NAG Library Routine Document

E01BGF

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

E01BGF evaluates a piecewise cubic Hermite interpolant and its first derivative at a set of points.

2 Specification

```
SUBROUTINE E01BGF (N, X, F, D, M, PX, PF, PD, IFAIL)
INTEGER                 N, M, IFAIL
REAL (KIND=nag_wp) X(N), F(N), D(N), PX(M), PF(M), PD(M)
```

3 Description

E01BGF evaluates a piecewise cubic Hermite interpolant, as computed by E01BEF, at the points PX(i), for $i = 1, 2, \ldots, m$. The first derivatives at the points are also computed. If any point lies outside the interval from X(1) to X(N), values of the interpolant and its derivative are extrapolated from the nearest extreme cubic, and a warning is returned.

If values of the interpolant only, and not of its derivative, are required, E01BFF should be used.

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

4 References

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

5 Arguments

1: N – INTEGER *Input*

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

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

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

On entry: N, X, F and D must be unchanged from the previous call of E01BEF.

5: M – INTEGER *Input*

On entry: m, the number of points at which the interpolant is to be evaluated.

Constraint: M ≥ 1.

6: PX(M) – REAL (KIND=nag_wp) array *Input*

On entry: the m values of x at which the interpolant is to be evaluated.

7: PF(M) – REAL (KIND=nag_wp) array *Output*

On exit: PF(i) contains the value of the interpolant evaluated at the point PX(i), for $i = 1, 2, \ldots, m$.

8: PD(M) – REAL (KIND=nag_wp) array *Output*

On exit: PD(i) contains the first derivative of the interpolant evaluated at the point PX(i), for $i = 1, 2, \ldots, m$.

9: 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 $= 3$

On entry, M < 1.

IFAIL $= 4$

At least one of the points PX(i), for $i = 1, 2, \ldots,$ M, lies outside the interval [X(1), X(N)], and extrapolation was performed at all such points. Values computed at these points may be very unreliable.

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 arrays PF and PD should be negligible in most practical situations.

8 Parallelism and Performance

E01BGF is not threaded in any implementation.

9 Further Comments

The time taken by E01BGF is approximately proportional to the number of evaluation points, m. The evaluation will be most efficient if the elements of PX are in nondecreasing order (or, more generally, if they are grouped in increasing order of the intervals $[\mathrm{X}(r-1), \mathrm{X}(r)]$). A single call of E01BGF with $m > 1$ is more efficient than several calls with $m = 1$.

10 Example

This example reads in values of N, X, F and D, and calls E01BGF to compute the values of the interpolant and its derivative at equally spaced points.

10.1 Program Text

```
    Program e01bgfe

!     E01BGF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: e01bgf, 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(:), pd(:), pf(:), px(:),     &
                                          x(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: min, real
!     .. Executable Statements ..
      Write (nout,*) 'E01BGF 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), d(r)
      End Do

      Read (nin,*) m
      Allocate (pd(m),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 e01bgf(n,x,f,d,m,px,pf,pd,ifail)

      Write (nout,*)
      Write (nout,*) '                 Interpolated   Interpolated'
```

```
      Write (nout,*) '       Abscissa          Value     Derivative'

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

99999 Format (1X,2F15.4,1P,E15.3)
    End Program e01bgfe
```

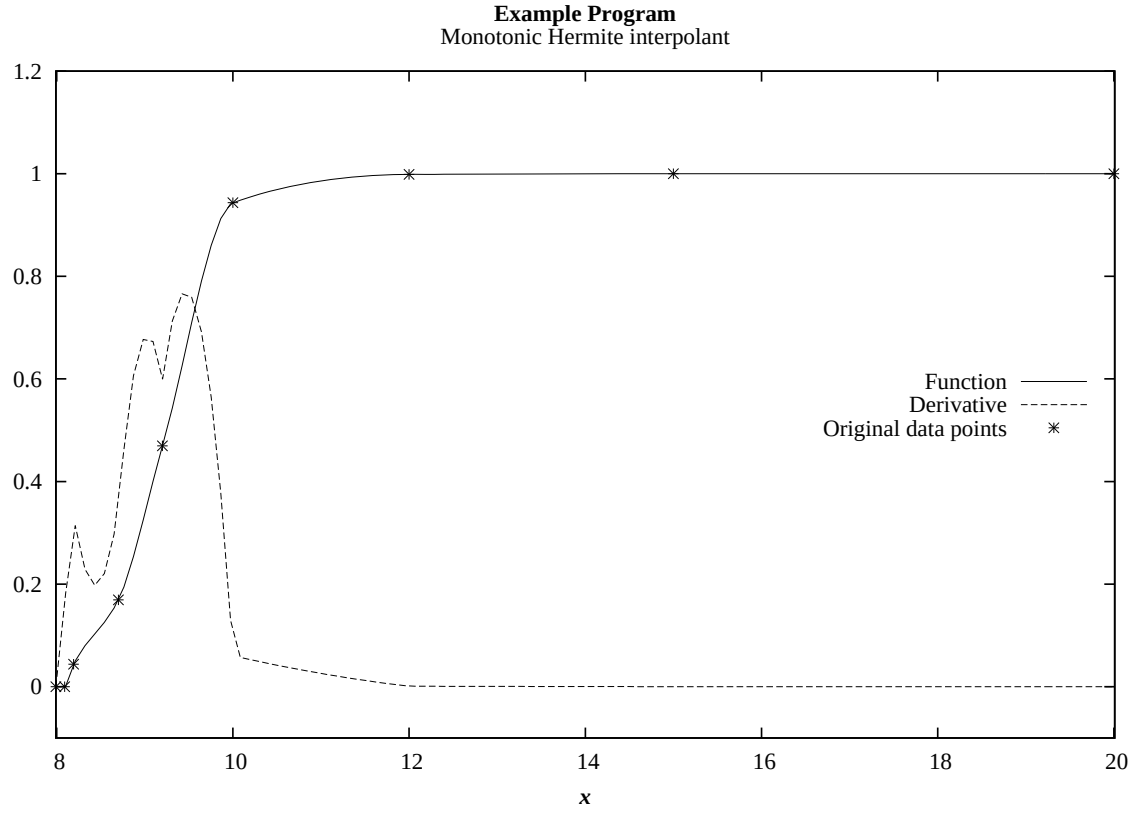
10.2 Program Data

```
E01BGF Example Program Data
   9                            N, the number of data points
  7.990  0.00000E+0  0.00000E+0  X(R), F(R), D(R)
  8.090  0.27643E-4  5.52510E-4
  8.190  0.43749E-1  0.33587E+0
  8.700  0.16918E+0  0.34944E+0
  9.200  0.46943E+0  0.59696E+0
  10.00  0.94374E+0  6.03260E-2
  12.00  0.99864E+0  8.98335E-4
  15.00  0.99992E+0  2.93954E-5
  20.00  0.99999E+0  0.00000E+0  End of data points
   11                           M, the number of evaluation points
```

10.3 Program Results

```
 E01BGF Example Program Results

                 Interpolated   Interpolated
       Abscissa         Value     Derivative
         7.9900        0.0000      0.000E+00
         9.1910        0.4640      6.060E-01
        10.3920        0.9645      4.569E-02
        11.5930        0.9965      9.917E-03
        12.7940        0.9992      6.249E-04
        13.9950        0.9998      2.708E-04
        15.1960        0.9999      2.809E-05
        16.3970        1.0000      2.034E-05
        17.5980        1.0000      1.308E-05
        18.7990        1.0000      6.297E-06
        20.0000        1.0000     -3.388E-21
```


Example Program
Monotonic Hermite interpolant
1.2
1
0.8
0.6
0.4
0.2
0
8
10
12
14
16
18
20
x
Function
Derivative
Original data points