ALGORITHM 579
CPSC: Complex Power Series Coefficients [D4]

BENGT FORNBERG
California Institute of Technology

Key Words and Phrases: numerical differentiation, Taylor series coefficients, analytic functions
CR Categories: 5.16
Language: FORTRAN

DESCRIPTION

An algorithm CPSC is presented here. It evaluates numerically the leading coefficients in a power series expansion of an analytic function (or, equivalently, a number of leading derivatives of an analytic function). A detailed description of the theoretical background of the code is given in [1], together with some warnings about cases in which full accuracy may not be reached.

Such cases are

(1) very low-order polynomials (for example, $f(z) = 1 + z$);
(2) functions whose Taylor expansions contain very large isolated terms (for example, $f(z) = 10^6 + (1/(1 - z))$);
(3) certain entire functions (for example, $f(z) = e^z$);
(4) functions whose radius of convergence is limited by a branch point at which the function remains many times differentiable (for example, $f(z) = (1 + z)^{10} \log(1 + z)$ expanded around $z = 0$).

In the case 1, the routine will normally fail to even approximate the correct answer. In cases 2, 3, and 4, problems are normally encountered only if large numbers of coefficients are wanted (e.g., more than 30). The supplied error estimate will, in most of these cases, give correct information about the lowered accuracy.

Inevitably there is a risk that the routine will attempt to evaluate the given function exactly at a singularity. This will happen, for example, for $f(z) = 1/$

Received 28 August 1978; revised 6 February 1980; accepted 19 June 1981

This work was supported by Control Data Corporation and by the Department of Energy (Office of Basic Energy Sciences).
Author's address: Department of Applied Mathematics, 217-50 Caltech, Pasadena, CA 91125.

(1 − z) expanded around zero if the initial radius is of the form $r = 2^p$, with p any integer. The risk for such exact coincidences in floating point can be minimized by choosing "irregular" initial radii. A practical procedure may be to start each case by the obtained final radius from a previous case.

REFERENCES

1. FORNBERG, B. Numerical differentiation of analytic functions. *ACM Trans. Math. Softw.* 7, 4 (Dec. 1981), 512–526.

ALGORITHM

The user must enter the machine accuracy in a data statement. The routine adjusts the calculations accordingly in order to obtain the best possible accuracy. However, since a few decimal places are always lost in this routine, we do not recommend its general use if the machine accuracy is less than 10 significant decimal places. For example, for use on IBM System 370 machines, we recommend double or quadruple precision. The code below is for IBM double precision.

[A part of the listing is printed here. The complete listing is available from the ACM Algorithms Distribution Service (see page 555 for order form).]

```
C CPSC - COMPLEX POWER SERIES COEFFICIENTS
C
C BY BENGT FORNBERG
C
C ACM TRANSACTIONS ON MATHEMATICAL SOFTWARE, DECEMBER 1981
C
      PROGRAM TEST (OUTPUT)
C
C  THIS PROGRAM WITH THE FUNCTION F BELOW USES CPSC TO EVALUATE
C  THE LEADING DERIVATIVES OF F(Z) = CEXP(Z)/(CSIN(Z)**3+CCOS(Z)**3)
C  AT Z = Ø.
      EXTERNAL F
      DIMENSION A(51),IRANGE(4),ER(51)
      COMPLEX A
      DATA IRANGE/6,12,25,51/
      DO 1Ø I=1,4
        IR = IRANGE(I)
        R = 1.
        CALL CPSC(F,(Ø.,Ø.),IR,1,R,A,ER)
        WRITE(6,2Ø) I
        WRITE(6,3Ø) (A(J),J=1,IR)
   1Ø   WRITE(6,4Ø) (ER(J),J=1,IR)
   2Ø FORMAT(/2ØH DERIVATIVES , RANGE,I3)
   3Ø FORMAT(4(1X,E18.1Ø,E9.1))
   4Ø FORMAT(/17H ESTIMATED ERRORS/(1X,16E8.1))
      STOP
      END
      COMPLEX FUNCTION F(Z)
C
C  TEST FUNCTION FOR USE WITH CPSC.
C
      COMPLEX Z
      F = CEXP(Z)/(CSIN(Z)**3+CCOS(Z)**3)
      RETURN
      END
```

```
      SUBROUTINE CPSC(F,Z,N,IC,R,RS,ER)
C
C  EVALUATION OF COMPLEX POWER SERIES COEFFICIENTS OR DERIVATIVES.
C
C  *** INPUT PARAMETERS ***
C    F  COMPLEX FUNCTION, OF WHICH THE COEFFICIENTS OR DERIVATIVES
C       ARE SOUGHT. THIS FUNCTION MUST BE DECLARED EXTERNAL IN THE
C       CALLING PROGRAM.
C    Z  COMPLEX POINT AROUND WHICH F SHALL BE EXPANDED OR AT WHICH
C       DERIVATIVES SHALL BE EVALUATED.
C    N  INTEGER, NUMBER OF COEFFICIENTS OR DERIVATIVES WANTED.
C       N MUST BE GE 1 AND LE 51.
C    IC SELECTS BETWEEN POWER SERIES COEFFICIENTS AND DREIVATIVES.
C       IC .EQ. Ø  ROUTINE RETURNS POWER SERIES COEFFICIENTS IN  RS
C       IC .NE. Ø  ROUTINE RETURNS DERIVATIVES IN RS .
C  *** INPUT AND OUTPUT PARAMETER ***
C    R  INITIAL RADIUS USED IN SEARCH FOR OPTIMAL RADIUS. THE RESULTING
C       RADIUS IS LEFT IN R. THE PROVIDED GUESS MAY BE IN ERROR WITH AT
C       MOST A FACTOR OF 3.E4 .
C  *** OUTPUT PARAMETERS ***
C    RS COMPLEX ARRAY RS(N) CONTAINING THE N FIRST
C       COEFFICIENTS (CORRESPONDING TO THE POWERS Ø TO N-1) OR DERIVA-
C       TIVES (ORDERS Ø TO N-1) .
C    ER REAL ARRAY ER(N) CONTAINING ERROR ESTIMATES FOR THE
C       NUMBERS IN RS(N).
C
      DIMENSION IP(32),A(64),RS(N),ER(N),RT(51,3),FV(6),
     *  IW(7),SC(4),RV(3),C(4),FC(3)
      COMPLEX F,A,V,RS,RT,FV,U,W,T,Z,RV,RQ,S,XK,MULT,CO
C
C  LIST OF THE VARIABLES INITIALIZED IN THE DATA STATEMENT BELOW.
C   EPS  MACHINE ACCURACY. THIS CONSTANT HAS TO BE SUPPLIED BY
C        THE USER. IN THIS LIST, IT IS GIVEN AS 1.E-14 CORRESPONDING
C        TO THE 48 BIT FLOATING POINT MANTISSAS ON CDC CYBER MACHINES.
C   IND  INTEGER FLAG.
C   L2   INTEGER FLAG.
C   IW   2**( Ø , 1 , 2 , 3 , 4 , 5 , 6 ) .
C   IP   PERMUTATION CONSTANTS FOR THE FFT.
C   RV   CONSTANTS FOR THE LAURENT SERIES TEST .
C
      DATA EPS/1.E-14/,IND/Ø/,L2/1/,IW/1,2,4,8,16,32,64/,
     *  IP/64,32,48,16,56,24,4Ø,8,6Ø,28,44,12,52,2Ø,36,4,62,3Ø,46,14,
     *    54,22,38,6,58,26,42,1Ø,5Ø,18,34,2/,
     *  RV/(-.4,.3),(.7,.2),(.Ø2,-.Ø6)/
C
C  STATEMENT FUNCTION FOR MULTIPLICATION OF A COMPLEX NUMBER
C  BY A REAL NUMBER.
C
      MULT(RE,CO) = CMPLX(RE*REAL(CO),RE*AIMAG(CO))
C
C  EVALUATE SOME CONSTANTS THE FIRST TIME THE CODE IS EXECUTED.
C
      IF(IND.EQ.1) GOTO 2Ø
      IND = 1
      SC(1) = .125
      C(1) = EPS**(1./28.)
      EP6 = C(1)**6
      PI = 4.*ATAN(1.)
```

```
      FV(1) = (-1.,Ø.)
      FV(2) = (Ø.,-1.)
      R1 = SQRT(.5)
      RA = 1./R1
      FV(3) = CMPLX(R1,-R1)
      DO 1Ø I=2,4
        SC(I) = .5*SC(I-1)
        C(I) = SQRT(C(I-1))
        ANG = PI*SC(I-1)
   1Ø   FV(I+2) = CMPLX(COS(ANG),-SIN(ANG))
   2Ø CONTINUE
C
C  START EXECUTION.
C
      IF(N.GT.51.OR.N.LT.1) GOTO 26Ø
      LF = Ø
      NP = Ø
      M = Ø
      NR = -1
C
C  FIND IF A FFT OVER 8, 16, 32, OR 64 POINTS SHOULD BE USED.
C
      KL = 1
      IF(N.GT.6) KL=2
      IF(N.GT.12) KL=3
      IF(N.GT.25) KL=4
      KM = KL+2
      KN = 7-KM
      IX = IW(KM+1)
      IS = IW(KN)
   3Ø V = CMPLX(R,Ø.)
C
C  FUNCTION VALUES OF F ARE STORED READY PERMUTATED FOR THE FFT.
C
      DO 4Ø I=IS,32,IS
        IQ = IP(I)
        V = V*FV(KM)
        A(IQ) = F(Z+V)
   4Ø   A(IQ-1) = F(Z-V)
      LN = 2
      JN = 1
C
C  THE LOOP  DO 7Ø ... CONSTITUTES THE FFT.
C
      DO 7Ø L=1,KM
        U = (1.,Ø.)
        W = FV(L)
        DO 6Ø J=1,JN
          DO 5Ø I=J,IX,LN
            IT = I+JN
            T = A(IT)*U
            A(IT) = A(I)-T
   5Ø       A(I) = A(I)+T
   6Ø     U = U*W
        LN = LN+LN
   7Ø   JN = JN+JN
      CX = Ø.
      B = 1.
C
```

```
C  TEST ON HOW FAST THE COEFFICIENTS OBTAINED DECREASE.
C
      DO 8Ø I=1,IX
        CT = CABS(A(I))/B
        IF(CT.LT.CX) GOTO 8Ø
        CX = CT
        INR = I
   8Ø   B = B*C(KL)
      IF(M.LE.1) GOTO 1ØØ
C
C  ESTIMATE OF THE ROUNDING ERROR LEVEL FOR THE LAST RADIUS.
C
      ER(1) = CX*EPS
      DO 9Ø I=2,N
   9Ø   ER(I) = ER(I-1)/R
  1ØØ SF = SC(KL)
      DO 11Ø I=1,IX
        A(I) = MULT(SF,A(I))
  11Ø   SF = SF/R
      L1 = L2
      L2 = 1
      IF(INR.GT.IW(KM)) GOTO 15Ø
      IF(LF.EQ.1) GOTO 14Ø
C
C  TEST IF THE SERIES IS A TAYLOR OR A LAURENT SERIES.
C
      SR = Ø.
      SP = Ø.
      DO 13Ø J=1,3
        RQ = MULT(R,RV(J))
        S = A(IX)
        DO 12Ø I=2,IX
          IA = IX+1-I
  12Ø     S = S*RQ+A(IA)
        CP = CABS(S)
        IF(CP.GT.SP) SP=CP
        CM = CABS(S-F(Z+RQ))
  13Ø   IF(CM.GT.SR) SR=CM
      IF(SR.GT.1.E-3*SP) GOTO 15Ø
      LF = 1
  14Ø L2 = -1
C
C  DETERMINATION OF THE NEXT RADIUS TO BE USED.
C
  15Ø IF(NR.GE.Ø) GOTO 16Ø
      FACT = 2.
      IF(L2.EQ.1) FACT=.5
      L1 = L2
      NR = Ø
  16Ø IF(L1.NE.L2) GOTO 18Ø
      IF(NR.GT.Ø) GOTO 17Ø
      NP = NP+1
      IF(NP-15) 19Ø,19Ø,26Ø
  17Ø FACT = 1./FACT
  18Ø FACT = 1./SQRT(FACT)
      NR = NR+1
  19Ø R = R*FACT
      M = NR-KL-1
      IF(M.LE.Ø) GOTO 3Ø
```

```
C
C  THE RESULTS FOR THE LAST THREE RADII ARE STORED.
C
      DO 200 I=1,N
  200   RT(I,M) = A(I)
      IF(M.EQ.1) GOTO 220
C
C  EXTRAPOLATION.
C
      DO 210 I=1,N
        XK = RT(I,M-1)-RT(I,M)
  210   RT(I,M-1) = RT(I,M)-MULT(FC(M-1),XK)
      IF(M.EQ.3) GOTO 230
C
C  CALCULATION OF THE EXTRAPOLATION CONSTANTS.
C
  220 FC(M) = 1.5+SIGN(.5,FACT-1.)
      IF(M.EQ.2) FC(M)=FC(M)+RA
      IF(FACT.GT.1.) FC(M)=-FC(M)
      GOTO 30
  230 FC(3) = FC(1)*FC(2)/(FC(1)+FC(2)+1.)
C
C  FINAL EXTRAPOLATION AND ERROR ESTIMATE.
C
      DO 240 I=1,N
        XK = RT(I,1)-RT(I,2)
        ER(I) = ER(I)+EP6*CABS(XK)
  240   RS(I) = RT(I,2)-MULT(FC(3),XK)
C
C  MULTIPLY POWER SERIES COEFFICIENTS AND ERROR ESTIMATE BY FACTORIALS
C  IF DERIVATIVES WANTED.
C
      IF(IC.EQ.0) RETURN
      FAC = 0.
      FACT = 1.
      DO 250 I=1,N
        RS(I) = MULT(FACT,RS(I))
        ER(I) = FACT*ER(I)
        FAC = FAC+1.
  250   FACT = FACT*FAC
      RETURN
C
C  ERROR RETURN.
C
  260 DO 270 I=1,N
        RS(I) = (0.,0.)
  270   ER(I) = 1.E10
      RETURN
      END
```