NAG Library Routine Document

E01BEF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

E01BEF computes a monotonicity-preserving piecewise cubic Hermite interpolant to a set of data points.

2 Specification

```
SUBROUTINE E01BEF (N, X, F, D, IFAIL)

INTEGER            N, IFAIL
REAL (KIND=nag_wp) X(N), F(N), D(N)
```

3 Description

E01BEF estimates first derivatives at the set of data points (x_r, f_r), for $r = 1, 2, \ldots, n$, which determine a piecewise cubic Hermite interpolant to the data, that preserves monotonicity over ranges where the data points are monotonic. If the data points are only piecewise monotonic, the interpolant will have an extremum at each point where monotonicity switches direction. The estimates of the derivatives are computed by a formula due to Brodlie, which is described in Fritsch and Butland (1984), with suitable changes at the boundary points.

The routine is derived from routine PCHIM in Fritsch (1982).

Values of the computed interpolant, and of its first derivative and definite integral, can subsequently be computed by calling E01BFF, E01BGF and E01BHF, as described in Section 9.

4 References

Fritsch F N (1982) PCHIP final specifications *Report UCID-30194* Lawrence Livermore National Laboratory

Fritsch F N and Butland J (1984) A method for constructing local monotone piecewise cubic interpolants *SIAM J. Sci. Statist. Comput.* **5** 300–304

5 Arguments

1: N – INTEGER *Input*

On entry: n, the number of data points.

Constraint: $\mathrm{N} \geq 2$.

2: X(N) – REAL (KIND=nag_wp) array *Input*

On entry: $\mathrm{X}(r)$ must be set to x_r, the rth value of the independent variable (abscissa), for $r = 1, 2, \ldots, n$.

Constraint: $\mathrm{X}(r) < \mathrm{X}(r+1)$.

3: F(N) – REAL (KIND=nag_wp) array *Input*

On entry: $\mathrm{F}(r)$ must be set to f_r, the rth value of the dependent variable (ordinate), for $r = 1, 2, \ldots, n$.

4: D(N) – REAL (KIND=nag_wp) array *Output*

On exit: estimates of derivatives at the data points. D(r) contains the derivative at X(r).

5: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, N < 2.

IFAIL $= 2$

The values of X(r), for $r = 1, 2, \ldots,$N, are not in strictly increasing order.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

The computational errors in the array D should be negligible in most practical situations.

8 Parallelism and Performance

E01BEF is not threaded in any implementation.

9 Further Comments

The time taken by E01BEF is approximately proportional to n.

The values of the computed interpolant at the points PX(i), for $i = 1, 2, \ldots,$M, may be obtained in the real array PF, of length at least M, by the call:

```
CALL E01BFF(N,X,F,D,M,PX,PF,IFAIL)
```

where N, X and F are the input arguments to E01BEF and D is the output argument from E01BEF.

The values of the computed interpolant at the points PX(i), for $i = 1, 2, \ldots, \mathrm{M}$, together with its first derivatives, may be obtained in the real arrays PF and PD, both of length at least M, by the call:

```
CALL E01BGF(N,X,F,D,M,PX,PF,PD,IFAIL)
```

where N, X, F and D are as described above.

The value of the definite integral of the interpolant over the interval A to B can be obtained in the real variable PINT by the call:

```
CALL E01BHF(N,X,F,D,A,B,PINT,IFAIL)
```

where N, X, F and D are as described above.

10 Example

This example reads in a set of data points, calls E01BEF to compute a piecewise monotonic interpolant, and then calls E01BFF to evaluate the interpolant at equally spaced points.

10.1 Program Text

```
    Program e01befe

!     E01BEF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: e01bef, e01bff, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: step
      Integer                          :: i, ifail, m, n, r
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: d(:), f(:), pf(:), px(:), x(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: min, real
!     .. Executable Statements ..
      Write (nout,*) 'E01BEF Example Program Results'

!     Skip heading in data file
      Read (nin,*)

      Read (nin,*) n
      Allocate (d(n),f(n),x(n))

      Do r = 1, n
        Read (nin,*) x(r), f(r)
      End Do

      ifail = 0
      Call e01bef(n,x,f,d,ifail)

      Read (nin,*) m
      Allocate (pf(m),px(m))

!     Compute M equally spaced points from X(1) to X(N).

      step = (x(n)-x(1))/real(m-1,kind=nag_wp)

      Do i = 1, m
        px(i) = min(x(1)+real(i-1,kind=nag_wp)*step,x(n))
      End Do
      ifail = 0
      Call e01bff(n,x,f,d,m,px,pf,ifail)

      Write (nout,*)
      Write (nout,*) '                Interpolated'
      Write (nout,*) '     Abscissa          Value'

      Do i = 1, m
        Write (nout,99999) px(i), pf(i)
      End Do

99999 Format (1X,F13.4,2X,F13.4)
    End Program e01befe
```

10.2 Program Data

```
E01BEF Example Program Data
   9                 N, the number of data points
  7.99   0.00000E+0  X(R), F(R), independent and dependent variable
  8.09   0.27643E-4
  8.19   0.43750E-1
  8.70   0.16918E+0
  9.20   0.46943E+0
 10.00   0.94374E+0
 12.00   0.99864E+0
 15.00   0.99992E+0
 20.00   0.99999E+0  End of data points
  11                 M, the number of evaluation points
```

10.3 Program Results

```
 E01BEF Example Program Results

                Interpolated
      Abscissa         Value
        7.9900        0.0000
        9.1910        0.4640
       10.3920        0.9645
       11.5930        0.9965
       12.7940        0.9992
       13.9950        0.9998
       15.1960        0.9999
       16.3970        1.0000
       17.5980        1.0000
       18.7990        1.0000
       20.0000        1.0000
```