

# NAG Library Routine Document

## E01ABF

**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

E01ABF interpolates a function of one variable at a given point  $x$  from a table of function values evaluated at equidistant points, using Everett's formula.

### 2 Specification

```
SUBROUTINE E01ABF (N, P, A, G, N1, N2, IFAIL)
  INTEGER          N, N1, N2, IFAIL
  REAL (KIND=nag_wp) P, A(N1), G(N2)
```

### 3 Description

E01ABF interpolates a function of one variable at a given point

$$x = x_0 + ph,$$

where  $-1 < p < 1$  and  $h$  is the interval of differencing, from a table of values  $x_m = x_0 + mh$  and  $y_m$  where  $m = -(n-1), -(n-2), \dots, -1, 0, 1, \dots, n$ . The formula used is that of Frïberg (1970), neglecting the remainder term:

$$y_p = \sum_{r=0}^{n-1} \left( \frac{1-p+r}{2r+1} \right) \delta^{2r} y_0 + \sum_{r=0}^{n-1} \left( \frac{p+r}{2r+1} \right) \delta^{2r} y_1.$$

The values of  $\delta^{2r} y_0$  and  $\delta^{2r} y_1$  are stored on exit from the routine in addition to the interpolated function value  $y_p$ .

### 4 References

Frïberg C E (1970) *Introduction to Numerical Analysis* Addison–Wesley

### 5 Arguments

- 1: N – INTEGER *Input*  
*On entry:*  $n$ , half the number of points to be used in the interpolation.  
*Constraint:*  $N > 0$ .
- 2: P – REAL (KIND=nag\_wp) *Input*  
*On entry:* the point  $p$  at which the interpolated function value is required, i.e.,  $p = (x - x_0)/h$  with  $-1.0 < p < 1.0$ .  
*Constraint:*  $-1.0 < P < 1.0$ .
- 3: A(N1) – REAL (KIND=nag\_wp) array *Input/Output*  
*On entry:*  $A(i)$  must be set to the function value  $y_{i-n}$ , for  $i = 1, 2, \dots, 2n$ .  
*On exit:* the contents of A are unspecified.

- 4:  $G(N2)$  – REAL (KIND=nag\_wp) array *Output*  
*On exit:* the array contains  
 $y_0$  in  $G(1)$   
 $y_1$  in  $G(2)$   
 $\delta^{2r}y_0$  in  $G(2r + 1)$   
 $\delta^{2r}y_1$  in  $G(2r + 2)$ , for  $r = 1, 2, \dots, n - 1$ .  
The interpolated function value  $y_p$  is stored in  $G(2n + 1)$ .
- 5:  $N1$  – INTEGER *Input*  
*On entry:* the value  $2n$ , that is,  $N1$  is equal to the number of data points.
- 6:  $N2$  – INTEGER *Input*  
*On entry:* the value  $2n + 1$ , that is,  $N2$  is one more than the number of data points.
- 7:  $IFAIL$  – 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,  $P \leq -1.0$ ,  
or  $P \geq 1.0$ .

$IFAIL = -99$

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

In general, increasing  $n$  improves the accuracy of the result until full attainable accuracy is reached, after which it might deteriorate. If  $x$  lies in the central interval of the data (i.e.,  $0.0 \leq p < 1.0$ ), as is desirable, an upper bound on the contribution of the highest order differences (which is usually an upper bound on the error of the result) is given approximately in terms of the elements of the array  $G$  by  $a \times (|G(2n-1)| + |G(2n)|)$ , where  $a = 0.1, 0.02, 0.005, 0.001, 0.0002$  for  $n = 1, 2, 3, 4, 5$  respectively, thereafter decreasing roughly by a factor of 4 each time.

## 8 Parallelism and Performance

E01ABF is not threaded in any implementation.

## 9 Further Comments

The computation time increases as the order of  $n$  increases.

## 10 Example

This example interpolates at the point  $x = 0.28$  from the function values

$$\begin{pmatrix} x_i & -1.00 & -0.50 & 0.00 & 0.50 & 1.00 & 1.50 \\ y_i & 0.00 & -0.53 & -1.00 & -0.46 & 2.00 & 11.09 \end{pmatrix}.$$

We take  $n = 3$  and  $p = 0.56$ .

### 10.1 Program Text

```

Program e01abfe

!      E01ABF Example Program Text
!
!      Mark 26 Release. NAG Copyright 2016.
!
!      .. Use Statements ..
!      Use nag_library, Only: e01abf, nag_wp
!      .. Implicit None Statement ..
!      Implicit None
!      .. Parameters ..
!      Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
!      Real (Kind=nag_wp)          :: p
!      Integer                     :: i, ifail, n, n1, n2, r
!      .. Local Arrays ..
!      Real (Kind=nag_wp), Allocatable :: a(:), g(:)
!      .. Executable Statements ..
!      Write (nout,*) 'E01ABF Example Program Results'

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

!      Read (nin,*) n, p
!      n1 = 2*n
!      n2 = n1 + 1
!      Allocate (a(n1),g(n2))

!      Read (nin,*)(a(i),i=1,n1)

!      ifail = 0
!      Call e01abf(n,p,a,g,n1,n2,ifail)

!      Write (nout,*)

!      Do r = 0, n - 1
!         Write (nout,99999) 'Central differences order ', r, ' of Y0 =', &

```

```
      g(2*r+1)
      Write (nout,99998) '
End Do

      Y1 =', g(2*r+2)

      Write (nout,*)
      Write (nout,99998) 'Function value at interpolation point =', g(n2)

99999 Format (1X,A,I1,A,F12.5)
99998 Format (1X,A,F12.5)
      End Program e01abfe
```

## 10.2 Program Data

E01ABF Example Program Data

```
3      0.56
      0.00  -0.53  -1.00  -0.46   2.00  11.09
```

## 10.3 Program Results

E01ABF Example Program Results

```
Central differences order 0 of Y0 =  -1.00000
                                   Y1 =  -0.46000
Central differences order 1 of Y0 =   1.01000
                                   Y1 =   1.92000
Central differences order 2 of Y0 =  -0.04000
                                   Y1 =   3.80000
```

```
Function value at interpolation point =  -0.83591
```

---