Non-linear Equations

The table of contents leads us to the zero finding routine FZERO. We will use its D variant whose API (file **dfzero8.f.html**) is shown below:

dfzero.f

```
      SUBROUTINE DFZERO (F, B, C, R, RE, AE, IFLAG)
C***BEGIN PROLOGUE  DFZERO
C***PURPOSE  Search for a zero of a function F(X) in a given interval
C            (B,C).  It is designed primarily for problems where F(B)
C            and F(C) have opposite signs.
C***LIBRARY   SLATEC
C***CATEGORY  F1B
C***TYPE      DOUBLE PRECISION (FZERO-S, DFZERO-D)
C***KEYWORDS  BISECTION, NONLINEAR, ROOTS, ZEROS
C***AUTHOR  Shampine, L. F., (SNLA)
C           Watts, H. A., (SNLA)
C***DESCRIPTION
C
C     DFZERO searches for a zero of a DOUBLE PRECISION function F(X)
C     between the given DOUBLE PRECISION values B and C until the width
C     of the interval (B,C) has collapsed to within a tolerance
C     specified by the stopping criterion,
C        ABS(B-C) .LE. 2.*(RW*ABS(B)+AE).
C     The method used is an efficient combination of bisection and the
C     secant rule and is due to T. J. Dekker.
C
C     Description Of Arguments
C
C   F     :EXT   - Name of the DOUBLE PRECISION external function.
This
C                  name must be in an EXTERNAL statement in the calling
C                  program.  F must be a function of one DOUBLE
C                  PRECISION argument.
C
C   B     :INOUT - One end of the DOUBLE PRECISION interval (B,C).  The
C                  value returned for B usually is the better
C                  approximation to a zero of F.
C
C   C     :INOUT - The other end of the DOUBLE PRECISION interval (B,C)
C
C   R     :IN    - A (better) DOUBLE PRECISION guess of a zero of F
C                  which could help in speeding up convergence.  If
F(B)
C                  and F(R) have opposite signs, a root will be found
in
C                  the interval (B,R);  if not, but F(R) and F(C) have
C                  opposite signs, a root will be found in the interval
C                  (R,C);  otherwise, the interval (B,C) will be
C                  searched for a possible root.  When no better guess
```

```
C                  is known, it is recommended that R be set to B or C,
C                  since if R is not interior to the interval (B,C), it
C                  will be ignored.
C
C   RE    :IN    - Relative error used for RW in the stopping
criterion.
C                  If the requested RE is less than machine precision,
C                  then RW is set to approximately machine precision.
C
C   AE    :IN    - Absolute error used in the stopping criterion.  If
C                  the given interval (B,C) contains the origin, then a
C                  nonzero value should be chosen for AE.
C
C   IFLAG :OUT   - A status code.  User must check IFLAG after each
C                  call.  Control returns to the user from DFZERO in
all
C                  cases.
C
C               1  B is within the requested tolerance of a zero.
C                  The interval (B,C) collapsed to the requested
C                  tolerance, the function changes sign in (B,C), and
C                  F(X) decreased in magnitude as (B,C) collapsed.
C
C               2  F(B) = 0.  However, the interval (B,C) may not have
C                  collapsed to the requested tolerance.
C
C               3  B may be near a singular point of F(X).
C                  The interval (B,C) collapsed to the requested tol-
C                  erance and the function changes sign in (B,C), but
C                  F(X) increased in magnitude as (B,C) collapsed,
i.e.
C                    ABS(F(B out)) .GT. MAX(ABS(F(B in)),ABS(F(C in)))
C
C               4  No change in sign of F(X) was found although the
C                  interval (B,C) collapsed to the requested
tolerance.
C                  The user must examine this case and decide whether
C                  B is near a local minimum of F(X), or B is near a
C                  zero of even multiplicity, or neither of these.
C
C               5  Too many (.GT. 500) function evaluations used.
C
C***REFERENCES  L. F. Shampine and H. A. Watts, FZERO, a root-solving
C                  code, Report SC-TM-70-631, Sandia Laboratories,
C                  September 1970.
C                T. J. Dekker, Finding a zero by means of successive
C                  linear interpolation, Constructive Aspects of the
C                  Fundamental Theorem of Algebra, edited by B. Dejon
C                  and P. Henrici, Wiley-Interscience, 1969.
C***ROUTINES CALLED  D1MACH
C***REVISION HISTORY  (YYMMDD)
C   700901  DATE WRITTEN
C   890531  Changed all specific intrinsics to generic.  (WRB)
C   890531  REVISION DATE from Version 3.2
C   891214  Prologue converted to Version 4.0 format.  (BAB)
C   920501  Reformatted the REFERENCES section.  (WRB)
C***END PROLOGUE  DFZERO
```

As an example, let us write a program to compute the intersection of the graph of y = x with that of y = cos(x); i.e. the root of the equation:

```
    y = x - cos(x)
```

```
program zero
implicit none
real*8 from, upto, guess, EPS
integer*2 status
parameter (EPS = 1.E-8)
external myFun

print*, "Enter from/to range for zero search ..."
read*, from, upto
guess = from

call dfZero(myFun, from, upto, guess, EPS, EPS, status)

print*, from, upto, status
end


real*8 function myFun(x)
implicit none
real*8 x
myFun = x - cos(x)
end
```

Running the above program yields:

```
Enter from/to range for zero search ...
0
2
0.739085137  0.73908512 1
```