(N2+I)
      200 W(I) =W(N2+I)
1000 CONTINUE
2000 FORMAT('/' N,ITERATIONS, DETERMINANT, SCALING FACTOR FOR AX=B (2
1 ROWS) / FOLLOWED BY ERRORS IN THE 1ST 2N MOMENTS')
3000 FORMAT(2I5,2G20.5)
4000 FORMAT(8G15.2)
      RETURN
      END

```

APPENDIX II

FORTRAN IV DOUBLE PRECISION PROGRAM FOR EXAMPLES 1 AND 2

A. GAUSS-LEGENDRE QUADRATURE

```

C      THIS PROGRAM CONSTRUCTS GAUSSIAN WEIGHTS AND COORDINATES
C      TO INTEGRATE F(X) FROM X=-1 TO X=1
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(138),W(138),XM(34)
C
      NPRINT = 1
      N = 16
      BL = -1.0D0
      BR = 1.0D0
      EPS = 0.5D-6
      DO 50 III = 1, 2
      WRITE(6,1000)N, EPS
      DO 10 K = 1, N
      XM(2*K-1) = 2.0D0/(2*K-1)
10    XM(2*K) = 0.0D0
      CALL GAUSSQ(N,XM,BL,BR,W,X,EPS,NPRINT)
      K1 = 1
      K2 = 1
      WRITE(6,100)
      DO 20 K = 1, N
      WRITE(6,1000)(K,X(I),W(I),I=K1,K2)
      K1 = K2 + 1
      K2 = K1 + K
20    WRITE(6,2000)
50    EPS = 0.5D-12
100   FORMAT(/' GAUSSIAN COORDINATES AND WEIGHTS, I=1,...,N
1      1'/'      N              X(I)              W(I)')
1000  FORMAT(15,2G30.15)
2000  FORMAT(//)
      RETURN
      END

```

B. WEIGHT FUNCTION WITH A LOGARITHMIC SINGULARITY

```

C      THIS PROGRAM CONSTRUCTS GAUSSIAN WEIGHTS AND COORDINATES
C      FOR LOG(1/X) USING SUBROUTINE GAUSSQ
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(120),W(120),XM(34)
C
      NPRINT = 1
      N = 10
      BL =0.0D0
      BR =1.0D0
      EPS=0.5D-6
      WRITE(6,1000)N,EPS
      N1= 20
      DO 10 K=1,N1
100  XM(K) =1.0D0/K**2
      CALL GAUSSQ(N,XM,BL,BR,W,X,EPS,NPRINT)
      K1=1
      K2=1
      WRITE(6,100)
      DO 20 K=1,N
      WRITE(6,1000)(K,X(I),W(I),I=K1,K2)
      K1=K2+1
      K2=K1+K
20  WRITE(6,2000)
100  FORMAT(/' GAUSSIAN COORDINATES AND WEIGHTS, I=1,...,N
1'/'      N              X(I)              W(I)')
1000 FORMAT(I5,2G30.15)
2000 FORMAT(//)
      RETURN
      END

```

REFERENCES

1. D. G. Anderson, "Gaussian quadrature formulae for $\int_0^1 -\ln(x)f(x)dx$," Math. Comput., 19 (1965), 477-481.
2. E. H. Bareiss, "Numerical Solution of the Transport Equation," in Numerical Solution of Integral Equations, SIAM Proceedings (to be published).
3. E. H. Bareiss, "Numerical Approaches to the Transport Equation and Computational Experimentation," Proceedings of Second Conference on Transport Theory, CONF-710107, Los Alamos Scientific Laboratory, 1971.
4. E. H. Bareiss, "Computational Solutions of Matrix Problems over an Integral Domain," Argonne National Laboratory AMD Report TM-222, 1970.
5. P. J. Davis and P. Rabinowitz, Numerical Integration, Ginn Blaisdell Pub. Co., Waltham, Massachusetts, 1967.
6. B. S. Garbow, "MATINV, Matrix Inversion with Accompanying Solution of Linear Equations," Argonne National Laboratory, Applied Mathematics Division SYSTEM/360 Library Subroutine ANL F402S, 1967.
7. B. S. Garbow, "SYMINV, Symmetric Matrix Inversion with Accompanying Solution of Linear Equations," Argonne National Laboratory, Applied Mathematics Division SYSTEM/360 Library Subroutine ANL F152S, 1968.
8. W. Gautschi, "Construction of Gauss-Christoffel quadrature formulas," Math. Comput., 22 (1968), 251-270.
9. W. Gautschi, "Algorithm 331, Gaussian quadrature formulas," Comm. ACM, 11 (1968), 432-436.
10. G. H. Golub, "Numerical methods for solving linear least squares problems," Numer. Math., 7 (1965), 206-216.

11. K. Hillstrom, "GAUSS, OS/360-Fortran IV routine for Gaussian quadrature formula generation," Argonne National Laboratory, Applied Mathematics Division SYSTEM/360 Library Subroutine ANL D161S, 1968 (Based on Ref. 9 above).
12. J. Kaganove and D. Rocke, "GAUSSLGN, Legendre-Gauss Quadrature," Argonne National Laboratory, Applied Mathematics Division CDC 3600 Library Subroutine ANL D159, 1968.
13. V. I. Krylov, Approximate Calculation of Integrals. New York, Macmillan 1962.
14. V. I. Krylov and A. A. Pal'tev, "Tables for Numerical Integration of Functions with Logarithmic and Power Singularities," Translated from Russian by IPST Staff, Jerusalem, 1971.
15. A. H. Stroud and D. Secrest, Gaussian Quadrature Formulas, Prentice-Hall, Inc., 1966.
16. J. H. Wilkinson, "The Algebraic Eigenvalue Problem," Clarendon Press, Oxford 1965.