NAG Fortran Library Routine Document

E02GAF

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

E02GAF calculates an l_1 solution to an over-determined system of linear equations.

2 Specification

```
 SUBROUTINE E02GAF(M, A, LA, B, NPLUS2, TOLER, X, RESID, IRANK, ITER,
1                  IWORK, IFAIL)
 INTEGER           M, LA, NPLUS2, IRANK, ITER, IWORK(M), IFAIL
 real              A(LA,NPLUS2), B(M), TOLER, X(NPLUS2), RESID
```

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 minimizes 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. The matrix A need not be of full rank.

Typically in applications to data fitting, data consisting of m points with co-ordinates (t_i, y_i) are to be approximated in the l_1 norm by a linear combination of known functions $\phi_j(t)$,

$$\alpha_1\phi_1(t) + \alpha_2\phi_2(t) + \cdots + \alpha_n\phi_n(t).$$

This is equivalent to fitting an l_1 solution to the over-determined system of equations

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

Thus if, for each value of i and j, the element a_{ij} of the matrix A in the previous paragraph is set equal to the value of $\phi_j(t_i)$ and b_i is set equal to y_i, 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 ϕ_i is a function of a different variable, or set of variables).

The algorithm is a modification of the simplex method of linear programming applied to the primal formulation of the l_1 problem (see Barrodale and Roberts (1973) and Barrodale and Roberts (1974)). The modification allows several neighbouring simplex vertices to be passed through in a single iteration, providing a substantial improvement in efficiency.

4 References

Barrodale I and Roberts F D K (1973) An improved algorithm for discrete l_1 linear approximation *SIAM J. Numer. Anal.* **10** 839–848

Barrodale I and Roberts F D K (1974) Solution of an overdetermined system of equations in the l_1-norm *Comm. ACM* **17 (6)** 319–320

5 Parameters

1: M – INTEGER *Input*

On entry: the number of equations, m (the number of rows of the matrix A).

Constraint: $\mathrm{M} \geq n \geq 1$.

2: A(LA,NPLUS2) – ***real*** array *Input/Output*

On entry: $\mathrm{A}(i,j)$ must contain a_{ij}, the element in the ith row and jth column of the matrix A, for $i = 1,2,\ldots,m$ and $j = 1,2,\ldots,n$. The remaining elements need not be set.

On exit: A contains the last simplex tableau generated by the simplex method.

3: LA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which E02GAF is called.

Constraint: $\mathrm{LA} \geq \mathrm{M}+2$.

4: B(M) – ***real*** array *Input/Output*

On entry: b_i, the ith element of the vector b, for $i = 1,2,\ldots,m$.

On exit: the ith residual r_i corresponding to the solution vector x, for $i = 1,2,\ldots,m$.

5: NPLUS2 – INTEGER *Input*

On entry: $n+2$, where n is the number of unknowns (the number of columns of the matrix A).

Constraint: $3 \leq \mathrm{NPLUS2} \leq \mathrm{M}+2$.

6: TOLER – ***real*** *Input*

On entry: a non-negative value. In general TOLER specifies a threshold below which numbers are regarded as zero. The recommended threshold value is $\epsilon^{2/3}$ where ϵ is the ***machine precision***. The recommended value can be computed within the routine by setting TOLER to zero. If premature termination occurs a larger value for TOLER may result in a valid solution.

Suggested value: 0.0.

7: X(NPLUS2) – ***real*** array *Output*

On exit: $\mathrm{X}(j)$ contains the jth element of the solution vector x, for $j = 1,2,\ldots,n$. The elements $\mathrm{X}(n+1)$ and $\mathrm{X}(n+2)$ are unused.

8: RESID – ***real*** *Output*

On exit: the sum of the absolute values of the residuals for the solution vector x.

9: IRANK – INTEGER *Output*

On exit: the computed rank of the matrix A.

10: ITER – INTEGER *Output*

On exit: the number of iterations taken by the simplex method.

11: IWORK(M) – INTEGER array *Workspace*

12: 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: 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 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 $= 1$

An optimal solution has been obtained but this may not be unique.

IFAIL $= 2$

The calculations have terminated prematurely due to rounding errors. Experiment with larger values of TOLER or try scaling the columns of the matrix (see Section 8).

IFAIL $= 3$

On entry, NPLUS2 < 3,
or NPLUS2 $>$ M $+ 2$,
or LA $<$ M $+ 2$.

7 Accuracy

Experience suggests that the computational accuracy of the solution x is comparable with the accuracy that could be obtained by applying Gaussian elimination with partial pivoting to the n equations satisfied by this algorithm (i.e., those equations with zero residuals). The accuracy therefore varies with the conditioning of the problem, but has been found generally very satisfactory in practice.

8 Further Comments

The effects of m and n on the time and on the number of iterations in the Simplex Method vary 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.

It is recommended that, before the routine is entered, the columns of the matrix A are scaled so that the largest element in each column is of the order of unity. This should improve the conditioning of the matrix, and also enable the parameter TOLER to perform its correct function. The solution x obtained will then, of course, relate to the scaled form of the matrix. Thus if the scaling is such that, for each $j = 1, 2, \ldots, n$, the elements of the jth column are multiplied by the constant k_j, the element x_j of the solution vector x must be multiplied by k_j if it is desired to recover the solution corresponding to the original matrix A.

9 Example

Suppose we wish to approximate a set of data by a curve of the form

$$y = Ke^t + Le^{-t} + M$$

where K, L and M are unknown. Given values y_i at 5 points t_i we may form the over-determined set of equations for K, L and M

$$e^{x_i}K + e^{-x_i}L + M = y_i, \quad i = 1, 2, \ldots, 5.$$

E02GAF is used to solve these in the l_1 sense.

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.

```
*     E02GAF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          MMAX, LA, NPLUS2
      PARAMETER        (MMAX=5,LA=MMAX+2,NPLUS2=5)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real             RESID, T, TOL
      INTEGER          I, IFAIL, ITER, M, RANK
*     .. Local Arrays ..
      real             A(LA,NPLUS2), B(MMAX), X(NPLUS2)
      INTEGER          IWORK(MMAX)
*     .. External Subroutines ..
      EXTERNAL         E02GAF
*     .. Intrinsic Functions ..
      INTRINSIC        EXP
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E02GAF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) M
      IF (M.GT.0 .AND. M.LE.MMAX) THEN
         DO 20 I = 1, M
            READ (NIN,*) T, B(I)
            A(I,1) = EXP(T)
            A(I,2) = EXP(-T)
            A(I,3) = 1.0e0
   20    CONTINUE
         TOL = 0.0e0
         IFAIL = 1
*
         CALL E02GAF(M,A,LA,B,NPLUS2,TOL,X,RESID,RANK,ITER,IWORK,IFAIL)
*
         IF (IFAIL.LE.1) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,99999) 'Resid = ', RESID, '  Rank = ', RANK,
     +        '  Iterations = ', ITER, '  IFAIL =', IFAIL
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Solution'
            WRITE (NOUT,99998) (X(I),I=1,NPLUS2-2)
         ELSE
            WRITE (NOUT,99997) 'E02GAF fails with error', IFAIL
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,A,e10.2,A,I5,A,I5,A,I5)
99998 FORMAT (1X,6F10.4)
99997 FORMAT (1X,A,I2)
      END
```

9.2 Program Data

```
E02GAF Example Program Data
  5
   0.0 4.501
   0.2 4.360
   0.4 4.333
   0.6 4.418
   0.8 4.625
```

9.3 Program Results

```
 E02GAF Example Program Results

 Resid =   0.28E-02  Rank =     3  Iterations =     5  IFAIL =    0

 Solution
     1.0014    2.0035    1.4960
```