255

ANL-7511

3/26/69

732

ANL-7511

Argonne National Laboratory

COMPARISON OF SEVERAL ADAPTIVE NEWTON-COTES QUADRATURE ROUTINES IN EVALUATING DEFINITE INTEGRALS WITH PEAKED INTEGRANDS

by

K. E. Hillstrom

DISCLAIMER

Printed in the United States of America
Available from
Clearinghouse for Federal Scientific and Technical Information
National Bureau of Standards, U. S. Department of Commerce
Springfield, Virginia 22151
Price: Printed Copy $3.00; Microfiche $0.65

ANL-7511
Mathematics and
Computers

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

COMPARISON OF SEVERAL ADAPTIVE NEWTON-COTES QUADRATURE ROUTINES IN EVALUATING DEFINITE INTEGRALS WITH PEAKED INTEGRANDS

by

K. E. Hillstrom

Applied Mathematics Division

November 1968

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

COMPARISON OF SEVERAL ADAPTIVE NEWTON-COTES QUADRATURE ROUTINES IN EVALUATING DEFINITE INTEGRALS WITH PEAKED INTEGRANDS

by

K. E. Hillstrom

ABSTRACT

This report compares the performance of five different adaptive quadrature schemes, based on Newton-Cotes $(2N+1)$-point rules $(N = 1, 2, 3, 4, 5)$, in approximating the sets of definite integrals

$$\int_{-1}^{1} (x^2+p^2)^{-1}\, dx,\ \int_0^1 x/(1.1 - x)^p\, dx, \text{ and } \int_0^3 x^p(x-2)(x-3)\, dx$$

with relative accuracy ϵ.

I. INTRODUCTION

High-degree Newton-Cotes quadrature rules have seldom been used in practice because they occasionally fail to converge or they contain weights of different signs. This report shows, by numerical experiment only, that there are certain quadratures for which moderate degree Newton-Cotes rules, used adaptively, may be superior to other rules based on equally spaced abscissas.

McKeeman[1-3] and Davis and Rabinowitz[4] describe the use of quadrature rules in an adaptive manner by means of algorithms. Lyness[5] gives a thorough description of the Adaptive Simpson Rule, together with suggested modifications.

II. QUADRATURE SCHEMES USED

The schemes used in these investigations are adaptive Newton-Cotes rules of degree 3, 5, 7, 9, and 11, incorporating the modifications of types 1, 2, and 3 described in Ref. 5, adjusted for use with the particular rule. The results are, therefore, of polynomial degree 5, 7, 9, 11, and 13, respectively.

In particular, if ϵ is the total absolute error allowed and

$$Q_N[a, a+h]\ f(x) = \sum_{j=1}^{2N+1} a_j f(x_j)$$

is the (2N + 1) Newton-Cotes Rule, convergence is achieved over the interval $[a, a+h]$ if for

$$\Delta = Q_N[a, a+h]\ f(x) - \left(Q_N\left[a, a+\frac{h}{2}\right] f(x) + Q_N\left[a+\frac{h}{2}, a+h\right] f(x)\right),$$

$$|\Delta| \leq \frac{\left(2^{2N+1} - 1\right)\epsilon}{2^r},$$

where $h = (B - A)\, 2^{-r}$. If this convergence criterion is not satisfied and if $r < 30$, the interval is bisected, r is replaced by $r + 1$, and, after function evaluations at the new mesh points, the above test is repeated. If the convergence criterion is satisfied or if $r = 30$,

$$Q_N\left[a, a+\frac{h}{2}\right] f(x) + Q_N\left[a+\frac{h}{2}, a+h\right] f(x) + \Delta\Big/\left(2^{2N+1} - 1\right)$$

is accepted as an approximation for the integral over $[a, a+h]$, this approximation being of degree $2N + 3$. Finally, these component approximations and error estimates are summed to obtain a final or total approximation over $[A, B]$.

III. INVESTIGATIONS CONDUCTED

Numerical integrations were carried out using the five different adaptive Newton-Cotes rules in approximating three sets of definite integrals. In each case, an input parameter ϵ prescribes the error. A routine is termed the most efficient if its result satisfies the error criterion ϵ while requiring the fewest function evaluations. The actual error in each result is usually much smaller than ϵ.

IV. COMPUTATIONS

All the computations conducted were performed on an IBM 360/50-75 in double-precision floating point.

The sets of definite integrals

$$I_1 = \int_{-1}^{1} (x^2+p^2)^{-1},$$

$$I_2 = \int_0^1 x/(1.1 - x)^p \, dx,$$

and

$$I_3 = \int_0^3 x^p(x - 2)(x - 3) \, dx$$

have been evaluated with p ranges $(1, 10^{-4})$, $(1, 9)$, and $(1, 16)$, respectively, and ϵ ranging from 1 to 10^{-8}.

For small p, the I_1 integrand has a peak of height p^{-2} at the origin and is approximately 1 at the end points, and I_1 is approximately equal to πp^{-1}.

For large p, the I_2 integrand has a maximum of 10^p at the upper limit and is zero at the origin, and I_2 is approximately equal to $10^{p-1}/(p - 1)$.

For large p, the I_3 integrand has a maximum near 2, has a minimum near 3, and is zero at the origin and at $x = 2$ and $x = 3$, and I_3 is approximately equal to $-3^{p+2}/p^2$.

The appendix presents FORTRAN listings of the comparison routine, the integrand and integral evaluation subroutines, and the five adaptive quadrature subroutines.

V. RESULTS

Results using the adaptive Newton-Cotes rules for the quadratures I_1, I_2, and I_3 are displayed in Figs. 1, 2, and 3, respectively, and are to the required accuracy ϵI. If the point (p,ϵ) lies in the zone numbered N, the adaptive Newton-Cotes $(2N+1)$-point rule is the most efficient of those tested.

The actual demarcation lines between zones are not regular in Figs. 1 and 2. These irregularities are partly due to the fact that the number of points used by the adaptive Newton-Cotes rule of degree $2N + 1$ in approximating an integral over $(a, a+h)$ is restricted to numbers of the form $8kN + 1$ $(k = 1, 2, 3, \ldots)$. If the integrand function is altered slightly, the demarcation line is different in detail, but has the same general configuration. In Figs. 1 and 2, the "buffer zones" between distinct zones indicate the general width of these irregularities.

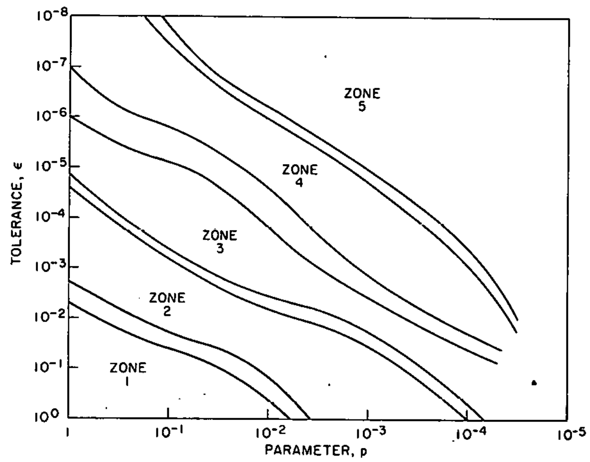

Fig. 1. Performance Comparison in Approximating $\int_{-1}^{1} (x^2 + p^2)^{-1}\, dx$ for p Ranging from 1 to 10^{-4} and ε Ranging from 1 to 10^{-8}. If the point (p, ε) lies in the zone numbered N, the adaptive Newton-Cotes (2N + 1)-point rule is the most efficient of those tested.

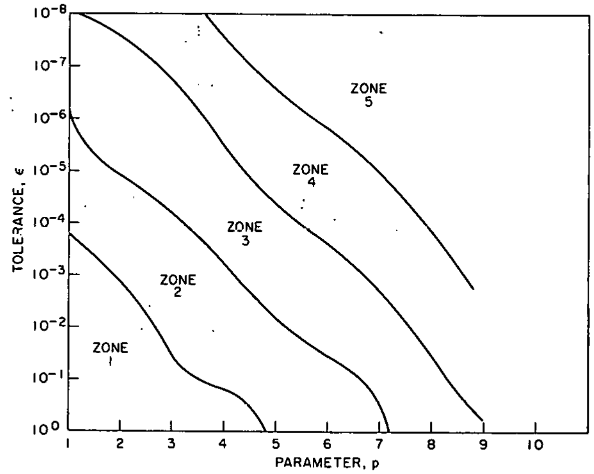

Fig. 2. Performance Comparison in Approximating $\int_{0}^{1} x/(1.1 - x)^p\, dx$ for p Ranging from 1 to 9 and ε Ranging from 1 to 10^{-8}. If the point (p, ε) lies in the zone numbered N, the adaptive Newton-Cotes (2N + 1)-point rule is the most efficient of those tested.

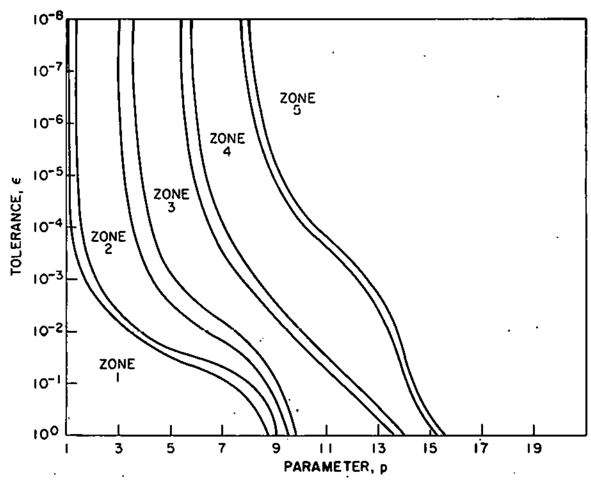

Fig. 3

Performance Comparison in Approximating $\int_{0}^{3} x^p(x - 2)(x - 3)\, dx$ for p Ranging from 1 to 16 and ε Ranging from 1 to 10^{-8}. If the point (p, ε) lies in the zone numbered N, the adaptive Newton-Cotes (2N + 1)-point rule is the most efficient of those tested.

These results are of interest because they indicate that if the integrand has a high, sharp peak, or if great accuracy is required, an adaptive high-degree rule is most efficient. This is in contradistinction to the more familiar state of affairs in which sharp peaks are associated with inefficient polynomial approximations and the use of low-degree rules.

In addition, error curves of the adaptive Newton-Cotes rules for the quadratures I_1, I_2, and I_3 and a particular value of p are displayed in Figs. 4, 5, and 6, respectively. The error curves of the adaptive Newton-Cotes (2N + 1)-point rules are labeled EN; the projections of the pairs of intersection points (p_{N-1}, p_N) onto the ϵ axis define, for fixed p, the log ϵ interval over which the $(2N+1)$-point rule is most efficient, in terms of points M required; and the lines S1, S2, S3, S4, and S5 with slopes 6, 8, 10, 12, and 14, respectively, indicate the rate of convergence of these rules.

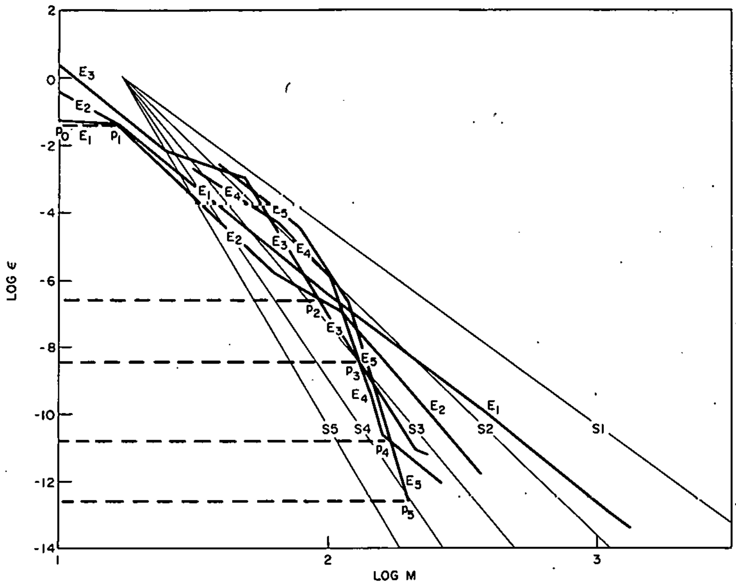

Fig. 4. Error Curves of the Adaptive Newton-Cotes $(2N+1)$-point Rules in Approximating $I_1 = \int_{-1}^{1} (x^2 + 10^{-8})^{-1}\, dx$. A point (M,ε) located on line E_N between intersection points (p_{N-1}, p_N) indicates that the $2N+1$ adaptive Newton-Cotes routine with input error εI_1 required the least number M of function evaluations of the routines tested.

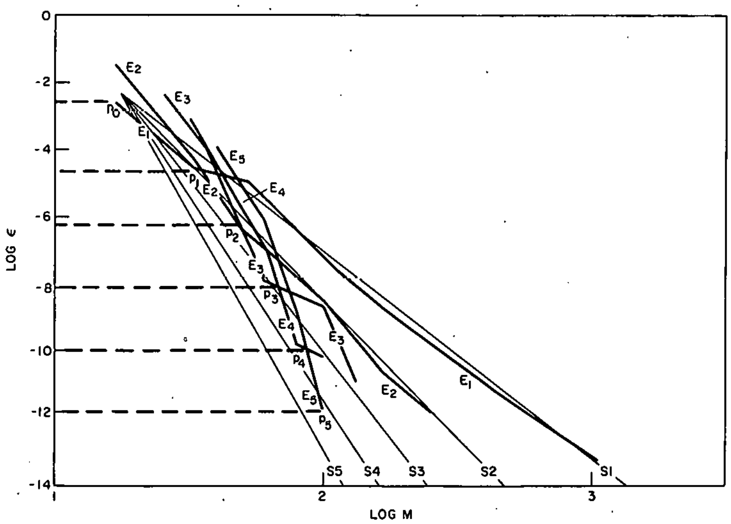

Fig. 5. Error Curves of the Adaptive Newton-Cotes $(2N+1)$-point Rules in Approximating $I_2 = \int_{0}^{1} x/(1.1-x)^9\, dx$. A point (M,ε) located on line E_N between intersection points (p_{N-1}, p_N) indicates that the $2N+1$ adaptive Newton-Cotes routine with input error εI_2 required the least number M of function evaluations of the routines tested.

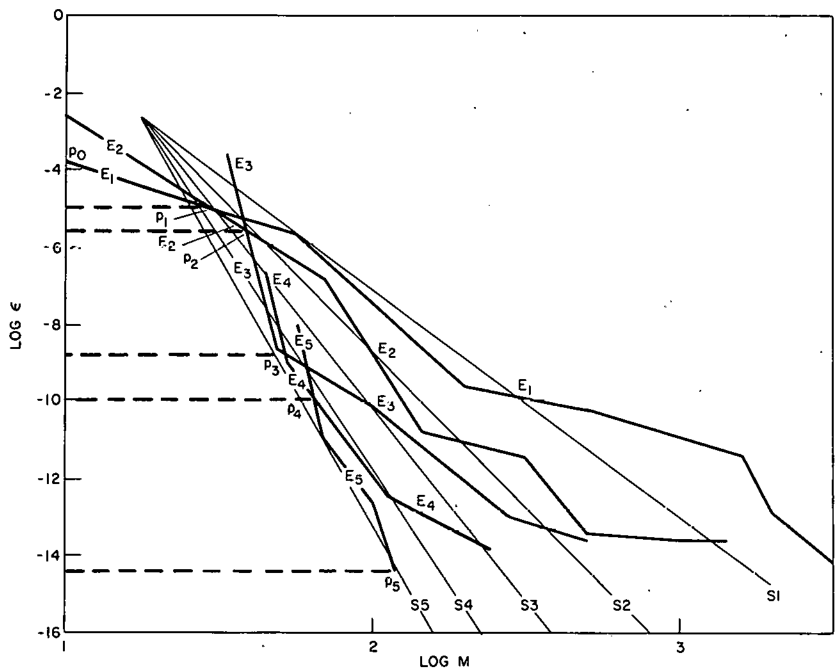

Fig. 6. Error Curves of the Adaptive Newton-Cotes (2N + 1)-point Rules in Approximating $I_3 = \int_0^3 x^{18}(x-2)(x-3)\,dx$. A point (M,ε) located on line E_N between intersection point (p_{N-1}, p_N) indicates that the 2N + 1 adaptive Newton-Cotes routine with input error εI_3 required the least number M of function evaluations of the routines tested.

These results define (for a particular p) the ε interval of greatest efficiency of the (2N + 1)-point rules and demonstrate that the modified Newton-Cotes rules, being of polynomial degree 5, 7, 9, 11, and 13 have error curves with slopes approximately equal to 6, 8, 10, 12, and 14, respectively.

APPENDIX

FORTRAN Listings of the Comparison Routine, the Integrand and Integral Evaluation Subroutines, and the Five Adaptive Quadrature Subroutines

1. Routine for Comparing Adaptive Newton-Cotes Quadrature Routines

```
FORTRAN IV G LEVEL 1, MOD 1              MAIN              DATE = 68288              16/56/53

          C      QUADRATURE ROUTINE COMPARISON DRIVER
          C
                 DOUBLE PRECISION A,B,EI,FE,PI,DP,FP,P,EPS,QI,RI,RE,RES,QUAD,T,EVAL
                 DOUBLE PRECISION DE
                 COMMON P,IFI,IFC,ICC
                 DIMENSION IVC(5),QDR(6),TITLE(18)
                 DATA QDR/'2','4','6','8','0','X'/
                 PRINT 15
              15 FORMAT('1TEST OF ADAPTIVE NEWTON COTES 2,4,6,8,10 HAVIE AND RUMBER
                XG'///)
               1 READ 1000,TITLE
                 PRINT 1000,TITLE
                 READ 1001,IFI,INC
                 PRINT 1001,IFI,INC
                 DO 11 N=1,INC
                 READ 1002,A,B,EI,DE,PI,DP
                 PRINT 1006,A,B,EI,DE,PI,DP
                 READ 1004,FE,FP,IE,IP,ICVC
                 PRINT 1007,FE,FP,IE,IP,ICVC
                 DO 10 M=1,ICVC
                 READ (5,1003) (IVC(I),I=1,5)
                 READ 1005,K,ICC
                 IFCALL=1
                 XMAX=FP
                 XPINCH=(FP-PI)/10.0D0
                 YMAX=FE
                 YMIN=EI
                 XMIN=PI
                 XP=PI
                 GO TO (2,3),K
               2 P=PI
                 GO TO 4
               3 P=10.0D0**(-PI)
               4 DO 9 L=1,IP
                 YP=EI
                 EPS=10.D0**(-EI)
                 PRINT 20,A,B,EPS,P
              20 FORMAT(' A=',D12.5,' B=',D12.5,' EPS=',D12.5,' P=',D12.5,///)
                 RI=EVAL(A,B)
                 DO 8 J=1,IE
                 IFCS=2147483647
                 RES=1.D0+74
                 SYM=QDR(6)
                 DO 7  I=1,5
                 IF(IVC(I).EQ.0)GO TO 7
                 QI=QUAD(A,B,EPS,I)
                 RE=DABS(RI-QI)/RI
                 PRINT 30,QI,RI,RE,IFC
              30 FORMAT(' QI=',D22.15,' RI=',D22.15,' RERR=',D22.15,' FCOUNT=',I8,/
                X/)
                 IF(RE.GT.EPS)GO TO 7
                 IF(IFCS.LT.IFC)GO TO 7
                 IF(IFCS.GT.IFC)GO TO 6
                 IF(RES.LE.RE)GO TO 7
               6 IFCS=IFC
```

```
FORTRAN IV G LEVEL 1, MOD 1             MAIN               DATE = 68288           16/56/53

 0052                 RES=RE
 0053                 SYM=QDR(I)
 0054               7 CONTINUE
 0055                 PRINT 50
 0056              50 FORMAT(' **************************************************************'///)
 0057                 GO TO (16,17),K
 0058              16 CALL YOLYPL(XP,YP,XMAX,XPINCH,YMAX,0,1,K,8HPARAMETR,8H-LOG(EP),IFC
                     XALL,1,1,XMIN,YMIN,SYM,TITLE,1)
 0059                 GO TO 18
 0060              17 CALL YOLYPL(XP,YP,XMAX,XPINCH,YMAX,0,1,K,8H-LOG(P) ,8H-LOG(EP),IFC
                     XALL,1,1,XMIN,YMIN,SYM,TITLE,1)
 0061              18 IF(IFCALL.EQ.2)GO TO 12
 0062                 IFCALL=2
 0063              12 EPS=EPS*10.0D0**(-DE)
 0064               8 YP=YP+DE
 0065                 XP=XP+DP
 0066                 GO TO (19,21),K
 0067              19 P=P+DP
 0068                 GO TO 9
 0069              21 P=P*10.0D0**(-DP)
 0070               9 CONTINUE
 0071              10 CONTINUE
 0072              11 CONTINUE
 0073                 CALL PLOT(15.,0,-3)
 0074                 CALL YOLYPL(XP,YP,XMAX,XPINCH,YMAX,-1,1,K,8HPARAMETR,8H-LOG(EP),IF
                     XCALL,1,1,XMIN,YMIN,SYM,TITLE,1)
 0075                 STOP
 0076            1000 FORMAT(18A4)
 0077            1001 FORMAT(2I2)
 0078            1002 FORMAT(6D12.5)
 0079            1003 FORMAT (5I1)
 0080            1004 FORMAT(2D12.5,I3,I3,I2)
 0081            1005 FORMAT(I3,I2)
 0082            1006 FORMAT(' A=',D12.5,' B=',D12.5,' IEP=',D12.5,' DEP=',D12.5,' IP=',
                     XD12.5,' DP=',D12.5)
 0083            1007 FORMAT(' FE=',D12.5,' FP=',D12.5,' EC= ',I3,' PC= ',I3,' VC= ',I2,
                     X//)
 0084                 END

FORTRAN IV G LEVEL 1, MOD 1             MAIN               DATE = 68270           12/16/41

            C         QUADRATURE ROUTINE SELECTOR
            C
 0001                 FUNCTION QUAD(A,B,EPS,I)
 0002                 DOUBLE PRECISION A,B,EPS,EP,FCN,QUAD,ANC2,ANC4,ANC6,ANC8,ANC10,P
 0003                 COMMON P,IFI,IFC,ICC
 0004                 EXTERNAL FCN
 0005                 IFC=0
 0006                 EP=EPS
 0007                 M=15
 0008                 N=ICC
 0009                 GO TO(10,20,30,40,50),I
 0010              10 QUAD=ANC2(A,B,EP,M,N,FCN)
 0011                 RETURN
 0012              20 QUAD=ANC4(A,B,EP,M,N,FCN)
 0013                 RETURN
 0014              30 QUAD=ANC6(A,B,EP,M,N,FCN)
 0015                 RETURN
 0016              40 QUAD=ANC8(A,B,EP,M,N,FCN)
 0017                 RETURN
 0018              50 QUAD=ANC10(A,B,EP,M,N,FCN)
 0019                 RETURN
 0020                 END
```

```
FORTRAN IV G LEVEL 1, MOD 1              MAIN              DATE = 68270              12/16/41

      C     EVALUATES INTEGRAND FUNCTION
      C
            FUNCTION FCN(X)
            DOUBLE PRECISION X,P,FCN
            COMMON P,IFI,IFC,ICC
            IFC=IFC+1
            GO TO(10,20,30),IFI
         10 FCN=1.0D0/(X*X+P*P)
            RETURN
         20 FCN=X/(1.1D0-X)**P
            RETURN
         30 FCN=X**P*(X-2.0D0)*(X-3.0D0)
            RETURN
            END
```

```
FORTRAN IV G LEVEL 1, MOD 1              MAIN              DATE = 68270              12/16/41

      C     EVALUATES DEFINITE INTEGRAL
      C
            FUNCTION EVAL(A,B)
            DOUBLE PRECISION A,B,P,EVAL,T1,T2,T3,T4,Q1,Q2
            COMMON P,IFI,IFC,ICC
            GO TO(10,20,30),IFI
         10 EVAL=(DATAN(B/P)-DATAN(A/P))/P
            RETURN
         20 IF(P.EQ.1.0D0) GO TO 24
            IF(P.EQ.2.0D0) GO TO 28
            Q1=P-1.0D0
            Q2=P-2.0D0
            T1=(1.1D0-B)**Q2
            T2=(1.1D0-B)**Q1
            T3=(1.1D0-A)**Q2
            T4=(1.1D0-A)**Q1
            EVAL=-1.0D0/(Q2*T1)+1.1D0/(Q1*T2)+1.0D0/(Q2*T3)-1.1D0/(Q1*T4)
            RETURN
         24 T1=DABS(1.1D0-B)
            T2=DABS(1.1D0-A)
            EVAL=A-B-1.1D0*(DLOG(T1)-DLOG(T2))
            RETURN
         28 T1=DABS(1.1D0-B)
            T2=DABS(1.1D0-A)
            EVAL=DLOG(T1)-DLOG(T2)+1.1D0/(1.1D0-B)-1.1D0/(1.1D0-A)
            RETURN
         30 T1=B*B
            T2=A*A
            T3=P+1.0D0
            EVAL=B**T3*(T1/(P+3.0D0)-5.0D0*B/(P+2.0D0)+6.0D0/(P+1.0D0))-A**T3*
           X(T2/(P+3.0D0)-5.0D0*A/(P+2.0D0)+6.0D0/(P+1.0D0))
            RETURN
            END
```

2. Adaptive Quadrature Routine Based on Newton-Cotes 3-point Rule

```
FORTRAN IV G LEVEL 1, MOD 1              MAIN              DATE = 68270          12/16/41

      C      ANC2 INTEGRATION
      C      ADAPTIVE QUADRATURE ROUTINE BASED ON NEWTON-COTES 3 POINT RULE
      C
             FUNCTION ANC2(A1,B1,EP,M,N,FUN)
             DOUBLE PRECISION  ANC2,A1,B1,EP,FUN,A,B,EPS,ABSAR,EST,FA,FM,FB,XB,
            1F1,F2,FBP,EST2,DIFF,EST1,SUM,DAFT,ESUM,TSUM,DA,SX,SA
             DOUBLE PRECISION AEST2,FTST,FMAX,AEST1,DELTA,AEST
             DIMENSION F2(30) ,FBP(30),EST2(30),NRTR(30)
             DIMENSION AEST2(30),FTST(3),XB(30)
      C      THE PARAMETER SETUP FOR THE INITIAL CALL
             IF(N.LE.0)GO TO 210
             IF(N.GT.3)GO TO 211
             A=A1
             B=B1
             EPS=EP*15.0D0
             ESUM=0.0D0
             TSUM=0.0D0
             LVL=1
             DA=B-A
             FA=FUN(A)
             FM=FUN((A+B)*0.5D0)
             FB=FUN(B)
             M=3
             FMAX=DABS(FA)
             FTST(1)=FMAX
             FTST(2)=DABS(FM)
             FTST(3)=DABS(FB)
             DO 100 I=2,3
             IF(FMAX.GE.FTST(I))GO TO 100
             FMAX=FTST(I)
         100 CONTINUE
             EST=(FA+4.0D0*FM+FB)*DA/6.0D0
             ABSAR=(FTST(1)+4.0D0*FTST(2)+FTST(3))*DA/6.0D0
             AEST=ABSAR
      C      1=RECUR
           1 SX=(DA/(2.0D0**LVL))/6.0D0
             F1=FUN((3.0D0*A+B)/4.0D0)
             F2(LVL)=FUN((A+3.0D0*B)/4.0D0)
             EST1=SX*(FA+4.0D0*F1+FM)
             FBP(LVL)=FB
             XB(LVL)=B
             EST2(LVL)=SX*(FM+4.0D0*F2(LVL)+FB)
             SUM=EST1+EST2(LVL)
             FTST(1)=DABS(F1)
             FTST(2)=DABS(F2(LVL))
             FTST(3)=DABS(FM)
             AEST1=SX*(DABS(FA)+4.0D0*FTST(1)+FTST(3))
             AEST2(LVL)=SX*(FTST(3) +4.0D0*FTST(2)+DABS(FB))
             ABSAR=ABSAR-AEST+AEST1+AEST2(LVL)
             M=M+2
             GO TO (201,200,202),N
         200 DELTA=ABSAR
             GO TO 205
         210 PRINT 39
          39 FORMAT(' ERROR RETURN-N.LE.0')
```

```
              RETURN
          211 PRINT 40
           40 FORMAT(' ERROR RETURN-N.GT.3')
              RETURN
          201 DELTA=1.0D0
              GO TO 205
          202 DO 203 I=1,2
              IF(FMAX.GE.FTST(I))GO TO 203
              FMAX=FTST(I)
          203 CONTINUE
              DELTA=FMAX
          205 DAFT=EST-SUM
              DIFF=DABS(DAFT)
              DAFT=DAFT/15.0D0
              IF(DIFF-EPS*DELTA)6,6,3
            3 IF(LVL-30)4,2,2
            6 IF(LVL-1)2,4,2
C             2=UP
            2 A=B
              ESUM=ESUM+DAFT
              TSUM=TSUM+SUM
            9 LVL=LVL-1
              L=NRTR(LVL)
              GO TO (11,12),L
C             11=R1,12=R2
    4         NRTR(LVL)=1
              EST=EST1
              AEST=AEST1
              FB=FM
              FM=F1
              B=(A+B)/2.0D0
              EPS=EPS/2.0D0
            7 LVL=LVL+1
              GO TO 1
   11         NRTR(LVL)=2
              FA=FB
              FM=F2(LVL)
              FB=FBP(LVL)
              B=XB(LVL)
              EST=EST2(LVL)
              AEST=AEST2(LVL)
              GO TO 7
           12 EPS=2.0D0*EPS
              IF(LVL-1) 5,5,9
            5 ANC2=TSUM-ESUM
              RETURN
              END
```

3. Adaptive Quadrature Routine Based on Newton-Cotes 5-point Rule

```
FORTRAN IV G LEVEL 1, MOD 1               MAIN                 DATE = 68270          12/16/41

                C      ANC4 INTEGRATION
                C      ADAPTIVE QUADRATURE ROUTINE BASED ON NEWTON-COTES 5 POINT RULE
                C
 0001                  FUNCTION ANC4(A1,B1,EP,M,N,FUN)
 0002                  DOUBLE PRECISION  ANC4,A1,B1,EP,FUN,A,B,EPS,ESUM,TSUM,DA,XB,SX,FA
                      1,F1,FS,F3,FM,F2,FT,F4,FB,FTP,FBP,FMAX,FTST,EST,AEST,EST1,EST2,AEST
                      21,AEST2,ABSAR,DELTA,DIFF,DAFT,SUM,SA
 0003                  DIMENSION F2(30),F4(30),FTP(30),FBP(30),FTST(5),EST2(30),NRTR(30)
 0004                  DIMENSION AEST2(30),XB(30)
                C      THE PARAMETER SETUP FOR THE INITIAL CALL
 0005                  IF(N.LE.0)GO TO 210
 0006                  IF(N.GT.3)GO TO 211
 0007                  A=A1
 0008                  B=B1
 0009                  EPS=EP*63.0D0
 0010                  ESUM=0.0D0
 0011                  TSUM=0.0D0
 0012                  LVL=1
 0013                  DA=B-A
 0014                  FA=FUN(A)
 0015                  FS=FUN((3.0D0*A+B)/4.0D0)
 0016                  FM=FUN((A+B)*0.5D0)
 0017                  FT=FUN((A+3.0D0*B)/4.0D0)
 0018                  FB=FUN(B)
 0019                  M=5
 0020                  FMAX=DABS(FA)
 0021                  FTST(1)=FMAX
 0022                  FTST(2)=DABS(FS)
 0023                  FTST(3)=DABS(FM)
 0024                  FTST(4)=DABS(FT)
 0025                  FTST(5)=DABS(FB)
 0026                  DO 100 I=2,5
 0027                  IF(FMAX.GE.FTST(I))GO TO 100
 0028                  FMAX=FTST(I)
 0029              100 CONTINUE
 0030                  EST=(7.0D0*(FA+FB)+32.0D0*(FS+FT)+12.0D0*FM)*DA/90.0D0
 0031                  ABSAR=(7.0D0*(FTST(1)+FTST(5))+32.0D0*(FTST(2)+FTST(4))+12.0D0*FTS
                      1T(3))*DA/90.0D0
 0032                  AEST=ABSAR
                C      1=RECUR
 0033                1 SX=(DA/(2.0D0**LVL))/90.0D0
 0034                  F1=FUN((7.0D0*A+B)/8.0D0)
 0035                  F3=FUN((5.0D0*A+3.0D0*B)/8.0D0)
 0036                  F2(LVL)=FUN((3.0D0*A+5.0D0*B)/8.0D0)
 0037                  F4(LVL)=FUN((A+7.0D0*B)/8.0D0)
 0038                  EST1=SX*(7.0D0*(FA+FM)+32.0D0*(F1+F3)+12.0D0*FS)
 0039                  FBP(LVL)=FB
 0040                  FTP(LVL)=FT
 0041                  XB(LVL)=B
 0042                  EST2(LVL)=SX*(7.0D0*(FM+FB)+32.0D0*(F2(LVL)+F4(LVL))+12.0D0*FT)
 0043                  SUM=EST1+EST2(LVL)
 0044                  FTST(1)=DABS(F1)
 0045                  FTST(2)=DABS(F2(LVL))
 0046                  FTST(3)=DABS(F3)
 0047                  FTST(4)=DABS(F4(LVL))
```

```
 0048              FTST(5)=DABS(FM)
 0049              AEST1=SX*(7.0D0*(DABS(FA)+FTST(5))+32.0D0*(FTST(1)+FTST(3))+12.0D0
                  X*DABS(FS))
 0050              AEST2(LVL)=SX*(7.0D0*(FTST(5)+DABS(FB))+32.0D0*(FTST(2)+FTST(4))+1
                  X2.0D0*DABS(FT))
 0051              ABSAR=ABSAR-AEST+AEST1+AEST2(LVL)
 0052              M=M+4
 0053              GO TO (201,200,202),N
 0054          200 DELTA=ABSAR
 0055              GO TO 205
 0056          210 PRINT 39
 0057           39 FORMAT(' ERROR RETURN-N.LE.0')
 0058              RETURN
 0059          211 PRINT 40
 0060           40 FORMAT(' ERROR RETURN-N.GT.3')
 0061              RETURN
 0062          201 DELTA=1.0D0
 0063              GO TO 205
 0064          202 DO 203 I=1,4
 0065              IF(FMAX.GE.FTST(I))GO TO 203
 0066              FMAX=FTST(I)
 0067          203 CONTINUE
 0068              DELTA=FMAX
 0069          205 DAFT=EST-SUM
 0070              DIFF=DABS(DAFT)
 0071              DAFT=DAFT/63.0D0
 0072              IF(DIFF-EPS*DELTA)6,6,3
 0073            3 IF(LVL-30)4,2,2
 0074            6 IF(LVL-1)2,4,2
            C        2=UP
 0075            2 A=B
 0076              ESUM=ESUM+DAFT
 0077              TSUM=TSUM+SUM
 0078            9 LVL=LVL-1
 0079              L=NRTR(LVL)
 0080              GO TO (11,12),L
            C        11=R1,12=R2
 0081            4 NRTR(LVL)=1
 0082              EST=EST1
 0083              AEST=AEST1
 0084              FB=FM
 0085              FT=F3
 0086              FM=FS
 0087              FS=F1
 0088              B=(A+B)/2.0D0
 0089              EPS=EPS/2.0D0
 0090            7 LVL=LVL+1
 0091              GO TO 1
 0092           11 NRTR(LVL)=2
 0093              FA=FB
 0094              FS=F2(LVL)
 0095              FM=FTP(LVL)
 0096              FT=F4(LVL)
 0097              FB=FBP(LVL)
 0098              B=XB(LVL)
 0099              EST=EST2(LVL)
 0100              AEST=AEST2(LVL)
 0101              GO TO 7
 0102           12 EPS=2.0D0*EPS
 0103              IF(LVL-1)5,5,9
 0104            5 ANC4=TSUM-ESUM
 0105              RETURN
 0106              END
```

4. Adaptive Quadrature Routine Based on Newton-Cotes 7-point Rule

```
FORTRAN IV G LEVEL 1, MOD 1              MAIN              DATE = 68270          12/16/41

              C       ANC6 INTEGRATION
              C       ADAPTIVE QUADRATURE ROUTINE BASED ON NEWTON-COTES 7 POINT RULE
              C
 0001                 FUNCTION ANC6(A1,B1,EP,M,N,FUN)
 0002                 DOUBLE PRECISION  ANC6,A1,B1,EP,FUN,A,B,EPS,SUM,ESUM,TSUM,DA,XB,SX
                     1,F1,FS,F3,FM,F2,FT,F4,FB,FTP,FBP,FMAX,FTST,EST,AEST,EST1,EST2,AEST
                     21,AEST2,ABSAR,DELTA,DIFF,DAFT,FA,SA
 0003                 DOUBLE PRECISION FR,F5,FU,F6,XR,X5,XU,X6,FUP
 0004                 DIMENSION F2(30),F4(30),FTP(30),FBP(30),FTST(7),EST2(30),NRTR(30)
 0005                 DIMENSION AEST2(30),XB(30)
 0006                 DIMENSION FUP(30),F6(30)
              C       THE PARAMETER SETUP FOR THE INITIAL CALL
 0007                 IF(N.LE.0)GO TO 210
 0008                 IF(N.GT.3)GO TO 211
 0009                 A=A1
 0010                 B=B1
 0011                 EPS=EP*255.0D0
 0012                 ESUM=0.0D0
 0013                 TSUM=0.0D0
 0014                 LVL=1
 0015                 DA=B-A
 0016                 FA=FUN(A)
 0017                 FR=FUN((5.0D0*A+B)/6.0D0)
 0018                 FS=FUN((2.0D0*A+B)/3.0D0)
 0019                 FM=FUN((A+B)*0.5D0)
 0020                 FT=FUN((A+2.0D0*B)/3.0D0)
 0021                 FU=FUN((A+5.0D0*B)/6.0D0)
 0022                 FB=FUN(B)
 0023                 M=7
 0024                 FMAX=DABS(FA)
 0025                 FTST(1)=FMAX
 0026                 FTST(2)=DABS(FR)
 0027                 FTST(3)=DABS(FS)
 0028                 FTST(4)=DABS(FM)
 0029                 FTST(5)=DABS(FT)
 0030                 FTST(6)=DABS(FU)
 0031                 FTST(7)=DABS(FB)
 0032                 DO 100 I=2,7
 0033                 IF(FMAX.GE.FTST(I))GO TO 100
 0034                 FMAX=FTST(I)
 0035             100 CONTINUE
 0036                 EST=(41.0D0*(FA+FB)+216.0D0*(FR+FU)+27.0D0*(FS+FT)+272.0D0*FM)*DA/
                     1840.0D0
 0037                 ABSAR=(41.0D0*(FTST(1)+FTST(7))+216.0D0*(FTST(2)+FTST(6))+27.0D0*(
                     1FTST(3)+FTST(5))+272.0D0*FTST(4))*DA/840.0D0
 0038                 AEST=ABSAR
              C       1=RECUR
 0039               1 SX=(DA/(6.0D0*2.0D0**LVL))/140.0D0
 0040                 F1=FUN((11.0D0*A+B)/12.0D0)
 0041                 F3=FUN((9.0D0*A+3.0D0*B)/12.0D0)
 0042                 F5=FUN((7.0D0*A+5.0D0*B)/12.0D0)
 0043                 F2(LVL)=FUN((5.0D0*A+7.0D0*B)/12.0D0)
 0044                 F4(LVL)=FUN((3.0D0*A+9.0D0*B)/12.0D0)
 0045                 F6(LVL)=FUN((A+11.0D0*B)/12.0D0)
 0046                 EST1=SX*(41.0D0*(FA+FM)+216.0D0*(F1+F5)+27.0D0*(FR+FS)+272.0D0*F3)
 0047                 FBP(LVL)=FB
 0048                 XB(LVL)=B
 0049                 FTP(LVL)=FT
 0050                 FUP(LVL)=FU
 0051                 EST2(LVL)=SX*(41.0D0*(FM+FB)+216.0D0*(F2(LVL)+F6(LVL))+27.0D0*(FT+
                     1FU)+272.0D0*F4(LVL))
 0052                 SUM=EST1+EST2(LVL)
 0053                 FTST(1)=DABS(F1)
 0054                 FTST(2)=DABS(F3)
 0055                 FTST(3)=DABS(F5)
 0056                 FTST(4)=DABS(F2(LVL))
```

```
             FTST(5)=DABS(F4(LVL))
             FTST(6)=DABS(F6(LVL))
             FTST(7)=DABS(FM)
             AEST1=SX*(41.0D0*(DABS(FA)+FTST(7))+216.0D0*(FTST(1)+FTST(3))+27.0
            1D0*(DABS(FR)+DABS(FS))+272.0D0*FTST(2))
             AEST2(LVL)=SX*(41.0D0*(FTST(7)+DABS(FB))+216.0D0*(FTST(4)+FTST(6))
            1+27.0D0*(DABS(FT)+DABS(FU))+272.0D0*FTST(5))
             ABSAR=ABSAR-AEST+AEST1+AEST2(LVL)
             M=M+6
             GO TO (201,200,202),N
         200 DELTA=ABSAR
             GO TO 205
         210 PRINT 39
          39 FORMAT(' ERROR RETURN-N.LE.0')
             RETURN
         211 PRINT 40
          40 FORMAT(' ERROR RETURN-N.GT.3')
             RETURN
         201 DELTA=1.0D0
             GO TO 205
         202 DO 203 I=1,6
             IF(FMAX.GE.FTST(I))GO TO 203
             FMAX=FTST(I)
         203 CONTINUE
             DELTA=FMAX
         205 DAFT=EST-SUM
             DIFF=DABS(DAFT)
             DAFT=DAFT/255.0D0
             IF(DIFF-EPS*DELTA)6,6,3
           3 IF(LVL-30)4,2,2
           6 IF(LVL-1)2,4,2
      C      2=UP
           2 A=B
             ESUM=ESUM+DAFT
             TSUM=TSUM+SUM
           9 LVL=LVL-1
             L=NRTR(LVL)
             GO TO (11,12),L
      C      11=R1,12=R2
           4 NRTR(LVL)=1
             EST=EST1
             AEST=AEST1
             FB=FM
             FU=F5
             FT=FS
             FM=F3
             FS=FR
             FR=F1
             B=(A+B)/2.0D0
             EPS=EPS/2.0D0
           7 LVL=LVL+1
             GO TO 1
          11 NRTR(LVL)=2
             FA=FB
             FR=F2(LVL)
             FS=FTP(LVL)
             FM=F4(LVL)
             FT=FUP(LVL)
             FU=F6(LVL)
             FB=FBP(LVL)
             B=XB(LVL)
             EST=EST2(LVL)
             AEST=AEST2(LVL)
             GO TO 7
          12 EPS=EPS*2.0D0
             IF(LVL-1)5,5,9
           5 ANC6=TSUM-ESUM
             RETURN
             END
```

5. Adaptive Quadrature Routine Based on Newton-Cotes 9-point Rule

```
FORTRAN IV G LEVEL 1, MOD 1            MAIN            DATE = 68270         21/22/06

              C      ANC8 INTEGRATION
              C      ADAPTIVE QUADRATURE ROUTINE BASED ON NEWTON-COTES 9 POINT RULE
              C
 0001                FUNCTION ANC8(A1,B1,EP,M,N,FUN)
 0002                DOUBLE PRECISION  ANC8,A1,B1,EP,FUN,A,B,EPS,ESUM,TSUM,DA,XB,SX,FA
                    1,F1,FS,F3,FM,F2,FT,F4,FB,FTP,FBP,FMAX,FTST,EST,AEST,EST1,EST2,AEST
                    21,AEST2,ABSAR,DELTA,DIFF,DAFT,SUM,SA
 0003                DOUBLE PRECISION FR,F5,FU,F6,XR,X5,XU,X6,FUP
 0004                DOUBLE PRECISION FQ,F7,FV,F8,XQ,X7,XV,X8,FVP
 0005                DIMENSION F2(30),F4(30),FTP(30),FBP(30),EST2(30),NRTR(30),FTST(9)
 0006                DIMENSION AEST2(30),XB(30)
 0007                DIMENSION FUP(30),F6(30)
 0008                DIMENSION F8(30),FVP(30)
              C      THE PARAMETER SETUP FOR THE INITIAL CALL
 0009                IF(N.LE.0)GO TO 210
 0010                IF(N.GT.3)GO TO 211
 0011                A=A1
 0012                B=B1
 0013                EPS=EP*1023.0D0
 0014                ESUM=0.0D0
 0015                TSUM=0.0D0
 0016                LVL=1
 0017                DA=B-A
 0018                FA=FUN(A)
 0019                FQ=FUN((7.0D0*A+B)/8.0D0)
 0020                FR=FUN((3.0D0*A+B)/4.0D0)
 0021                FS=FUN((5.0D0*A+3.0D0*B)/8.0D0)
 0022                FM=FUN((A+B)*0.5D0)
 0023                FT=FUN((3.0D0*A+5.0D0*B)/8.0D0)
 0024                FU=FUN((A+3.0D0*B)/4.0D0)
 0025                FV=FUN((A+7.0D0*B)/8.0D0)
 0026                FB=FUN(B)
 0027                M=9
 0028                FMAX=DABS(FA)
 0029                FTST(1)=FMAX
 0030                FTST(2)=DABS(FQ)
 0031                FTST(3)=DABS(FR)
 0032                FTST(4)=DABS(FS)
 0033                FTST(5)=DABS(FM)
 0034                FTST(6)=DABS(FT)
 0035                FTST(7)=DABS(FU)
 0036                FTST(8)=DABS(FV)
 0037                FTST(9)=DABS(FB)
 0038                DO 100 I=2,9
 0039                IF(FMAX.GE.FTST(I))GO TO 100
 0040                FMAX=FTST(I)
 0041            100 CONTINUE
 0042                EST=(989.0D0*(FA+FB)+5888.0D0*(FQ+FV)-928.0D0*(FR+FU)+10496.0D0*(F
                    XS+FT)-4540.0D0*FM)*DA/28350.0D0
 0043                ABSAR=(989.0D0*(FTST(1)+FTST(9))+5888.0D0*(FTST(2)+FTST(8))-928.0D
                    X0*(FTST(3)+FTST(7))+10496.0D0*(FTST(4)+FTST(6))-4540.0D0*FTST(5))*
                    XDA/28350.0D0
 0044                AEST=ABSAR
              C      1=RECUR
 0045              1 SX=DA/(28350.0D0*2.0D0**LVL)
```

```
0046              F1=FUN((15.0D0*A+B)/16.0D0)
0047              F3=FUN((13.0D0*A+3.0D0*B)/16.0D0)
0048              F5=FUN((11.0D0*A+5.0D0*B)/16.0D0)
0049              F7=FUN((9.0D0*A+7.0D0*B)/16.0D0)
0050              F2(LVL)=FUN((7.0D0*A+9.0D0*B)/16.0D0)
0051              F4(LVL)=FUN((5.0D0*A+11.0D0*B)/16.0D0)
0052              F6(LVL)=FUN((3.0D0*A+13.0D0*B)/16.0D0)
0053              F8(LVL)=FUN((A+15.0D0*B)/16.0D0)
0054              EST1=SX*(989.0D0*(FA+FM)+5888.0D0*(F1+F7)-928.0D0*(FQ+FS)+10496.0D
                 X0*(F3+F5)-4540.0D0*FR)
0055              FBP(LVL)=FB
0056              XB(LVL)=B
0057              FTP(LVL)=FT
0058              FUP(LVL)=FU
0059              FVP(LVL)=FV
0060              EST2(LVL)=SX*(989.0D0*(FM+FB)+5888.0D0*(F2(LVL)+F8(LVL))-928.0D0*(
                 XFT+FV)+10496.0D0*(F4(LVL)+F6(LVL))-4540.0D0*FU)
0061              SUM=EST1+EST2(LVL)
0062              FTST(1)=DABS(F1)
0063              FTST(2)=DABS(F3)
0064              FTST(3)=DABS(F5)
0065              FTST(4)=DABS(F7)
0066              FTST(5)=DABS(F2(LVL))
0067              FTST(6)=DABS(F4(LVL))
0068              FTST(7)=DABS(F6(LVL))
0069              FTST(8)=DABS(F8(LVL))
0070              FTST(9)=DABS(FM)
0071              AEST1=SX*(989.0D0*(DABS(FA)+FTST(9))+5888.0D0*(FTST(1)+FTST(4))-92
                 X8.0D0*(DABS(FQ)+DABS(FS))+10496.0D0*(FTST(2)+FTST(3))-4540.0D0*DAB
                 XS(FR))
0072              AEST2(LVL)=SX*(989.0D0*(FTST(9)+DABS(FB))+5888.0D0*(FTST(5)+FTST(8
                 X))-928.0D0*(DABS(FT)+DABS(FV))+10496.0D0*(FTST(6)+FTST(7))-4540.0D
                 X0*DABS(FU))
0073              ABSAR=ABSAR-AEST+AEST1+AEST2(LVL)
0074              M=M+8
0075              GO TO (201,200,202),N
0076          200 DELTA=ABSAR
0077              GO TO 205
0078          210 PRINT 39
0079           39 FORMAT(' ERROR RETURN-N.LE.0')
0080              RETURN
0081          211 PRINT 40
0082           40 FORMAT(' ERROR RETURN-N.GT.3')
0083              RETURN
0084          201 DELTA=1.0D0
0085              GO TO 205
0086          202 DO 203 I=1,8
0087              IF(FMAX.GE.FTST(I))GO TO 203
0088              FMAX=FTST(I)
0089          203 CONTINUE
0090              DELTA=FMAX
0091          205 DAFT=EST-SUM
0092              DIFF=DABS(DAFT)
0093              DAFT=DAFT/1023.0D0
0094              IF(DIFF-EPS*DELTA)6,6,3
```

```
0095           3 IF(LVL-30)4,2,2
0096           6 IF(LVL-1)2,4,2
         C       2=UP
0097           2 A=B
0098             ESUM=ESUM+DAFT
0099             TSUM=TSUM+SUM
0100           9 LVL=LVL-1
0101             L=NRTR(LVL)
0102             GO TO (11,12),L
         C       11=R1,12=R2
0103           4 NRTR(LVL)=1
0104             EST=EST1
0105             AEST=AEST1
0106             FB=FM
0107             FV=F7
0108             FU=FS
0109             FT=F5
0110             FM=FR
0111             FS=F3
0112             FR=FQ
0113             FQ=F1
0114             B=(A+B)/2.0D0
0115             EPS=EPS/2.0D0
0116           7 LVL=LVL+1
0117             GO TO 1
0118          11 NRTR(LVL)=2
0119             FA=FB
0120             FQ=F2(LVL)
0121             FR=FTP(LVL)
0122             FS=F4(LVL)
0123             FM=FUP(LVL)
0124             FT=F6(LVL)
0125             FU=FVP(LVL)
0126             FV=F8(LVL)
0127             FB=FBP(LVL)
0128             B=XB(LVL)
0129             EST=EST2(LVL)
0130             AEST=AEST2(LVL)
0131             GO TO 7
0132          12 EPS=2.0D0*EPS
0133             IF(LVL-1)5,5,9
0134           5 ANC8=TSUM-ESUM
0135             RETURN
0136             END
```

6. Adaptive Quadrature Routine Based on Newton-Cotes 11-point Rule

```
FORTRAN IV G LEVEL 1, MOD 1              MAIN              DATE = 68270          21/22/06

      C      ANC10 INTEGRATION
      C      ADAPTIVE QUADRATURE ROUTINE BASED ON NEWTON-COTES 11 POINT RULE
      C
             FUNCTION ANC10(A1,B1,EP,M,N,FUN)
             DOUBLE PRECISION   ANC10,A1,B1,EP,FUN,A,B,EPS,SUM,ESUM,TSUM,DA,XB,S
            1X,F1,FS,F3,FM,F2,FT,F4,FB,FTP,FBP,FMAX,FTST,EST,AEST,EST1,EST2,AES
            2T1,AEST2,ABSAR,DELTA,DIFF,DAFT,FA,SA
             DOUBLE PRECISICN FR,F5,FU,F6,XR,X5,XU,X6,FUP
             DOUBLE PRECISION FQ,F7,FV,F8,XQ,X7,XV,X8,FVP
             DOUBLE PRECISICN FP,F9,FW,F10,XP,X9,XW,X10,FWP
             DIMENSICN F2(30),F4(30),FTP(30),FBP(30),EST2(30),NRTR(30),FTST(11)
             DIMENSION AEST2(30),XB(30)
             DIMENSICN FUP(30),F6(30)
             DIMENSION F8(30),FVP(30)
             DIMENSION F10(30),FWP(30)
      C      THE PARAMETER SETUP FOR THE INITIAL CALL
             IF(N.LE.0)GO TO 210
             IF(N.GT.3)GO TC 211
             A=A1
             B=B1
             EPS=EP*4095.0D0
             ESUM=0.0D0
             TSUM=C.0D0
             LVL=1
             DA=B-A
             FA=FUN(A)
             FP=FUN((9.0D0*A+B)/10.0D0)
             FQ=FUN((4.0D0*A+B)/5.0D0)
             FR=FUN((7.0D0*A+3.0D0*B)/10.0D0)
             FS=FUN((3.0D0*A+2.0D0*B)/5.0D0)
             FM=FUN((A+B)*0.5D0)
             FT=FUN((2.0D0*A+3.0D0*B)/5.0D0)
             FU=FUN((3.0D0*A+7.0D0*B)/10.0D0)
             FV=FUN((A+4.0D0*B)/5.0D0)
             FW=FUN((A+9.0D0*B)/10.0D0)
             FB=FUN(B)
             M=11
             FMAX=DABS(FA)
             FTST(1)=FMAX
             FTST(2)=DABS(FP)
             FTST(3)=DABS(FQ)
             FTST(4)=DABS(FR)
             FTST(5)=DABS(FS)
             FTST(6)=DABS(FM)
             FTST(7)=DABS(FT)
             FTST(8)=DABS(FU)
             FTST(9)=DABS(FV)
             FTST(10)=DABS(FW)
             FTST(11)=DABS(FB)
             DO 100I=2,11
             IF(FMAX.GE.FTST(I))GO TO 100
             FMAX=FTST(I)
         100 CONTINUE
             EST=(16067.0D0*(FA+FB)+106300.0D0*(FP+FW)-48525.0D0+(FQ+FV)+272400
            X.0D0*(FR+FU)-260550.0D0*(FS+FT)+427368.0D0*FM)*DA/598752.0D0
```

```
0049           ABSAR=(16067.0D0*(FTST(1)+FTST(11))+106300.0D0*(FTST(2)+FTST(10))-
              X48525.0D0*(FTST(3)+FTST(9))+272400.0D0*(FTST(4)+FTST(8))-260550.0D
              X0*(FTST(5)+FTST(7))+427368.0D0*FM)*DA/598752.0D0
0050           AEST=ABSAR
         C     1=RECUR
0051         1 SX=DA/(598752.0D0*2.0D0**LVL)
0052           F1=FUN((19.0D0*A+B)/20.0D0)
0053           F3=FUN((17.0D0*A+3.0D0*B)/20.0D0)
0054           F5=FUN((15.0D0*A+5.0D0*B)/20.0D0)
0055           F7=FUN((13.0D0*A+7.0D0*B)/20.0D0)
0056           F9=FUN((11.0D0*A+9.0D0*B)/20.0D0)
0057           F2(LVL)=FUN((9.0D0*A+11.0D0*B)/20.0D0)
0058           F4(LVL)=FUN((7.0D0*A+13.0D0*B)/20.0D0)
0059           F6(LVL)=FUN((5.0D0*A+15.0D0*B)/20.0D0)
0060           F8(LVL)=FUN((3.0D0*A+17.0D0*B)/20.0D0)
0061           F10(LVL)=FUN((A+19.0D0*B)/20.0D0)
0062           EST1=SX*(16067.0D0*(FA+FM)+106300.0D0*(F1+F9)-48525.0D0*(FP+FS)+27
              X2400.0D0*(F3+F7)-260550.0D0*(FQ+FR)+427368.0D0*F5)
0063           FBP(LVL)=FB
0064           XB(LVL)=B
0065           FTP(LVL)=FT
0066           FUP(LVL)=FU
0067           FVP(LVL)=FV
0068           FWP(LVL)=FW
0069           EST2(LVL)=SX*(16067.0D0*(FM+FB)+106300.0D0*(F2(LVL)+F10(LVL))-4852
              X5.0D0*(FT+FW)+272400.0D0*(F4(LVL)+F8(LVL))-260550.0D0*(FU+FV)+4273
              X68.0D0*F6(LVL))
0070           SUM=EST1+EST2(LVL)
0071           FTST(1)=DABS(F1)
0072           FTST(2)=DABS(F3)
0073           FTST(3)=DABS(F5)
0074           FTST(4)=DABS(F7)
0075           FTST(5)=DABS(F9)
0076           FTST(6)=DABS(F2(LVL))
0077           FTST(7)=DABS(F4(LVL))
0078           FTST(8)=DABS(F6(LVL))
0079           FTST(9)=DABS(F8(LVL))
0080           FTST(10)=DABS(F10(LVL))
0081           FTST(11)=DABS(FM)
0082           AEST1=SX*(16067.0D0*(DABS(FA)+FTST(11))+106300.0D0*(FTST(1)+FTST(5
              X))-48525.0D0*(DABS(FP)+DABS(FS))+272400.0D0*(FTST(2)+FTST(4))-2605
              X50.0D0*(DABS(FQ)+DABS(FR))+427368.0D0*FTST(3))
0083           AEST2(LVL)=SX*(16067.0D0*(FTST(11)+DABS(FB))+106300.0D0*(FTST(6)+F
              XTST(10))-48525.0D0*(DABS(FT)+DABS(FW))+272400.0D0*(FTST(7)+FTST(9)
              X)-260550.0D0*(DABS(FU)+DABS(FV))+427368.0D0*FTST(8))
0084           ABSAR=ABSAR-AEST+AEST1+AEST2(LVL)
0085           M=M+10
0086           GO TO (201,200,202),N
0087       200 DELTA=ABSAR
0088           GO TO 205
0089       210 PRINT 39
0090        39 FORMAT(' ERROR RETURN-N.LE.0')
0091           RETURN
0092       211 PRINT 40
0093        40 FORMAT(' ERROR RETURN-N.GT.3')
```

```
      RETURN
  201 DELTA=1.0D0
      GO TO 205
  202 DO 203 I=1,10
      IF(FMAX.GE.FTST(I))GO TO 203
      FMAX=FTST(I)
  203 CONTINUE
      DELTA=FMAX
  205 DAFT=EST-SUM
      DIFF=DABS(DAFT)
      DAFT=DAFT/4095.0D0
      IF(DIFF-EPS*DELTA)6,6,3
    3 IF(LVL-30)4,2,2
    6 IF(LVL-1)2,4,2
C     2=UP
    2 A=B
      ESUM=ESUM+DAFT
      TSUM=TSUM+SUM
    9 LVL=LVL-1
      L=NRTR(LVL)
      GO TO (11,12),L
C     11=R1,12=R2
    4 NRTR(LVL)=1
      EST=EST1
      AEST=AEST1
      FB=FM
      FW=F9
      FV=FS
      FU=F7
      FT=FR
      FM=F5
      FS=FQ
      FR=F3
      FQ=FP
      FP=F1
      B=(A+B)/2.0D0
      EPS=EPS/2.0D0
    7 LVL=LVL+1
      GO TO 1
   11 NRTR(LVL)=2
      FA=FB
      FP=F2(LVL)
      FQ=FTP(LVL)
      FR=F4(LVL)
      FS=FUP(LVL)
      FM=F6(LVL)
      FT=FVP(LVL)
      FU=F8(LVL)
      FV=FWP(LVL)
      FW=F10(LVL)
      FB=FBP(LVL)
      B=XB(LVL)
      EST=EST2(LVL)
      AEST=AEST2(LVL)
      GO TO 7
   12 EPS=EPS*2.0D0
      IF(LVL-1)5,5,9
    5 ANC10=TSUM-ESUM
      RETURN
      END
```

REFERENCES

1. McKeeman, W. M., Algorithm 145, *Adaptive Numerical Integration by Simpson's Rule,* Comm. ACM, Vol. 5 (1962), p. 604.

2. McKeeman, W. M., and Tesler, L., Algorithm 182, *Nonrecursive Adaptive Integration,* Comm. ACM, Vol. 6 (1963), p. 315.

3. McKeeman, W. M., Algorithm 198, *Adaptive Integration and Multiple Integration,* Vol. 6 (1963), p. 443.

4. Davis, P., and Rabinowitz, P., *Numerical Integration,* Blaisdell Publ. Co., London, 1967.

5. Lyness, J. N., *Notes on the Adaptive Simpson Quadrature Routine,* submitted to JACM.