DECUS PROGRAM LIBRARY

DECUS NO.	FOCAL8-232
TITLE	ROOTS BY INVERSE INTERPOLATION
AUTHOR	H. Bradford Thompson
COMPANY	University of Toledo Toledo, Ohio
DATE	May 3, 1972
SOURCE LANGUAGE	FOCAL

ROOTS BY INVERSE INTERPOLATION

DECUS Program Library Write-up DECUS NO. FOCAL8-232

I. INTRODUCTION

Whenever a continuous function y=f(x) has values of y of opposite sign at two points x=a and x=b, then at least one solution of the equation y=0 must lie between these points. Thus one method of seeking roots of a function is to evaluate y at spaced values of x, and when y values of differing sign are found for adjacent x values, to seek out the intervening root. Several search procedures are known, among them the method of inverse interpolation or false-position (regula falsi). The inverse interpolation procedure is usually quite rapid, but for certain functions can converge very slowly. A modified method with improved convergence properties is employed here: details of the method and modification are clearly described in "Introduction to Applied Numerical Analysis" by R. W. Hamming, McGraw-Hill, 1971, pp 45-48.

II. USAGE

The inverse interpolation is performed by a subprogram located in block 2 of FOCAL® instructions in the examples given here. The user provides a master program, a subprogram in block 3 which contains the function to be evaluated, and a subprogram in block 4 which uses each root in any desired manner. The properties required of each of these programs follow.

1. Master Program.

The user's program will normally read in any required parameters, perform initial calculations, then enter the interpolation subprogram via a FOR statement of the form:

```
FOR XX=XØ,XD,XM;DO 2
```

The quantities XX and XØ are used by the interpolation subprogram. XD and XM are not, and may be replaced by other symbols or by numerical values. The master program must also furnish a convergence parameter DX, defining the accuracy to which the argument X is desired.

The master program may, of course, print headings, calculate further results from the roots obtained, and so on.

2. Block 3.

The interpolation program enters this block via a DO 3. This block must calculate a value of Y for the value of X provided. The calculation may consist of as many steps as necessary, and could consist of several lines. However, since this block is performed a number of times, care should be taken to shorten its execution time as much as possible.

During testing and debugging it is useful to include in Block 3 a statement which types the values of Y and X each time the block is executed. In this way, the user can see how rapidly convergence is obtained. This may be particularly helpful in choosing values of DX and XD. Such a TYPE statement has been included in the test program listed below. Of course, this TYPE statement would be removed after the trial period.

3. Block 4.

Upon finding a root to within the precision given by DX, the interpolation program performs a DO 4 with X equal to the value of the root. A DO 4 can result from one of the following conditions:

a. The value of Y was found to be zero to within the precision of the floating point package;

b. Two values of X differing by less than DX were found to give Y values differing in sign.

The user may simply print this root, may test it in various ways, or may store it for use in a later stage of the master calculation.

4. Variables used in Block 2.

The following are used to link Block 2 to the other portions of the program and should be used as described above: X, XX, XØ, DX, and Y.

The following are used internally in Block 2 and should not be used in other sections of the program: X(1), X(2), XT, Y(1), Y(2), YT, DY, and IQ

III. LIMITATIONS

This interpolation scheme is fast and efficient when the value of XD is small enough to isolate each root. It may fail for discontinuous functions, and may converge slowly for functions with sharp localized curvature. The modification of the false-position algorithm used here will usually require many fewer determinations of the function than the bisection method. As a demonstration the function

Y = FEXP(-C*X)-D

has been chosen, using particularly unsatisfactory values of C and D. The standard false-position algorithm fails badly on this function; however, the modification recommended by Hamming still converges more rapidly than a bisection technique.

The rapid convergence in the case of the sine function in the first example below is typical of a normally tractable problem. While the extended precision (Four-Word) version of FOCAL is used here, the roots differ from $\pi/2$, π, and $3\pi/2$ due to the limited accuracy of the FOCAL sine function.

IV. MODIFICATIONS

1. Convergence Test.

The rest fōr convergence is contained in line 2.60. If a relative precision rather than an absolute precision in X is desired, this statement might be charged to:

```
IF ((X(2)-X(1))/FABS(X)-DX)2.95
```

Note that this procedure has dangers if X can become zero. Other tests may be appropriate to specific applications.

2. The Unmodified False-Position Method.

In this program the false-position method is modified by halving one of the Y values used for the inverse interpolation whenever the trial X value has been found to fall twice on the same side of the root sought. This feature can be removed by deleting the first SET command in line 2.80. For tractable functions this may slightly speed the calculation. Normally, the saving, if any, is minimal.

V. LISTING WITH EXAMPLES

The FOCAL listing appended contains a simple master program, the interpolation routine in Block 2, and a printout of the root in Block 4. Two different functions have been inserted in Block 3, and in each case a type statement has been included, so that the search process can be followed in detail.

The examples have been run on the 4-WORD modification of FOCAL. This is by no means required, and is used here merely to help illustrate the convergence properties of the program. Of course, the value of DX must be chosen with the precision of the FOCAL floating point package in mind. It may be noted that while DX was set at 1E-6, the root has commonly been found to 1E-8. This is common with this procedure; the final cycle is often a very nearly linear interpolation, and thus very precise. There is, however, no guarantee that precision will exceed that specified by DX.

```
*W
C-8K4WTV4A @1969

01.10 SET DX=1E-6
01.20 ASK !,?C?,?D?,?K?
01.30 SET X0=0;SET XD=.5
01.40 FOR XX=X0,XD,5;DO 2
01.50 T ! "SEARCH TERMINATED";Q

02.05 I (X0-XX) 2.1; D 2.2; R
02.10 S Y(1)=YT; S X(1)=XT
02.20 S X=XX; D 2.72;S YT=Y;S XT=X
02.30 I (Y(1)*Y) 2.4;R
02.40 S IQ=2
02.50 S X(IQ)=X; S Y(IQ)=Y
02.60 I (X(2)-X(1)-DX)2.95
02.70 S DY=Y(1)-Y(2);S X=(X(2)*Y(1)-X(1)*Y(2))/DY
02.72 D 3; I (Y↑2)2.95,2.95
02.80 S Y(3-IQ)=.5*Y(3-IQ);S IQ=1;I (Y/DY)2.4,2.95,2.5
02.95 D 4;R

03.10 SET Y=FSIN(C*X)*K
03.20 T %,!,X,Y

04.10 T %,!,"ROOT=",X
*GO

C:2 D:0 K:1
 0.0000000000E+00 0.0000000000E+00
ROOT= 0.0000000000E+00
 0.5000000000E+00 0.8414710163E+00
 0.1000000000E+01 0.9092975623E+00
 0.1500000000E+01 0.1411200538E+00
 0.2000000000E+01-0.7568024737E+00
 0.1578581420E+01-0.1556950971E-01
 0.1564376372E+01 0.1283960363E-01
 0.1570796395E+01-0.9057112033E-07
 0.1570796350E+01 0.0000000000E+00
ROOT= 0.1570796350E+01
 0.2500000000E+01-0.9589246993E+00
 0.3000000000E+01-0.2794155875E+00
 0.3500000000E+01 0.6569865319E+00
 0.3149196366E+01 0.1520674594E-01
 0.3134550928E+01-0.1408307822E-01
 0.3141592720E+01 0.3949878870E-07
 0.3141592700E+01-0.9313225738E-09
ROOT= 0.3141592700E+01
 0.4000000000E+01 0.9893594992E+00
 0.4500000000E+01 0.4121186122E+00
 0.5000000000E+01-0.5440209955E+00
 0.4715511735E+01-0.6245329189E-02
 0.4709172063E+01 0.6433930094E-02
 0.4712389049E+01 0.1396983861E-08
 0.4712389051E+01-0.2328306434E-09
ROOT= 0.4712389051E+01
SEARCH TERMINATED*
```

```
 W 3
03.10 S Y=FEXP(-C*X)-D
03.20 T !,X,Y
*GO

C:5 D:.02 K:0
 0.0000000000E+00 0.9800000001E+00
 0.5000000000E+00 0.6208500700E-01
 0.1000000000E+01-0.1326205011E-01
 0.9119935760E+00-0.9537601804E-02
 0.8151621201E+00-0.3021550308E-02
 0.7638064405E+00 0.1949038327E-02
 0.7839437310E+00-0.1533177528E-03
 0.7824751862E+00-0.7052671207E-05
 0.7823410498E+00 0.6360744010E-05
 0.7824046582E+00-0.1120497473E-08
 0.7824046469E+00 0.1637090463E-10
ROOT= 0.7824046469E+00
 0.1500000000E+01-0.1944691533E-01
 0.2000000000E+01-0.1995460003E-01
 0.2500000000E+01-0.1999627335E-01
 0.3000000000E+01-0.1999969410E-01
 0.3500000000E+01-0.1999997489E-01
 0.4000000000E+01-0.1999999794E-01
 0.4500000000E+01-0.1999999983E-01
 0.5000000000E+01-0.1999999999E-01
SEARCH TERMINATED*
*
*GO

C:10 D:.02 K:0
 0.0000000000E+00 0.9800000001E+00
 0.5000000000E+00-0.1326205011E-01
 0.4933239923E+00-0.1279686894E-01
 0.4807682211E+00-0.1183323130E-01
 0.4586174988E+00-0.9808229394E-02
 0.4246194198E+00-0.5681370404E-02
 0.3885762282E+00 0.5321763165E-03
 0.3916632466E+00-0.9197250620E-04
 0.3912083537E+00-0.1205999069E-05
 0.3911964778E+00 0.1169180905E-05
 0.3912023237E+00-0.2910383046E-10
 0.3912023235E+00 0.9094947018E-11
ROOT= 0.3912023235E+00
 0.1000000000E+01-0.1995460003E-01
 0.1500000000E+01-0.1999969410E-01
 0.2000000000E+01-0.1999999794E-01
 0.2500000000E+01-0.1999999999E-01
 0.3000000000E+01-0.2000000000E-01
 0.3500000000E+01-0.2000000000E-01
 0.4000000000E+01-0.2000000000E-01
 0.4500000000E+01-0.2000000000E-01
 0.5000000000E+01-0.2000000000E-01
SEARCH TERMINATED*
```