NAG Library Routine Document

E02ACF

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

E02ACF calculates a minimax polynomial fit to a set of data points.

2 Specification

```
SUBROUTINE E02ACF (X, Y, N, A, M1, REF)
INTEGER            N, M1
REAL (KIND=nag_wp) X(N), Y(N), A(M1), REF
```

3 Description

Given a set of data points (x_i, y_i), for $i = 1, 2, \ldots, n$, E02ACF uses the exchange algorithm to compute an mth-order polynomial

$$\mathrm{P}(x) = a_1 + a_2 x + a_3 x^2 + \cdots + a_{m+1} x^m$$

such that $\max_i 2|\mathrm{P}(x_i) - y_i|$ is a minimum.

The routine also returns a number whose absolute value is the final reference deviation (see Section 6). The routine is an adaptation of Boothroyd (1967).

4 References

Boothroyd J B (1967) Algorithm 318 *Comm. ACM* **10** 801

Stieffel E (1959) Numerical methods of Tchebycheff approximation *On Numerical Approximation* (ed R E Langer) 217–232 University of Wisconsin Press

5 Arguments

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

On entry: the values of the x coordinates, x_i, for $i = 1, 2, \ldots, n$.

Constraint: $x_1 < x_2 < \cdots < x_n$.

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

On entry: the values of the y coordinates, y_i, for $i = 1, 2, \ldots, n$.

3: N – INTEGER *Input*

On entry: the number n of data points.

4: A(M1) – REAL (KIND=nag_wp) array *Output*

On exit: the coefficients a_i of the final polynomial, for $i = 1, 2, \ldots, m + 1$.

5: M1 – INTEGER *Input*

On entry: $m + 1$, where m is the order of the polynomial to be found.

Constraint: $\mathrm{M1} < \min(\mathrm{N}, 100)$.

6: REF – REAL (KIND=nag_wp) *Output*

On exit: the final reference deviation (see Section 6).

6 Error Indicators and Warnings

If an error is detected in an input argument E02ACF will act as if a soft noisy exit has been requested (see Section 3.4.4 in How to Use the NAG Library and its Documentation).

7 Accuracy

This is wholly dependent on the given data points.

8 Parallelism and Performance

E02ACF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The time taken increases with m.

10 Example

This example calculates a minimax fit with a polynomial of degree 5 to the exponential function evaluated at 21 points over the interval $[0, 1]$. It then prints values of the function and the fitted polynomial.

10.1 Program Text

```
    Program e02acfe

!     E02ACF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: e02acf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: m1 = 6, n = 21, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: ref, s, t, z
      Integer                          :: i, j
!     .. Local Arrays ..
      Real (Kind=nag_wp)               :: a(m1), x(n), y(n)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: exp, real
!     .. Executable Statements ..
      Write (nout,*) 'E02ACF Example Program Results'

      x(1:n) = real((/(i-1,i=1,n)/),kind=nag_wp)/real(n-1,kind=nag_wp)
      y(1:n) = exp(x(1:n))

      Call e02acf(x,y,n,a,m1,ref)

      Write (nout,*)
```

```
      Write (nout,*) '   Polynomial coefficients'
      Write (nout,99998)(a(i),i=1,m1)
      Write (nout,*)
      Write (nout,99997) '   Reference deviation = ', ref
      Write (nout,*)
      Write (nout,*) '  X     exp(X)    Fit     Residual'

      Do j = 1, n, 2
        z = x(j)

        s = a(m1)

        Do i = m1 - 1, 1, -1
          s = s*z + a(i)
        End Do

        t = y(j)
        Write (nout,99999) z, s, t, s - t
      End Do

99999 Format (1X,F5.2,2F9.4,E11.2)
99998 Format (6X,E12.4)
99997 Format (1X,A,E10.2)
    End Program e02acfe
```

10.2 Program Data

None.

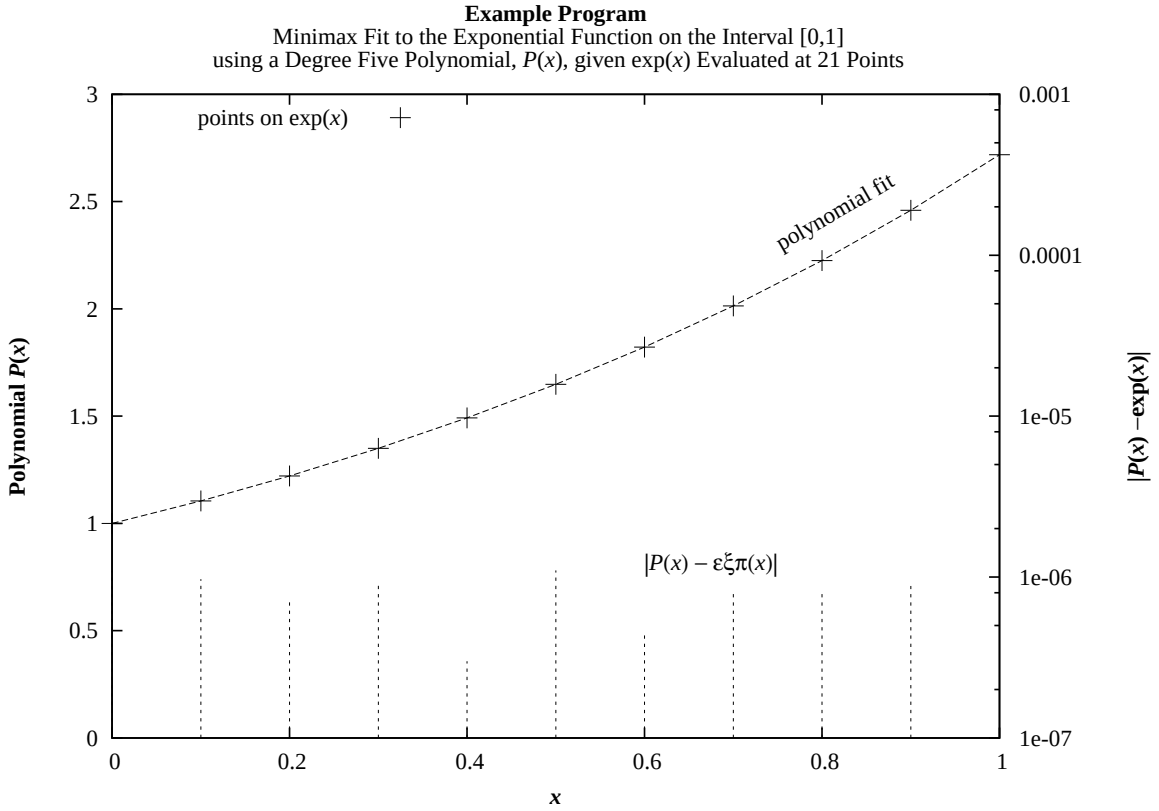
10.3 Program Results

```
 E02ACF Example Program Results

    Polynomial coefficients
        0.1000E+01
        0.1000E+01
        0.4991E+00
        0.1704E+00
        0.3478E-01
        0.1391E-01

    Reference deviation =   0.11E-05

   X     exp(X)    Fit     Residual
  0.00   1.0000   1.0000  -0.11E-05
  0.10   1.1052   1.1052   0.97E-06
  0.20   1.2214   1.2214  -0.74E-06
  0.30   1.3499   1.3499  -0.92E-06
  0.40   1.4918   1.4918   0.30E-06
  0.50   1.6487   1.6487   0.11E-05
  0.60   1.8221   1.8221   0.46E-06
  0.70   2.0138   2.0138  -0.82E-06
  0.80   2.2255   2.2255  -0.84E-06
  0.90   2.4596   2.4596   0.88E-06
  1.00   2.7183   2.7183  -0.11E-05
```


Example Program
Minimax Fit to the Exponential Function on the Interval [0,1]
using a Degree Five Polynomial, P(x), given exp(x) Evaluated at 21 Points
points on exp(x)
polynomial fit
|P(x) − εξπ(x)|
Polynomial P(x)
|P(x) −exp(x)|
x
3
2.5
2
1.5
1
0.5
0
0.001
0.0001
1e-05
1e-06
1e-07
0
0.2
0.4
0.6
0.8
1