MATHEMATICA — REVUE D'ANALYSE NUMÉRIQUE
ET DE THÉORIE DE L'APPROXIMATION
L'ANALYSE NUMÉRIQUE ET LA THÉORIE DE L'APPROXIMATION
Tome 9, N° 1, 1980, pp. 81–92

NUMERICAL INTEGRATION BY MEANS OF GAUSS-LEGENDRE FORMULA

by

ALEXANDRU LUPAŞ
(Sibiu)

1. Introduction. The aim of this paper is to construct an algorithm written in FORTRAN — IV for the composite Gauss-Legendre quadrature formula. This FORTRAN algorithm consists of a commended subprogram which furnishes us the Gauss rule of integration constructed at the roots of the Legendre polynomial of the degree N, N being arbitrary.

If N is a fixed natural number let us denote by x_i, $i = 1, 2, \ldots, N$, $-1 < x_N < \ldots < x_2 < x_1 < +1$, the roots of the Legendre polynomial P_N which may be defined by the recursion relation

$$P_0(x) = 1, \quad P_1(x) = x$$

$$P_j(x) = \frac{2j-1}{j} x P_{j-1}(x) - \frac{j-1}{j} P_{j-2}(x), \quad j = 2, 3, \ldots, N.$$

For an integrable function $f\colon [a, b] \to \mathbf{R}$ the Gauss-Legendre formula may be written as

$$\int_a^b f(x)\,dx = \frac{b-a}{2} \sum_{k=1}^{N} c_k f\left(\frac{a+b}{2} + \frac{b-a}{2} x_k\right) + R(f) \tag{1}$$

where $c_1, \ldots, c_N$ are the Christoffel weights defined as

$$c_k = \frac{2}{(1-x_k^2)\,|P_n'(x_k)|^2},$$

and $R(f) = R(f; [a, b], N)$ is the remainder. It is known that if f belongs to the space $C^{(2N)}[a, b]$ then there exists at least a point θ, $\theta \in [a, b]$, such that (see [3], [5]).

$$R(f) = \frac{(b-a)^{2N+1}}{2N+1}\binom{2N}{N}^{-2}\frac{f^{(2N)}(\theta)}{(2N)!}. \tag{2}$$

The simplest composite formula is generated by repeated application of the quadrature formula (1) at each subinterval $[a_i, a_{i+1}]$, $a_i = a + \frac{i-1}{L}(b-a)$, $i = 1, 2, \ldots, L$. Let us denote

$$H = \frac{b-a}{2L}, \quad h_i = a + (2i-1)H, \quad M = \left[\frac{N}{1}\right],$$
$$T_{ik}(f) = f(h_i + Hx_k) + f(h_i - Hx_k).$$

In terms of L, the number of applications of the formula (1), and a and b, the integration limits, the composite formula is

$$\int_a^b f(x)\,dx = H\left[c(N)\sum_{i=1}^{L} f(h_i) + \sum_{k=1}^{M} c_k \sum_{i=1}^{L} T_{ik}(f)\right] + R_{N,L}(f) \tag{3}$$

where

$$c(N) = \begin{cases} 0 & if \quad N = 2M \\ \frac{2^{2N-1}}{N^2}\binom{2M}{M}^{-2} & if \quad N = 2M+1 \end{cases}$$

2. The remainder $R_{N,L}$. If $f \in C^{(2N)}[a, b]$ then from (1) — (2) it follows that there are the points $\theta_1, \ldots, \theta_L$, $\theta_i \in [a_i, a_{i+1}]$, such that

$$R_{N,L}(f) = \frac{(2H)^{2N+1}}{2N+1}\binom{2N}{N}^{-2}\frac{1}{(2N)!}\sum_{i=1}^{L} f^{(2N)}(\theta_i).$$

Using the inequalities

$$\frac{4^n}{\sqrt{\pi n + a}} < \binom{2n}{n} < \frac{4^n}{\sqrt{\pi n + b}}, \quad a = \frac{8}{9}, \quad b = \frac{\pi}{4}, \; n > 2,$$

one proves the following

Lemma 1. *a) If $f \in C^{(2N)}[a, b]$ then there exists θ, $\theta \in [a, b]$, such that*

$$R_{N,L}(f) = \frac{(2H)^{2N+1}}{2N+1}\binom{2N}{N}^{-2} \cdot L \cdot \frac{f^{(2N)}(\theta)}{(2N)!}; \tag{4}$$

b) For $f \in C^{(2N)}[a, b]$ we have

$$|R_{N,L}(f)| < \frac{2\pi}{(2N)!}\left(\frac{H}{2}\right)^{2N+1} ||f^{(2N)}||, \quad N > 2,$$

where $||\cdot||$ denotes the uniform norm.

In the following we find the remainder $R_{N,L}$ on the space of continuous functions. The following notation is used:

$$e_k(t) = t^k, \qquad k = 0, 1, \ldots,$$

$$Q_{n,x}(t) = |t - x|_+^n = \begin{cases} (t-x)^n & if \; t \geq x, \\ 0 & if \; t < x, \end{cases}$$

$[z_1, z_2, \ldots, z_n; f]$ — denotes the divided difference at the knots z_i, $i = 1, 2, \ldots, n$.

Let us consider a bound linear functional $R : C[a, b] \to \mathbf{R}$. If for each $h \in C^{(n+1)}[a, b]$ there exists a point $\theta = \theta(h) \in [a, b]$ for which

$$R(h) = \frac{h^{(n+1)}(\theta)}{(n+1)!} R(e_{n+1}), \quad R(e_{n+1}) \neq 0,$$

then we say that R has a *simple form on the space* $C^{(n+1)}[a, b]$. The functional R has a *simple form on* $C[a, b]$ if for each $f \in C(a, b]$ there is a system $\theta_1, \ldots, \theta_{n+2}$ of distinct points from $[a, b]$ such that

$$R(f) = R(e_{n+1}) \cdot [\theta_1, \theta_2, \ldots, \theta_{n+2}; f].$$

THEOREM 2. *If $R : C[a, b] \to \mathbf{R}$ is a bounded linear functional which has a simple form on the subspace $C^{(n+1)}[a, b]$ then R has a simple form on the whole space $C[a, b]$.*

Proof. If R verifies the prescribeb hypothesis then

$$R(e_k) = 0 \quad \text{for} \quad k = 0, 1, \ldots, n. \tag{5}$$

According to Peano (see [1], theorem 3.7.1) the equalities (5) imply that

$$R(h) = \int_a^b h^{(n+1)}(x) K(x)\,dx, \quad h \in C^{(n+1)}[a, b], \tag{6}$$

where

$$K(x) = \frac{1}{n!} R_t(|t - x|_+^n) = \frac{1}{n!} R(Q_{n,x}).$$

The continuity of K on $[a, b]$ implies that the function

$$h_0(x) = \int_a^x \int_a^{x_1} \cdots \int_a^{x_n} [K(x_{n+1}) - |K(x_{n+1})|]\,dx_1\,dx_2 \ldots dx_{n+1}, \; x \in [a, b],$$

verifies

$$h_0 \in C^{(n+1)}[a, b], \quad h_0^{(n+1)}(x) = K(x) - |K(x)|.$$

Without loss of generality, say $R(e_{n+1}) > 0$. From the behaviour of the functional R on the space $C^{(n+1)}[a, b]$ it follows that there is a point θ on $[a, b]$ such that

$$R(h_0) = [K(\theta) - |K(\theta)|]\,\frac{R(e_{n+1})}{(n+1)\,!} \le 0.$$

On the other hand, from (6)

$$R(h_0) = \int_a^b [K^2(x) - K(x)|K(x)|]\,dx = \frac{1}{2}\int_a^b (K(x) - |K(x)|)^2\,dx \ge 0.$$

Therefore $R(h_0) = 0$, i.e.

$$\int_a^b [K(x) - |K(x)|]^2\,dx = 0$$

or $K(x) = |K(x)|$, $x \in [a, b]$. In conclusion, K is non-negative on $[a, b]$. Therefore, the bounded linear functional $R : C[a, b] \to \mathbf{R}$ has the properties

i) $R(e_k) = 0$ *for* $k = 0, 1, \ldots, n$, $\quad R(e_{n+1}) > 0$,

ii) $R(Q_{n,x}) \ge$ *for* $x \in [a, b]$.

An earlier result by T. POPOVICIU ([7], theorem 12) asserts that in this case R has a simple form on the space $C[a, b]$. This completes the proof (see also [4]).

The equality (4) enables us to assert that $R_{N,L} : C[a, b] \to \mathbf{R}$ which is a bounded linear functional ([3]), has a simple form on the subspace $C^{(2N)}[a, b]$. From the above theorem we conclude with the following:

THEOREM 3. *If* f: $[a, b] \to \mathbf{R}$ *is continuous on* $[a, b]$ *then there is a system* $\theta_1, \theta_2, \ldots, \theta_{2N+1}$ *of distinct points from* $[a, b]$ *such that*

$$R_{N,L}(f) = L \cdot \frac{(2H)^{2N+1}}{2BN+1}\binom{2N}{N}^{-2} \cdot [\theta_1, \theta_2, \ldots, \theta_{2N+1}; f].$$

3. The roots of the Legendre polynomial

$$I_j = \left(\cos\frac{j\pi}{N+1},\ \cos\frac{4j-1}{4N+2}\,\pi\right).$$

If x_i, $-1 < x_N < \ldots < x_2 < x_1 < +1$, are the roots of the Legendre polynomial P_N, then it is known that (see [9])

$$x_j \in I_j,\ x_{N+1-j} = -x_j,\ j = 1, 2, \ldots, M.$$

Moreover $I_j \cup I_i = \varnothing$, $i \ne j$. In order to find the roots $x_1, \ldots, x_M$ we have used the so-called „square-root iteration method" ([6]), namely

$$x_j^{(0)} = \cos\frac{4j-1}{4N+1}\,\pi,\quad x_j^{(0)} \in I_j,\ 1 \le j \le M,$$
$$x_j^{(k)} = F(x_j^{(k-1)}),\qquad k = 1, 2, \ldots,$$

where

$$F(x) = x - \frac{P_N(x)}{P_N'(x)\sqrt{1 - \dfrac{P_N(x)P_N''(x)}{P_N'(x)^2}}}.$$

An iteration $x_j^{(k_0)}$ was considered as a „good approximation" of the root x_j if one of the following inequalities

$$|x_j^{(k_0)} - x_j^{(k_0-1)}| \le 10^{-18},\ |P_N(x_j^{(k_0)})| \le 10^{-20},\ k_0 \ge 10.$$

was verified. This iteration method was tested for $N = 2{,}200{,}1$. We note that the first inequality was satisfied for $k_0 = 3$. On the other hand, for all values of N

$$|P_N(x_j^{(3)})| \le 10^{-15},\quad j = 1, 2, \ldots, M.$$

By means of these approximative values of the roots there are determined the corresponding Christoffel weights c_j. Let us denote

$$S_N(k) = \frac{1 - (-1)^{k+1}}{k+1} - \sum_{j=1}^{N} (x_j)^k \cdot c_j = R(e^k;\ [-1, +1], N).$$

Taking into account that the quadrature formula (1) is exact for all polynomials of the degree $\le 2N - 1$, in order to test the accuracy of the iterative method, we have asked if the values $S_N(2k)$, $k = 0, N - 1$, 1, are zero; note that $S_N(2k - 1) = 0$ for $1 \le k \le N$. Using this precision criterion it was observed that

$$|S_N(2k) \le 10^{-14},\ k = 0, 1, \ldots, N - 1.$$

4. FORTRAN implementation.

Let us denote

$$I(f) = \int_a^b f(x)\,dx$$

$$G(N, L, f) = H \cdot \left[c(N)\sum_{i=1}^{L} f(h_i) + \sum_{k=1}^{M} c_k \sum_{i=1}^{L} T_{ik}(f)\right].$$

The composite quadrature formula (3) furnishes us the approximation

$$I(f) \approx G(N, L, f).$$

In order to evaluate $I(f)$ we have constructed a subroutine

GAUSS1 (F, A, B, N, L, EPS, X, W, R, ITER, KOD)

where the input parameters are:

- FON = the function f which must be integrated
- A, B = the integration limits a, b
- N = degree of the Legendre polynomial
- L = the (initial) number of applications of the Gauss-Legendre quadrature formula
- EPS = a tolerance value
- ITER = the maximum number of iterative cycles permitted, i.e., L (final) $\leq$ ITER

The output parameters are:

- R = the final approximative value for $I(f)$
- X = a vector of dimension $1+\left[\frac{N}{2}\right]$ which contains the non-negative roots $X(1) > X(2) > \ldots > X(N1)$ of the Legendre polynomial; Note that $N1 = [N/2]$ if N is even and $N1 = 1 + [N/2]$ for N odd.
- W = an array of dimension $1 + [N/2]$ which includes the Christoffel numbers $W(I) := c_i$ corresponding to $X(I)$
- KOD = an error code, KOD $\in \{0, 1\}$.

The program was compiled and executed on a FELIX C-256 computer using double precision arithmetic. The principal steps of this FORTRAN algorithm are:

1) Test if $N > 3$. If this is false then $N := 4$;
2) Check for $0 < \text{EPS} < 10^{-7}$. If these inequalities are not satisfied, then EPS $:= 10^{-8}$;
3) Compute an initial approximation RINIT for $I(f)$. More precisely RINIT $:= G(3, 1, f)$;
4) Compute the (approximative) roots $X(1)$, $X(2)$, $\ldots$, $X(N1)$ of the Legendre polynomial as well as the corresponding Christoffel weights $W(1)$, $W(2)$, $\ldots$, $W(N1)$;
5) Calculate the approximation R of the integral $I(f)$. This means that $R := G(N, L, f)$;
6) Test of accuracy: more precisely check for

(7) $$|R - \text{RINIT}| \leq \text{EPS} \min(1, |R|).$$

If (7) is verified then KOD = 0 and RETURN.

7) Iteration of the integration method: when the accuracy criterion (7) is not satisfied then test if $2L <$ ITER: if this is true then $L := 2L$, RINIT $:= R$ and a new approximation R is constructed, i.e., $R := G(2L, N, f)$. This iteration proceeds until the accuracy test (7) is passed or until L, the number of iterations, exceeds ITER. If $2L \geq$ ITER and (7) is not verified then KOD = 1 and RETURN. Note that L may be destroyed during computation; in the following we shall denote by LFINAL the final value of L.

The program listing is the following:

```
      SUBROUTINE GAUSS1 (F,A,B,N,L,EPS,X,W,R,ITER,KOD)
C     = = = = = = = = = = = = = = = = = = = = = = =
C      PURPOSE:
C         THE APPROXIMATIVE INTEGRATION OF A FUNCTION
C         F ON AN INTERVAL (A,B)
C
C
C      METHOD:
C         GAUSS-LEGENDRE QUADRATURE FORMULA
C
C
C      USAGE:
C         CALL GAUSS1 (F,A,B,N,L,EPS,X,W,R,ITER,KOD)
C
C
C      DESCRIPTION OF PARAMETERS:
C         F    - THE FUNCTION WHICH IS INTEGRATED
C         A,B  - ENDPOINTS OF THE INTERVAL OF INTE-
C                GRATION
C         N    - THE DEGREE OF THE LEGENDRE POLY-
C                NOMIAL
C         L    - NUMBER OF EQUIDISTANT SUBINTERVALS
C                FROM (A,B)
C         EPS  - A TOLERANCE VALUE
C         X    - VECTOR OF DIMENSION 1+N/2 WHICH CON-
C                TAINS THE NON-NEGATIVE ROOTS OF THE
C                LEGENDRE POLYNOMINAL
C         W    - AN ARRAY OF DIMENSION 1+N/2 WHICH
C                INCLUDES THE CHRISTOFFEL NUMBERS
C         R    - THE APPROXIMATIVE VALUE OF THE
C                INTEGRAL
C         ITER - THE MAXIMUM NUMBER OF ITERATIVE
C                CYCLES PERMITTED
C         KOD  - AN ERROR CODE : KOD = 0 MEANS THAT
C                A CERTAIN ACCURACY TEST IS VERIFIED.
C                OTHERWISE, KOD = 1.
C
C
C       INPUT PARAMETERS : F,A,B,N,L,EPS,ITER
C       OUTPUT PARAMETERS: X,W,R,KOD
C
C       REMARKS:
C         (1) L IS DESTROYED DURING COMPUTATION
C         (2) A DECLARATION EXTERNAL F MUST BE USED
C             BEFORE CALL GAUSS1 (F, A, B, N, L, EPS, X, W, R,
C             ITER, KOD)
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(1), W(1)
C        = = = TEST ON N = = =
      IF(N-3) 1,1,8
 1    N = 4
C        = = = TEST ON EPS = = =
 8    EPX=EPS - .1D-6
      IF(EPS*EPX) 2,9,9
 9    EPS = .1D-7
C        =+= THE INITIAL VALUE OF R, NAMELY RINIT, IS
      CONSTRUCTED=+=
 2    H1 = (B-A)*0.5
      P = (A+B)*0.5
      ROOT=H1*0.774596669241483
      P1 = P + ROOT
      P2 = P - ROOT
      RINIT = (F(P1)+F(P2))*0.555555555555556
      RINIT = RINIT + 0.888888888888889*F(P)
C        =+= INITIAL VALUES =+=
      M = N/2
      TN=DEFLOAT(N)
      AN = 1./TN
      CSI = 1./(4.*TN + 1.)
      DEV = (1. + TN)*TN
C        =+= START OF THE ITERATION METHOD WHICH
         FURNISHES US
C     THE ROOTS X(1),...,X(M) AND THE WEIGHTS W(1),...,
      W(M) =+=
      DO 100 K = 1,M
        TED = (4.*DFLOAT(K)-1.)*CSI
        V = TED*3.1415926535897932
        X(K) = DCOS(V)
        NTER = 0
 300    P2 = X(K)
        P1 = 1.
C     = = COMPUTE P3, THE VALUE OF THE LEGENDRE
C          POLYNOMIAL PN(X) AT X(K) = =
        DO 70 IT = 2,N
        ZI = 1./DFLOAT(IT)
        P3 = (2.-ZI)*X(K)*P2 - (1.-ZI)*P1
        P1 = P2
        P2 = P3
 70     CONTINUE
C     = = CALCULATE THE SUCCESIVE APPROXIMATION OF
C          THE ROOT X(K) = =
        U = 1. - X(K)*X(K)
        U1 = P3*X(K) - P1
```



```
        Q = U*AN/U1
        GW = P3*Q
        DE = U + 2.*X(K)*GW + GW*GW*DEV
        DER = DABS(U)/DE)
        EPSI = GW*DSQRT(DER)
        IF(DABS(P3) - .1D-19) 100,100,5
 5      IF(DABS(EPSI)- .1D-17) 100,100,6
 6      IF(NTER-10) 7,7,100
 7      NTER = NTER + 1
        X(K) = X(K) + EPSI
        GO TO 300
 100    W(K) = 2.*Q*Q/U
C     = = CALCULATE THE APPROXIMATION R OF THE
        INTEGRAL = = IF(2*M - N) 3,4,3
 3      AM = 1.
        DO 50 I=1,M
        TIX = 2.*DFLOAT(I) - 1.
 50     AM = AM*(1. + 1./TIX)
        NM = M + 1
        W(NM) = AM*AM*AN*AN
        X(NM) = 0
        GO TO 44
 4      NM = M
 44     H = H1/DFLOAT(L)
        R = 0
        DO 444 K = 1,NM
        ZW = H* X(K)
        S = O
        DO 80 I = 1,L
        ZI = 2.*DFLOAT(I) - 1.
        SI = A + ZI*H
 80     S = S + F(SI + ZW) + F(SI - ZW)
 444    R = R + W(K)*S
        R = R*H
        RABS = DABS(R)
        IF(RABS - 1.) 90,90,91
 90     DEZ = DABS(R-RINIT) - EPS*RABS
        GO TO 92
 91       DEZ = DABS(R-RINIT) - EPS
C     = = TEST ON ACCURACY = =
 92     IF(DEZ) 93,93,94
 92     L = 2*L
        IF(L-ITER) 95,96,96
 95     RINIT = R
C     = = ITERATION OF THE INTEGRATION METHOD = =
        GO TO 44
 93     KOD = 0
```

```
      GO TO 99
96    KOD = 1
      L = L/2
99    IF(N-NM) 401,500,401
C  =  = FINDING THE TRUE WEIGHT W(M+1) = =
401   W(NM) = 2.*W(NM)
500   RETURN
      END
```

5. Numerical examples. The above algorithm was programmed and examples were computed for several functions. The computations were done in double precision.

5.1. $f(x) = x + \pi \cdot \sin(\pi x)$, $a = 0$, $b = 1$, $N = 10$, $L = 128$, ITER=1000, EPS = .1D−14.

Results: $R = .250000000000000E+01$, LFINAL=512, KOD= 1
$I(f) = .250000000000000E+01$.

5.2. $f(x) = 1 + e^x$, $a = 0$, $b = 1$, $N = 4$, $L = 40$, ITER= 500, EPS = .1D−5.

Results: $R = .271828182845905E+01$, LFINAL=160, KOD = 0
$I(f) = .2718281828459045\ldots E+01 = e$.

5.3. $f(x) = 1/(1 + x)$, $a = 1$, $b = 0$, $N = 5$, $L = 20$, ITER = 500, EPS = .1D−7.

Results: $R = -.693147180559945E+00$, LFINAL=160, KOD =0
$I(f) = -.6931471805599453\ldots E+00 = -\log_e 2$.

5.4. $f(x) = 6\sqrt{2\pi}\,(\sin x)^{3/2}$, $a = 0$, $b = \pi/2$, ITER=4000, EPS= = .1D−7.

Results: 1) With $N = 4$, $L = 200$
$R = .131450472058757E+02$, LFINAL=400, KOD=0;
2) With $N = 5$, $L = 400$
$R = .131450472063418E+02$, LFINAL=400, KOD=0;
3) If $N = 44$, $L = 200$, then
$R = .131450472065968E+02$, LFINAL=200, KOD=0;
4) For $N = 100$, $L = 100$ one finds
$R = .131450472065969E+02$, LFINAL=100, KOD=0;
$I(f) = .131450472065969\ldots E+02 = \left[\Gamma\left(\frac{1}{4}\right)\right]^2$.

5.5. $f(x) = x \cdot \operatorname{arctg} x$, $a = 1$, $b = 0$, $N = 15$, $L=10$, ITER=50, EPS = .1D−8.

Results: $R = -.285398163397448E+00$, LFINAL=40, KOD=0
$I(f) = -.285398163397448\ldots E+00 = 0.5 - \operatorname{arctg}(1.)$

5.6. $f(x) = x \cdot \log_e x$, $a=1$, $b=2$, ITER=300, EPS= .1D−7.

Results: 1) With $N = 200$, $L = 20$, we obtain
$R = .636294361119892E+00$, LFINAL=20, KOD=0;
2) If $N = 150$, $L = 256$, then
$R = .636294361119892E+00$, LFINAL=256, KOD=0,
$I(f) = .6362943611198906188\ldots E+00$.

5.7. $f(x) = (0.5x + \sqrt{1 + 0.25x^2})^{13} \cdot P_{10}(x)$, $a = -1$, $b = 1$, ITER = =4000, EPS = .1D−8. Note that in the following we denote by P_n the n-th Legendre polynomial.

Results: 1) For $N = 90$, $L = 80$ one finds
$R = .118828159060659 5E-01$, LFINAL=80, KOD=0;
2) If $L = 400$ and $N = 5$, respectively $N = 6$, then LFINAL = 800, KOD = 0 and
$R = .118281590606931E-01$, resp.,
$R = .118281590606417E-01$
$I(f) = .118281590606678\ldots E-01 = 65\sqrt{5}/12288$.

5.8. $f(x) = (1 - x - \log_e x)/((1 - x) \cdot \log x)$, $a = 0$, $b = 1$, ITER = = 4000, $L = 1400$.

Results: 1) With $N = 10$, EPS = .1D−9 one otbtains
$R = .577215655361789E+00$, LFINAL=2800, KOD=1;
2) For $N = 50$, EPS = .1D−8 the computer furnishes
$R = .577215664180130E+00$, LFINAL=2800,KOD=1
$I(f) = .57721566490153286\ldots E+00 = \gamma$ = Euler's constant.

5.9. $f(x) = \log_e 2 + \dfrac{x}{(\sin x + \cos x)\sin x}$, $a = 0$, $b = \pi/4$, ITER = 400, EPS = 1.D−9, $N = 10$, $L = 1200$.

Results: $R = .915965594177221E+00$, LFINAL=2400, KOD=0
$I(f) = .915965594177219015\ldots E+00 = G$ = Catalan's constant.

5.10. $f(x) = 9728 \cdot P_9(x)(1.25 - x)^{-0.5}$, $a = -1$, $b = 1$, ITER = 4000, EPS = .1D−8.

Results: 1) If $N = 8$, $L = 1000$, then
$R = .200000000000189E+01$, KOD=0, LFINAL= = 2000;
2) For $N = 9$, $L = 1000$, one finds
$R = .200000000000120E+01$, KOD=0, LFINAL= = 2000,
$I(f) = .200000000000000E+01$.

Taking into account that the knots at which the integrand is evaluated are chosen in a way that depends on the nature of the function f, we note that this program named GAUSS1 furnishes us an adaptive iterative scheme for automatic integration. An important property of the above subroutine is the fact that the integration abscissas $X(I)$, as well as the values of the corresponding weights $W(I)$, are output parameters. Comparison of the calculated roots $X(I)$ and weights $W(I)$ with those listed in [2]−[3], [8], shows that our algorithm yields results accurate to 14 figures or more.

REFERENCES

[1] Davis Philip, J., *Interpolation and approximation*. Blaisdell, New York, 1965.
[2] Kronrod, A. S., *Nodes and weights of quadrature formulas*. Consultants Bureau Enterprises, Inc., 1965.
[3] Krylov, V. I., *Approximate calculation of integrals*. Macmillan, New York, 1962.
[4] Lupaş, A., *Contributions to the theory of approximation by linear operators*. (Dissertation, Romanian), Cluj, 1975.
[5] Natanson, I. P., *Constructive functtion theory*. Frederick Ungar Publ. Co., New York, 1965.
[6] Ostrowski, A. M., *Solutions of equations in euclidean and Banach spaces*. Academic Press, New York, 1973.
[7] Popoviciu, T., *Notes sur les fonctions convexes d'ordre suprieur (IX)*. Mathematica (Cluj) **43**, 85–141 (1942).
[8] Stroud, A. H. and Secrest, D., *Gaussian quadrature formulas*, Prentice Hall, Englewood Cliffs, N.J., 1966.
[9] Szegö, G., *Orthogonal polynomials*, Amer. Math. Soc. Colloq. Publ. vol. XXIII, New New York, 1959.

Received 28. XII. 1978

Facultatea de mecanică
Str. I. Raţiu nr. 7
2400 SIBIU