32

An Optimal Adaptative Numerical Integration Method

A. Nicolet, A. Genon, W. Legros, M. Umé, F. Delincé*

* This author is a Research Assistant with the Belgian National Fund for Scientific Research

University of Liège - Dept. of Electrical Engineering - Institut Montefiore
Sart Tilman B28, B-4000 Liège, Belgium

ABSTRACT

This paper presents an adaptative extension of the Gaussian integration method. It is well known that the Gaussian integration method is optimal for sufficiently smooth functions (i.e. which may be approximated by a polynomial) in the sense that it gives the maximum accuracy for a given number of nodes. Unfortunately it is not always possible to choose a priori the number of nodes for the integration. One alternative is to try successive Gaussian formulae with an increasing number of points until they agree with the required accuracy. In this case, most of the advantages of the method are lost. A less accurate but naturally adaptative method such as the Romberg method may become a better solution.

The idea of the optimal adaptative method is to find a series of integration formulae with an increasing number of nodes in order that the set of abscissae of lower order formulae is a subset of abscissae of higher order formulae. Then, the sequential evaluation of formulae of increasing order only requires the addition of new points. Under this constraint, the remaining degrees of freedom (the new abscissae and all the weight factors) are used to obtain formulae of maximum order.

INTRODUCTION

Some well-known and classical methods [1] are reviewed in order to situate the new method. Those methods are valid for sufficiently smooth integrands.

Trapezoidal method

Amongst the simplest, this method consists in choosing equally-spaced points between the endpoints and to approximate the function by piecewise linear functions. The trapezoidal rule is :

$$I(h) = \int_a^b f(x)\,dx \approx h\left[\frac{1}{2} f_0 + f_1 + \cdots f_i + \cdots f_{N-1} + \frac{1}{2} f_N\right] \qquad (1)$$

with : $h = (b - a) / N$ and $f_i = f(a + i\,h)$

Romberg method

The basic idea is to use the results from successive refinements of the trapezoidal rule :

$$I_0 = I(b - a)\ ,\ I_1 = I((b - a) / 2)\ ,\ \ldots\ ,\ I_i = I((b - a) / 2^i)\ ,\ \ldots \qquad (2)$$

The Richardson extrapolation is applied to this sequence in order to eliminate high order error terms[2].

This method is naturally adaptative. For each refinement, a new trapezoidal approximation is computed with the number of points multiplied by two and reusing the previously computed values of the function. Then the Richardson extrapolation is applied to the new sequence. This process is repeated until the required accuracy is reached. The number of function evaluations is not known a priori and depends on the integrand.

General rule

Most of the integration rules to integrate the following expression :

$$\int_a^b K(x)\ f(x)\ dx \qquad (3)$$

have the form :

$$\sum_{i=1}^{n} w_i\ f(x_i) \qquad (4)$$

The approximation is a linear combination, with weight factors w_i, of values of the function f(x), for n absissae x_i.

In the rest of the paper, the integral (3), i.e. the integral on the interval [a,b] of the product of the function f(x) with the kernel K(x), will be referred as "the integral of f(x)". Formulae (4) are tabulated in the litterature for some kernels K(x) and an associated interval [a,b].

Orthogonal polynomials [3] associated to the set 'kernel K(x) - interval [a,b]' play an important role in the theory (table 1).

Kernel K(x)	Interval a	b	Associated orthogonal polynomials
1.	-1.	1.	Legendre
e^{-x}	0.	∞	Laguerre
e^{-x^2}	$-\infty$	∞	Hermite
$1/\sqrt{1-x^2}$	-1.	1.	Tschebychev (first kind)
ln(x)	0.	1.	Orthogonal polynomials associated to ln(x) on [0,1] (Berthod-Zaborowski formulae)

***Table 1.** Orthogonal polynomials [4]*

The orthogonality of polynomials $P_i(x)$, is expressed by [3] :

$$\int_a^b K(x)\, P_i(x)\, P_j(x)\, dx = \begin{cases} 0 & \text{if } i \neq j \\ \lambda_i & \text{if } i = j \end{cases} \tag{5}$$

A numerical integration rule is characterized by an integer p such that all the polynomials $Q_k(x)$ of order k less than or equal to p are exactly integrated :

$$\int_a^b K(x)\, Q_k(x)\, dx = \sum_{i=1}^{n} w_i\, Q_k(x_i) \tag{6}$$

Gaussian method

Gaussian integration rules are optimal in the sense that p is maximal. An n point rule that has 2n degrees of freedom (n abscissae and n weight factors) can integrate exactly all the polynomials up to the order 2n-1 (i.e. polynomials that have up to 2n coefficients).

The necessary and sufficient condition is that all the powers of x up to the order 2n-1 are correctly integrated on the considered interval :

$$\int_a^b K(x)\, x^k\, dx = m_k = \sum_{i=1}^{n} w_i\, x_i^k \qquad k = 0, \ldots, 2n-1 \tag{7}$$

The relations (7) constitute a system of 2n equations with 2n unknowns (the x_i and the w_i) whose solution gives the parameters of the integration rule. This system is unfortunately non linear and difficult to solve in this form. The classical approach is to introduce an auxiliary polynomial $\pi(x)$ whose roots are the abscissae x_i :

$$\pi(x) = \prod_{i=1}^{n} (x-x_i) = \sum_{i=0}^{n} c_i\, x^i \qquad (c_n=1.) \tag{8}$$

Sums of equations (7) weighted by the coefficients c_i are constructed, in order to do appear $\pi(x_i)$:

$$\sum_{i=0}^{n} c_i\, m_{i+p} = \sum_{i=0}^{n} c_i \left(\sum_{j=1}^{n} w_j\, x_j^{i+p} \right) = \sum_{j=1}^{n} w_j \left(\sum_{i=0}^{n} c_i\, x_j^{i+p} \right) \tag{9}$$

$$= \sum_{j=1}^{n} w_j\, \pi(x_j)\, x_j^p = 0 \qquad \text{for} \quad p = 0, \ldots, n-1$$

Relations (9) constitute a system of n equations with n unknowns that gives the coefficients c_i :

$$\sum_{i=0}^{n-1} c_i\, m_{i+p} = m_{n+p} \qquad \text{for} \quad p = 0, \ldots, n-1 \tag{10}$$

The solution of the system (10) determines the polynomial $\pi(x)$ whose roots are the abscissae x_i of the integration rule.

A relationship between this polynomial and the orthogonal polynomials associated to the problem may be found.

$P_n(x)$ is the orthogonal polynomial of degree n associated to the problem. Any polynomial $f_{2n-1}(x)$ of degree 2n-1 may be expressed as:

$$f_{2n-1}(x) = g(x)\, P_n(x) + r(x) \tag{11}$$

with a quotient polynomial g(x) and a rest polynomial r(x), both of degree at most equal to n-1.

The polynomial $f_{2n-1}(x)$ is integrated exactly :

$$\int_a^b K(x)\, g(x)\, P_n(x)\, dx + \int_a^b K(x)\, r(x)\, dx = \sum_{i=1}^{n} w_i\, g(x_i)\, P_n(x_i) + \sum_{i=1}^{n} w_i\, r(x_i) \tag{12}$$

The polynomial $P_n(x)$ being orthogonal to all the polynomials of degree less or equal to n , the first term of the left hand member is equal to zero. The relation (12) is always true only if the first term of the right hand member is cancelled, what is assured if $P_n(x_i)$ is equal to zero for i = 1, ..., n. Thus $P_n(x)$ is equal to $\pi(x)$ apart from a constant factor.

The theory of the orthogonal polynomials assure that their roots are simple, real and situated in the interval [a , b]. The abscissae being determined, the w_i must be computed in order that any polynomial of order less than or equal to n-1 is exactly integrated. This problem is solved in the next paragraph independently of the Gaussian method.

If the number of points required to compute an integral with the satisfactory accuracy is known a priori, the Gaussian rule is the best method. Unfortunately, this is not the case in most practical problems. A possiblity is to try a sequence of Gaussian rules of increasing order until the difference between two approximations is less than the required accuracy. But, as the abscissae differ from one rule to the other, the advantages are lost and a naturally adaptative method such as the Romberg method becomes more relevant.

WEIGHT COEFFICIENT DETERMINATION

The problem of determining the 'optimal' weighting factors w_i associated to any given n abscissae x_i may be solved in a general way [5]. The n factors w_i are computed in order that any polynomial up to order n-1 is integrated exactly.

A polynomial $f_{n-1}(x)$ of order n-1 is integrated by the rule :

$$\int_a^b K(x)\, f_{n-1}(x)\, dx = \sum_{i=1}^{n} w_i\, f_{n-1}(x_i) \tag{13}$$

The Lagrange interpolation polynomials $L_i(x)$ are defined by :

$$L_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{(x - x_j)}{(x_i - x_j)} \tag{14}$$

Any polynomial $f_{n-1}(x)$ may be written

$$f_{n-1}(x) = \sum_{i=1}^{n} L_i(x)\, f_{n-1}(x_i) \tag{15}$$

Introducing expression (15) in (13), gives, for any set of $f_{n-1}(x_i)$:

$$\int_a^b K(x) \left(\sum_{i=1}^{n} L_i(x)\, f_{n-1}(x_i) \right) dx = \sum_{i=1}^{n} w_i\, f_{n-1}(x_i) \tag{16}$$

Identifying the coefficients of $f_{n-1}(x_i)$ in the two members gives :

$$w_i = \int_a^b K(x)\, L_i(x)\, dx \tag{17}$$

In the particular case of the Gaussian method, it can be shown that this determination of the w_i is equivalent to the solution of the n first equations of (7), linear with respect to w_i, with the x_i given. The w_i obtained for the Gaussian rules are always positive if K(x) is positive on the interval [a,b]. Indeed, the polynomial $L_i^2(x)$ of degree 2(n-1) (the square of a Lagrange interpolating polynomial) may be integrated exactly by the Gaussian rule :

$$\int_a^b K(x)\, L_i^2(x)\, dx = \sum_{j=1}^{n} L_i^2(x_j)\, w_j \tag{18}$$

The first member is positive because $K(x) > 0$ on [a , b] and $L_i^2(x) > 0$. The right hand member terms are all equal to zero except for i=j and then :

$$w_i = \frac{\int_a^b K(x)\, L_i^2(x)\, dx}{L_i^2(x_i)} > 0 \tag{19}$$

This property assures a good response to Gaussian rules from the rounding error point of view.

OPTIMAL ADDITION OF POINTS

Starting from a given integration rule, it is possible to add points in an optimal way [5,6], i.e. to combine the already computed values of the function with new values in order to integrate exactly polynomials of degree as high as possible. An existing integration rule may be extended without wasting any integrand computation. This is particularly important for an adaptative integration. If a n point formula is extended with p new points, n+2p degrees of freedom are available (the p new abscissae and the n+p weighting factors for all the abscissae). A new formula may be found that integrates exactly all the polynomials up to the order n+2p-1 and, for $F_{n+2p-1}(x)$, such a polynomial of this order :

$$\int_a^b K(x)\, F_{n+2p-1}(x)\, dx = \sum_{i=1}^{n+p} w_i\, F_{n+2p-1}(x_i) \tag{20}$$

A polynomial $G_{n+p}(x)$ is introduced whose roots are the abscissae of the extended rule (The new ones as well as the old ones) :

$$G_{n+p}(x) = \prod_{i=1}^{n+p} (x-x_i) \tag{21}$$

the polynomial $F_{n+2p-1}(x)$ may be expressed :

$$F_{n+2p-1}(x) = R_{n+p-1}(x) + G_{n+p}(x)\, Q_{p-1}(x) \tag{22}$$

where $R_{n+p-1}(x)$ is the rest polynomial of degree at most equal to n+p-1 and $Q_{p-1}(x)$ the quotient polynomial of degree p-1.

The equality (20) is then :

$$\int_a^b K(x)\, R_{n+p-1}(x)\, dx + \int_a^b K(x)\, G_{n+p}(x)\, Q_{p-1}(x)\, dx$$
$$= \sum_{i=1}^{n+p} w_i\, R_{n+p-1}(x_i) + \sum_{i=1}^{n+p} w_i\, G_{n+p}(x_i)\, Q_{p-1}(x_i) \tag{23}$$

By definition of G_{n+p}, the second term of the right hand member is equal to zero. Moreover, if the n+p weight factors w_i have been computed in order to integrate exactly all the polynomials up to degree n+p-1 (see above), the first term of both members of (23) are equal.

Thus :

$$\int_a^b K(x)\, G_{n+p}(x)\, Q_{p-1}(x)\, dx = 0 \tag{24}$$

for any polynomial $Q_{p-1}(x)$ of order less or equal to p-1.

As a particular case, the associated orthogonal polynomials $P_k(n)$ may be introduced in (24) :

$$\int_a^b K(x)\, G_{n+p}(x)\, P_k(x)\, dx = 0 \qquad \text{pour } k = 0, ..., p-1 \tag{25}$$

The polynomial $G_{n+p}(x)$ may be expressed as a linear combination of associated orthogonal polynomials :

$$G_{n+p}(x) = \sum_{i=0}^{n+p} t_i P_i(x) \tag{26}$$

By introducing expression (26) in equation (25) , the orthogonality property of polynomials P_i gives directly $t_k = 0$ for k = 0, ..., p-1. Then :

$$G_{n+p}(x) = \sum_{i=p}^{n+p} t_i P_i(x) \tag{27}$$

The coefficient t_{n+p} is chosen equal to 1 and the n remaining coefficients are determined by expressing that the n initial abscissae x_j' are already known roots of G_{n+p}.

$$G_{n+p}(x_i') = \sum_{i=p}^{n+p-1} t_i P_i(x_j') + P_{n+p}(x_j') = 0 \qquad j = 1, ..., n \tag{28}$$

The relations (28) constitute a linear system of n equations with n unknowns whose solution yield the expression of G_{n+p} as a function of orthogonal polynomials. The finding of the p supplementary roots of G_{n+p} yields the p new abscissae x_i. Finally , all the abscissae being known, the weighting factors may be determined in a classical way (see above). It may not be guaranteed, in a general way, that the obtained rule has practicable characteristics (simple roots, real and all situated in the interval of integration, positive weighting factors). It may be shown that a n point Gaussian rule must be extended with at least n+1 points [6]. In this case, the interpolant polynomial in (17) is the product of an orthogonal polynomial of degree n corresponding to the initial abscissae by a polynomial of order p-1 corresponding to the added abscissae. If p-1 is less than n, the expression (17) gives a weighting factor equal to zero because the an orthogonal polynomial is orthogonal to all the polynomials of inferior degree .

In the case of a kernel K(x) and of an interval [a,b] both symmetrical with respect to the origin (i.e. K(x) = K(-x) and a = -b), the equalities $w_i=w_{n-i+1}$ and $x_i = -x_{n-i+1}$ are true for any n point rule.

Thanks to the symmetry of abscissae and to the parity of the involved polynomials (26) is reduced to [5]:

$$G_{n+p}(x) = \sum_{i=1}^{[n/2]+1} c_i P_{2i-2+p+q}(x) \tag{29}$$

with :

q=n-2[n/2]

[n/2] = integer part of n/2

and the system (28) is simplified to a system of [n/2] equations with [n/2] unknowns :

$$\sum_{i=1}^{[n/2]} c_i \, P_{2i-2+p+q}(x_j') = - P_{n+p}(x_j') \qquad j = 1, ..., [n/2] \qquad (30)$$

Only the first [n/2] supplementary roots of G_{n+p} have to be determined, the remaining ones are obtained directly by symmetry.

PATTERSON METHOD

A practical problem is to find a particular sequence of practicable rules by applying the preceding theory. The Patterson method [7] is an example: the starting point is a one point formula (the value of the function at the center of the interval multiplied by the length of the interval), two points are optimally added to obtain a 3 point formula (which is the 3 point Gaussian rule), then 4 points are added to obtain a 7 point formula (which is not a Gaussian rule), and so on. At step n, n+1 points interlaced with the previous ones (figure 1) are added.

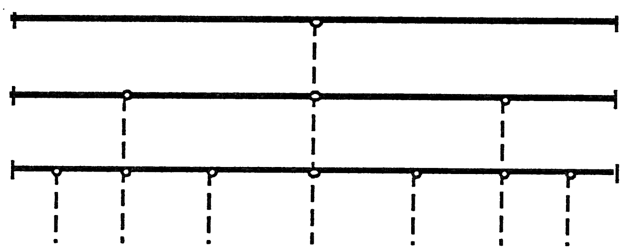

Figure 1. Optimal addition of points

A sequence of rules with 1, 3, 7, 15, 31, 63, 127, .. points is obtained, where each rule reuses the previous computed values of the integrand. An n point rule, obtained by optimal addition of points, integrate exactly the polynomials up to degree (3n-1)/2, what is not so far from the n point Gaussian rule which is exact up to degree 2n-1. The comparison between successive approximations gives an error estimate. This method is naturally adaptative and a Fortran program may be found in annex. Another Fortran program with 20 figure coefficients and formulae up to 255 points may be found in [7].

SINGULAR AND QUASI-SINGULAR INTEGRALS

The boundary element method involves the integration of singular and quasi-singular kernels. For the numerical integration method, the integrands were supposed to be smooth enough functions to be well approximated by polynomials.

Nevertheless, the Patterson method may be efficiently applied if a change of variable is made in order to even out peaks or singularities [8,9].

For instance, the integral of a function f(t) with a singularity or a peak at $t = t_{min}$ is considered (in practice, this occurs when an influenced point is close to an influencing element, t_{min} corresponds to the parametric coordinate of the point of the influencing element at a minimum distance of or the same as the influenced point) :

$$I = \int_0^1 f(t)\, dt \qquad (31)$$

The following change of variable is performed :

$$t - t_{min} = u^3 \qquad (32)$$

The integral (31) becomes :

$$I = 3 \int_{-\sqrt[3]{t_{min}}}^{+\sqrt[3]{1-t_{min}}} f(u^3 + t_{min})\, u^2\, du \qquad (33)$$

In expression (33), the point corresponding to $t = t_{min}$ is $u = 0$. Thus the singularity or the peak is eliminated by the term u^2 in (33).

The practical algorithm is :

- Check if the influenced point is on or close enough to the influencing element (The proximity criterion has been empirically chosen: a point is close enough to an element if its distance is less than one tenth of the length of the element);

- If the point is close enough, choose the expression (33), otherwise, choose expression (31);

- Apply the adaptative Patterson method on the chosen expression. The criterion for stopping the process is to have two successive approximations with a relative difference less than 10^{-4}.

CONCLUSION

One of the characteristics of the boundary element method is that it involves the computations of integrals ranging from very easy to singular. The proposed algorithm allows for the use of an efficient method with an adaptative order, and workable for the whole set of integrations. This provides accuracy and security (error is controlled) for a rather low computationnal cost.

REFERENCES

[1] P.J. DAVIS, P. RABINOWITZ. "Numerical integration". Blaisdell, 1967.

[2] W.H. PRESS, B.P. FLANNERY, S.A. TEUKOLSKY, W.T. VETTERLING. "Numerical recipes : the art of scientific computing". Cambridge University Press, 1989.

[3] H. HOCHSTADT. "The functions of mathematical physics". Dover, 1986.

[4] M. ABRAMOWITZ, I.A. STEGUN ed. "Handbook of mathematical functions". Dover, 1972.

[5] T.N.L. PATTERSON. "The optimum addition of points to quadrature formulae". Math. Comp. 22, 1968, pp.847-856.

[6] T.N.L. PATTERSON. "On some Optimally and Lobatto based quadrature formulae". Math. Comp. 22, 1968, pp.877-881.

[7] T.N.L. PATTERSON. "Algorithm for automatic numerical integration over a finite interval". Communications of the ACM, Vol. 16, n°11, November 1973, pp.694-699.

[8] J.P. ADRIAENS, P. BOURMANNE, F. DELINCE, A. GENON, A. NICOLET, W. LEGROS. "Numerical computations of eddy currents in thin plates". IEEE Transactions on Magnetics, Vol. MAG-26, n°5, September 1990, pp. 2376-2378.

[9] A. NICOLET. "Modélisation du champ magnétique dans les systèmes comprenant des milieux non linéaires". Ph.D. Thesis, University of Liège, 1991.

ANNEX

```
C
C/*INTEG
C-------------------------------------------------------------
      FUNCTION INTEG(A,B,F,EPS,ERR)
C-------------------------------------------------------------
C     Integrate the function F between A and B
C     using the adaptative Patterson method.
C     The required relative accuracy is EPS
C     and the obtained relative accuracy is ERR
C
C     written by A. Nicolet
C
      IMPLICIT NONE
C
      REAL C(63),CO(189),F,INTEG,A,B,F0
      REAL EPS,ERR,VAL,VAL1,ABS0,ABS1,ABS2
      INTEGER I,N,N1,N2,IP,NLIM
      EXTERNAL F
C
      DATA CO(  1),CO(  2)/.7745966692414834,.8888888888888889/
      DATA CO(  3),CO(  4)/.5555555555555556,.4342437493468026/
      DATA CO(  5),CO(  6)/.9604912687080203,.4509165386584741/
      DATA CO(  7),CO(  8)/.4013974147759622,.2684880898683334/
      DATA CO(  9),CO( 10)/.1046562260264673,.2233866864289669/
      DATA CO( 11),CO( 12)/.6211029467372264,.8884592328722570/
      DATA CO( 13),CO( 14)/.9938319632127550,.2255104997982067/
      DATA CO( 15),CO( 16)/.2191568584015875,.2006285293769890/
      DATA CO( 17),CO( 18)/.1715119091363914,.1344152552437842/
      DATA CO( 19),CO( 20)/
     &     9.2927195315124535E-02,5.1603282997079735E-02/
      DATA CO( 21),CO( 22)/
     &     1.7001719629940262E-02,.1124889431331866/
      DATA CO( 23),CO( 24)/.3311353932579768,.5313197436443756/
      DATA CO( 25),CO( 26)/.7024962064915271,.8367259381688687/
      DATA CO( 27),CO( 28)/.9296548574297401,.9815311495537401/
      DATA CO( 29),CO( 30)/.9990981249676676,.1127552567207687/
      DATA CO( 31),CO( 32)/.1119568730209535,.1095784210559246/
      DATA CO( 33),CO( 34)/.1056698935802348,.1003142786117956/
      DATA CO( 35),CO( 36)/
     &     9.3627109981264474E-02,8.5755920049990352E-02/
      DATA CO( 37),CO( 38)/
     &     7.6879620499003530E-02,6.7207754295990704E-02/
      DATA CO( 39),CO( 40)/
     &     5.6979509494123357E-02,4.6462893261757987E-02/
      DATA CO( 41),CO( 42)/
     &     3.5957103307129322E-02,2.5807598096176654E-02/
      DATA CO( 43),CO( 44)/
     &     1.6446049854387805E-02,8.4345657393211071E-03/
      DATA CO( 45),CO( 46)/
     &     2.5447807915618750E-03,5.6344313046592790E-02/
      DATA CO( 47),CO( 48)/.1682352515522075,.2777498220218243/
      DATA CO( 49),CO( 50)/.3833593241987303,.4836180269458410/
      DATA CO( 51),CO( 52)/.5771957100520458,.6629096600247806/
      DATA CO( 53),CO( 54)/.7397560443526948,.8069405319502176/
```

```
 DATA CO( 55),CO( 56)/.8639079381936905,.9103711569570043/
 DATA CO( 57),CO( 58)/.9463428583734029,.9721828747485818/
 DATA CO( 59),CO( 60)/.9886847575474295,.9972062593722220/
 DATA CO( 61),CO( 62)/
&     .9998728881203576,5.6377628360384718E-02/
 DATA CO( 63),CO( 64)/
&     5.6277699831254301E-02,5.5978436510476320E-02/
 DATA CO( 65),CO( 66)/
&     5.5481404356559364E-02,5.4789210527962865E-02/
 DATA CO( 67),CO( 68)/
&     5.3905499335266064E-02,5.2834946790116520E-02/
 DATA CO( 69),CO( 70)/
&     5.1583253952048459E-02,5.0157139305899537E-02/
 DATA CO( 71),CO( 72)/
&     4.8564330406673199E-02,4.6813554990628012E-02/
 DATA CO( 73),CO( 74)/
&     4.4914531653632198E-02,4.2877960025007734E-02/
 DATA CO( 75),CO( 76)/
&     4.0715510116944319E-02,3.8439810249455532E-02/
 DATA CO( 77),CO( 78)/
&     3.6064432780782573E-02,3.3603877148207731E-02/
 DATA CO( 79),CO( 80)/
&     3.1073551111687965E-02,2.8489754745833549E-02/
 DATA CO( 81),CO( 82)/
&     2.5869679327214747E-02,2.3231446639910270E-02/
 DATA CO( 83),CO( 84)/
&     2.0594233915912711E-02,1.7978551568128270E-02/
 DATA CO( 85),CO( 86)/
&     1.5406750466559498E-02,1.2903800100351266E-02/
 DATA CO( 87),CO( 88)/
&     1.0498246909621322E-02,8.2230079572359297E-03/
 DATA CO( 89),CO( 90)/
&     6.1155068221172465E-03,4.2176304415588542E-03/
 DATA CO( 91),CO( 92)/
&     2.5790497946856862E-03,1.2651565562300700E-03/
 DATA CO( 93),CO( 94)/
&     3.6322148184553001E-04,2.8184648949745694E-02/
 DATA CO( 95),CO( 96)/
&     8.4454040083710883E-02,.1404242331525602/
 DATA CO( 97),CO( 98)/.1958975027111002,.2506787303034832/
 DATA CO( 99),CO(100)/.3045764415567140,.3574038378315322/
 DATA CO(101),CO(102)/.4089798212298887,.4591300119898323/
 DATA CO(103),CO(104)/.5076877575337166,.5544951326319325/
 DATA CO(105),CO(106)/.5994039302422429,.6422766425097595/
 DATA CO(107),CO(108)/.6829874310910792,.7214230853700989/
 DATA CO(109),CO(110)/.7574839663805136,.7910849337998484/
 DATA CO(111),CO(112)/.8221562543649804,.8506444947683503/
 DATA CO(113),CO(114)/.8765134144847053,.8997448997769400/
 DATA CO(115),CO(116)/.9203400254700124,.9383203977795929/
 DATA CO(117),CO(118)/.9537300064257611,.9666378515584166/
 DATA CO(119),CO(120)/.9771415146397057,.9853714995985204/
 DATA CO(121),CO(122)/.9914957211781061,.9957241046984070/
 DATA CO(123),CO(124)/.9983166353184119,.9995987996718457/
 DATA CO(125),CO(126)/
&     .9999824303550674,2.8188814180192359E-02/
 DATA CO(127),CO(128)/
&     2.8176319033016602E-02,2.8138849915627151E-02/
```

```
 DATA CO(129),CO(130)/
&      2.8076455793817246E-02,2.7989218255238160E-02/
 DATA CO(131),CO(132)/
&      2.7877251476613702E-02,2.7740702178279682E-02/
 DATA CO(133),CO(134)/
&      2.7579749566481873E-02,2.7394605263981432E-02/
 DATA CO(135),CO(136)/
&      2.7185513229624792E-02,2.6952749667633032E-02/
 DATA CO(137),CO(138)/
&      2.6696622927450360E-02,2.6417473395058260E-02/
 DATA CO(139),CO(140)/
&      2.6115673376706098E-02,2.5791626976024230E-02/
 DATA CO(141),CO(142)/
&      2.5445769965464766E-02,2.5078569652949769E-02/
 DATA CO(143),CO(144)/
&      2.4690524744487677E-02,2.4282165203336599E-02/
 DATA CO(145),CO(146)/
&      2.3854052106038540E-02,2.3406777495314006E-02/
 DATA CO(147),CO(148)/
&      2.2940964229387749E-02,2.2457265826816099E-02/
 DATA CO(149),CO(150)/
&      2.1956366305317825E-02,2.1438980012503867E-02/
 DATA CO(151),CO(152)/
&      2.0905851445812024E-02,2.0357755058472159E-02/
 DATA CO(153),CO(154)/
&      1.9795495048097500E-02,1.9219905124727766E-02/
 DATA CO(155),CO(156)/
&      1.8631848256138790E-02,1.8032216390391286E-02/
 DATA CO(157),CO(158)/
&      1.7421930159464174E-02,1.6801938574103865E-02/
 DATA CO(159),CO(160)/
&      1.6173218729577720E-02,1.5536775555843982E-02/
 DATA CO(161),CO(162)/
&      1.4893641664815182E-02,1.4244877372916774E-02/
 DATA CO(163),CO(164)/
&      1.3591571009765547E-02,1.2934839663607374E-02/
 DATA CO(165),CO(166)/
&      1.2275830560082770E-02,1.1615723319955135E-02/
 DATA CO(167),CO(168)/
&      1.0955733387837902E-02,1.0297116957956356E-02/
 DATA CO(169),CO(170)/
&      9.6411777297025368E-03,8.9892757840641358E-03/
 DATA CO(171),CO(172)/
&      8.3428387539681576E-03,7.7033752332797418E-03/
 DATA CO(173),CO(174)/
&      7.0724899954335555E-03,6.4519000501757369E-03/
 DATA CO(175),CO(176)/
&      5.8434498758356395E-03,5.2491234548088592E-03/
 DATA CO(177),CO(178)/
&      4.6710503721143218E-03,4.1115039786546928E-03/
 DATA CO(179),CO(180)/
&      3.5728927835172987E-03,3.0577534101755354E-03/
 DATA CO(181),CO(182)/
&      2.5687649437940377E-03,2.1088152457265515E-03/
 DATA CO(183),CO(184)/
&      1.6811428654211222E-03,1.2895240826120425E-03/
 DATA CO(185),CO(186)/
```

```
     &      9.3836984854889166E-04,6.3260731933381262E-04/
      DATA CO(187),CO(188)/
     &      3.7774664625235091E-04,1.8073956470813311E-04/
      DATA CO(189)/5.0536095040845502E-05/
C
      IF (EPS.LE.0.) RETURN
C
C     initialization
      N1   = 64
      N2   = 32
      NLIM = 63
      N    = 0
      IP   = 0
C
      F0 = F(.5*(A+B))
      VAL1 = 2.*F0
C
 1000 N  = N + 1
C
C     computation of the new values of the integrand
      DO 10 I=N2,NLIM,N1
         IP   = IP + 1
         ABS0 = CO(IP)
         ABS1 = .5*((1.-ABS0)*A+(1.+ABS0)*B)
         ABS2 = .5*((1.+ABS0)*A+(1.-ABS0)*B)
         C(I) = F(ABS1) + F(ABS2)
 10   CONTINUE
C
C     weighted sum of old and new values
      IP  = IP+1
      VAL = F0 *CO(IP)
      DO 20 I=N2,NLIM,N2
         IP  = IP + 1
         VAL = VAL + C(I) *CO(IP)
 20   CONTINUE
C
      N1 = N2
      N2 = N2 /2
C
C     error estimation
      ERR = ABS(VAL1-VAL)
      IF (VAL.NE.0.) ERR = ERR /ABS(VAL)
      IF (ERR.GT.EPS) THEN
         IF(N.EQ.6) GOTO 100
         VAL1 = VAL
         GOTO 1000
      ENDIF
C
 100  INTEG = VAL * .5 *(B-A)
C
      RETURN
      END
```