E02GBF – NAG Fortran Library Routine Document

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

E02GBF calculates an l_1 solution to an over-determined system of linear equations, possibly subject to linear inequality constraints.

2 Specification

```
 SUBROUTINE E02GBF(M, N, MPL, E, IE, F, X, MXS, MONIT, IPRINT, K,
1                  EL1N, INDX, W, IW, IFAIL)
 INTEGER           M, N, MPL, IE, MXS, IPRINT, K, INDX(MPL), IW,
1                  IFAIL
 real              E(IE,MPL), F(MPL), X(N), EL1N, W(IW)
 EXTERNAL          MONIT
```

3 Description

Given a matrix A with m rows and n columns $(m \geq n)$ and a vector b with m elements, the routine calculates an l_1 solution to the over-determined system of equations

$$Ax = b.$$

That is to say, it calculates a vector x, with n elements, which minimises the l_1 norm (the sum of the absolute values) of the residuals

$$r(x) = \sum_{i=1}^{m} |r_i|,$$

where the residuals r_i are given by

$$r_i = b_i - \sum_{j=1}^{n} a_{ij} x_j, \quad i = 1, 2, \ldots, m.$$

Here a_{ij} is the element in row i and column j of A, b_i is the ith element of b and x_j the jth element of x.

If, in addition, a matrix C with l rows and n columns and a vector d with l elements, are given, the vector x computed by the routine is such as to minimize the l_1 norm $r(x)$ subject to the set of inequality constraints $Cx \geq d$.

The matrices A and C need not be of full rank.

Typically in applications to data fitting, data consisting of m points with co-ordinates (t_i, y_i) is to be approximated by a linear combination of known functions $\phi_i(t)$,

$$\alpha_1\phi_1(t) + \alpha_2\phi_2(t) + \ldots + \alpha_n\phi_n(t),$$

in the l_1 norm, possibly subject to linear inequality constraints on the coefficients α_j of the form $C\alpha \geq d$ where α is the vector of the α_j and C and d are as in the previous paragraph. This is equivalent to finding an l_1 solution to the over-determined system of equations

$$\sum_{j=1}^{n} \phi_j(t_i)\alpha_j = y_i, \quad i = 1, 2, \ldots, m,$$

subject to $C\alpha \geq d$.

Thus if, for each value of i and j, the element a_{ij} of the matrix A above is set equal to the value of $\phi_j(t_i)$ and b_i is equal to y_i and C and d are also supplied to the routine, the solution vector x will contain the required values of the α_j. Note that the independent variable t above can, instead, be a vector of several independent variables (this includes the case where each of ϕ_i is a function of a different variable, or set of variables).

The algorithm follows the Conn–Pietrzykowski approach (see Bartels *et al.* [1] and Conn and Pietrzykowski [2]), which is via an exact penalty function

$$g(x) = \gamma r(x) - \sum_{i=1}^{l} \min(0, c_i^T x - d_i),$$

where γ is a penalty parameter, c_i^T is the ith row of the matrix C, and d_i is the ith element of the vector d. It proceeds in a step-by-step manner much like the simplex method for linear programming but does not move from vertex to vertex and does not require the problem to be cast in a form containing only non-negative unknowns. It uses stable procedures to update an orthogonal factorization of the current set of active equations and constraints.

4 References

[1] Bartels R H, Conn A R and Sinclair J W (1978) Minimisation techniques for piecewise differentiable functions – the l_1 solution to an overdetermined linear system *SIAM J. Numer. Anal.* **15** 224–241

[2] Conn A R and Pietrzykowski T (1977) A penalty-function method converging directly to a constrained optimum *SIAM J. Numer. Anal.* **14** 348–375

[3] Bartels R H, Conn A R and Charalambous C (1976) Minimisation techniques for piecewise Differentiable functions – the l_∞ solution to an overdetermined linear system *Technical Report No. 247, CORR 76/30* Mathematical Sciences Department, The John Hopkins University

[4] Bartels R H, Conn A R and Sinclair J W (1976) A Fortran program for solving overdetermined systems of linear equations in the l_1 Sense *Technical Report No. 236, CORR 76/7* Mathematical Sciences Department, The John Hopkins University

5 Parameters

1: M — INTEGER *Input*

On entry: the number of equations in the over-determined system, m (i.e., the number of rows of the matrix A).

Constraint: M $\geq$ 2.

2: N — INTEGER *Input*

On entry: the number of unknowns, n (the number of columns of the matrix A).

Constraint: M $\geq$ N $\geq$ 2.

3: MPL — INTEGER *Input*

On entry: $m + l$, where l is the number of constraints (which may be zero).

Constraint: MPL $\geq$ M.

4: E(IE,MPL) — ***real*** array *Input/Output*

On entry: the equation and constraint matrices stored in the following manner:

The first m columns contain the m rows of the matrix A; element E(i,j) specifying the element a_{ji} in the jth row and ith column of A (the coefficient of the ith unknown in the jth equation), for $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, m$. The next l columns contain the l rows of the constraint matrix C; element E$(i, j + m)$ containing the element c_{ji} in the jth row and ith column of C (the coefficient of the ith unknown in the jth constraint), for $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, l$.

On exit: unchanged, except possibly to the extent of a small multiple of the ***machine precision***. (See Section 8.)

5: IE — INTEGER *Input*

On entry: the first dimension of the array E as declared in the (sub)program from which E02GBF is called.

Constraint: IE $\geq$ N.

6: F(MPL) — ***real*** array *Input*

On entry: F(i), for $i = 1, 2, \ldots, m$ must contain b_i (the ith element of the right-hand side vector of the over-determined system of equations) and F($m + i$), for $i = 1, 2, \ldots, l$ must contain d_i (the ith element of the right-hand side vector of the constraints), where l is the number of constraints.

7: X(N) — ***real*** array *Input/Output*

On entry: X(i) must contain an estimate of the ith unknown, for $i = 1, 2, \ldots, n$. If no better initial estimate for X(i) is available, set X(i) = 0.0.

On exit: the latest estimate of the ith unknown, for $i = 1, 2, \ldots, n$. If IFAIL = 0 on exit, these are the solution values.

8: MXS — INTEGER *Input*

On entry: the maximum number of steps to be allowed for the solution of the unconstrained problem. Typically this may be a modest multiple of n. If, on entry, MXS is zero or negative, the value returned by X02BBF is used.

9: MONIT — SUBROUTINE, supplied by the user. *External Procedure*

Monit can be used to print out the current values of any selection of its parameters. The frequency with which MONIT is called in E02GBF is controlled by IPRINT (see below).

Its specification is:

```
SUBROUTINE MONIT(N, X, NITER, K, EL1N)
INTEGER          N, NITER, K
real             X(N), EL1N
```

1: N — INTEGER *Input*

On entry: the number n of unknowns (the number of columns of the matrix A).

2: X(N) — ***real*** array *Input*

On entry: the latest estimate of the unknowns.

3: NITER — INTEGER *Input*

On entry: the number of iterations so far carried out.

4: K — INTEGER *Input*

On entry: the total number of equations and constraints which are currently active (i.e., the number of equations with zero residuals plus the number of constraints which are satisfied as equations).

5: EL1N — ***real*** *Input*

On entry: the l_1 norm of the current residuals of the over-determined system of equations.

MONIT must be declared as EXTERNAL in the (sub)program from which E02GBF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

10: IPRINT — INTEGER *Input*

On entry: the frequency of iteration print out. If IPRINT > 0, then MONIT is called every IPRINT iterations and at the solution. If IPRINT = 0, then information is printed out at the solution only. Otherwise MONIT is not called (but a dummy routine must still be provided).

11: K — INTEGER *Output*

On exit: the total number of equations and constraints which are then active (i.e., the number of equations with zero residuals plus the number of constraints which are satisfied as equalities).

12: EL1N — ***real*** *Output*

On exit: the l_1 norm (sum of absolute values) of the equation residuals.

13: INDX(MPL) — INTEGER array *Output*

On exit: specifies which columns of E relate to the inactive equations and constraints. INDX(1) up to INDX(K) number the active columns and INDX(K + 1) up to INDX(MPL) number the inactive columns.

14: W(IW) — ***real*** array *Workspace*

15: IW — INTEGER *Input*

On entry: the dimension of the array W as declared in the (sub)program from which E02GBF is called.

Constraint: $\mathrm{IW} \geq 3 \times \mathrm{MPL} + 5 \times \mathrm{N} + \mathrm{N}^2 + (\mathrm{N}+1) \times (\mathrm{N}+2)/2$.

16: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

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

6 Error Indicators and Warnings

Errors detected by the routine:

IFAIL = 1

The constraints cannot all be satisfied simultaneously: they are not compatible with one another. Hence no solution is possible.

IFAIL = 2

The limit imposed by MXS has been reached without finding a solution. Consider restarting from the current point by simply calling E02GBF again without changing the parameters.

IFAIL = 3

The routine has failed because of numerical difficulties; the problem is too ill-conditioned. Consider rescaling the unknowns.

IFAIL = 4

On entry, one or more of the following conditions are violated:

$\mathrm{M} \geq \mathrm{N} \geq 2$,
$\mathrm{MPL} \geq \mathrm{M}$,
$\mathrm{IW} \geq 3 \times \mathrm{MPL} + 5 \times \mathrm{N} + \mathrm{N}^2 + (\mathrm{N}+1) \times (\mathrm{N}+2)/2$,
$\mathrm{IE} \geq \mathrm{N}$.

Alternatively elements 1 to M of one of the first MPL columns of the array E are all zero – this corresponds to a zero row in either of the matrices A or C.

7 Accuracy

The method is stable.

8 Further Comments

The effect of m and n on the time and on the number of iterations varies from problem to problem, but typically the number of iterations is a small multiple of n and the total time taken by the routine is approximately proportional to mn^2.

Linear dependencies among the rows or columns of A and C are not necessarily a problem to the algorithm. Solutions can be obtained from rank-deficient A and C. However, the algorithm requires that at every step the currently active columns of E form a linearly independent set. If this is not the case at any step, small, random perturbations of the order of rounding error are added to the appropriate columns of E. Normally this perturbation process will not affect the solution significantly. It does mean, however, that results may not be exactly reproducible.

9 Example

Suppose we wish to approximate in $[0, 1]$ a set of data by a curve of the form

$$y = ax^3 + bx^2 + cx + d$$

which has non-negative slope at the data points. Given points (t_i, y_i) we may form the equations

$$y_i = at_i^3 + bt_i^2 + ct_i + d$$

for the 6 data points, $i = 1, 2, \ldots, 6$. The requirement of a non-negative slope at the data points demands

$$3at_i^2 + 2bt_i + c \geq 0$$

for each t_i and these form the constraints.

(Note that, for fitting with polynomials, it would usually be advisable to work with the polynomial expressed in Chebyshev series form (see the Chapter Introduction). The power series form is used here for simplicity of exposition.)

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.

```
*     E02GBF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          N, MPLMAX, IE, IW
      PARAMETER        (N=4,MPLMAX=12,IE=N,IW=3*MPLMAX+N*N+5*N+(N+1)
     +                 *(N+2)/2)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real             EL1N, T, XI
      INTEGER          I, IFAIL, IPRINT, K, L, M, MXS
*     .. Local Arrays ..
      real             E(IE,MPLMAX), F(MPLMAX), W(IW), X(N)
      INTEGER          IWORK(MPLMAX)
*     .. External Subroutines ..
      EXTERNAL         E02GBF, MONIT
*     .. Intrinsic Functions ..
      INTRINSIC        real
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E02GBF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
```

```
      READ (NIN,*) M
      L = M
      IF (M.GT.0 .AND. M+L.LE.MPLMAX) THEN
         DO 20 I = 1, M
            READ (NIN,*) T, F(I)
            XI = 0.1e0*real(I-1)
            E(1,I) = 1.0e0
            E(2,I) = T
            E(3,I) = T*T
            E(4,I) = T*T*T
            E(1,M+I) = 0.0e0
            E(2,M+I) = 1.0e0
            E(3,M+I) = 2.0e0*T
            E(4,M+I) = 3.0e0*T*T
            F(M+I) = 0.0e0
   20    CONTINUE
         DO 40 I = 1, N
            X(I) = 0.0e0
   40    CONTINUE
         MXS = 50
*        * Set IPRINT=1 to obtain output from MONIT at each iteration *
         IPRINT = 0
         IFAIL = 1
*
         CALL E02GBF(M,N,M+L,E,IE,F,X,MXS,MONIT,IPRINT,K,EL1N,IWORK,W,
     +               IW,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,99999) 'IFAIL = ', IFAIL
      END IF
      STOP
*
99999 FORMAT (1X,A,I2)
      END
*
      SUBROUTINE MONIT(N,X,NITER,K,EL1N)
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Scalar Arguments ..
      real             EL1N
      INTEGER          K, N, NITER
*     .. Array Arguments ..
      real             X(N)
*     .. Executable Statements ..
      WRITE (NOUT,*)
      WRITE (NOUT,99999) 'Results at iteration ', NITER
      WRITE (NOUT,*) 'X-values'
      WRITE (NOUT,99998) X
      WRITE (NOUT,99997) 'Norm of residuals =', EL1N
      RETURN
*
99999 FORMAT (1X,A,I5)
99998 FORMAT (1X,4F15.4)
99997 FORMAT (1X,A,e12.5)
      END
```

9.2 Program Data

```
E02GBF Example Program Data
  6
  0.00  0.00
  0.20  0.07
  0.40  0.07
  0.60  0.11
  0.80  0.27
  1.00  0.68
```

9.3 Program Results

```
 E02GBF Example Program Results

 Results at iteration    10
 X-values
          0.0000         0.6943        -2.1482         2.1339
 Norm of residuals = 0.95714E-02

 IFAIL =  0
```