NAG Fortran Library Routine Document

E02AGF

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

E02AGF computes constrained weighted least-squares polynomial approximations in Chebyshev-series form to an arbitrary set of data points. The values of the approximations and any number of their derivatives can be specified at selected points.

2 Specification

```
 SUBROUTINE E02AGF(M, KPLUS1, NROWS, XMIN, XMAX, X, Y, W, MF, XF, YF,
1                  LYF, IP, A, S, NP1, WRK, LWRK, IWRK, LIWRK, IFAIL)
 INTEGER           M, KPLUS1, NROWS, MF, LYF, IP(MF), NP1, LWRK,
1                  IWRK(LIWRK), LIWRK, IFAIL
 real              XMIN, XMAX, X(M), Y(M), W(M), XF(MF), YF(LYF),
1                  A(NROWS,KPLUS1), S(KPLUS1), WRK(LWRK)
```

3 Description

This routine determines least-squares polynomial approximations of degrees up to k to the set of data points (x_r, y_r) with weights w_r, for $r = 1, 2, \ldots, m$. The value of k, the maximum degree required, is prescribed by the user. At each of the values XF_r, for $r = 1, 2, \ldots, \mathrm{MF}$, of the independent variable x, the approximations and their derivatives up to order p_r are constrained to have one of the user-specified values YF_s, for $s = 1, 2, \ldots, n$, where $n = \mathrm{MF} + \sum_{r=1}^{\mathrm{MF}} p_r$.

The approximation of degree i has the property that, subject to the imposed constraints, it minimizes Σ_i, the sum of the squares of the weighted residuals ϵ_r for $r = 1, 2, \ldots, m$ where

$$\epsilon_r = w_r(y_r - f_i(x_r))$$

and $f_i(x_r)$ is the value of the polynomial approximation of degree i at the rth data point.

Each polynomial is represented in Chebyshev-series form with normalised argument $\bar{x}$. This argument lies in the range -1 to $+1$ and is related to the original variable x by the linear transformation

$$\bar{x} = \frac{2x - (x_{\mathrm{max}} + x_{\mathrm{min}})}{(x_{\mathrm{max}} - x_{\mathrm{min}})}$$

where x_{min} and x_{max}, specified by the user, are respectively the lower and upper end-points of the interval of x over which the polynomials are to be defined.

The polynomial approximation of degree i can be written as

$$\tfrac{1}{2}a_{i,0} + a_{i,1}T_1(\bar{x}) + \cdots + a_{ij}T_j(\bar{x}) + \cdots + a_{ii}T_i(\bar{x})$$

where $T_j(\bar{x})$ is the Chebyshev polynomial of the first kind of degree j with argument $\bar{x}$. For $i = n, n+1, \ldots, k$, the routine produces the values of the coefficients a_{ij}, for $j = 0, 1, \ldots, i$, together with the value of the root mean square residual, S_i, defined as

$$\sqrt{\frac{\sum_i}{(m' + n - i - 1)}},$$

where m' is the number of data points with non-zero weight.

Values of the approximations may subsequently be computed using E02AEF or E02AKF.

First E02AGF determines a polynomial $\mu(\bar{x})$, of degree $n-1$, which satisfies the given constraints, and a polynomial $\nu(\bar{x})$, of degree n, which has value (or derivative) zero wherever a constrained value (or derivative) is specified. It then fits $y_r - \mu(x_r)$, for $r = 1, 2, \ldots, m$ with polynomials of the required degree in $\bar{x}$ each with factor $\nu(\bar{x})$. Finally the coefficients of $\mu(\bar{x})$ are added to the coefficients of these fits to give the coefficients of the constrained polynomial approximations to the data points (x_r, y_r), for $r = 1, 2, \ldots, m$. The method employed is given in Hayes (1970): it is an extension of Forsythe's orthogonal polynomials method (Forsythe (1957)) as modified by Clenshaw (Clenshaw (1960)).

4 References

Clenshaw C W (1960) Curve fitting with a digital computer *Comput. J.* **2** 170–173

Forsythe G E (1957) Generation and use of orthogonal polynomials for data fitting with a digital computer *J. Soc. Indust. Appl. Math.* **5** 74–88

Hayes J G (ed.) (1970) Curve fitting by polynomials in one variable *Numerical Approximation to Functions and Data* Athlone Press, London

5 Parameters

1: M – INTEGER *Input*

On entry: the number m of data points to be fitted.

Constraint: M ≥ 1.

2: KPLUS1 – INTEGER *Input*

On entry: $k+1$, where k is the maximum degree required.

Constraint: $n + 1 \leq$ KPLUS1 $\leq m'' + n$, where n is the total number of constraints and m'' is the number of data points with non-zero weights and distinct abscissae which do not coincide with any of the XF(r).

3: NROWS – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which E02AGF is called.

Constraint: NROWS $\geq$ KPLUS1.

4: XMIN – ***real*** *Input*
5: XMAX – ***real*** *Input*

On entry: the lower and upper end-points, respectively, of the interval $[x_{\text{min}}, x_{\text{max}}]$. Unless there are specific reasons to the contrary, it is recommended that XMIN and XMAX be set respectively to the lowest and highest value among the x_r and XF(r). This avoids the danger of extrapolation provided there is a constraint point or data point with non-zero weight at each end-point.

Constraint: XMAX $>$ XMIN.

6: X(M) – ***real*** array *Input*

On entry: the value x_r of the independent variable at the rth data point, for $r = 1, 2, \ldots, m$.

Constraint: the X(r) must be in non-decreasing order and satisfy XMIN $\leq$ X$(r) \leq$ XMAX.

7: Y(M) – ***real*** array *Input*

On entry: Y(r) must contain y_r, the value of the dependent variable at the rth data point, for $r = 1, 2, \ldots, m$.

8: W(M) – ***real*** array *Input*

On entry: the weights w_r to be applied to the data points x_r, for $r = 1, 2 \ldots, m$. For advice on the choice of weights see Chapter E02. Negative weights are treated as positive. A zero weight causes the corresponding data point to be ignored. Zero weight should be given to any data point whose x and y values both coincide with those of a constraint (otherwise the denominators involved in the root-mean-square residuals s_i will be slightly in error).

9: MF – INTEGER *Input*

On entry: the number of values of the independent variable at which a constraint is specified.

Constraint: MF ≥ 1.

10: XF(MF) – ***real*** array *Input*

On entry: the rth value of the independent variable at which a constraint is specified, for $r = 1, 2, \ldots$, MF.

Constraint: these values need not be ordered but must be distinct and satisfy XMIN $\leq$ XF(r) $\leq$ XMAX.

11: YF(LYF) – ***real*** array *Input*

On entry: the values which the approximating polynomials and their derivatives are required to take at the points specified in XF. For each value of XF(r), YF contains in successive elements the required value of the approximation, its first derivative, second derivative, $\ldots, p_r$th derivative, for $r = 1, 2, \ldots$, MF. Thus the value which the kth derivative of each approximation ($k = 0$ referring to the approximation itself) is required to take at the point XF(r) must be contained in YF(s), where

$$s = r + k + p_1 + p_2 + \cdots + p_{r-1},$$

for $k = 0, 1, \ldots, p_r$ and $r = 1, 2, \ldots$, MF. The derivatives are with respect to the user's variable x.

12: LYF – INTEGER *Input*

On entry: the dimension of the array YF as declared in the (sub)program from which E02AGF is called.

Constraint: LYF $\geq n$, where $n = \mathrm{MF} + p_1 + p_2 + \cdots + p_{\mathrm{MF}}$.

13: IP(MF) – INTEGER array *Input*

On entry: IP(r) must contain p_r, the order of the highest-order derivative specified at XF(r), for $r = 1, 2, \ldots$, MF. $p_r = 0$ implies that the value of the approximation at XF(r) is specified, but not that of any derivative.

Constraint: IP$(r) \geq 0$, for $r = 1, 2, \ldots$, MF.

14: A(NROWS,KPLUS1) – ***real*** array *Output*

On exit: A$(i+1, j+1)$ contains the coefficient a_{ij} in the approximating polynomial of degree i, for $i = n, n+1, \ldots, k$; $j = 0, 1, \ldots, i$.

15: S(KPLUS1) – ***real*** array *Output*

On exit: S$(i+1)$ contains s_i, for $i = n, n+1, \ldots, k$, the root-mean-square residual corresponding to the approximating polynomial of degree i. In the case where the number of data points with non-zero weight is equal to $k + 1 - n$, s_i is indeterminate: the routine sets it to zero. For the interpretation of the values of s_i and their use in selecting an appropriate degree, see Section 3.1 of the E02 Chapter Introduction.

16: NP1 – INTEGER *Output*

On exit: $n + 1$, where n is the total number of constraint conditions imposed: $n = \mathrm{MF} + p_1 + p_2 + \cdots + p_{\mathrm{MF}}$.

17: WRK(LWRK) – ***real*** array *Output*

On exit: WRK contains weighted residuals of the highest degree of fit determined (k). The residual at x_r is in element $2(n+1)+3(m+k+1)+r$, for $r=1,2,\ldots,m$. The rest of the array is used as workspace.

18: LWRK – INTEGER *Input*

On entry: the dimension of the array WRK as declared in the (sub)program from which E02AGF is called.

Constraint: $\mathrm{LWRK} \geq \max(4\times\mathrm{M}+3\times\mathrm{KPLUS1}, 8\times n+5\times\mathrm{IPMAX}+\mathrm{MF}+10)+2\times n+2$, where $\mathrm{IPMAX} = \max(\mathrm{IP}(\mathrm{R}))$.

19: IWRK(LIWRK) – INTEGER array *Workspace*
20: LIWRK – INTEGER *Input*

On entry: the dimension of the array IWRK as declared in the (sub)program from which E02AGF is called.

Constraint: $\mathrm{LIWRK} \geq 2\times\mathrm{MF}+2$.

21: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: $\mathrm{IFAIL} = 0$ 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, for users 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 $\mathrm{IFAIL} = 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

$\mathrm{IFAIL} = 1$

On entry, $\mathrm{M} < 1$,
or $\mathrm{KPLUS1} < n+1$,
or $\mathrm{NROWS} < \mathrm{KPLUS1}$,
or $\mathrm{MF} < 1$,
or $\mathrm{LYF} < n$,
or LWRK is too small (see Section 5),
or $\mathrm{LIWRK} < 2\times\mathrm{MF}+2$.

(Here n is the total number of constraint conditions.)

$\mathrm{IFAIL} = 2$

$\mathrm{IP}(r) < 0$ for some $r = 1,2,\ldots,\mathrm{MF}$.

$\mathrm{IFAIL} = 3$

$\mathrm{XMIN} \geq \mathrm{XMAX}$, or $\mathrm{XF}(r)$ is not in the interval XMIN to XMAX for some $r=1,2,\ldots,\mathrm{MF}$, or the $\mathrm{XF}(r)$ are not distinct.

$\mathrm{IFAIL} = 4$

$\mathrm{X}(r)$ is not in the interval XMIN to XMAX for some $r=1,2,\ldots,\mathrm{M}$.

IFAIL = 5

X(r) < X($r-1$) for some $r = 2,3,\ldots,$M.

IFAIL = 6

KPLUS1 > $m'' + n$, where m'' is the number of data points with non-zero weight and distinct abscissae which do not coincide with any XF(r). Thus there is no unique solution.

IFAIL = 7

The polynomials $\mu(x)$ and/or $\nu(x)$ cannot be determined. The problem supplied is too ill-conditioned. This may occur when the constraint points are very close together, or large in number, or when an attempt is made to constrain high-order derivatives.

7 Accuracy

No complete error analysis exists for either the interpolating algorithm or the approximating algorithm. However, considerable experience with the approximating algorithm shows that it is generally extremely satisfactory. Also the moderate number of constraints, of low-order, which are typical of data fitting applications, are unlikely to cause difficulty with the interpolating routine.

8 Further Comments

The time taken by the routine to form the interpolating polynomial is approximately proportional to n^3, and that to form the approximating polynomials is very approximately proportional to $m(k+1)(k+1-n)$.

To carry out a least-squares polynomial fit without constraints, use E02ADF. To carry out polynomial interpolation only, use E01AEF.

9 Example

The example program reads data in the following order, using the notation of the parameter list above:

MF

IP(i), XF(i), Y-value and derivative values (if any) at XF(i), for $i = 1,2,\ldots,$MF

M

X(i), Y(i), W(i), for $i = 1,2,\ldots,$M

k, XMIN, XMAX

The output is:

the root-mean-square residual for each degree from n to k;

the Chebyshev coefficients for the fit of degree k;

the data points, and the fitted values and residuals for the fit of degree k.

The program is written in a generalized form which will read any number of data sets.

The data set supplied specifies 5 data points in the interval [0.0,4.0] with unit weights, to which are to be fitted polynomials, p, of degrees up to 4, subject to the 3 constraints:

$p(0.0) = 1.0, \quad p'(0.0) = -2.0, \quad p(4.0) = 9.0.$

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     E02AGF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          MFMAX, MMAX, KP1MAX, NROWS, LA, LIWRK, LYF, LWRK
      PARAMETER        (MFMAX=5,MMAX=20,KP1MAX=6,NROWS=KP1MAX,
     +                 LA=NROWS*KP1MAX,LIWRK=2*MFMAX+2,LYF=15,LWRK=200)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real             FITI, XMAX, XMIN
      INTEGER          I, IFAIL, IY, J, K, M, MF, NP1
*     .. Local Arrays ..
      real             A(NROWS,KP1MAX), S(KP1MAX), W(MMAX), WRK(LWRK),
     +                 X(MMAX), XF(MFMAX), Y(MMAX), YF(LYF)
      INTEGER          IP(MFMAX), IWRK(LIWRK)
*     .. External Subroutines ..
      EXTERNAL         E02AGF, E02AKF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E02AGF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
   20 READ (NIN,*,END=100) MF
      IF (MF.GT.0 .AND. MF.LE.MFMAX) THEN
         IY = 1
         DO 40 I = 1, MF
            READ (NIN,*) IP(I), XF(I), (YF(J),J=IY,IY+IP(I))
            IY = IY + IP(I) + 1
   40    CONTINUE
         READ (NIN,*) M
         IF (M.GT.0 .AND. M.LE.MMAX) THEN
            READ (NIN,*) (X(I),Y(I),W(I),I=1,M)
            READ (NIN,*) K, XMIN, XMAX
            IFAIL = 0
*
            CALL E02AGF(M,K+1,NROWS,XMIN,XMAX,X,Y,W,MF,XF,YF,LYF,IP,A,S,
     +                  NP1,WRK,LWRK,IWRK,LIWRK,IFAIL)
*
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Degree  RMS residual'
            WRITE (NOUT,99999) (I,S(I+1),I=NP1-1,K)
            WRITE (NOUT,*)
            WRITE (NOUT,99996) 'Details of the fit of degree ', K
            WRITE (NOUT,*)
            WRITE (NOUT,*) '  Index   Coefficient'
            DO 60 I = 1, K + 1
               WRITE (NOUT,99997) I - 1, A(K+1,I)
   60       CONTINUE
            WRITE (NOUT,*)
            WRITE (NOUT,*)
     +        '     I      X(I)       Y(I)       Fit     Residual'
            DO 80 I = 1, M
*
               CALL E02AKF(K+1,XMIN,XMAX,A(K+1,1),NROWS,LA-K,X(I),FITI,
     +                     IFAIL)
*
               WRITE (NOUT,99998) I, X(I), Y(I), FITI, FITI - Y(I)
   80       CONTINUE
            GO TO 20
         END IF
      END IF
  100 STOP
*
99999 FORMAT (1X,I4,1P,e15.2)
99998 FORMAT (1X,I6,3F11.4,e11.2)
99997 FORMAT (1X,I6,F11.4)
```

```
99996 FORMAT (1X,A,I2)
      END
```

9.2 Program Data

```
E02AGF Example Program Data
    2
    1       0.0       1.0      -2.0
    0       4.0       9.0
    5
       0.5      0.03       1.0
       1.0     -0.75       1.0
       2.0     -1.0        1.0
       2.5     -0.1        1.0
       3.0      1.75       1.0
    4       0.0       4.0
```

9.3 Program Results

```
 E02AGF Example Program Results

 Degree  RMS residual
    3       2.55E-03
    4       2.94E-03

 Details of the fit of degree  4

   Index   Coefficient
      0     3.9980
      1     3.4995
      2     3.0010
      3     0.5005
      4    -0.0000

      I      X(I)       Y(I)       Fit     Residual
      1     0.5000     0.0300     0.0310   0.10E-02
      2     1.0000    -0.7500    -0.7508  -0.78E-03
      3     2.0000    -1.0000    -1.0020  -0.20E-02
      4     2.5000    -0.1000    -0.0961   0.39E-02
      5     3.0000     1.7500     1.7478  -0.22E-02
```