NAG Library Routine Document

E01BFF

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

E01BFF evaluates a piecewise cubic Hermite interpolant at a set of points.

2 Specification

```
SUBROUTINE E01BFF(N, X, F, D, M, PX, PF, IFAIL)
INTEGER           N, M, IFAIL
double precision  X(N), F(N), D(N), PX(M), PF(M)
```

3 Description

E01BFF evaluates a piecewise cubic Hermite interpolant, as computed by E01BEF, at the points PX(i), for $i = 1, 2, \ldots, m$. If any point lies outside the interval from X(1) to X(N), a value is extrapolated from the nearest extreme cubic, and a warning is returned.

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

4 References

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

5 Parameters

1: N – INTEGER *Input*

2: X(N) – ***double precision*** array *Input*

3: F(N) – ***double precision*** array *Input*

4: D(N) – ***double precision*** 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) – ***double precision*** array *Input*

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

7: PF(M) – ***double precision*** 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: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

On exit: IFAIL $= 0$ unless the routine detects an error (see Section 6).

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 parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

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 such points may be very unreliable.

7 Accuracy

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

8 Further Comments

The time taken by E01BFF 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 [X($r-1$), X(r)]). A single call of E01BFF with $m > 1$ is more efficient than several calls with $m = 1$.

9 Example

This example reads in values of N, X, F and D, and then calls E01BFF to evaluate the interpolant at equally spaced points.

9.1 Program Text

```
*     E01BFF Example Program Text
*     Mark 14 Revised. NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NMAX
      PARAMETER        (MMAX=21,NMAX=50)
*     .. Local Scalars ..
      DOUBLE PRECISION STEP
      INTEGER          I, IFAIL, M, N, R
*     .. Local Arrays ..
      DOUBLE PRECISION D(NMAX), F(NMAX), PF(MMAX), PX(MMAX), X(NMAX)
*     .. External Subroutines ..
      EXTERNAL         E01BFF
```

```
*     .. Intrinsic Functions ..
      INTRINSIC        MIN
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E01BFF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.GT.0 .AND. N.LE.NMAX) THEN
         DO 20 R = 1, N
            READ (NIN,*) X(R), F(R), D(R)
   20    CONTINUE
         READ (NIN,*) M
         IF (M.GT.0 .AND. M.LE.MMAX) THEN
*           Compute M equally spaced points from X(1) to X(N).
            STEP = (X(N)-X(1))/(M-1)
            DO 40 I = 1, M
               PX(I) = MIN(X(1)+(I-1)*STEP,X(N))
   40       CONTINUE
            IFAIL = 1
*
            CALL E01BFF(N,X,F,D,M,PX,PF,IFAIL)
*
            IF (IFAIL.EQ.0) THEN
               WRITE (NOUT,*)
               WRITE (NOUT,*) '                  Interpolated'
               WRITE (NOUT,*) '       Abscissa          Value'
               DO 60 I = 1, M
                  WRITE (NOUT,99999) PX(I), PF(I)
   60          CONTINUE
            ELSE
               WRITE (NOUT,99998) IFAIL
            END IF
         END IF
      END IF
*
99999 FORMAT (1X,3F15.4)
99998 FORMAT (1X,' ** E01BFF returned with IFAIL = ',I5)
      END
```

9.2 Program Data

```
E01BFF 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
```

9.3 Program Results

```
 E01BFF 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
```