3 4456 0555900 8
OAK RIDGE NATIONAL LABORATORY LIBRARIES

CENTRAL RESEARCH LIBRARY

ORNL-TM-4588

cy 2

SUBROUTINE CAUCHY — COMPLEX ROOTS OF A FUNCTION USING A CAUCHY INTEGRAL TECHNIQUE

C. O. Beasley, Jr.
H. K. Meier

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
DOCUMENT COLLECTION
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this document, send in name with document and the library will arrange a loan.
UCN-7969
(3 3-67)

OAK RIDGE NATIONAL LABORATORY
OPERATED BY UNION CARBIDE CORPORATION • FOR THE U.S. ATOMIC ENERGY COMMISSION

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

ORNL-TM-4588

Contract No. W-7405-eng-26

Thermonuclear Division

SUBROUTINE CAUCHY - COMPLEX ROOTS OF A FUNCTION USING A CAUCHY INTEGRAL TECHNIQUE

C. O. Beasley, Jr. and H. K. Meier
Oak Ridge National Laboratory, Oak Ridge, Tennessee

AUGUST 1974

NOTICE This document contains information of a preliminary nature and was prepared primarily for internal use at the Oak Ridge National Laboratory. It is subject to revision or correction and therefore does not represent a final report.

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
U.S. ATOMIC ENERGY COMMISSION

3 4456 0555900 8

OAK RIDGE NATIONAL LABORATORY LIBRARIES

SUBROUTINE CAUCHY - COMPLEX ROOTS OF A FUNCTION USING A CAUCHY INTEGRAL TECHNIQUE

C. O. Beasley, Jr. and H. K. Meier
Oak Ridge National Laboratory, Oak Ridge, Tennessee

ABSTRACT

A description of the subroutine CAUCHY, the program for finding zeros of an analytic function f(z) within a contour in the z-plane, is given. Examples of types of functions for which this program is most advantageous as well as types of functions which show its weaknesses, are given.

I. INTRODUCTION

The purpose of this report is to describe in detail the subroutine CAUCHY and its subprograms. This routine is used to find the zeros of an arbitrary function f(z) within a contour c in the complex z-plane.

The method used has been documented.[1] It relies on calculating integrals of the form

$$S_N = \frac{1}{2\pi i} \oint z^N \frac{f'(z)}{f(z)} dz = \sum_{i=1}^{M} z_i^N - \sum_{j=1}^{K} w_i^N \tag{1}$$

where z_i are the zeros and w_i are the poles of f within c. Thus if there are no poles, S_0 gives the number of zeros, S_1 the sum of the zeros, S_2 the sum of their squares, etc.

*Research sponsored by U. S. Atomic Energy Commission under contract with Union Carbide Corporation.

This method becomes numerically practical when c contains a small number of zeros. Thus some knowledge of f is desirable, and in principle, if sufficiently small integration steps are taken so that the function is sufficiently smooth, the answer may be obtained to arbitrary accuracy. Since the method does not rely on iteration, it will work even when well-known iteration methods will not.

The problem is then to perform the integral with suitable accuracy. One straight-forward method is found in the programs LSQNK1 and LSQNK2 from the Massachusetts Institute of Technology Information Processing Center.[2] The method used by these programs is to integrate (1) by parts, and then use a Simpson's rule integration to perform the numerical integration around c. The program is most useful when large numbers of points of c may be calculated and when a large number of roots may exist ($M \leq 4$ for this program). An automatic increasing or decreasing of step size along c is included in the program.

In many cases of numerical interest, it is costly to compute the function, and one would like to locate one or two zeros with fair accuracy with a minimum number of points. The presence of nearby poles or zeros in f introduces poles on the integrand of equation 1 and may make normal integration procedures impractical because of the large number of contour points required. It is these cases for which the subroutine CAUCHY is designed.

In the next section, we will discuss the method. In the third section, we describe the subroutines. The fourth section gives some sample results, and the fifth a listing of the subroutine and its subprograms

II. METHOD

Suppose we are given three values of a function f_j at three points z_j. We can form a best-fit binomial

$$f_{approx} = a\, z^2 + b\, z + c \tag{2}$$

by determining a, b, and c. Now we may factor f_{approx}

$$f_{approx} = (z - z_1)(z - z_2) \tag{3}$$

where

$$z_{1,2} = - \frac{b \pm \sqrt{b^2 - 4\, a\, c}}{2a}$$

Then

$$f'_{approx} = (z - z_2) + (z - z_1)$$

and

$$\frac{f'_{approx}}{f_{approx}} = \frac{1}{z - z_1} + \frac{1}{z - z_2} \tag{4}$$

So we may perform our integral over the segment of the contour c over which (2) is valid; the integrals are of the form

$$\int_{z_a}^{z_b} dz\, \frac{z^n}{z - z_\alpha} \qquad n = 0,1,2 \ldots \tag{5}$$

$$= \ln\left(\frac{z_b - z_\alpha}{z_a - z_\alpha}\right) \qquad n = 0$$

$$= (z_b - z_a) - \ln\left(\frac{z_b - z_\alpha}{z_a - z_\alpha}\right) \qquad n = 1$$

$$= \frac{1}{2}(z_b - z_a)(z_b + z_a - 2\, z_\alpha) - 2(z_b - z_a) + \ln\left(\frac{z_b - z_\alpha}{z_a - z_\alpha}\right) \qquad n = 2$$

However, (2) is not a good approximation to f if f has nearby singularities. A much better approximation obtains if f_j is known at five points z_j:

$$f_{approx} = \frac{c_1 + c_2 z + c_3 z^2}{c_4 + c_5 z + c_6 z^2} \tag{6}$$

We may factor both numerator and denominator

$$f_{approx} = \frac{(z-z_{n1})(z-z_{n2})}{(z-z_{d1})(z-z_{d2})} \frac{c_3}{c_6} \tag{7}$$

So

$$\frac{f'_{approx}}{f_{approx}} = \frac{1}{z-z_{n1}} + \frac{1}{z-z_{n2}} - \frac{1}{z-z_{d1}} - \frac{1}{z-z_{d2}} \tag{8}$$

Thus in calculating the integral (5), one adds the contributions from the roots of the numerator and subtracts those of the denominator.

We must first solve for the c_i. We can make the numerical computation more accurate by transforming to a coordinate system where the center point of the five z_j, z_o, is zero; thus we know our function at five points f_{-2}, f_{-1}, f_o, f_1, and f_2. We can then write (6) as

$$c_1 + c_2 z_j + c_3 z_j^2 = (c_4 + c_5 z_j + c_6 z_j^2) f_j \tag{9}$$

We note that (9) is a set of five linear equations for the six c_i. We may write the c_i as follows. We first note that at $z_o = 0$,

$$c_1 = c_4 f_o \tag{10}$$

Using this, we may write the other numerator coefficients c_2 and c_3 in terms of the denominator coefficients:

$$c_2 = \frac{z_1 z_{-1} (f_{-1}-f_1)}{z_1 - z_{-1}} c_6 + \frac{(z_1 f_{-1} - z_{-1} f_1)}{z_1 - z_{-1}} c_5 + \tag{11}$$

$$+ \frac{1}{z_1 - z_{-1}} \left[\frac{z_1}{z_{-1}} (f_{-1} - f_o) - \frac{z_{-1}}{z_1} (f_1 - f_o) \right] c_4$$

$$c_3 = \frac{z_1 f_1 - z_{-1} f_{-1}}{z_1 - z_{-1}} c_6 + \frac{f_1 - f_{-1}}{z_1 - z_{-1}} c_5 + \frac{1}{z_1 - z_{-1}} \left[\frac{f_1 - f_o}{z_1} - \frac{f_{-1} - f_o}{z_{-1}} \right] c_4 \tag{12}$$

If we put c_1, c_2, and c_3 in (9)and divide by z_j,

$$A^j c_5 + B^j c_6 = C^j c_4 \quad (j = \pm 2) \tag{13}$$

where

$$A^j = \frac{z_j (f_1 - f_{-1})}{z_1 - z_{-1}} + \frac{(z_1 f_{-1} - z_{-1} f_1)}{z_1 - z_{-1}} - f_j \tag{14}$$

$$B^j = \frac{z_j (z_1 f_1 - z_{-1} f_{-1})}{z_1 - z_{-1}} + \frac{z_1 z_{-1} (f_{-1} - f_1)}{z_1 - z_{-1}} - z_j f_j \tag{15}$$

$$C^j = \frac{f_j - f_o}{z_j} - \frac{1}{z_1 - z_{-1}} \left[\frac{z_1}{z_{-1}} (f_{-1} - f_o) - \frac{z_{-1}}{z_1} (f_1 - f_o) \right]$$

$$- \frac{z_j}{z_1 - z_{-1}} \left[\frac{f_1 - f_o}{z_1} - \frac{f_{-1} - f_o}{z_{-1}} \right] \tag{16}$$

$A^j = B^j = C^j = 0$ for $j = 0, \pm 1$. We now solve (13) for c_5 and c_6:

$$c_5 = \frac{c_4}{DET} \begin{vmatrix} C^2 & B^2 \\ C^{-2} & B^{-2} \end{vmatrix} \tag{17}$$

$$c_6 = \frac{c_4}{DET} \begin{vmatrix} A^2 & C^2 \\ A^{-2} & C^{-2} \end{vmatrix} \tag{18}$$

where

$$DET = A^2 B^{-2} - B^2 A^{-2} \tag{19}$$

If we now set c_4 = DET, we have a unique solution for the c_j.

Problems do arise if DET = 0. This will happen when f is of the form

$$f = \frac{\alpha_o + \alpha_1 z}{\beta_o + \beta_1 z} \tag{20}$$

We examine three special cases:

1) $\alpha_1 = \beta_1 = 0$. In this case, $A^j = B^j = C^j = 0$. The simplest solution is $c_4 = 1$, $c_5 = c_6 = 0$.

2) $\beta_1 = 0$. In this case, $A^j = C^j = 0 \neq B^j$. Again the simplest solution is to set $c_4 = 1$, $c_5 = c_6 = 0$.

3) $\alpha_1 \neq 0 \neq \beta_1$. In this case we are free to choose $C_6 = 0$; then from (13), $A^j c_5 = C^j c_4$, and since the determinant is zero (A^{-2}, C^{-2}) equals (A^2, C^2) within a constant factor so we can set $c_5 + C^2$ and $c_4 = A^2$. One could, of course, set c_5 and c_6 to any numbers not equal to 0; this would amount to multiplying numerator and denominator by the same factor, provided the overall polynominal remained of degree $\lesssim 2$.

Once the coefficients c_i have been obtained, the roots z_{n1}, z_{n2}, z_{d1} and z_{d2} are easily obtained. The integral is then performed from z_a to z_b where $z_a = \frac{z_{-1} + z_o}{2}$ and $z_b + \frac{z_o + z_1}{2}$. If z_o is not a corner point, that is, if $\Delta^2 z/\Delta z = 0$, then the integral is performed in one step: z_a to z_b: If $\Delta^2 z/\Delta z \neq 0$, it is performed in two steps, z_a to z_o, and z_o to z_b. The integrals are performed in the transformed coordinate system ($z_3 = 0$), the same one that the coefficients are found in.

III. DESCRIPTION OF THE SUBROUTINE

A. CAUCHY

The main subroutine begins by initializing IND (D = double precision), where N is defined in equation (1). The initialization is necessary, since the subroutine which performs the integral adds incrementally to IND.

Setting $C(1) = 10^{15}$ signals COEF that it is the first time that COEF has been called for a given contour (see description of COEF).

The array ZSC(I) is set equal to $|\Delta Z_i|$. If any ZSC(I) equals zero, an error message is printed out and a return initiated.

A check is made to see if the contour contains at least five points. If not, an error message is printed out and a return initiated.

The DO 160 loop is now started; this ultimately performs the incremental integral for each Z(I). Each set of five z_j = ZD(J) and f_j = FD(J), $1 \le J \le 5$ are then set up, with special provisions at the "beginning" and "ending" of the contour. For these five ZD(J) and FD(J), the coefficients c_ℓ = CD(L), $1 \le L \le 6$, are calculated (see equation (6)) in subroutine COEF. The roots ZRN(1) and ZRN(2) of the numerator, and ZRD(1) and ZRD(2) of the denominator, are calculated in ROOT2.

A check is then made to see if $\Delta^2 Z_3/\Delta Z_3 \simeq 0$; if so, the point ZD(3) is not a "corner point", and the integral is performed from [ZD(2)+ZD(3)]/2=Z1 to [ZD(3)+ZD(4)]/2=Z2. If ZD(3) is a corner point, the integral is done in two steps: from Z1 to ZD(3), and from ZD(3) to Z2.

The integrals IN = IND, $0 \le N \le 2$ are divided by $2\pi i$ and then a return is made to the calling program.

B. COEF

Given the function F(I) at Z(I), $1 \le I \le 5$, this subroutine calculates the c_j = C(J), $1 \le J \le 6$ as defined in (6).

A check is first made to see if C(1) = 10^{15}. If so, it is the first call of COEF for a given contour, and all five $|f_i|$ = FM(I), $1 \le I \le 5$ are calculated; if not, the FM(I) are set equal to the FM(I+1), $1 \le I \le 4$, and FM(5) is calculated from F(5).

From the FM(I), a scale factor is formed by finding the largest FM(I). If Max (FM(I)) = 0, F(I) = 0, $1 \le I \le 5$, and the scale factor is not needed; otherwise the scale factor SCF = 1/max(FM(I)).

All FM(J) are now multiplied by SCF. Then the second largest FM(I), TSCFM is found. If it is 0, then the largest F(I) = F(J) is a singular point, and the points of the contour have to be redefined. Hence an error message and stop are provided. If two or more FM(I) are equal to the maximum FM(I), or if the TSCFM $\neq$ 0, the program proceeds. Note that the presence of two singular points at two contour points (within five consecutive points) would cause the program to fail without any error message.

The program now proceeds to calculate the A^j, B^j, C^j defined in equations (14), (15), and (16) ($A^{-2} \equiv$ AN2, etc.). From these, C(4), C(5), and C(6) are calculated. We now test to see if C(4) = DET = 0. Since we have used the scaled FF(I) to calculate C(4), we define "0" as a small number to take care of roundoff errors in subtraction. If |C(4)| = "0", then we have a function of the type shown in equation (20).

We then set C(5) = C2 and C(4) = A2 as discussed in case 3). In addition $|C(5)| = 0$, we have either case 2) or case 1), and in either case, the denominator is a constant, set equal to 1. The numerator coefficient C(1), C(2), C(3) are then calculated, and the return is made to CAUCHY.

C. ROOT2

This program calculates the roots ZR(1) and ZR(2) of $f = A*Z^2 + B*Z + C$. Roots at infinity are set to 10^{27}. The subroutine then calculates $Z(1) = -B + \sqrt{B^2 - 4*A*C}$ and $Z(2) = -B - \sqrt{B^2 - 4*A*C}$. If $T_1 = |Z(1)| = 0$ and $T_2 = |Z(2)| = 0$ special cases must be treated. In that case, if $A = 0$, then f = constant, and the two roots are at "infinity"; a return is made. If $C = 0$ also (which could not happen in CAUCHY, but might arise in some other use of ROOT2), an error stop and message are instituted. If $A = 0$ when $T_1 = T_2 20$, then both roots ZR(1) = ZR(2) = 0.

If none of the special cases occurs, we determine the larger of T1 and T2. The larger Z(1,2) is then used to calculate ZR(1) = 2C/Z(1,2). A check is made to see whether the second root is at "infinity", corresponding to A = 0. If it is, a return occurs, if not the second root is calculated: ZR(2) + Z(1,2)/2A.

D. INCIN

Incin increments the I_0, I_1 and I_2 integrals by forming

$$I_0 = I_0 + FND \cdot \sum_{j=1}^{IR} \Delta I_{0j}, \tag{21.1}$$

$$I_1 = I_1 + FND \cdot \sum_{j=1}^{IR} \Delta I_{2j}, \tag{21.2}$$

and

$$I_2 = I_2 + FND \cdot \sum_{j=1}^{IR} \Delta I_{2j} \tag{21.3}$$

where

$$\Delta I_{0j} = \int_{Z1}^{Z2} \frac{dz}{(z - Z_j)} \tag{22.1}$$

$$\Delta I_{1j} = \int_{Z1}^{Z2} \frac{z\ dz}{(z - Z_j)} \tag{22.2}$$

and

$$\Delta I_{2j} = \int_{Z1}^{Z2} \frac{z^2\ dz}{(z-Z_j)} \tag{22.3}$$

Z1, Z2, FND, IØ, I1, I2 and IR are arguments of INCIN. The Z_j's are obtained from

$$Z_j = ZC + ZR(J) \tag{23}$$

where ZC and ZR(j) are additional arguments of the subroutine.

Starting with equations (22), ΔI_{1j} can be expressed as

$$\Delta I_{1j} = \int_{Z1}^{Z2} dz\ \frac{[(z-Z_j) + Z_j]}{(z - Z_j)}$$

$$= \int_{Z1}^{Z2} dz\ \left[1 + \frac{Z_j}{(z-Z_j)}\right]$$

$$= (Z2 - Z1) + Z_j\ \Delta I_{10}$$

$$= DZ + Z_j\ \Delta I_{10} \tag{24.1}$$

and

$$\Delta I_{2j} = \int_{Z_1}^{Z_2} dz \, \frac{[(z^2 - Z_J^2) + Z_j^2]}{(z - Z_j)}$$

$$= \int_{Z_1}^{Z_j} dz \, [z + Z_j + \frac{Z_j^2}{z - Z_j}]$$

$$= \frac{1}{2} (z_2^{\,2} - Z_1^{\,2}) + Z_j (Z_2 - Z_1)$$

$$+ Z_j^2 \;\; \Delta I_{0j}$$

$$= DZ \cdot Z_B + Z_j \cdot DZ + Z_j^2 \; \Delta I_{0j}$$

$$= DZ \cdot Z_B + Z_j \cdot \Delta I_{1j} \tag{24.2}$$

where

$$DZ = Z_2 - Z_1 \tag{25.1}$$

and

$$Z_B = \frac{1}{2} (Z_1 + Z_2). \tag{25.2}$$

So the only difficulty is the evaluation of ΔI_0. If

$$z = \frac{1}{2} DZ \cdot x + Z_B$$

is substituted into equation (2.1) it is seen that

$$\Delta I_{0j} = \int_{-1}^{1} \frac{dx}{x - \frac{2(Z_j - Z_B)}{DZ}} \tag{26}$$

or

$$\Delta I_{0_j} \equiv \int_{-1}^{1} \frac{dx}{x - a - i\,b} \tag{27}$$

where

$$a + i\,b = \frac{2\,(Z_j - Z_B)}{DZ} \tag{28}$$

and

a and b are real.

This can then be expressed as

$$\Delta I_{0j} = \int_{-1}^{1} \frac{dx\,(x - a + i\,b)}{(x-a)^2 + b^2}$$

$$\frac{1}{2}\,\ln\left[(x-a)^2 + b^2\right]_{-1}^{1} + ib\int_{-1}^{1} \frac{dx}{(x-a)^2+b^2} \tag{29}$$

$$= \frac{1}{2}\,\ln\left[\frac{(1-a)^2 + b^2}{(1+a)^2+b^2}\right] + i\,b\int_{-1}^{1} \frac{dx}{(x-a)^2+b^2}\ ,$$

and

$$\mathrm{Im}\ \Delta\ I_{0j} = b\int_{-1}^{1} \frac{dx}{(x-a)^2+b^2}\ . \tag{30}$$

Now make the substitution

$$x = a + |b|\ \tan\varphi \tag{31}$$

with

$$\tan\varphi_{-1} = -\frac{(1+a)}{|b|}\ ,$$

$$\tan\varphi_1 = \frac{1-a}{|b|}\ ,$$

and

$$|\varphi| \le \frac{\pi}{2}\ , \tag{32}$$

into equation (30).

It is now seen that

$$\mathrm{Im}\ \Delta I_{0j} = \frac{b}{|b|}\int_{\varphi_{-1}}^{\varphi_1} d\varphi$$

$$= \frac{b}{|b|}\ (\varphi_1 - \varphi_{-1})\ . \tag{33}$$

Since $-1 < 1$ it follows from equation (31) and (32) that

$$0 \le \varphi_1 - \varphi_{-1} \le \pi$$

hence

$$|\mathrm{Im}\ \Delta\ I_{0j}| \le \pi\ . \tag{34}$$

Starting again with equation (26) it follows that

$$\begin{aligned}
\Delta I_{0j} &= \int_{-1}^{1} \frac{dx}{x - \frac{2}{DZ}(Z_j - Z_B)} &, \\
&= \ln\left(x - \frac{2}{DZ}(Z_j - Z_B)\right]_{-1}^{1} &, \\
&= \ln\left(\frac{1 - \frac{2}{DZ}(A_j - Z_B)}{-1 - \frac{2}{DZ}(Z_j - Z_B)}\right) &, \\
&= \ln\left(\frac{\frac{2}{DZ}(Z_j - Z_B) - 1}{\frac{2}{DZ}\ Z_j - Z_B) + 1}\right) &,
\end{aligned} \tag{35}$$

with the branch specified by equation (34).

If

$$\left|\frac{2(Z_j - Z_B)}{DZ}\right| \le 1$$

define

$$R = \frac{2(Z_j - Z_B)}{DZ} \tag{36}$$

and then

$$\Delta\ I_{0j} = \ln\left(\frac{1-R}{1+R}\right) + \ln(-1),$$

and

$$\Delta\ I_{0j} = \ln\left(\frac{1-R}{1+R}\right) - i\pi\ . \tag{37}$$

It makes no difference which branch of $\ln(-1)$ is used here since the imaginary part of $\Delta\ I_{0j}$ is adjusted to satisfy equation (34) by adding or subtracting $2\pi\ i$ to the final result.

If

$$1 \leq \left| \frac{2\,(Z_j - Z_B)}{DZ} \right|$$

then R is defined as

$$R = \frac{DZ}{2(Z_j - Z_B)} \tag{38}$$

and

$$\Delta\, I_{0j} = \ln\left(\frac{\frac{1}{R} - 1}{\frac{1}{R} + 1}\right)$$

$$= \ln\left(\frac{1 - R}{1 + R}\right) \tag{39}$$

This is the same as equation (37) without the $- i\,\pi$ term.

The program sets R according to the value of $\left|\frac{(Z_j - Z_B}{DZ}\right|$ to either the R.H.S. of equation (36) or(38) and $\Delta\, I_{0j}$ to $- i\pi$ or 0. Then in either case the fortran statement

$$\Delta\, I_{0j} = \Delta\, I_{0j} + \text{CDLOG}\left(\frac{1 - R}{1 + R}\right)$$

produces $\Delta\, I_{0j}$. If R is less than .1 an expansion of the $\ln\left(\frac{1 - R}{1 + R}\right)$ is used otherwise a fortran supplied subrouting computes the value. The imaginary part of $\Delta\, I_{0j}$ is adjusted to satisfy equation (34) by adding or subtracting $2\,\pi i$. Then ΔI_{1j} and ΔI_{2j} are computed from equations (24) and the integrals are incremented.

IV. SAMPLE RESULTS

In Table 1, we show results for various functions when integrated around a square in the complex Z-plane

$$-2 \leq \mathrm{Re},\ \mathrm{Im}(Z) \leq 2$$

While the function does not provide an exhaustive examination, it does illustrate several important points

1) There is a dramatic change in the winding number (IO) as a zero or pole crosses a contour. In all examples, this zero or pole was chosen to lie halfway between two grid points by choosing an appropriate y.
2) The error in any two runs should decrease as $\left(\frac{|\Delta Z|\ \text{Run B}}{|\Delta Z|\ \text{Run A}}\right)^4$; for the larger errors, this is borne out in practice.
3) Even a crude number of points, with zeros (or poles) right on the contour yield an interpretable IO.

In Table 1, the first column shows the function calculated. Results are listed for three different ΔZ. Below the calculated result is the correct result for comparison.

Table 1A

	$\lvert\Delta Z\rvert = 1$		
f(z)	I0	I1	I2
$1 - z$	1.0000+0.0000i 1+0i	1.0000+0.0000i 1+0i	1.000+0.000i 1+0i
$\sin(z/4)$	1.0001+0.0000i 1+0i	-0.0000+0.0000i 0+0i	-0.000+0.000i 0+0i
$\frac{\sin(z/4)}{z - 1}$	-0.0002-0.0000i 0+0i	-1.0003+0.0000i -1+0i	-1.005+0.000i -1+0i
$\frac{\sin\left[\frac{z-1.99-iy}{4}\right]}{z - 1}$	-0.0571+0.0048i 0+0i	0.8695+0.4814i .99+.5i	2.468+1.892i 2.710+1.99i
$\frac{\sin\left[\frac{z-2.01-iy}{4}\right]}{z - 1}$	-0.8860-0.0608i -1+0i	-0.7420-0.0659i -1+0i	-0.457-0.005i -1+0i
$\frac{\sin\left[\frac{z - 3}{4}\right]}{z - 1}$	-1.0020+0.0000i -1+0i	-1.0026-0.0000i -1+0i	-1.004+0.000i -1+0i
$\frac{\sin(z/4)}{z - 1.99 - iy}$	-0.0329+0.0041i 0+0i	-2.0586-0.5081i -1.99-.5i	-3.850-2.041i -3.71-1.99i
$\frac{\sin(z/4)}{z - 2.01 - iy}$	1.0332-0.0037i 1+0i	0.0675+0.0091i 0+0i	0.124+0.052i 0+0i
$\frac{\sin(z/4)}{(z - 1)(z + 1)}$	-1.0009+0.0000i -1+0i	0.0000-0.0000i 0+0i	-1.985+0.000i -2+0i
$\frac{\sin(z/4)}{(z - 1)^2}$	-1.0002+0.0000i -1+0i	-1.9992+0.0000i -2+0i	-1.979-0.000i -2+0i

Table 1B

f(z)	$\lvert\Delta z\rvert$ .5		
	I0	I1	I2
$1 - z$	1.000+0.000i 1 + 0i	1.0000+0.0000i 1 + 0i	1.000+0.000i 1 + 0i
$\sin(z/4)$	1.0000+0.0000i 1 + 0i	-0.0000+0.0000i 0 + 0i	-0.0000+0.000 0 + 0i
$\frac{\sin(z/4)}{-1}$	-0.0000+0.000i 0 + 0i	-1.0000+0.0000i -1 + 0i	-1.000+0.000i -1 + 0i
$\frac{\sin\left[\frac{z-1.99-iy}{4}\right]}{z-1}$	-00052+0.0010i 0 + 0i	0.9793+0.2508i .99 + .25i	2.875 + 0.994i 2.898 + 0.995i
$\frac{\sin\left[\frac{z-2.01-iy}{4}\right]}{z-1}$	-0.9941-0.0012i -1 + 0i	-0.9878-0.0011i -1 + 0i	-0.976+0.001i -1 + 0i
$\frac{\sin\left[\frac{z-3}{4}\right]}{z-1}$	-1.001+0.0000i -1 + 0i	-1.0002-0.0000i -1 + 0i	-1.000-0.000 -1 + 0i
$\frac{\sin(z/4)}{z-1.99-iy}$	-0.0011+0.003i 0 + 0i	-1.9923-0.2498i -1.99 - 0.25i	-3.903-0995i -3.898-0.995i
$\frac{\sin(z/4)}{z-2.01-iy}$	1.0012-0.0003i 1 + 0i	0.0025-0.0003i 0 + 0i	0.005+0.000i 0 + 0i
$\frac{\sin(z/4)}{(z-1)(z+1)}$	-1.0002-0.0000i - 1 + 0i	0.0000-0.0000i 0 + 0i	-1.999-0.000i -2 + 0
$\frac{\sin(z/4)}{(z-1)^2}$	-1.0001+0.0000i - 1 + 0i	-2.0001-0.0000i -2 + 0i	-1.998-0.000i - 2 +0i

Table 1C

f(z)	$\lvert\Delta z\rvert$ .25		
	I0	I1	I2
$1 - z$	1 + 0i	1 + 0i	1 + 0i
	1 + 0i	1 + 0i	1 + 0i
$\sin(z/4)$	1 + 0i	0 + 0i	0 + 0i
	1 + 0i	0 + 0i	0 + 0i
$\frac{\sin(z/4)}{z - 1}$	0 + 0i	-1 + 0i	-1 + 0i
	0 + 0i	-1 + 0i	-1 + 0i
$\frac{\sin\left[\frac{z-1.99-iy}{4}\right]}{z - 1}$	-0.0003+0.0001i	0.9894+0.1251i	2.943+0.498i
	0 + 0i	.99 + .125i	2.944+0.498i
$\frac{\sin\left[\frac{z-2.01-iy}{4}\right]}{z - 1}$	0.9996-0.0001i	-0.9993-0.0001i	-0.999-0.000i
	-1 + 0i	-1 + 0i	-1 + 0i
$\frac{\sin\left[\frac{z-3}{4}\right]}{z - 1}$	-1 + 0i	-1 + 0i	-1 + 0i
	-1 + 0i	-1 + 0i	-1 + 0i
$\frac{\sin(z/4)}{z-1.99-iy}$	0.0000+0.0000i	-1.99901-0.1250i	-3.945-0.498i
	0 + 0i	1.99-0.25i	-3.944-0.498i
$\frac{\sin(z/4)}{z-2.01-iy}$	1 + 0i	0 + 0i	0 + 0i
	1 + 0i	0 + 0i	0 + 0i
$\frac{\sin(z/4)}{(z-1)(z+1)}$	1 + 0i	0 + 0i	0 + 0i
	1 + 0i	0 + 0i	0 + 0i
$\frac{\sin(z/4)}{(z-1)^2}$	-1 + 0i	-2 + 0i	-2 + 0i
	-1 + 0i	-2 + 0i	-2 + 0i

V. PROGRAM LISTING

```
LEVEL  21.6  (DEC 72)                    OS/360  FORTRAN H                    DATE  74.060/12.21.46

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOLDIT,NOID,XREF
                C
  ISN 0002            SUBROUTINE CAUCHY(IMAX,Z,F,I0,I1,I2)
                C
                C     CAUCHY CALCULATES THE INTEGRALS I0=INT(D(LN(F))/DZ)/2*PI*I,
                C     I1=INT(Z*D(LN(F))/DZ)/2*PI*I, AND I2=INT(Z**2*D(LN(F))/DZ)/2*PI*I
                C     WHERE F IS DEFINED AT IMAX POINTS ON THE CONTOUR Z.  IN COEF
                C     THE CURVE IS FIRST FIT TO A FORM (A*X**2+B*X+C)/(D*X**2+E*X+F),
                C     THE ROOTS OF THE NUMERATOR AND DENOMINATOR ARE OBTAINED (ROOT2),
                C     AND THEN THE INTEGRAL IS PERFORMED (INCIN)
                C
  ISN 0003            COMPLEX*16 ZD(5),FD(5),CD(6),ZRN(2),ZRD(2),Z1,Z2,I0D,I1D,I2D
  ISN 0004            COMPLEX*8 Z(101),F(101),I0,I1,I2
  ISN 0005            REAL*4 ZSC(101)
                C
  ISN 0006            I0D=(0.D0,0.D0)
  ISN 0007            I1D=(0.D0,0.D0)
  ISN 0008            I2D=(0.D0,0.D0)
  ISN 0009            CD(I)=(1.D15,0.D0)
  ISN 0010            DO 110 I=2,IMAX
  ISN 0011            ZSC(I)=CABS(Z(I)-Z(I-1))
  ISN 0012            IF(ZSC(I).NE.0.) GO TO 110
  ISN 0014            WRITE (1,501)
  ISN 0015        501 FORMAT (19H ERROR IN Z-CONTOUR)
  ISN 0016            RETURN
  ISN 0017        110 CONTINUE
  ISN 0018            IF(IMAX.GE.5) GO TO 120
  ISN 0020            WRITE (1,502)
  ISN 0021        502 FORMAT (10H IMAX.LT.5)
  ISN 0022            RETURN
  ISN 0023        120 IMAXM1=IMAX-1
  ISN 0024            DO 160 I=1,IMAXM1
  ISN 0025            DO 130 J=1,5
  ISN 0026            JJ=I-3+J
  ISN 0027            IF(JJ.EQ.-1) JJ=IMAX-2
  ISN 0029            IF(JJ.EQ.0) JJ=IMAX-1
  ISN 0031            IF(JJ.GT.IMAX) JJ=2
  ISN 0033            ZD(J)=Z(JJ)
  ISN 0034        130 FD(J)=F(JJ)
  ISN 0035            CALL COEF(ZD,FD,CD)
  ISN 0036            CALL ROOT2(CD(1),CD(2),CD(3),ZRN)
  ISN 0037            CALL ROOT2(CD(4),CD(5),CD(6),ZRD)
  ISN 0038            IP=I+1
  ISN 0039            IM=I-1
  ISN 0040            IF(IM.EQ.0) IM=IMAX-1
  ISN 0042            IF(CABS(Z(IP)+Z(IM)-2.*Z(I)).LT.ZSC(IP)*1.E-6) GO TO 140
  ISN 0044            Z1=(Z(I)+Z(IM))*0.5
  ISN 0045            Z2=Z(1)
  ISN 0046            CALL INCIN(Z1,Z2,ZD(3),ZRN,2,I0D,I1D,I2D,1.D0)
  ISN 0047            CALL INCIN(Z1,Z2,ZD(3),ZRD,2,I0D,I1D,I2D,-1.D0)
  ISN 0048            Z1=Z(I)
  ISN 0049            Z2=(Z(IP)+Z(I))*0.5
  ISN 0050            GO TO 150
  ISN 0051        140 Z1=(Z(I)+Z(IM))*0.5
  ISN 0052            Z2=(Z(IP)+Z(I))*0.5
  ISN 0053        150 CALL INCIN(Z1,Z2,ZD(3),ZRN,2,I0D,I1D,I2D,1.D0)
  ISN 0054            CALL INCIN(Z1,Z2,ZD(3),ZRD,2,I0D,I1D,I2D,-1.D0)
```

PAGE 002

```
  ISN 0055        160 CONTINUE
  ISN 0056            I0=I0D/(0.D0,6.283185300)
  ISN 0057            I1=I1D/(0.D0,6.283185300)
  ISN 0058            I2=I2D/(0.D0,6.283185300)
  ISN 0059            RETURN
  ISN 0060            END
```

```
LEVEL  21.6  (DEC 72)                          OS/360  FORTRAN H                                    DATE  74.112/11.47.55
                                                                                                    Version 22 Apr 74
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,NOID,NOXREF
  ISN 0002              SUBROUTINE INCIN(Z1,Z2,ZC,ZR,IR,I0,I1,I2,FND)
  ISN 0003              IMPLICIT COMPLEX*16 (A-H,O-Z)
                C       INCIN ADDS THE INCREMENTAL INTEGRALS FROM Z1 TO Z2 TO THE
                C       INTEGRALS I0, I1, AND I2.
                C       I-SUB-N = I-SUB-N + FND * SUM OVER J,J=1,IR OF INT(J), WHERE
                C       INT(J) = INTEGRAL FROM Z1 TO Z2 OF (DZ*(Z**N)/(Z-ZC-ZR(J))
  ISN 0004              REAL*8 AR,PI,TWOPI,FND,TEST
  ISN 0005              COMPLEX*16 I0,I1,I2,ZR(1)
  ISN 0006              DATA PI/3.14159265358979D0/,TWOPI/6.28318530717958D0/
  ISN 0007              DZ=Z2-Z1
  ISN 0008              ZB=.5D0*(Z2+Z1)
  ISN 0009              DO 50 J=1,IR
  ISN 0010              ZJ=ZR(J)+ZC
  ISN 0011              ZRB=2.D0*(ZJ-ZB)
  ISN 0012              R=ZRB/DZ
  ISN 0013              AR=CDABS(R)
  ISN 0014              IF(AR.GT.1.D0) GO TO 10
  ISN 0016              DI=DCMPLX(0.D0,-PI)
  ISN 0017              GO TO 20
  ISN 0018         10   R=DZ/ZRB
  ISN 0019              AR=1.D0/AR
  ISN 0020              DI=(0.D0,0.D0)
  ISN 0021         20   IF(AR.GT. .1D0) GO TO 30
  ISN 0023              IF(AR.LT.1.D-25) GO TO 50
  ISN 0025              RS=R*R
  ISN 0026              DI=DI-R*(2.D0+(.66666666666666667D0+(.4D0+(.28571428571428570D0
                       1 +(.22222222222222222D0+.18181818181818181818D0*RS)*RS)*RS)*RS)*RS)
  ISN 0027              TEST=DIMAG(DI)
  ISN 0028              IF(TEST.LT.-PI) DI=DI+DCMPLX(0.D0,TWOPI)
  ISN 0030              GO TO 40
  ISN 0031         30   DI=DI+CDLOG((1.D0-R)/(1.D0+R))
  ISN 0032              TEST=DIMAG(DI)
  ISN 0033              IF(TEST.GT.PI) DI=DI-DCMPLX(0.D0,TWOPI)
  ISN 0035              IF(TEST.LT.-PI) DI=DI+DCMPLX(0.D0,TWOPI)
  ISN 0037         40   I0=I0+FND*DI
  ISN 0038              DI=DZ+ZJ*DI
  ISN 0039              I1=I1+FND*DI
  ISN 0040              I2=I2+FND*(DZ*ZB+ZJ*DI)
  ISN 0041         50   CONTINUE
  ISN 0042              RETURN
  ISN 0043              END
```

```
LEVEL  21.6  (DEC 72)                          OS/360  FORTRAN H                                    DATE  74.060/12.22.03

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,NOID,XREF
                C
  ISN 0002              SUBROUTINE ROOT2(C,B,A,ZR)
                C
                C       FINDS ROOTS OF A*Z**2+B*Z+C
                C
  ISN 0003              COMPLEX*16 A,B,C,ZR(2),Z(2),RAD
  ISN 0004              REAL*4 T1,T2,TOVFL
                C
                C       INFINITY = 1.D27
                C
  ISN 0005              ZR(1)=(1.D27,0.D0)
  ISN 0006              ZR(2)=(1.D27,0.D0)
  ISN 0007              RAD=CDSQRT(B*B-4*A*C)
  ISN 0008              Z(1)=-B+RAD
  ISN 0009              Z(2)=-B-RAD
  ISN 0010              T1=CDABS(Z(1))
  ISN 0011              T2=CDABS(Z(2))
  ISN 0012              IF(T1+T2.NE.0.) GO TO 30
  ISN 0014              IF(CDABS(A).NE.0.) GO TO 20
  ISN 0016              IF(CDABS(C).NE.0.) GO TO 10
  ISN 0018              WRITE (51,501)
  ISN 0019          501 FORMAT (' ERROR STOP  -  A = B = C = 0')
  ISN 0020              STOP
  ISN 0021           10 RETURN
  ISN 0022           20 ZR(1)=(0.D0,0.D0)
  ISN 0023              ZR(2)=(0.D0,0.D0)
  ISN 0024              RETURN
  ISN 0025           30 I=1
  ISN 0026              IF(T2.GT.T1) I=2
  ISN 0028              ZR(1)=(C+C)/Z(I)
  ISN 0029              TOVFL=CDABS(1.D-27*Z(I))-CDABS(A+A)
  ISN 0030              IF(TOVFL.GT.0.) RETURN
  ISN 0032              ZR(2)=Z(I)/(A+A)
  ISN 0033              RETURN
  ISN 0034              END
```

```
LEVEL  21.6  (DEC 72)                    CS/360  FORTRAN H                    DATE  74.080/15.21.30
                                                                              Version 21 Mar 74
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,NOID,XREF
                C
ISN 0002                SUBROUTINE COEF(Z,F,C)
                C
                C       PROGRAM TO FIND COEFFICIENTS FOR THE FUNCTION
                C       F=(C(1)+C(2)*Z+C(3)*Z**2)/(C(4)+C(5)*Z+C(6)*Z**2)
                C       GIVEN THE FUNCTION F AT FIVE POINTS Z.
                C
ISN 0003                IMPLICIT COMPLEX*16 (A-H,C-Z)
ISN 0004                DIMENSION Z(5),F(5),C(6),FF(5)
ISN 0005                REAL*8 ZERO/1.D-10/,SCF,FM(5),TSCF(5),TSCFM
                C
                C       DETERMINE SCALE FACTOR (SCF) FOR F SO F IS OF ORDER ONE
                C
ISN 0006                IF(CDABS(C(1)).NE.CDABS((1.D15,0.D0))) GO TO 20
ISN 0008                DO 10 I=1,5
ISN 0009             10 FM(I)=CDABS(F(I))
ISN 0010                GO TO 40
ISN 0011             20 DO 30 I=1,4
ISN 0012             30 FM(I)=FM(I+1)
ISN 0013                FM(5)=CDABS(F(5))
ISN 0014             40 SCF=DMAX1(FM(1),FM(2),FM(3),FM(4),FM(5))
ISN 0015                IF(SCF.NE.0.D0) SCF=1.D0/SCF
                C
                C       DETERMINE IF FUNCTION IS SINGULAR (TSCFM=0).  IF ITSCF.GT.1,
                C       FUNCTION IS CONSTANT AT TWO OR MORE POINTS.
                C
ISN 0017                ITSCF=0
ISN 0018                DO 50 I=1,5
ISN 0019                TSCF(I)=SCF*FM(I)
ISN 0020                IF(DABS(TSCF(I)-1.D0).GT.ZERO) GO TO 50
ISN 0022                ITSCF=ITSCF+1
ISN 0023                J=I
ISN 0024             50 CONTINUE
ISN 0025                IF(ITSCF.GE.2) GO TO 70
ISN 0027                TSCFM=0.D0
ISN 0028                DO 60 I=1,5
ISN 0029                IF(I.EQ.J) GO TO 60
ISN 0031                IF(TSCF(I).GT.TSCFM) TSCFM=TSCF(I)
ISN 0033             60 CONTINUE
ISN 0034                IF(TSCFM.GT.ZERO) GO TO 70
ISN 0036                WRITE (51,501) (F(I),I=1,5)
ISN 0037            501 FORMAT (' ERROR STOP  -  SINGULAR FUNCTION0',/1P10E12.4)
ISN 0038                STOP
ISN 0039             70 DO 80 I=1,5
ISN 0040             80 FF(I)=SCF*F(I)
                C
ISN 0041                ZN2=Z(1)-Z(3)
ISN 0042                ZN1=Z(2)-Z(3)
ISN 0043                Z1=Z(4)-Z(3)
ISN 0044                Z2=Z(5)-Z(3)
                C
ISN 0045                GN2=(FF(1)-FF(3))/ZN2
ISN 0046                GN1=(FF(2)-FF(3))/ZN1
ISN 0047                G1=(FF(4)-FF(3))/Z1
ISN 0048                G2=(FF(5)-FF(3))/Z2
                C
```

PAGE 002

```
ISN 0049                ZDENOM=1.D0/(Z1-ZN1)
ISN 0050                F1M1=FF(4)-FF(2)
ISN 0051                ZF1=Z1*FF(4)
ISN 0052                ZF1M1=ZF1-ZN1*FF(2)
ISN 0053                G1M1=G1-GN1
ISN 0054                RAT=(ZN2-Z1)*ZDENOM
ISN 0055                AN2=RAT*F1M1-FF(1)+FF(4)
ISN 0056                BN2=RAT*ZF1M1-ZN2*FF(1)+ZF1
ISN 0057                CN2=GN2-G1-RAT*G1M1
ISN 0058                RAT=(Z2-Z1)*ZDENOM
ISN 0059                A2=RAT*F1M1-FF(5)+FF(4)
ISN 0060                B2=RAT*ZF1M1-Z2*FF(5)+ZF1
ISN 0061                C2=G2-G1-RAT*G1M1
                C
ISN 0062                C(4)=A2*BN2-AN2*B2
ISN 0063                C(5)=C2*BN2-CN2*B2
ISN 0064                C(6)=A2*CN2-AN2*C2
                C
                C       TEST FOR CASES WHERE DENOMINATOR COEFFICIENTS VANISH.
                C
ISN 0065                IF(CDABS(C(4)).GT.ZERO) GO TO 90
ISN 0067                C(4)=A2
ISN 0068                C(5)=C2
ISN 0069                C(6)=(0.D0,0.D0)
ISN 0070                IF(CDABS(C(5)).LE.ZERO) C(4)=(1.D0,0.D0)
ISN 0072             90 C(3)=ZDENOM*(G1M1*C(4)+F1M1*C(5)+ZF1M1*C(6))
ISN 0073                C(2)=-Z1*C(3)+G1*C(4)+FF(4)*(C(5)+Z1*C(6))
ISN 0074                C(1)=FF(3)*C(4)
ISN 0075                RETURN
ISN 0076                END
```

REFERENCES

1. J. E. McCune, Phys.Fluids 9, 2082 (1966).

2. Applications Program Series AP-26, Mass. Inst. of Tech. Information Processing Center, March 28, 1973.