Algorithm 413

ENTCAF and ENTCRE: Evaluation of Normalized Taylor Coefficients of an Analytic Function [C5]

J.N. Lyness,* Argonne National Laboratory, Argonne, IL 60439, and G. Sande,† Department of Statistics, The University of Chicago, Chicago, IL 60637 (Recd. 17 June 1968, 12 Feb 1970, and 20 July 1970)

Key Words and Phrases: Taylor coefficients, Taylor series, Cauchy integral, numerical integration, numerical differentiation, interpolation, complex variable, complex arithmetic, fast Fourier transform

CR Categories: 5.12, 5.13, 5.16

Description

Introduction. Two subroutines, *ENTCAF* and *ENTCRE*, coded in ANSI FORTRAN are described here. *ENTCAF* may be used to calculate approximations $r^s a_s^{(m)}$ to a set of normalized Taylor coefficients

$$r^s a_s = r^s f^{(s)}(\zeta)/s! \quad s = 0,1,2,\ldots. \tag{1.1}$$

The values of r and ζ, a complex number, are provided by the user together with a function subprogram that represents $f(z)$ as a complex-valued function of a complex variable. The user also provides a value of ϵ_{req}, the required absolute accuracy. The routine returns an accuracy estimate ϵ_{est} together with approximations $r^s a_s^{(m)}$ and a number m. These are supposed to satisfy

$$\begin{aligned} &| r^s a_s^{(m)} - r^s a_s | < \epsilon_{est} \quad s = 0,1,2,\ldots, m-1, \\ &| r^s a_s | < \epsilon_{est} \qquad\qquad\quad s = m, m+1, \ldots. \end{aligned} \tag{1.2}$$

A result status indicator *NCODE* is output. If $\epsilon_{est} > \epsilon_{req}$ this gives a brief indication of why the required accuracy was not achieved.

ENTCRE carries out the same task as *ENTCAF* in the case that ζ is real and also that $f(z)$ is real when z is real. In this special and common case, *ENTCRE* is about twice as economic as *ENTCAF*.

Outline of method. The Taylor coefficients a_s occur in the Taylor series

$$f(z) = \sum_{s=0}^{\infty} a_s (z-\zeta)^s, \quad | z - \zeta | < R_c, \tag{2.1}$$

where R_c is the radius of convergence of the Taylor series. Cauchy's theorem provides a set of integral representations. One of these is

$$r^s a_s = \frac{r^s}{2\pi i} \int_{C_r} \frac{f(z)}{(z-\zeta)^{s+1}} dz, \quad r < R_c, \tag{2.2}$$

where C_r is the circle $| z - \zeta | = r$. The approximation $r^s a_s^{(m)}$ is obtained by replacing the integral in (2.2) by an approximation based on an m-point trapezoidal rule approximation. Specifically,

$$r^s a_s \simeq r^s a_s^{(m)} = m^{-1} \sum_{j=0}^{m-1} \exp(-2\pi i j s/m) f(\zeta + r \exp(2\pi i j/m)), \quad s = 0,1,\ldots, m-1. \tag{2.3}$$

* Work performed under the auspices of the U.S. Atomic Energy Commission. †This research was supported in part by National Science Foundation research grant number NSF-GP-16071 from the Division of Mathematical and Physical Sciences.

The calculation is in two parts. The first part (stages 1, 2, and 3) is iterative in nature. Using (2.3) the approximations $a_0^{(m)}$ with $m = 1,2,4,8, \cdots$ are calculated. The function values are retained. The convergence criterion is based on the circumstance that the true value

$$a_0 = f(\zeta) \tag{2.4}$$

of one of the approximations $a_0^{(m)}$ may be determined by a single function evaluation. A rather involved convergence criterion based on the orderly approach of the sequence $a_0^{(m)}$, $m = 1,2,4, \ldots$, to its limiting value a_0 is used. This is described in some detail by Lyness [8].

When the convergence of $a_0^{(m)}$ to a_0 has been achieved the routine carries out the second part (stage 4). This consists of evaluating $r^s a_s^{(m)}$ from (2.3) for $s = 0,1, \cdots, m-1$ using the function values calculated and retained during the first part. A fast Fourier transform technique is used for this calculation. This is particularly appropriate since m is a power of two. The derivation and implementation of this technique is described in Gentleman and Sande [5, pp. 566–7]. The specialized version used in *ENTCRE* is described in Sande [9].

Restrictions: theoretical. There are two restrictions of a theoretical nature.

1. The value of r must be less than the radius of convergence, R_c, of the Taylor series. So long as this condition is satisfied, it can be shown (see [5] and [8]) that

$$\begin{aligned} &| r^s a_s | < K\rho^s, \\ &| r^s a_s^{(m)} - r^s a_s | < K\rho^{m+s}/(1-\rho^m), \end{aligned} \tag{3.1}$$

where ρ is any number greater than r/R_c and K depends on ρ. Thus the approximations approach their limiting values and there are only a finite number of normalized Taylor coefficients whose magnitude exceeds ϵ_{req}. If this restriction is violated, that is, a value of $r \geq R_c$ is chosen, then in general the sequence $r^s a_s^{(m)}$ converges, but not to $r^s a_s$. Instead it converges to the integral on the right in (2.2), but (2.2) is not generally vailid if $r \geq R_c$. Thus the routine itself fails to converge since $a_0^{(m)}$ does not approach $f(\zeta)$ in the limit of increasing m.

2. The function $f(z)$ must not be an odd function of $(z - \zeta)$. While the convergence criterion based on (2.4) has much to recommend it, it does have one serious drawback. If it happens (as it does in the case $f(z) = sin\ (z)$; $\zeta = 0$) that

$$f(z - \zeta) = -f(\zeta - z), \tag{3.2}$$

then every approximation $a_0^{(m)}$ is zero, as is the true value a_0. The routine then finds that it converges immediately. In this case the problem should be reformulated. One defines $g(z) = f(z)/(z - \zeta)$ or $g(z) = (z - \zeta)f(z)$. The Taylor coefficients A_s of $g(z) = \zeta$ are then calculated using *ENTCAF*. A_s is the same as a_{s+1} or a_{s-1} as the case may be.

Restrictions: practical. There are two principal practical restrictions. These arise because (1) the computer uses finite length floating-point arithmetic; (2) execution cannot be allowed to continue indefinitely; at some stage it has to terminate whether or not the calculation is complete.

An output status parameter *NCODE* indicates to the user whether the results have been significantly affected by either of these restrictions.

1. *Roundoff error.* The routine requires as an input parameter the machine accuracy parameter ϵ_M. The approximations $r^s a_s^{(m)}$ given by (2.3) are of such a form that an estimate of the roundoff error level is

$$\epsilon_{r.o}^{(m)} = \epsilon_M \max_{j=0,\ \ldots\ ,m-1} | f(\zeta + r \exp(2\pi i j/m)) |. \tag{3.3}$$

If, at any stage it appears that

$$\epsilon_{req} < 10\ \epsilon_{r.o}^{(m)}, \tag{3.4}$$

the routine internally replaces ϵ_{req} by $10\ \epsilon_{r.o}^{(m)}$ and either terminates (input *NCODE* negative) or continues with the calculation (input *NCODE* nonnegative).

2. *Physical upper limit.* This is defined by an input parameter *NMAX*. Iterations in the first part to calculate $a_0^{(m)}$, $m = 1,2,4,8, \ldots$, with $m < NMAX$ are possible. If convergence has not been achieved by this stage, the calculation is completed.

The output status parameter *NCODE* is +1 if all went well. In general $NCODE = 0$ if the calculation was terminated; is positive if it converged and negative if it did not converge; has magnitude 1 if roundoff error was not observed; and has magnitude 2 if roundoff error was observed.

If $NCODE \neq 0$, the returned value ϵ_{est} corresponds to the estimated accuracy of *TCOF(J)* whether or not convergence or roundoff error occurred. If $NCODE = 0$, the quantity $10\ \epsilon_{r.o}^{(m)}$ is returned in place of ϵ_{est}.

Comments. The algorithms described here deliver approximations to a set of normalized Taylor coefficients $r^s a_s$. It is natural to ask why this choice of output was made, rather than perhaps a set of Taylor coefficients a_s or a set of derivatives $f^{(s)}(\zeta)$. The most immediate reason is that the algorithm naturally provides a set or normalized Taylor coefficients to a uniform absolute accuracy. The user specifies r and ϵ_{req} only. If, for example, one is interested in a set of derivatives, the specification of the accuracy requirements becomes very much more complicated. However, if one looks ahead to the use to which the Taylor coefficients are to be put, one finds in many cases that uniform accuracy in normalized Taylor coefficients corresponds to the sort of accuracy requirement which is most convenient.

As an illustration we consider a very trivial problem. We wish to represent $f''(x)$ as a polynomial in the interval $(-l,l)$ to an accuracy E. Clearly

$$f''(x) = \sum_{s=2}^{\infty} s(s-1)a_s x^{s-2} = \frac{1}{r^2} \sum_{s=2}^{\infty} s(s-1)a_s r^s \left(\frac{x}{r}\right)^{s-2}. \tag{4.1}$$

A very crude approach might be to take $r = l$ and $\epsilon = r^2E/6$. In this case the error in the sth term is less than $s(s-1)E(x/l)^{s-2}/6$. One cannot be assured that for $x \simeq l$ these errors may not cooperate in such a way as to lose the required accuracy. However, if r is chosen to be greater than l and $\epsilon = r^2(1 - l/r)^3E/2$ then it follows at once that if the allowed error in a_s, r^s is less thah ϵ, the error in $f''(x)$ is less than E. These two approaches represent extremes. Neither take into account that the sequence $a_s r^s$ itself approaches zero and for high values of s it is unnecessary to bound the error in omitting such a term by ϵ. A more complicated formula based on (3.1) is derived by Lyness and Delves [5], eq. (2.9). But the underlying feature of any of these approaches to approximating (4.1) is that a uniform absolute accuracy for $a_s r^s$, $s = 0,1,2, \ldots$, is very convenient for this problem. If the algorithm instead calculated $f^{(s)}(0)$ to a specified relative accuracy, the determination of the accuracy to use in this problem would be very much more involved.

Possible modifications. The general approach to a numerical calculation by means of the numerical evaluation of contour integrals is at present an open field for investigation. The algorithms described here may be used in several problems known to the authors. These are: (a) determination of zeros of analytic function [7, 1, and 5]; (b) numerical differentiation [7, 6]; (c) numerical quadrature [8].

In particular applications, modifications of *ENTCAF* or *ENTCRE* can lead to more efficient calculations. Possible modifications include: (a) Provision for calculation of only some of the Taylor coefficients, for example, s even or $s \leq 12$; (b) Provision for a "subsequent return option" which allows the same calculation to be taken up at a later stage if it is found subsequently that higher accuracy is required; (c) Provision for an "early exit." Used in conjunction with (b) this would enable the program to consider intermediate results to determine whether to continue with the current values of r and ϵ, before a high investment of computer time has been made.

In fact, *ENTCRE* is a special modification of *ENTCAF* designed for a particular application, ζ real, $f(x)$ real. The output status parameter *NCODE* is of particular use in these applications since it allows appropriate remedial action to be taken under program control.

Algorithms which include modifications (b) and (c) above have been used by the first author. However, these involve complicated logic and are strongly connected with the particular application. The algorithms listed here may be modified by the user in particular applications for any large scale use. However, in pilot runs or small scale calculations they are adequate as they stand.

Comparisons and examples. In [6] and [8], several numerical examples are given, and comparisons with other methods are made. So far as the determination of zeros of an analytic function is concerned, the method described in [6] has some advantages in a global situation, but should not be used locally. For numerical quadrature, the method described [8] is definitely superior to standard methods if there is a nearby pole or singularity of a special type. In these cases a proper evaluation depends on the details of the problem under consideration.

It is in problems involving numerical differentiation that the method on which these algorithms are based show up to great advantage. This is simply because, once the use of complex function values is allowed, the numerical instability associated with numerical differentiation may be avoided.

In [6], a different but related method for numerical differentiation is described. The remarks about the roundoff error given there apply to these routines also. There as an example, the calculation of $f^{(5)}(0)$ was considered for

$$f(x) = e^x/(\sin^3(x) + \cos^3(x)).$$

The actual value of this derivative is an integer, namely

$$f^{(5)}(0) = -164.$$

In order to provide some sort of comparison, a special algorithm for numerical differentiation based on polynomial interpolation was written using only function values at real abscissas. A set of several dozen numerical experiments were carried out on a machine for which $\epsilon_M = 3 \times 10^{-11}$. The closest result was in error by 10^{-2}; the worst result had the wrong sign.

ENTCRE was then used for the same problem in an attempt to obtain seven-digit accuracy, i.e. an absolute accuracy of $E = 10^{-4}$. A sequence of values of r was used, with in each case $\epsilon_{req} = r^5 \times 10^{-4}/5!$ and input parameter $NCODE = -1$ to secure immediate termination if roundoff error prevented a sufficiently accurate result from being attained. With $r = 0.1$ and $r = 0.2$, execution terminated using in each case one complex and three real function values. With $r = 0.4$, the result

$$f^{(5)}(0) = -164.00000013$$

was obtained at a cost of 15 complex and three real function values ($m = 32$); the accuracy estimate given by the algorithm was

$$E_{est} = \epsilon_{est}\, 5!/r^5 = 6 \times 10^{-6}.$$

Incidentally, an absolute accuracy of less than 10^{-4} was estimated and a better accuracy obtained for $r = 0.3, 0.4, 0.5, 0.6, 0.7$ with $m = 32, 32, 64, 64, 128$, respectively. For $r = 0.8$ and $r = 0.9$ the routine failed to converge with $m = 128$ giving absurd results and estimates. These latter values of r are greater than the radius of convergence $R_c = \pi/4$.

The role played by the output status parameter *NCODE* is illustrated in this example. With $r = 0.1$ and $r = 0.2$, the value of *NCODE* indicated immediately that the results were not to be taken seriously because of roundoff error. With $r = 0.8$ and $r = 0.9$, the value of *NCODE* indicated that the results were not to be taken seriously because of lack of convergence. Thus the calculation could have been carried out completely under program control, with a driver program finding for itself an appropriate value of r. An efficient program for this application would require modifications (a), (b), and (c) of the previous section.

The testing of the algorithm included the calculation of high-order derivatives. In general, it frequently happens that even when analytic closed expressions are known for such derivatives, these expressions are difficult to evaluate because of excessive subtraction error. Cases in point include the functions e^x/x and $\sin(x)/x$. Programs were written to evaluate the first 80 derivatives of these functions at $x = 5, 10, 20, 40$, and 80. It turned out that meaningful results could be obtained. For example, for $f(x) = e^x/x$, using $r = 32$ and $\epsilon_{req} = 10^{-10}$, *ENTCRE* gives

$$f^{(25)}(40) = 3.6560469 \times 10^{16}$$

with an estimated relative accuracy of 2.5×10^{-9}. These results were compared with those obtained using an algorithm due to Gautschi and Klein [2, 3]. In all cases examined corresponding results agreed to within the calculated error estimate.

References

1. Delves, L.M., and Lyness, J.N. A numerical method for locating the zeros of an analytic function. *Math. Comput. 21* (1967), 543–560.
2. Gautschi, W., and Klein, B.J. Recursive computation of certain derivatives—A study of error propagation, *Comm. ACM 13* (Jan. 1970), 7–9.
3. Gautschi, W., and Klein, B.J. R282 Derivatives of e^x/x, $\cos(x)$ and $\sin(x)/x$. *Comm. ACM 13* (Jan. 1970), 53–54.
4. Gentleman, W.M., and Sande, G. Fast Fourier transforms—for fun and profit. Proc. AFIPS 1966 FJCC, Vol. 29, Spartan Books, New York, pp. 563–578.
5. Lyness, J.N., and Delves, L.M. On numerical contour integration round a closed contour. *Math. Comput. 21* (1967), 561–577.
6. Lyness, J.N., and Moler, C.B., Numerical differentiation of analytic functions. *SIAM J. Numer. Anal. 4* (1967), 202–210.
7. Lyness, J.N. Numerical algorithms based on the theory of complex variables. Proc. ACM 22nd Nat. Conf. 1967, pp. 125–134.
8. Lyness, J.N. Quadrature methods based on complex function values. *Math. Comput. 23* (1969), 601–620.
9. Sande, G., Fast Fourier transform—A globaly complex algorithm with locally real implementation. Proc. 4th Ann. Princeton Symp. on Information Sciences and Systems, 1970, pp. 136–142.

Algorithm

```
      SUBROUTINE ENTCRE ( CFUN, ZETA, RCIRC, EPREQ, EPMACH, NMAX, NCODE,
     . EPEST, NTCOF, TCOF, WORK, NTAB, SINTAB )
C
C ** EVALUATION OF NORMALIZED TAYLOR COEFFICIENTS **
C **          OF A REAL ANALYTIC FUNCTION           **
C
C      **  GENERAL PURPOSE  **
C THIS ROUTINE EVALUATES A SET OF NORMALIZED TAYLOR COEFFICIENTS
C TCOF(J+1) = (RCIRC**J) * (J-TH DERIVATIVE OF CFUN(Z) AT Z=ZETA)
C DIVIDED BY FACTORIAL(J) ... J = 0,1,2,3...NMAX-1.
C TO A UNIFORM ABSOLUTE ACCURACY **EPEST** USING FUNCTION
C VALUES OF CFUN(Z) AT POINTS IN THE COMPLEX PLANE LYING ON
C THE CIRCLE OF RADIUS **RCIRC** WITH CENTER AT Z = ZETA.
C THIS ROUTINE IS A SPECIAL VERSION OF ENTCAF FOR USE WHEN
C ZETA IS REAL AND ALSO CFUN(Z) IS REAL WHEN Z IS REAL.
C
C      ** THEORETICAL RESTRICTIONS **
C RCIRC MUST BE SMALLER THAN THE RADIUS OF CONVERGENCE OF
C THE TAYLOR SERIES.  THE PROBLEM HAS TO BE REFORMULATED
C SHOULD CFUN(Z) HAPPEN TO BE AN ODD FUNCTION
C OF  (Z - ZETA)  . THAT IS IF THE RELATION
C ** -CFUN(-(Z-ZETA))=CFUN(Z-ZETA) **  IS AN IDENTITY.
C
C      ** REQUIREMENTS FOR CALLING PROGRAM **
C CALLING PROGRAM MUST CONTAIN CONTROL STATEMENTS DESCRIBED
C NOTES (3) AND (4) BELOW.  IT MUST ALSO ASSIGN VALUES TO
C INPUT PARAMETERS.  THE ROUTINE REQUIRES TWO SUBPROGRAMS.
C HFCOF  (LISTED AFTER ENTCRE) AND CFUN (SEE NOTE (4) BELOW).
C
C      **INPUT PARAMETERS**
C (1) CFUN   NAME OF COMPLEX FUNCTION SUBPROGRAM.
C (2) ZETA   REAL POINT ABOUT WHICH TAYLOR EXPANSION IS REQUIRED
C (3) RCIRC  RADIUS (REAL)
C (4) EPREQ  THE ABSOLUTE ACCURACY (REAL) TO WHICH THE
C            NORMALIZED TAYLOR COEFFICENTS, TCOF(J), ARE REQUIRED
C (5) EPMACH THE MACHINE ACCURACY PARAMETER (REAL)
C            (OR AN UPPER BOUND ON THE RELATIVE ACCURACY OF
C            QUANTITES LIKELY TO BE ENCOUNTERED).
C (6) NMAX   PHYSICAL UPPER LIMIT ON THE SIZE AND LENGTH
C            OF THE CALCULATION.  THE MAXIMUM NUMBER OF
C            COEFFICIENTS CALCULATED WILL BE THAT POWER OF TWO
C            LESS THAN OR EQUAL TO NMAX.  NMAX IS ASSUMED TO
C            BE AT LEAST 8.  (SEE NOTE(3) BELOW).
C (7) NCODE  .GE.0  THE ROUTINE WILL DO AS WELL AS IT CAN.
C            .LT.0  THE ROUTINE WILL ABORT AT AN EARLY STAGE
C            IF THE REQUIRED ACCURACY CANNOT BE ATTAINED
C            BECAUSE OF ROUND OFF ERROR.
```

```
C (12) NTAB   IN NORMAL RUNNING, NTAB SHOULD BE SET TO ZERO
C             BEFORE THE FIRST CALL TO ENTCRE, BUT LEFT ALONE
C             AFTER THAT.  (FOR MORE SOPHISTICATED USE, SEE
C             OUTPUT PARAMETERS (12) AND (13) AND NOTE(2) BELOW)
C
C     ** OUTPUT PARAMETERS **
C  (1),(2),(3),(4),(5),(6) IDENTICAL WITH INPUT VALUES.
C (7)   NCODE  RESULT STATUS INDICATOR.
C             TAKES ONE OF FIVE VALUES AS FOLLOWS,
C             = +1. CONVERGED NORMALLY.
C             = -1. DID NOT CONVERGE. NO ROUND OFF ERROR
C             TROUBLE.
C             = +2. CONVERGED, BUT WITH A HIGHER TOLERANCE
C             SET BY THE ROUND OFF LEVEL. (EPEST.GT.EPREQ)
C             = -2. DID NOT CONVERGE IN SPITE OF HIGHER
C             TOLERANCE SET BY ROUND OFF LEVEL.
C             =  0. RUN WAS ABORTED BECAUSE EPREQ IS
C             UNATTAINABLE DUE TO ROUND OFF LEVEL AND INPUT
C             NCODE IS NEGATIVE.
C (8)   EPEST  ESTIMATE OF ACTUAL UNIFORM ABSOLUTE ACCURACY
C             IN ALL TCOF.  EXCEPT, IF NCODE.EQ.0  ESTIMATE
C             OF ROUND OFF LEVEL.
C (9)   NTCOF  NUMBER OF NONTRIVIAL VALUES OF TCOF ACTUALLY
C             CALCULATED.  THEY ARE BASED ON NTCOF/2+2 CALLS
C             OF CFUN (THREE CALLS WERE FOR PURELY
C             REAL ARGUMENT).
C (10)  TCOF   REAL DIMENSION (DIM).  APPROXIMATIONS TO THE
C             NORMALIZED TAYLOR COEFFICIENTS, EXCEPT WHEN
C             OUTPUT NCODE = 0.  (SEE NOTE(3) BELOW)
C (11)  WORK   INTERNAL WORKING AREA OF REAL DIMENSION (DIM)
C             (SEE NOTE(3) BELOW.) CONTENTS IS IDENTICAL WITH
C             THAT OF TCOF.
C (12)  NTAB   NUMBER OF VALUES OF SINTAB AVAILABLE
C             (SEE NOTE (2) BELOW).
C (13)  SINTAB REAL DIMENSION (DIM/4).  (SEE NOTES (2) AND (3)
C             BELOW.) SINTAB(J+1) = SIN(PI*J/2*NTAB)
C             J = 0,1,2,...NTAB-1.
C             (A QUARTER CYCLE) OTHER LOCATIONS ARE EMPTY.
C
C     ** NOTES ON INPUT/OUTPUT PARAMETERS **
C  NOTE(1)**  NCODE IS USED BOTH AS INPUT AND OUTPUT PARAMETER.
C NORMALLY IT RETAINS THE VALUE +1 AND NEED NOT BE RESET
C BETWEEN NORMAL RUNS.
C  NOTE(2)**  THE APPEARANCE OF NTAB AND SINTAB IN THE
C CALLING SEQUENCE ALLOWS THE USER TO MAKE USE OF - OR TO
C PRECOMPUTE - THESE NUMBERS IN ANOTHER PART OF THE PROGRAM
C SHOULD HE SO DESIRE.  NTAB MUST BE A POWER OF TWO OR 0.
C  NOTE(3)**  THE APPEARANCE OF NMAX,TCOF,WORK AND SINTAB IN
C THE CALLING SEQUENCE ALLOWS THE SCOPE OF THE SUBPROGRAM AND
C THE AMOUNT OF STORAGE TO BE ASSIGNED BY THE CALLING
C PROGRAM, WHICH SHOULD CONTAIN A CONTROL STATEMENT TO THE
C FOLLOWING EFFECT -
C  REAL TCOF(DIM), WORK(DIM), SINTAB(DIM/4)
C WHERE DIM IS NORMALLY A POWER OF TWO.  NMAX IS NORMALLY
C EQUAL TO DIM.  BUT MAY BE LESS THAN DIM.
C  NOTE(4)**  CFUN(Z)  IS A USER PROVIDED COMPLEX VALUED
C FUNCTION SUBPROGRAM WITH A COMPLEX VALUED ARGUMENT.  THE
C CALLING PROGRAM MUST CONTAIN CONTROL STATEMENTS AS FOLLOWS -
C  EXTERNAL CFUN      COMPLEX CFUN
C
C     ** BOOKKEEPING PARAMETERS FOR STAGE ONE **
C  NCONV   1  CONVERGENCE ACHIEVED.
C         -1  NO CONVERGENCE ACHIEVED.
C  NROUND  1  NO ROUND OFF TROUBLE OBSERVED.
C          2  ROUND OFF TROUBLE OBSERVED.
C  NABORT  0  UPDATE TOLERANCE AND CONTINUE ON APPEARANCE OF
C               OF ROUND OFF TROUBLE.
C          1  TERMINATE WHEN ROUND OFF TROUBLE OBSERVED.
C  EXACT   THE EXACT VALUE OF TCOF(1) WHICH IS CFUN(ZETA).
C  SAFETY  THIS IS A SAFETY FACTOR BY WHICH THE ROUTINE AVOIDS
C          THE ROUND OFF LEVEL.  IT IS SET TO 10.0 AND
C          APPEARS ONLY IN THE COMBINATION (SAFETY*EPMACH).
C          TO ALTER THIS FACTOR, OR TO REMOVE THE ROUND OFF
C          ERROR GUARD COMPLETELY, THE USER NEED ONLY ADJUST
C          THE INPUT PARAMETER EPMACH APPROPRIATELY.
C
C     ** QUANTITIES CALCULATED IN STAGE THREE(A) **
C  THIS IS THE FIRST PART OF ITERATION NUMBER NTCOF. PRESENTLY
C AVAILABLE ARE -
C SINTAB(J+1) = SIN(PI*J/2*NTAB) , J = 0,1,2,...NTAB-1.
C WE REQUIRE THE SEQUENCE SIN(PI*J/2*(NTCOF/4)),
C J = 1,3,5,...(NTCOF/4-1).
C IF (NTCOF.LE.4*NTAB) THESE NUMBERS ARE ALREADY AVAILABLE IN
C THE SINTAB TABLE SPACED AT AN INTERVAL 2*NSPACE = 8*NTAB/NTCOF.
C OTHERWISE, NTCOF = 8*NTAB AND THE SINTAB TABLE IS UPDATED.
C THIS INVOLVES REARRANGING THE NTAB VALUES AVAILABLE,
C CALCULATING AND STORING NTAB NEW VALUES AND UPDATING
C NTAB TO 2*NTAB.
C
C     ** QUANTITIES CALCULATED IN STAGE THREE(B) **
C  ITERATIONS ARE NUMBERED 8,16,32,...AT THE END OF ITERATION
C NTCOF, THE NTCOF/2 + 1 COMPLEX FUNCTION VALUES AT
C ABSCISSAS REGULARLY SPACED ON UPPER HALF OF CIRCLE ARE
C STORED IN THE TCOF VECTOR AS FOLLOWS.
C  TCOF(J+1)   =   REAL PART OF CFUN(Z(J))  J=0,1,2,...NTCOF/2.
C  TCOF(NTCOF-J+1) = IMAGINARY PART OF CFUN(Z(J))
C                        J=1,2,...(NTCOF/2-1).
C WHERE
C  Z(J) = ZETA + RCIRC*CEXP(2*PI*EYE*J/NTCOF)
C THIS INVOLVES A REARRANGEMENT OF THE NTCOF/4 + 1 FUNCTION
C VALUES AVAILABLE AT THE START OF THE ITERATION AND THE
C CALCULATION OF A FURTHER NTCOF/4 FUNCTION VALUES.  IN
C ADDITION FMAX AND APPROX ARE CALCULATED.  THESE ARE
C  FMAX   MAXIMUM MODULUS OF THE FUNCTION VALUES SO FAR
C         ENCOUNTERED.
C  APPROX AN APPROXIMATION TO TCOF(1)
C         BASED ON THESE FUNCTION VALUES.
C
C     ** QUANTITIES CALCULATED AT STAGE THREE(C) **
C  ERROR1  CURRENT VALUE OF THE ERROR = ABS(APPROX-EXACT).
C  ERROR2,ERROR3,ERROR4  VALUES OF ERROR AT END OF THREE
C          PREVIOUS ITERATIONS.
C  EPMACH  MACHINE ACCURACY PARAMETER. (INPUT PARAMETER)
C  EPREQ   REQUIRED ACCURACY. (INPUT PARAMETER)
C  EPRO    HIGHEST ACCURACY REASONABLY ATTAINABLE IN VIEW OF
C          THE SIZE OF THE FUNCTION VALUES SO FAR ENCOUNTERED.
C          (=10.0*EPMACH*FMAX)
C  EPCOF   CURRENTLY REQUIRED ACCURACY (=AMAX1(EPREQ,EPRO)).
C  EPEST   ESTIMATE OF CURRENT ACCURACY. (THE MAXIMUM OF EPRO
C          AND A FUNCTION OF ERRORS 1,2,3 AND 4) (OUTPUT PARAMETER)
C
C ** CONVERGENCE AND TERMINATION CHECKS IN STAGE THREE(C) **
C (1)  USES FMAX TO RAISE EPCOF ABOVE ROUND OFF LEVEL.  IF
C  THIS IS NECESSARY AND THE INPUT VALUE OF NCODE IS NEGATIVE,
C  IT TERMINATES SETTING NCODE = 0.
C (2)  USES APPROX TO EVALUATE CONVERGENCE OF TCOF(1) TOWARDS
C  EXACT.  IT MAY ASSIGN CONVERGENCE AND GO TO STAGE FOUR(A)
C  SETTING NCODE = +1 OR +2.
C (3)  USES NMAX TO CHECK PHYSICAL LIMIT.  IF THIS HAS BEEN
C  REACHED, IT GOES TO STAGE FOUR(A) SETTING NCODE = -1 OR -2.
C (4)  OTHERWISE CONTINUES NEXT ITERATION BY GOING TO STAGE THREE
C
C  ** CALCULATION OF FIRST NTCOF TAYLOR COEFFICIENTS IN
C  **   STAGE FOUR(A)
C A VERSION OF THE FAST FOURIER TRANSFORM USING A WORK ARRAY
C IS USED.  THE ARRAY **WORK** IS USED ONLY DURING THIS STAGE.
C THE WORK ARRAY ALLOWS THE PERMUTING OF INDICES ASSOCIATED
C WITH IN-PLACE FFTS TO BE SUPPRESSED.  THE FFT CALCULATES
C THE NECCESSARY SUMMATIONS EXCEPT FOR DIVIDING BY NTCOF.
C
C ** SETTING OF REMAINING TAYLOR COEFFICIENTS IN STAGE FOUR(B) **
C  THE CONVERGENCE CRITERION ALLOWS US TO INFER THAT THE
C NORMALIZED TAYLOR COEFFICIENTS OF ORDER GREATER THAN NTCOF
C ARE ZERO TO ACCURACY  EPEST. THEY ARE EVALUATED AS BEING
C EXACTLY ZERO.
      COMPLEX CFUN
      REAL ZETA,RCIRC,EPREQ,EPMACH,EPEST
      INTEGER NMAX,NCODE,NTCOF,NTAB
      REAL TCOF (1), WORK (1), SINTAB (1)
      INTEGER NABORT,NCONV,NDISP,NDOLIM,NPREV,NROUND,NSPACE
      INTEGER J,JCONJ,JCOS,JFROM,JRCONJ,JREFL,JSIN,JTO
      REAL APPROX,COSDIF,EPCOF,EPMIN,EPRO,EP32,EP42
      REAL ERROR1,ERROR2,ERROR3,ERROR4,EXACT,FMAX,FVALIM
      REAL FVALRE,RCOS,RSIN,SAFETY,SCALE,SUPPER,TWOPI
      COMPLEX FVAL,ZVAL
      COMPLEX CMPLX
C ***   STAGE ONE   ***
C ---------------------
C INITIALISE BOOKKEEPING PARAMETERS AND EXACT VALUE OF TCOF(1).
      NROUND = 1
      NABORT = 0
      IF (NCODE.LT.0) NABORT = 1
      EPCOF = EPREQ
      SAFETY = 10.0
      ZVAL = CMPLX(ZETA,0.0)
      FVAL = CFUN(ZVAL)
      FVALRE = REAL(FVAL)
      EXACT = FVALRE
C ***   STAGE TWO   ***
C ---------------------
C FIRST THREE ITERATIONS ( THOSE WITH NTCOF = 1,2,4 ).
      ZVAL = CMPLX(ZETA+RCIRC,0.0)
      FVAL = CFUN(ZVAL)
      FVALRE = REAL(FVAL)
      APPROX = FVALRE
      FMAX = ABS(FVALRE)
      TCOF(1) = FVALRE
      ERROR3 = ABS(APPROX-EXACT)
      ZVAL = CMPLX(ZETA-RCIRC,0.0)
      FVAL = CFUN(ZVAL)
      FVALRE = REAL(FVAL)
      APPROX = 0.5*(APPROX+FVALRE)
      FMAX = AMAX1(FMAX,ABS(FVALRE))
      TCOF(3) = FVALRE
      ERROR2 = ABS(APPROX-EXACT)
      ZVAL = CMPLX(ZETA,RCIRC)
      FVAL = CFUN(ZVAL)
      FVALRE = REAL(FVAL)
      FVALIM = AIMAG(FVAL)
      APPROX = 0.5*(APPROX+FVALRE)
      FMAX = AMAX1(FMAX,CABS(FVAL))
      TCOF(2) = FVALRE
      TCOF(4) = FVALIM
      ERROR1 = ABS(APPROX-EXACT)
      NTCOF = 4
      EPRO = FMAX*SAFETY*EPMACH
      IF (EPRO.LT.EPCOF) GO TO 300
        EPCOF = EPRO
        NROUND = 2
        IF (NABORT.EQ.0) GO TO 300
        NCODE = 0
        EPEST = EPRO
        GO TO 470
C ***   STAGE THREE   ***
C -----------------------
C COMMENCE ITERATION NUMBER NTCOF.
  300 CONTINUE
      NPREV = NTCOF
      NTCOF = 2*NTCOF
C ***   STAGE THREE(A)   ***
C --------------------------
C UPDATE SINTAB TABLE IF NECESSARY.
      IF (4*NTAB.GE.NTCOF) GO TO 340
      IF (NTAB.GE.2) GO TO 310
        SINTAB(1) = 0.0
        SINTAB(2) = SQRT(0.5)
        NTAB = 2
        GO TO 340
  310 CONTINUE
      NDOLIM = NTAB-1
      DO 320 J = 1,NDOLIM
         JFROM = NTAB-J
         JTO = 2*JFROM
        SINTAB(JTO+1) = SINTAB(JFROM+1)
  320 CONTINUE
      NTAB = 2*NTAB
      TWOPI = 8.0*ATAN(1.0)
      COSDIF = COS(TWOPI/FLOAT(4*NTAB))
      NDOLIM = NTAB-3
      DO 330 J = 1,NDOLIM,2
        SINTAB(J+1) = (0.5*SINTAB(J)+0.5*SINTAB(J+2))/COSDIF
  330 CONTINUE
      SINTAB(NTAB) = COSDIF
  340 CONTINUE
C ***   STAGE THREE(B)   ***
C --------------------------
C UPDATE LIST OF FUNCTION VALUES IN TCOF,
C CALCULATE FMAX AND APPROX.
      NDOLIM = NPREV-1
      DO 350 J = 1,NDOLIM
        JFROM = NPREV-J
        JTO = 2*JFROM
        TCOF(JTO+1) = TCOF(JFROM+1)
  350 CONTINUE
      SUPPER = 0.0
      NDOLIM = (NPREV/2)-1
      NSPACE = (4*NTAB)/NTCOF
      DO 360 J = 1,NDOLIM,2
        JSIN = J*NSPACE
        JCOS = NTAB-JSIN
        RSIN = RCIRC*SINTAB(JSIN+1)
        RCOS = RCIRC*SINTAB(JCOS+1)
        JCONJ = NTCOF-J
        ZVAL = CMPLX(ZETA+RCOS,RSIN)
        FVAL = CFUN(ZVAL)
        FVALRE = REAL(FVAL)
```

```
      FVALIM = AIMAG(FVAL)
      SUPPER = SUPPER+FVALRE
      FMAX = AMAX1(FMAX,CABS(FVAL))
      TCOF(J+1) = FVALRE
      TCOF(JCONJ+1) = FVALIM
      JREFL = NPREV-J
      JRCONJ = NTCOF-JREFL
      ZVAL = CMPLX(ZETA-RCOS,RSIN)
      FVAL = CFUN(ZVAL)
      FVALRE = REAL(FVAL)
      FVALIM = AIMAG(FVAL)
      SUPPER = SUPPER+FVALRE
      FMAX = AMAX1(FMAX,CABS(FVAL))
      TCOF(JREFL+1) = FVALRE
      TCOF(JRCONJ+1) = FVALIM
  360 CONTINUE
    APPROX = 0.5*APPROX+SUPPER/FLOAT(NPREV)
C ***    STAGE THREE(C)    ***
C --------------------------
C CONVERGENCE AND TERMINATION CHECK.
    ERROR4 = ERROR3
    ERROR3 = ERROR2
    ERROR2 = ERROR1
    ERROR1 = ABS(APPROX-EXACT)
    EPRO = FMAX*SAFETY*EPMACH
    IF (EPRO.LT.EPCOF) GO TO 370
      EPCOF = EPRO
      NROUND = 2
      IF (NABORT.EQ.0) GO TO 370
      NCODE = 0
      EPEST = EPRO
      GO TO 470
  370 CONTINUE
    ERROR4 = AMAX1(ERROR4,EPRO)
    ERROR3 = AMAX1(ERROR3,EPRO)
    EP42 = ERROR2*((ERROR2/ERROR4)**(4.0/3.0))
    EP32 = ERROR2*((ERROR2/ERROR3)**2)
    EPMIN = AMIN1(ERROR2,EP32,EP42)
    EPEST = AMAX1(ERROR1,EPMIN,EPRO)
    IF (EPEST.GT.EPCOF) GO TO 380
      NCONV = 1
      GO TO 400
  380 CONTINUE
    IF (2*NTCOF.LE.NMAX) GO TO 300
      NCONV = -1
C ***    STAGE FOUR(A)    ***
C --------------------------
C CALCULATION OF FIRST NTCOF TAYLOR COEFFICIENTS USING F.F.T.
  400 CONTINUE
    NCODE = NCONV*NROUND
    NDISP = NTCOF
  410 CONTINUE
    NDISP = NDISP/2
    CALL HFCOF (NTCOF,NDISP,TCOF,WORK,NTAB,SINTAB)
    IF (NDISP.GT.1) GO TO 430
    DO 420 J = 1,NTCOF
      TCOF(J) = WORK(J)
  420 CONTINUE
    GO TO 440
  430 CONTINUE
    NDISP = NDISP/2
    CALL HFCOF (NTCOF,NDISP,WORK,TCOF,NTAB,SINTAB)
    IF (NDISP.GT.1) GO TO 410
  440 CONTINUE
    SCALE = 1.0/FLOAT(NTCOF)
    DO 450 J = 1,NTCOF
      TCOF(J) = TCOF(J)*SCALE
      WORK(J) = TCOF(J)
  450 CONTINUE
C ***    STAGE FOUR(B)    ***
C --------------------------
C SETTING OF REMAINING TAYLOR COEFFICIENTS.
    IF (NTCOF.GE.NMAX) GO TO 470
    NDOLIM = NTCOF+1
    DO 460 J = NDOLIM,NMAX
      TCOF(J) = 0.0
      WORK(J) = 0.0
  460 CONTINUE
  470 CONTINUE
    RETURN
C END OF ENTCRE
    END
    SUBROUTINE HFCOF ( NTCOF, NDISP, TCOF, WORK, NTAB, SINTAB )
C
C ** HERMITIAN FOURIER COEFFICIENTS **
C
C     **  GENERAL PURPOSE  **
C THIS ROUTINE DOES ONE PASS OF A FAST FOURIER TRANSFORM.
C THE INDEXING IS ARRANGED SO THAT THE COEFFICIENTS ARE IN
C ORDER AT THE END OF THE LAST PASS.  THIS INDEXING REQUIRES
C THE USE OF SEPARATE ARRAYS FOR INPUT AND OUTPUT OF THE
C PARTIAL RESULTS.  THIS ROUTINE IS CALLED ONCE FOR EACH PASS.
C
C     **  INPUT PARAMETERS  **
C  (1)  NTCOF     NUMBER OF COEFFICIENTS TO BE PROCESSED.
C  (2)  NDISP     MAXIMUM VALUE OF DISPLACEMENT INDEX.
C  (3)  TCOF      (REAL) INPUT ARRAY.
C  (5)  NTAB      NUMBER OF ENTRIES IN SINTAB.
C  (6)  SINTAB    (REAL) TABLE OF VALUES OF SINE.
C                 SINTAB(J+1)=SIN(PI*J/2*NTAB), J=0,1,2...NTAB-1
C
```

```
C
C     **  OUTPUT PARAMETERS  **
C  (4)  WORK      (REAL) OUTPUT ARRAY.
C
C     **  INDEXING OF ARRAYS  **
C THE TWO POINT FOURIER TRANSFORM IS APPLIED TO THE POINTS
C OF TCOF WITH INDICES
C     JDISP*NPREV+JREPL  AND  JDISP*NPREV+JREPL+NHALF
C THE RESULTS ARE MODIFIED BY THE APPROPRIATE TWIDDLE FACTOR
C AND STORED IN WORK WITH INDICES
C     JDISP*NNEXT+JREPL  AND  JDISP*NNEXT+JREPL+NPREV
C  WHERE
C     NDISP       PRODUCT OF REMAINING FACTORS.
C     NPREV       PRODUCT OF PREVIOUS FACTORS.
C     NNEXT       PRODUCT OF PREVIOUS AND CURRENT FACTORS.
C     NHALF       PRODUCT OF PREVIOUS AND REMAINING FACTORS.
C     JREPL       REPLICATION INDEX = 1,2,...NPREV.
C     JDISP       HERMITIAN SYMMETRY IN THIS INDEX RESULTS IN
C                 THREE CASES.
C                 1)  INITIAL POINT - JDISP=0. INPUT POINTS
C                 ARE PURELY REAL AND OUTPUT POINTS ARE
C                 PURELY REAL.
C                 2)  MIDDLE POINT - JDISP=NDISP/2 - NOT
C                 ALWAYS PRESENT.  INPUT POINTS ARE COMPLEX AND
C                 OUTPUT POINTS ARE PURELY REAL.
C                 3)  INTERMEDIATE POINTS - JDISP=1,2,..(NDISP/2-1)
C                 - NOT ALWAYS PRESENT.  INPUT POINTS ARE
C                 COMPLEX AND OUTPUT POINTS ARE COMPLEX.
C
C ON INPUT, THE HERMITIAN SYMMETRY IS IN A BLOCK OF LENGTH
C 2*NDISP, I.E. THE POINT CONJUGATE TO JDISP IS 2*NDISP-JDISP.
C ON OUTPUT, THE HERMITIAN SYMMETRY IS IN A BLOCK OF LENGTH
C NDISP, I.E. THE POINT CONJUGATE TO JDISP IS NDISP-JDISP.
C A HERMITIAN SYMMETRIC BLOCK HAS REAL PARTS AT THE FRONT
C IMAGINARY PARTS (WHEN THEY EXIST) AT THE CONJUGATE
C POSITIONS AT THE BACK.
C
C THE TWIDDLE FACTOR CEXP(-PI*FYE*J/NDISP), J=1,2,..(NDISP/2-1)
C IS OBTAINED AS SEPARATE REAL AND IMAGINARY PARTS FROM
C THE SINTAB TABLE.  THE IMAGINARY PART SIN(PI*J/NDISP) IS
C FOUND AT A SPACING OF NSPACE=2*NTAB/NDISP IN SINTAB.
C THE REAL PART IS FOUND AT A CONJUGATE POSITION IN THE TABLE.
C
    INTEGER NTCOF,NDISP,NTAB
    REAL TCOF (1), WORK (1), SINTAB (1)
    REAL CS,IS,IU,I0,I1,RS,RU,R0,R1,SN
    INTEGER JCONJ,JCOS,JDISP,JREPL,JSIN,JT,JTC,JW,JWC,KT0,KT1
    INTEGER KT2,KT3,KW0,KW1,KW2,KW3,NHALF,NMIDL,NNEXT,NPREV,NSPACE
    NHALF = NTCOF/2
    NPREV = NTCOF/(2*NDISP)
    NNEXT = NTCOF/NDISP
    NMIDL = (NDISP-1)/2
    NSPACE = (2*NTAB)/NDISP
C INITIAL POINTS OF BLOCKS.
    DO 100 JREPL = 1,NPREV
      KT0 = JREPL
      KT1 = KT0+NHALF
      KW0 = JREPL
      KW1 = KW0+NPREV
      R0 = TCOF(KT0)
      R1 = TCOF(KT1)
      WORK(KW0) = R0+R1
      WORK(KW1) = R0-R1
  100 CONTINUE
C INTERMEDIATE POINTS OF BLOCKS.
    IF (NMIDL.LT.1) GO TO 400
    DO 300 JDISP = 1,NMIDL
      JCONJ = NDISP-JDISP
      JSIN = JDISP*NSPACE
      JCOS = NTAB-JSIN
      SN = SINTAB(JSIN+1)
      CS = SINTAB(JCOS+1)
      JT = JDISP*NPREV
      JTC = JCONJ*NPREV
      JW = JDISP*NNEXT
      JWC = JCONJ*NNEXT
      DO 200 JREPL = 1,NPREV
        KT0 = JT+JREPL
        KT1 = KT0+NHALF
        KT2 = JTC+JREPL
        KT3 = KT2+NHALF
        KW0 = JW+JREPL
        KW1 = KW0+NPREV
        KW2 = JWC+JREPL
        KW3 = KW2+NPREV
        R0 = TCOF(KT0)
        I0 = TCOF(KT3)
        R1 = TCOF(KT2)
        I1 = -TCOF(KT1)
        RS = R0+R1
        IS = I0+I1
        RU = R0-R1
        IU = I0-I1
        WORK(KW0) = RS
        WORK(KW2) = IS
        WORK(KW1) = RU*CS+IU*SN
        WORK(KW3) = IU*CS-RU*SN
  200   CONTINUE
  300 CONTINUE
  400 CONTINUE
C MIDDLE POINTS OF BLOCKS.
    IF (NDISP.LE.1) GO TO 600
    JT = (NDISP/2)*NPREV
    JW = (NDISP/2)*NNEXT
    DO 500 JREPL = 1,NPREV
      KT0 = JT+JREPL
      KT1 = KT0+NHALF
      KW0 = JW+JREPL
      KW1 = KW0+NPREV
      R0 = TCOF(KT0)
      I0 = TCOF(KT1)
      WORK(KW0) = 2.0*R0
      WORK(KW1) = 2.0*I0
  500 CONTINUE
  600 CONTINUE
    RETURN
C END OF HFCOF
    END
    SUBROUTINE ENTCAF ( CFUN, ZETA, RCIRC, EPREQ, EPMACH, NMAX, NCODE,
   . EPEST, NTCOF, TCOF, WORK, NTAB, EXPTAB )
C
```

```
C
C ** EVALUATION OF NORMALIZED TAYLOR COEFFICIENTS **
C **              OF AN ANALYTIC FUNCTION          **
C
C      **  GENERAL PURPOSE  **
C THIS ROUTINE EVALUATES A SET OF NORMALIZED TAYLOR COEFFICIENTS
C TCOF(J+1) = (RCIRC**J) * (J-TH DERIVATIVE OF CFUN(Z) AT Z=ZETA)
C DIVIDED BY FACTORIAL(J) ... J = 0,1,2,3...NMAX-1.
C TO A UNIFORM ABSOLUTE ACCURACY **EPEST** USING FUNCTION
C VALUES OF CFUN(Z) AT POINTS IN THE COMPLEX PLANE LYING ON
C THE CIRCLE OF RADIUS **RCIRC** WITH CENTER AT Z = ZETA.
C
C      ** THEORETICAL RESTRICTIONS **
C RCIRC MUST BE SMALLER THAN THE RADIUS OF CONVERGENCE OF
C THE TAYLOR SERIES.  THE PROBLEM HAS TO BE REFORMULATED
C SHOULD CFUN(Z) HAPPEN TO BE AN ODD FUNCTION OF (Z - ZETA),
C THAT IS IF THE RELATION **-CFUN(-(Z-ZETA))=CFUN(Z-ZETA)**
C IS AN IDENTITY.
C
C      ** REQUIREMENTS FOR CALLING PROGRAM **
C CALLING PROGRAM MUST CONTAIN CONTROL STATEMENTS DESCRIBED
C IN NOTES (3) AND (4) BELOW.  IT MUST ALSO ASSIGN VALUES TO
C INPUT PARAMETERS.  THE ROUTINE REQUIRES TWO SUBPROGRAMS,
C CFCOF (LISTED AFTER ENTCAF) AND CFUN (SEE NOTE(4) BELOW).
C
C      **INPUT PARAMETERS**
C   (1) CFUN   NAME OF COMPLEX FUNCTION SUBPROGRAM.
C   (2) ZETA   COMPLEX POINT ABOUT WHICH TAYLOR EXPANSION
C              IS REQUIRED.
C   (3) RCIRC  RADIUS (REAL)
C   (4) EPREQ  THE ABSOLUTE ACCURACY (REAL) TO WHICH THE
C              NORMALIZED TAYLOR COEFFICIENTS, TCOF(J), ARE REQUIRED
C   (5) EPMACH THE MACHINE ACCURACY PARAMETER (REAL) (OR AN
C              UPPER BOUND ON THE RELATIVE ACCURACY OF
C              QUANTITIES LIKELY TO BE ENCOUNTERED).
C   (6) NMAX   PHYSICAL UPPER LIMIT ON THE SIZE AND LENGTH OF
C              THE CALCULATION.  THE MAXIMUM NUMBER OF
C              COEFFICIENTS CALCULATED WILL BE THAT POWER OF
C              TWO LESS THAN OR EQUAL TO NMAX.  NMAX IS
C              ASSUMED TO BE AT LEAST 4. (SEE NOTE(3) BELOW.)
C   (7) NCODE  .GE.0 THE ROUTINE WILL DO AS WELL AS IT CAN.
C              .LT.0 THE ROUTINE WILL ABORT AT AN EARLY
C              STAGE IF THE REQUIRED ACCURACY CANNOT BE
C              ATTAINED BECAUSE OF ROUND OFF ERROR.
C  (12) NTAB   IN NORMAL RUNNING, NTAB SHOULD BE SET TO ZERO
C              BEFORE THE FIRST CALL TO ENTCAF, BUT LEFT ALONE
C              AFTER THAT. (FOR MORE SOPHISTICATED USE, SEE
C              OUTPUT PARAMETERS (12) AND (13) AND NOTE(2)
C              BELOW.)
C
C      ** OUTPUT PARAMETERS **
C   (1),(2),(3),(4),(5),(6) IDENTICAL WITH INPUT VALUES.
C   (7) NCODE  RESULT STATUS INDICATOR.  TAKES ONE OF FIVE
C              VALUES AS FOLLOWS.
C              =+1. CONVERGED NORMALLY.
C              =-1. DID NOT CONVERGE. NO ROUND OFF ERROR TROUBLE
C              =+2. CONVERGED, BUT WITH A HIGHER TOLERANCE SET
C              BY THE ROUND OFF LEVEL. (EPEST.GT.EPREQ)
C              =-2. DID NOT CONVERGE IN SPITE OF HIGHER
C              TOLERANCE SET BY ROUND OFF LEVEL.
C              = 0. RUN WAS ABORTED BECAUSE EPREQ IS
C              UNATTAINABLE DUE TO ROUND OFF LEVEL AND INPUT
C              NCODE IS NEGATIVE.
C   (8) EPEST  ESTIMATE OF ACTUAL UNIFORM ABSOLUTE ACCURACY
C              IN ALL TCOF. EXCEPT IF NCODE.EQ.0 ESTIMATE OF
C              ROUND OFF LEVEL.
C   (9) NTCOF  NUMBER OF NONTRIVIAL VALUES OF TCOF ACTUALLY
C              CALCULATED.  THEY ARE BASED ON NTCOF+1 CALLS
C              OF CFUN.
C  (10) TCOF   COMPLEX DIMENSION (DIM).  APPROXIMATIONS TO
C              THE NORMALIZED TAYLOR COEFFICIENTS, EXCEPT WHEN
C              OUTPUT NCODE = 0. (SEE NOTE(3) BELOW.)
C  (11) WORK   INTERNAL WORKING AREA OF COMPLEX DIMENSION (DIM).
C              (SEE NOTE(3) BELOW). CONTENTS IS IDENTICAL
C              WITH THAT OF TCOF.
C  (12) EXPTAB COMPLEX DIMENSION (DIM/2). (SEE NOTES (2) AND
C              (3) BELOW.)  EXPTAB(J+1) = CEXP(PI*EYE*J/NTAB)
C              J = 0,1,2,...,NTAB-1.   (A HALF CYCLE)
C              OTHER LOCATIONS ARE EMPTY.
C
C      ** NOTES ON INPUT/OUTPUT PARAMETERS **
C NOTE(1)** NCODE IS USED BOTH AS INPUT AND OUTPUT PARAMETER.
C  NORMALLY IT RETAINS THE VALUE +1 AND NEED NOT BE RESET
C  BETWEEN NORMAL RUNS.
C NOTE(2)** THE APPEARANCE OF NTAB AND EXPTAB IN THE CALLING
C  SEQUENCE ALLOWS THE USER TO MAKE USE OF - OR TO PRECOMPUTE -
C  THESE NUMBERS IN ANOTHER PART OF THE PROGRAM SHOULD HE
C  SO DESIRE.  NTAB MUST BE A POWER OF TWO OR 0.
C NOTE(3)** THE APPEARANCE OF NMAX, TCOF, WORK, AND EXPTAB
C  IN THE CALLING SEQUENCE ALLOWS THE SCOPE OF THE SUBPROGRAM
C  AND THE AMOUNT OF STORAGE TO BE ASSIGNED BY THE CALLING
C  PROGRAM, WHICH SHOULD CONTAIN A CONTROL STATEMENT TO THE
C  FOLLOWING EFFECT
C  COMPLEX TCOF(DIM), WORK(DIM), EXPTAB(DIM/2)
C  WHERE DIM IS NORMALLY A POWER OF TWO.  NMAX IS NORMALLY
C  EQUAL TO DIM, BUT MAY BE LESS THAN DIM.
C NOTE(4)** CFUN(Z) IS A USER PROVIDED COMPLEX VALUED
C  FUNCTION SUBPROGRAM WITH A COMPLEX VALUED ARGUMENT.  THE
C  CALLING PROGRAM MUST CONTAIN CONTROL STATEMENTS AS FOLLOWS
C  EXTERNAL CFUN
C  COMPLEX CFUN
C
C      ** BOOKKEEPING PARAMETERS FOR STAGE ONE **
C  NCONV    1  CONVERGENCE ACHIEVED.
C          -1  NO CONVERGENCE ACHIEVED.
C  NROUND   1  NO ROUND OFF TROUBLE OBSERVED.
C           2  ROUND OFF TROUBLE OBSERVED.
C  NABORT   0  UPDATE TOLERANCE AND CONTINUE ON APPEARANCE OF
C              ROUND OFF TROUBLE.
C           1  TERMINATE WHEN ROUND OFF TROUBLE OBSERVED.
C  EXACT    THE EXACT VALUE OF TCOF(1) WHICH IS CFUN(ZETA).
C  SAFETY   THIS IS A SAFETY FACTOR BY WHICH THE ROUTINE AVOIDS
C       THE ROUND OFF LEVEL.  IT IS SET TO 10.0 AND APPEARS
C       ONLY IN THE COMBINATION (SAFETY*EPMACH).  TO ALTER THIS
C       FACTOR, OR TO REMOVE THE ROUND OFF ERROR GUARD
C       COMPLETELY, THE USER NEED ONLY ADJUST THE INPUT
C       PARAMETER EPMACH APPROPRIATELY.
C
```

```
C
C      ** QUANTITIES CALCULATED IN STAGE THREE(A) **
C  THIS IS THE FIRST PART OF ITERATION NUMBER NTCOF. PRESENTLY
C  AVAILABLE ARE  EXPTAB(J+1) = CEXP(PI*EYE*J/NTAB).
C  J = 0,1,2,...NTAB-1.
C  WE REQUIRE THE SEQUENCE  CEXP(PI*EYE*J/NTCOF/2)),
C  J= 1,3,5,...(NTCOF/2-1).
C  IF (NTCOF.LE.2*NTAB) THESE NUMBERS ARE ALREADY AVAILABLE
C  IN THE EXPTAB TABLE SPACED AT AN INTERVAL  2*NSPACE = 4*NTAB/NTCOF.
C  OTHERWISE, NTCOF = 4*NTAB AND THE EXPTAB TABLE IS UPDATED.
C  THIS INVOLVES REARRANGING THE NTAB VALUES AVAILABLE,
C  CALCULATING AND STORING NTAB NEW VALUES AND UPDATING
C  NTAB TO 2*NTAB.
C
C      ** QUANTITIES CALCULATED IN STAGE THREE(B) **
C  ITERATIONS ARE NUMBERED 4,8,16,... AT THE END OF
C  ITERATION NUMBER NTCOF, THE NTCOF COMPLEX FUNCTION
C  VALUES AT ABCISSAS REGULARLY SPACED ON CIRCLE ARE STORED
C  IN THE TCOF VECTOR AS FOLLOWS
C   TCOF(J+1) = CFUN(Z(J))  J=0,1,2,...,NTCOF-1
C  WHERE
C   Z(J) = ZETA + RCIRC*CEXP(2*PI*EYE*J/NTCOF)
C  THIS INVLOVES A REARRANGEMENT OF THE NTCOF/2 FUNCTION
C  VALUES AVAILABLE AT THE START OF THE ITERATION AND THE
C  CALCULATION OF A FURTHER NTCOF/2 FUNCTION VALUES.  IN
C  ADDITION FMAX AND APPROX ARE CALCULATED.  THESE ARE
C   FMAX   MAXIMUM MODULUS OF THE FUNCTION VALUES SO FAR
C          ENCOUNTERED.
C   APPROX AN APPROXIMATION TO TCOF(1) BASED ON THESE
C          FUNCTION VALUES.
C
C      ** QUANTITIES CALCULATED AT STAGE THREE(C) **
C ERROR1 CURRENT VALUE OF THE ERROR = CABS(APPROX-EXACT).
C ERROR2, ERROR3, ERROR4 VALUES OF ERROR AT END OF THREE
C        PREVIOUS ITERATIONS.
C EPMACH MACHINE ACCURACY PARAMETER. (INPUT PARAMETER)
C EPREQ  REQUIRED ACCURACY. (INPUT PARAMETER)
C EPRO   HIGHEST  ACCURACY REASONABLY ATTAINABLE IN VIEW OF
C        THE SIZE OF THE FUNCTION VALUES SO FAR ENCOUNTERED.
C        (=10.0*EPMACH*FMAX)
C EPCOF  CURRENTLY REQUIRED ACCURACY (=AMAX1(EPREQ,EPRO)).
C EPEST  ESTIMATE OF CURRENT ACCURACY. (THE MAXIMUM OF EPRO AND
C        A FUNCTION OF ERRORS 1,2,3 AND 4. (OUTPUT PARAMETER)
C
C      ** CONVERGENCE AND TERMINATION CHECKS IN STAGE THREE(C) **
C (1) USES FMAX TO RAISE EPCOF ABOVE ROUND OFF LEVEL.
C  IF THIS NECESSARY AND THE INPUT VALUE OF NCODE IS NEGATIVE,
C  IT TERMINATES SETTING NCODE=0.
C (2) USES APPROX TO EVALUATE CONVERGENCE OF TCOF(1) TOWARDS
C  EXACT.  IT MAY ASSIGN CONVERGENCE AND GO TO STAGE FOUR(A)
C  SETTING NCODE=+1 OR +2. (CONVERGENCE IS NOT CHECKED FOR
C  FOUR OR FEWER POINTS).
C (3) USES NMAX TO CHECK PHYSICAL LIMIT.  IF THIS HAS BEEN
C  REACHED, IT GOES TO STAGE FOUR(A) SETTING NCODE=-1 OR -2.
C (4) OTHERWISE CONTINUES NEXT ITERATION BY GOING TO STAGE
C  THREE.
C
C **CALCULATION OF FIRST NTCOF TAYLOR COEFFICIENTS IN STAGE FOUR(A)
C  A VERSION OF THE FAST FOURIER TRANSFORM USING A WORK ARRAY
C IS USED.  THE ARRAY **WORK** IS USED ONLY DURING THIS STAGE.
C THE WORK ARRAY ALLOWS THE PERMUTING OF INDICES ASSOCIATED
C WITH IN-PLACE FFTS TO BE SUPPRESSED.  THE FFT CALCULATES
C THE NECCESSARY SUMMATIONS EXCEPT FOR DIVIDING BY NTCOF.
C
C **SETTING OF REMAINING TAYLOR COEFFICIENTS IN STAGE FOUR(B)
C  THE CONVERGENCE CRITERION ALLOWS US TO INFER THAT THE
C NORMALIZED TAYLOR COEFFICIENTS OF ORDER GREATER THAN NTCOF
C ARE ZERO TO ACCURACY EPEST.
C THEY ARE EVALUATED AS BEING EXACTLY ZERO.
      COMPLEX CFUN
      COMPLEX ZETA
      REAL RCIRC,EPREQ,EPMACH,EPEST
      INTEGER NMAX,NCODE,NTCOF,NTAB
      COMPLEX TCOF (1), WORK (1), EXPTAB (1)
      INTEGER NABORT,NCONV,NDISP,NDOLIM,NPREV,NROUND,NSPACE
      REAL COSDIF,EPCOF,EPMIN,EPRO,EP32,EP42,ERROR1,ERROR2
      REAL ERROR3,ERROR4,FMAX,SAFETY,SCALE,TWOPI
      COMPLEX APPROX,EXACT,FVAL,REXP,SUM,ZVAL
      INTEGER J,JCONJ,JFROM,JTAB,JTO
      COMPLEX CMPLX,CONJG
C ***    STAGE ONE    ***
C ---------------------
C INITIALISE BOOKKEEPING PARAMETERS AND EXACT VALUE OF TCOF(1).
      NROUND = 1
      NABORT = 0
      IF (NCODE.LT.0) NABORT = 1
      EPCOF = EPREQ
      SAFETY = 10.0
      ZVAL = ZETA
      FVAL = CFUN(ZVAL)
      EXACT = FVAL
C ***    STAGE TWO    ***
C ---------------------
C FIRST TWO ITERATIONS ( THOSE WITH NTCOF = 1,2 ).
      ERROR3 = 0.0
      ZVAL = ZETA+CMPLX(RCIRC,0.0)
      FVAL = CFUN(ZVAL)
      APPROX = FVAL
      FMAX = CABS(FVAL)
      TCOF(1) = FVAL
      ERROR2 = CABS(APPROX-EXACT)
      ZVAL = ZETA-CMPLX(RCIRC,0.0)
      FVAL = CFUN(ZVAL)
      APPROX = 0.5*(APPROX+FVAL)
      FMAX = AMAX1(FMAX,CABS(FVAL))
      TCOF(2) = FVAL
      ERROR1 = CABS(APPROX-EXACT)
      NTCOF = 2
C ***    STAGE THREE    ***
C -----------------------
C COMMENCE ITERATION NUMBER NTCOF.
  300 CONTINUE
      NPREV = NTCOF
      NTCOF = 2*NTCOF
C ***    STAGE THREE(A)    ***
C -------------------------
C UPDATE EXPTAB TABLE IF NECESSARY.
      IF (2*NTAB.GE.NTCOF) GO TO 340
      IF (NTAB.GE.2) GO TO 310
        EXPTAB(1) = (1.0,0.0)
        EXPTAB(2) = (0.0,1.0)
        NTAB = 2
        GO TO 340
```

```
  310 CONTINUE
      NDOLIM = NTAB-1
      DO 320 J = 1,NDOLIM
        JFROM = NTAB-J
        JTO = 2*JFROM
        EXPTAB(JTO+1) = EXPTAB(JFROM+1)
  320 CONTINUE
      NTAB = 2*NTAB
      TWOPI = 8.0*ATAN(1.0)
      COSDIF = COS(TWOPI/FLOAT(2*NTAB))
      NDOLIM = NTAB-3
      DO 330 J = 1,NDOLIM,2
        EXPTAB(J+1) = (0.5*EXPTAB(J)+0.5*EXPTAB(J+2))/COSDIF
  330 CONTINUE
      EXPTAB(NTAB) = (0.5*EXPTAB(NTAB-1)-(0.5,0.0))/COSDIF
  340 CONTINUE
C ***   STAGE THREE(B)    ***
C --------------------------
C UPDATE LIST OF FUNCTION VALUES IN TCOF.
C CALCULATE FMAX AND APPROX.
      NDOLIM = NPREV-1
      DO 350 J = 1,NDOLIM
        JFROM = NPREV-J
        JTO = 2*JFROM
        TCOF(JTO+1) = TCOF(JFROM+1)
  350 CONTINUE
      SUM = (0.0,0.0)
      NSPACE = (2*NTAB)/NTCOF
      DO 360 J = 1,NDOLIM,2
        JTAB = J*NSPACE
        REXP = RCIRC*EXPTAB(JTAB+1)
        ZVAL = ZETA+REXP
        FVAL = CFUN(ZVAL)
        SUM = SUM+FVAL
        FMAX = AMAX1(FMAX,CABS(FVAL))
        TCOF(J+1) = FVAL
        JCONJ = NTCOF-J
        ZVAL = ZETA+CONJG(REXP)
        FVAL = CFUN(ZVAL)
        SUM = SUM+FVAL
        FMAX = AMAX1(FMAX,CABS(FVAL))
        TCOF(JCONJ+1) = FVAL
  360 CONTINUE
      APPROX = 0.5*APPROX+SUM/FLOAT(NTCOF)
C ***   STAGE THREE(C)    ***
C --------------------------
C CONVERGENCE AND TERMINATION CHECK.
      ERROR4 = ERROR3
      ERROR3 = ERROR2
      ERROR2 = ERROR1
      ERROR1 = CABS(APPROX-EXACT)
      EPRO = FMAX*SAFETY*EPMACH
      IF (EPRO.LT.EPCOF) GO TO 370
        EPCOF = EPRO
        NROUND = 2
        IF (NABORT.EQ.0) GO TO 370
        NCODE = 0
        EPEST = EPRO
        GO TO 470
  370 CONTINUE
      IF (NTCOF.LE.4) GO TO 380
      ERROR4 = AMAX1(ERROR4,EPRO)
      ERROR3 = AMAX1(ERROR3,EPRO)
      EP42 = ERROR2*((ERROR2/ERROR4)**(4.0/3.0))
      EP32 = ERROR2*((ERROR2/ERROR3)**2)
      EPMIN = AMIN1(ERROR2,EP32,EP42)
      EPEST = AMAX1(ERROR1,EPMIN,EPRO)
      IF (EPEST.GT.EPCOF) GO TO 380
        NCONV = 1
        GO TO 400
  380 CONTINUE
      IF (2*NTCOF.LE.NMAX) GO TO 300
        NCONV = -1
C ***   STAGE FOUR(A)    ***
C --------------------------
C CALCULATION OF FIRST NTCOF TAYLOR COEFFICIENTS USING F.F.T.
  400 CONTINUE
      NCODE = NCONV*NROUND
      NDISP = NTCOF
  410 CONTINUE
      NDISP = NDISP/2
      CALL CFCOF (NTCOF,NDISP,TCOF,WORK,NTAB,EXPTAB)
      IF (NDISP.GT.1) GO TO 430
      DO 420 J = 1,NTCOF
        TCOF(J) = WORK(J)
  420 CONTINUE
      GO TO 440
  430 CONTINUE
      NDISP = NDISP/2
      CALL CFCOF (NTCOF,NDISP,WORK,TCOF,NTAB,EXPTAB)
      IF (NDISP.GT.1) GO TO 410
  440 CONTINUE
      SCALE = 1.0/FLOAT(NTCOF)
      DO 450 J = 1,NTCOF
        TCOF(J) = TCOF(J)*SCALE
        WORK(J) = TCOF(J)
  450 CONTINUE
C ***   STAGE FOUR(B)    ***
C --------------------------
C SETTING OF REMAINING TAYLOR COEFFICIENTS.
      IF (NTCOF.GE.NMAX) GO TO 470
      NDOLIM = NTCOF+1
      DO 460 J = NDOLIM,NMAX
        TCOF(J) = (0.0,0.0)
        WORK(J) = (0.0,0.0)
  460 CONTINUE
  470 CONTINUE
      RETURN
C END OF ENTCAF
      END
      SUBROUTINE CFCOF ( NTCOF, NDISP, TCOF, WORK, NTAB, EXPTAB )
C
C ** COMPLEX FOURIER COEFFICIENTS **
C     **  GENERAL PURPOSE  **
C THIS ROUTINE DOES ONE PASS OF A FAST FOURIER TRANSFORM.
C THE INDEXING IS ARRANGED SO THAT THE COEFFICIENTS ARE IN
C ORDER AT THE END OF THE LAST PASS.  THIS INDEXING REQUIRES
C THE USE OF SEPARATE ARRAYS FOR INPUT AND OUTPUT OF THE
C PARTIAL RESULTS.  THIS ROUTINE IS CALLED ONCE FOR
C EACH PASS.
C
C     **  INPUT PARAMETERS  **
C  (1)  NTCOF     NUMBER OF COEFFICIENTS TO BE PROCESSED.
C  (2)  NDISP     MAXIMUM VALUE OF DISPLACEMENT INDEX.
C  (3)  TCOF      (COMPLEX) INPUT ARRAY.
C  (5)  NTAB      NUMBER OF ENTRIES IN EXPTAB.
C  (6)  EXPTAB    (COMPLEX) TABLE OF VALUES OF COMPLEX EXPONENTIAL.
C                 EXPTAB(J+1) = CEXP(PI*EYE*J/NTAB),
C                 J = 0,1,2,...,NTAB-1.
C
C     **  OUTPUT PARAMETERS  **
C  (4)  WORK      (COMPLEX) OUTPUT ARRAY.
C
C     **  INDEXING OF ARRAYS  **
C THE TWO POINT FOURIER TRANSFORM IS APPLIED TO THE POINTS
C OF TCOF WITH INDICES
C  (JDISP-1)*NPREV+JREPL  AND  (JDISP-1)*NPREV+JREPL+NHALF.
C THE RESULTS ARE MODIFIED BY THE APPROPRIATE TWIDDLE FACTOR
C AND STORED IN WORK WITH INDICES
C  (JDISP-1)*NNEXT+JREPL  AND  (JDISP-1)*NNEXT+JREPL+NPREV
C WHERE
C      NDISP       PRODUCT OF REMAINING FACTORS.
C      NPREV       PRODUCT OF PREVIOUS FACTORS.
C      NNEXT       PRODUCT OF PREVIOUS AND CURRENT FACTORS.
C      NHALF       PRODUCT OF PREVIOUS AND REMAINING FACTORS.
C      JDISP       DISPLACEMENT INDEX = 1,2,...NDISP.
C      JREPL       REPLICATION INDEX = 1,2,...NPREV.
C
C THE TWIDDLE FACTOR CEXP(-PI*EYE*J/NDISP), J=0,1,99NDISP-1
C IS OBTAINED BY TAKING THE CONJUGATE OF ELEMENTS SPACED
C EVERY NSPACE=NTAB/NDISP OF EXPTAB.
      INTEGER NTCOF,NDISP,NTAB
      COMPLEX TCOF (1), WORK (1), EXPTAB (1)
      COMPLEX CONJG
      COMPLEX ROT,Z0,Z1
      INTEGER J,JDISP,JREPL,JTAB,JT,JW
      INTEGER KT0,KT1,KW0,KW1,NHALF,NNEXT,NPREV,NSPACE
      NHALF = NTCOF/2
      NPREV = NTCOF/(2*NDISP)
      NNEXT = NTCOF/NDISP
      NSPACE = NTAB/NDISP
      DO 200 JDISP = 1,NDISP
        J = JDISP-1
        JTAB = J*NSPACE
        ROT = CONJG(EXPTAB(JTAB+1))
        JT = J*NPREV
        JW = J*NNEXT
        DO 100 JREPL = 1,NPREV
          KT0 = JT+JREPL
          KT1 = KT0+NHALF
          KW0 = JW+JREPL
          KW1 = KW0+NPREV
          Z0 = TCOF(KT0)
          Z1 = TCOF(KT1)
          WORK(KW0) = Z0+Z1
          WORK(KW1) = (Z0-Z1)*ROT
  100   CONTINUE
  200 CONTINUE
      RETURN
C END OF CFCOF
      END
```