NAG Fortran Library Routine Document

D02GBF

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

D02GBF solves a general linear two-point boundary-value problem for a system of ordinary differential equations using a deferred correction technique.

2 Specification

```
 SUBROUTINE D02GBF (A, B, N, TOL, FCNF, FCNG, C, D, GAM, MNP, X, Y, NP,
1                   W, LW, IW, LIW, IFAIL)
 INTEGER            N, MNP, NP, LW, IW(LIW), LIW, IFAIL
 double precision   A, B, TOL, C(N,N), D(N,N), GAM(N), X(MNP), Y(N,MNP),
1                   W(LW)
 EXTERNAL           FCNF, FCNG
```

3 Description

D02GBF solves the linear two-point boundary-value problem for a system of n ordinary differential equations in the interval $[a, b]$. The system is written in the form

$$y' = F(x)y + g(x) \tag{1}$$

and the boundary conditions are written in the form

$$\mathrm{C}y(a) + Dy(b) = \gamma. \tag{2}$$

Here $F(x)$, C and D are n by n matrices, and $g(x)$ and γ are n-component vectors. The approximate solution to (1) and (2) is found using a finite-difference method with deferred correction. The algorithm is a specialisation of that used in (sub)program D02RAF which solves a nonlinear version of (1) and (2). The nonlinear version of the algorithm is described fully in Pereyra (1979).

You supply an absolute error tolerance and may also supply an initial mesh for the construction of the finite-difference equations (alternatively a default mesh is used). The algorithm constructs a solution on a mesh defined by adding points to the initial mesh. This solution is chosen so that the error is everywhere less than your tolerance and so that the error is approximately equidistributed on the final mesh. The solution is returned on this final mesh.

If the solution is required at a few specific points then these should be included in the initial mesh. If, on the other hand, the solution is required at several specific points, then you should use the interpolation routines provided in Chapter E01 if these points do not themselves form a convenient mesh.

4 References

Pereyra V (1979) PASVA3: An adaptive finite-difference Fortran program for first order nonlinear, ordinary boundary problems *Codes for Boundary Value Problems in Ordinary Differential Equations. Lecture Notes in Computer Science* (ed B Childs, M Scott, J W Daniel, E Denman and P Nelson) **76** Springer–Verlag

5 Parameters

1: A – ***double precision*** *Input*

On entry: a, the left-hand boundary point.

2: B – ***double precision*** *Input*

On entry: b, the right-hand boundary point.

Constraint: B $>$ A.

3: N – INTEGER *Input*

On entry: the number of equations; that is n is the order of system (1).

Constraint: N ≥ 2.

4: TOL – ***double precision*** *Input*

On entry: a positive absolute error tolerance. If

$$a = x_1 < x_2 < \cdots < x_{\mathrm{NP}} = b$$

is the final mesh, $z(x)$ is the approximate solution from D02GBF and $y(x)$ is the true solution of equations (1) and (2) then, except in extreme cases, it is expected that

$$\|z - y\| \leq \mathrm{TOL} \tag{3}$$

where

$$\|u\| = \max_{1 \leq i \leq \mathrm{N}} \max_{1 \leq j \leq \mathrm{NP}} \left|u_i(x_j)\right|.$$

Constraint: TOL > 0.0.

5: FCNF – SUBROUTINE, supplied by the user. *External Procedure*

FCNF must evaluate the matrix F(x) in (1) at a general point x.

Its specification is:

```
SUBROUTINE FCNF (X, F)
double precision  X, F(n,n)
```

where n is the actual value of N in the call of D02GBF.

1: X – ***double precision*** *Input*

On entry: the value of the independent variable x.

2: F(n,n) – ***double precision*** array *Output*

On exit: the (i,j)th element of the matrix F(x), for $i,j = 1, 2, \ldots, n$. (See Section 9 for an example.)

FCNF must be declared as EXTERNAL in the (sub)program from which D02GBF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

6: FCNG – SUBROUTINE, supplied by the user. *External Procedure*

FCNG must evaluate the vector $g(x)$ in (1) at a general point x.

Its specification is:

```
SUBROUTINE FCNG (X, G)
double precision  X, G(n)
```

where n is the actual value of N in the call of D02GBF.

1: X – ***double precision*** *Input*

On entry: the value of the independent variable x.

2: G(n) – ***double precision*** array *Output*

On exit: the ith element of the vector $g(x)$, for $i = 1, 2, \ldots, n$. (See Section 9 for an example.)

FCNG must be declared as EXTERNAL in the (sub)program from which D02GBF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

7: C(N,N) – ***double precision*** array *Input/Output*
8: D(N,N) – ***double precision*** array *Input/Output*
9: GAM(N) – ***double precision*** array *Input/Output*

On entry: the arrays C and D must be set to the matrices C and D in (2)). GAM must be set to the vector γ in (2).

On exit: the rows of C and D and the components of GAM are reordered so that the boundary conditions are in the order:

(i) conditions on $y(a)$ only;

(ii) condition involving $y(a)$ and $y(b)$; and

(iii) conditions on $y(b)$ only.

The routine will be slightly more efficient if the arrays C, D and GAM are ordered in this way before entry, and in this event they will be unchanged on exit.

Note that the problems (1) and (2) must be of boundary-value type, that is neither C nor D may be identically zero. Note also that the rank of the matrix $[C, D]$ must be n for the problem to be properly posed. Any violation of these conditions will lead to an error exit.

10: MNP – INTEGER *Input*

On entry: the maximum permitted number of mesh points.

Constraint: MNP ≥ 32.

11: X(MNP) – ***double precision*** array *Input/Output*

On entry: if NP ≥ 4 (see NP), the first NP elements must define an initial mesh. Otherwise the elements of x need not be set.

Constraint:

$$\mathrm{A} = \mathrm{X}(1) < \mathrm{X}(2) < \cdots < \mathrm{X}(\mathrm{NP}) = \mathrm{B}, \quad \mathrm{NP} \geq 4. \tag{4}$$

On exit: $\mathrm{X}(1), \mathrm{X}(2), \ldots, \mathrm{X}(\mathrm{NP})$ define the final mesh (with the returned value of NP) satisfying the relation (4).

12: Y(N,MNP) – ***double precision*** array *Output*

On exit: the approximate solution $z(x)$ satisfying (3), on the final mesh, that is

$$\mathrm{Y}(j, i) = z_j(x_i), \quad i = 1, 2, \ldots, \mathrm{NP}; j = 1, 2, \ldots, n$$

where NP is the number of points in the final mesh.

The remaining columns of Y are not used.

13: NP – INTEGER *Input/Output*

On entry: determines whether a default mesh or user-supplied mesh is used.

NP $= 0$

A default value of 4 for NP and a corresponding equispaced mesh $\mathrm{X}(1), \mathrm{X}(2), \ldots, \mathrm{X}(\mathrm{NP})$ are used.

NP ≥ 4

You must define an initial mesh X as in (4) above.

On exit: the number of points in the final (returned) mesh.

14: W(LW) – ***double precision*** array *Workspace*

15: LW – INTEGER *Input*

On entry: the dimension of the array W as declared in the (sub)program from which D02GBF is called.

Constraint: LW $\geq$ MNP $\times \left(3\text{N}^2 + 5\text{N} + 2\right) + 3\text{N}^2 + 5\text{N}$.

16: IW(LIW) – INTEGER array *Workspace*

17: LIW – INTEGER *Input*

On entry: the dimension of the array IW as declared in the (sub)program from which D02GBF is called.

Constraint: LIW $\geq$ MNP $\times (2\text{N} + 1) + \text{N}$.

18: IFAIL – INTEGER *Input/Output*

For this routine, the normal use of IFAIL is extended to control the printing of error and warning messages as well as specifying hard or soft failure (see Chapter P01).

On entry: IFAIL must be set to a value with the decimal expansion *cba*, where each of the decimal digits *c*, *b* and *a* must have a value of 0 or 1.

$a = 0$ specifies hard failure, otherwise soft failure;

$b = 0$ suppresses error messages, otherwise error messages will be printed (see Section 6);

$c = 0$ suppresses warning messages, otherwise warning messages will be printed (see Section 6).

The recommended value for inexperienced users is 110 (i.e., hard failure with all messages printed).

On exit: IFAIL $= 0$ unless the routine detects an error (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 $= 1$

One or more of the parameters N, TOL, NP, MNP, LW or LIW is incorrectly set, B $\leq$ A or the condition (4) on X is not satisfied.

IFAIL $= 2$

There are three possible reasons for this error exit to be taken:

(i) one of the matrices C or D is identically zero (that is the problem is of initial value and not boundary-value type). In this case, IW$(1) = 0$ on exit;

(ii) a row of C and the corresponding row of D are identically zero (that is the boundary conditions are rank deficient). In this case, on exit IW(1) contains the index of the first such row encountered; and

(iii) more than n of the columns of the n by $2n$ matrix $[C, D]$ are identically zero (that is the boundary conditions are rank deficient). In this case, on exit IW(1) contains minus the number of non-identically zero columns.

IFAIL = 3

The routine has failed to find a solution to the specified accuracy. There are a variety of possible reasons including:

(i) the boundary conditions are rank deficient, which may be indicated by the message that the Jacobian is singular. However this is an unlikely explanation for the error exit as all rank deficient boundary conditions should lead instead to error exits with either IFAIL = 2 or 5; see also (iv) below;

(ii) not enough mesh points are permitted in order to attain the required accuracy. This is indicated by NP = MNP on return from a call to D02GBF. This difficulty may be aggravated by a poor initial choice of mesh points;

(iii) the accuracy requested cannot be attained on the computer being used; and

(iv) an unlikely combination of values of $\mathrm{F}(x)$ has led to a singular Jacobian. The error should not persist if more mesh points are allowed.

IFAIL = 4

A serious error has occurred in a call to D02GBF. Check all array subscripts and (sub)program parameter lists in calls to D02GBF. Seek expert help.

IFAIL = 5

There are two possible reasons for this error exit which occurs when checking the rank of the boundary conditions by reduction to a row echelon form:

(i) at least one row of the n by $2n$ matrix $[C, D]$ is a linear combination of the other rows and hence the boundary conditions are rank deficient. The index of the first such row encountered is given by $\mathrm{IW}(1)$ on exit; and

(ii) as (i) but the rank deficiency implied by this error exit has only been determined up to a numerical tolerance. Minus the index of the first such row encountered is given by $\mathrm{IW}(1)$ on exit.

In case (ii) above there is some doubt as to the rank deficiency of the boundary conditions. However even if the boundary conditions are not rank deficient they are not posed in a suitable form for use with this routine.

For example, if

$$C = \begin{pmatrix} 1 & 0 \\ 1 & \epsilon \end{pmatrix}, \quad D = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, \quad \gamma = \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix}$$

and ϵ is small enough, this error exit is likely to be taken. A better form for the boundary conditions in this case would be

$$C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad D = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \gamma = \begin{pmatrix} \gamma_1 \\ \epsilon^{-1}(\gamma_2 - \gamma_1) \end{pmatrix}.$$

7 Accuracy

The solution returned by the routine will be accurate to your tolerance as defined by the relation (3) except in extreme circumstances. If too many points are specified in the initial mesh, the solution may be more accurate than requested and the error may not be approximately equidistributed.

8 Further Comments

The time taken by D02GBF depends on the difficulty of the problem, the number of mesh points (and meshes) used and the number of deferred corrections.

You are strongly recommended to set IFAIL to obtain self-explanatory error messages, and also monitoring information about the course of the computation. You may select the channel numbers on which this

output is to appear by calls of X04AAF (for error messages) or X04ABF (for monitoring information) – see Section 9 for an example. Otherwise the default channel numbers will be used, as specified in the Users' Note.

In the case where you wish to solve a sequence of similar problems, the use of the final mesh from one case is strongly recommended as the initial mesh for the next.

9 Example

We solve the problem (written as a first-order system)

$$\epsilon y'' + y' = 0$$

with boundary conditions

$$y(0) = 0, \quad y(1) = 1$$

for the cases $\epsilon = 10^{-1}$ and $\epsilon = 10^{-2}$ using the default initial mesh in the first case, and the final mesh of the first case as initial mesh for the second (more difficult) case. We give the solution and the error at each mesh point to illustrate the accuracy of the method given the accuracy request TOL = 1.0D − 3.

Note the call to X04ABF prior to the call to D02GBF.

9.1 Program Text

```
*     D02GBF Example Program Text
*     Mark 14 Revised. NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          N, MNP, LW, LIW
      PARAMETER        (N=2,MNP=70,LW=MNP*(3*N*N+5*N+2)+3*N*N+5*N,
     +                 LIW=MNP*(2*N+1)+N)
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Scalars in Common ..
      DOUBLE PRECISION EPS
*     .. Local Scalars ..
      DOUBLE PRECISION A, B, TOL
      INTEGER          I, IFAIL, J, NP
*     .. Local Arrays ..
      DOUBLE PRECISION C(N,N), D(N,N), GAM(N), W(LW), X(MNP), Y(N,MNP)
      INTEGER          IW(LIW)
*     .. External Subroutines ..
      EXTERNAL         D02GBF, FCNF, FCNG, X04ABF
*     .. Common blocks ..
      COMMON           EPS
*     .. Executable Statements ..
      WRITE (NOUT,*) 'D02GBF Example Program Results'
      TOL = 1.0D-3
      NP = 0
      A = 0.0D0
      B = 1.0D0
      CALL X04ABF(1,NOUT)
      DO 40 I = 1, N
         GAM(I) = 0.0D0
         DO 20 J = 1, N
            C(I,J) = 0.0D0
            D(I,J) = 0.0D0
   20    CONTINUE
   40 CONTINUE
      C(1,1) = 1.0D0
      D(2,1) = 1.0D0
      GAM(2) = 1.0D0
      DO 60 I = 1, 2
         EPS = 10.0D0**(-I)
         WRITE (NOUT,*)
         WRITE (NOUT,99999) 'Problem with epsilon = ', EPS
*        * Set IFAIL to 111 to obtain monitoring information *
         IFAIL = 11
*
```

```
         CALL D02GBF(A,B,N,TOL,FCNF,FCNG,C,D,GAM,MNP,X,Y,NP,W,LW,IW,LIW,
     +               IFAIL)
*
         WRITE (NOUT,*)
         IF (IFAIL.EQ.0) THEN
            WRITE (NOUT,99998) 'Approximate solution on final mesh of ',
     +        NP, ' points'
            WRITE (NOUT,*) '       X(I)     Y(1,I)'
            WRITE (NOUT,99997) (X(J),Y(1,J),J=1,NP)
         ELSE
            WRITE (NOUT,99996) ' IFAIL = ', IFAIL
            STOP
         END IF
   60 CONTINUE
      STOP
*
99999 FORMAT (1X,A,E10.2)
99998 FORMAT (1X,A,I2,A)
99997 FORMAT (1X,2F11.4)
99996 FORMAT (1X,A,I3)
      END
*
      SUBROUTINE FCNF(X,F)
*     .. Parameters ..
      INTEGER          N
      PARAMETER        (N=2)
*     .. Scalar Arguments ..
      DOUBLE PRECISION X
*     .. Array Arguments ..
      DOUBLE PRECISION F(N,N)
*     .. Scalars in Common ..
      DOUBLE PRECISION EPS
*     .. Common blocks ..
      COMMON           EPS
*     .. Executable Statements ..
      F(1,1) = 0.0D0
      F(1,2) = 1
      F(2,1) = 0.0D0
      F(2,2) = -1.0D0/EPS
      RETURN
      END
*
      SUBROUTINE FCNG(X,G)
*     .. Parameters ..
      INTEGER          N
      PARAMETER        (N=2)
*     .. Scalar Arguments ..
      DOUBLE PRECISION X
*     .. Array Arguments ..
      DOUBLE PRECISION G(N)
*     .. Executable Statements ..
      G(1) = 0.0D0
      G(2) = 0.0D0
      RETURN
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 D02GBF Example Program Results

 Problem with epsilon =   0.10E+00

 Approximate solution on final mesh of 15 points
        X(I)     Y(1,I)
      0.0000     0.0000
      0.0278     0.2425
```

```
      0.0556     0.4263
      0.1111     0.6708
      0.1667     0.8112
      0.2222     0.8917
      0.2778     0.9379
      0.3333     0.9644
      0.4444     0.9883
      0.5556     0.9962
      0.6667     0.9988
      0.7500     0.9995
      0.8333     0.9998
      0.9167     0.9999
      1.0000     1.0000

 Problem with epsilon =   0.10E-01

 Approximate solution on final mesh of 49 points
        X(I)     Y(1,I)
      0.0000     0.0000
      0.0009     0.0884
      0.0019     0.1690
      0.0028     0.2425
      0.0037     0.3095
      0.0046     0.3706
      0.0056     0.4262
      0.0065     0.4770
      0.0074     0.5232
      0.0083     0.5654
      0.0093     0.6038
      0.0111     0.6708
      0.0130     0.7265
      0.0148     0.7727
      0.0167     0.8111
      0.0185     0.8431
      0.0204     0.8696
      0.0222     0.8916
      0.0241     0.9100
      0.0259     0.9252
      0.0278     0.9378
      0.0306     0.9529
      0.0333     0.9643
      0.0361     0.9730
      0.0389     0.9795
      0.0417     0.9845
      0.0444     0.9883
      0.0472     0.9911
      0.0500     0.9933
      0.0528     0.9949
      0.0556     0.9961
      0.0648     0.9985
      0.0741     0.9994
      0.0833     0.9998
      0.0926     0.9999
      0.1019     1.0000
      0.1111     1.0000
      0.1389     1.0000
      0.1667     1.0000
      0.2222     1.0000
      0.2778     1.0000
      0.3333     1.0000
      0.4444     1.0000
      0.5556     1.0000
      0.6667     1.0000
      0.7500     1.0000
      0.8333     1.0000
      0.9167     1.0000
      1.0000     1.0000
```