NAG Toolbox

nag_lapack_dgbsvx (f07bb)

1 Purpose

nag_lapack_dgbsvx (f07bb) uses the LU factorization to compute the solution to a real system of linear equations

$$AX = B \quad \text{or} \quad A^{\mathrm{T}}X = B,$$

where A is an n by n band matrix with k_l subdiagonals and k_u superdiagonals, and X and B are n by r matrices. Error bounds on the solution and a condition estimate are also provided.

2 Syntax

```
[ab, afb, ipiv, equed, r, c, b, x, rcond, ferr, berr, work, info] =
nag_lapack_dgbsvx(fact, trans, kl, ku, ab, afb, ipiv, equed, r, c, b, 'n', n,
'nrhs_p', nrhs_p)
[ab, afb, ipiv, equed, r, c, b, x, rcond, ferr, berr, work, info] = f07bb(fact,
trans, kl, ku, ab, afb, ipiv, equed, r, c, b, 'n', n, 'nrhs_p', nrhs_p)
```

3 Description

nag_lapack_dgbsvx (f07bb) performs the following steps:

1. **Equilibration**

   The linear system to be solved may be badly scaled. However, the system can be equilibrated as a first stage by setting **fact** = 'E'. In this case, real scaling factors are computed and these factors then determine whether the system is to be equilibrated. Equilibrated forms of the systems $AX = B$ and $A^{\mathrm{T}}X = B$ are

   $$(D_R A D_C)\left(D_C^{-1}X\right) = D_R B$$

   and

   $$(D_R A D_C)^{\mathrm{T}}\left(D_R^{-1}X\right) = D_C B,$$

   respectively, where D_R and D_C are diagonal matrices, with positive diagonal elements, formed from the computed scaling factors.

   When equilibration is used, A will be overwritten by $D_R A D_C$ and B will be overwritten by $D_R B$ (or $D_C B$ when the solution of $A^{\mathrm{T}}X = B$ is sought).

2. **Factorization**

   The matrix A, or its scaled form, is copied and factored using the LU decomposition

   $$A = PLU,$$

   where P is a permutation matrix, L is a unit lower triangular matrix, and U is upper triangular.

   This stage can be by-passed when a factored matrix (with scaled matrices and scaling factors) are supplied; for example, as provided by a previous call to nag_lapack_dgbsvx (f07bb) with the same matrix A.

3. **Condition Number Estimation**

   The LU factorization of A determines whether a solution to the linear system exists. If some diagonal element of U is zero, then U is exactly singular, no solution exists and the function returns with a failure. Otherwise the factorized form of A is used to estimate the condition number

of the matrix A. If the reciprocal of the condition number is less than ***machine precision*** then a warning code is returned on final exit.

4. **Solution**

   The (equilibrated) system is solved for X ($D_C^{-1}X$ or $D_R^{-1}X$) using the factored form of A (D_RAD_C).

5. **Iterative Refinement**

   Iterative refinement is applied to improve the computed solution matrix and to calculate error bounds and backward error estimates for the computed solution.

6. **Construct Solution Matrix *X***

   If equilibration was used, the matrix X is premultiplied by D_C (if **trans** = 'N') or D_R (if **trans** = 'T' or 'C') so that it solves the original system before equilibration.

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia http://www.netlib.org/lapack/lug

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

Higham N J (2002) *Accuracy and Stability of Numerical Algorithms* (2nd Edition) SIAM, Philadelphia

5 Parameters

5.1 Compulsory Input Parameters

1: **fact** – CHARACTER(1)

Specifies whether or not the factorized form of the matrix A is supplied on entry, and if not, whether the matrix A should be equilibrated before it is factorized.

fact = 'F'
: **afb** and **ipiv** contain the factorized form of A. If **equed** $\neq$ 'N', the matrix A has been equilibrated with scaling factors given by **r** and **c**. **ab**, **afb** and **ipiv** are not modified.

fact = 'N'
: The matrix A will be copied to **afb** and factorized.

fact = 'E'
: The matrix A will be equilibrated if necessary, then copied to **afb** and factorized.

Constraint: **fact** = 'F', 'N' or 'E'.

2: **trans** – CHARACTER(1)

Specifies the form of the system of equations.

trans = 'N'
: $AX = B$ (No transpose).

trans = 'T' or 'C'
: $A^{\mathrm{T}}X = B$ (Transpose).

Constraint: **trans** = 'N', 'T' or 'C'.

3: **kl** – INTEGER

k_l, the number of subdiagonals within the band of the matrix A.

Constraint: **kl** ≥ 0.

4: **ku** – INTEGER

k_u, the number of superdiagonals within the band of the matrix A.

Constraint: **ku** ≥ 0.

5: **ab**$(ldab,:)$ – REAL (KIND=nag_wp) array

The first dimension of the array **ab** must be at least **kl** + **ku** + 1.

The second dimension of the array **ab** must be at least $\max(1,\mathbf{n})$.

The n by n coefficient matrix A.

The matrix is stored in rows 1 to $k_l + k_u + 1$, more precisely, the element A_{ij} must be stored in

$$\mathbf{ab}(k_u + 1 + i - j, j) \quad \text{for } \max(1, j - k_u) \leq i \leq \min(n, j + k_l).$$

See Section 9 for further details.

If **fact** = 'F' and **equed** $\neq$ 'N', A must have been equilibrated by the scaling factors in **r** and/or **c**.

6: **afb**$(ldafb,:)$ – REAL (KIND=nag_wp) array

The first dimension of the array **afb** must be at least $2 \times$ **kl** + **ku** + 1.

The second dimension of the array **afb** must be at least $\max(1,\mathbf{n})$.

If **fact** = 'N' or 'E', **afb** need not be set.

If **fact** = 'F', details of the LU factorization of the n by n band matrix A, as computed by nag_lapack_dgbtrf (f07bd).

The upper triangular band matrix U, with $k_l + k_u$ superdiagonals, is stored in rows 1 to $k_l + k_u + 1$ of the array, and the multipliers used to form the matrix L are stored in rows $k_l + k_u + 2$ to $2k_l + k_u + 1$.

If **equed** $\neq$ 'N', **afb** is the factorized form of the equilibrated matrix A.

7: **ipiv**$(:)$ – INTEGER array

The dimension of the array **ipiv** must be at least $\max(1,\mathbf{n})$

If **fact** = 'N' or 'E', **ipiv** need not be set.

If **fact** = 'F', **ipiv** contains the pivot indices from the factorization $A = LU$, as computed by nag_lapack_dgbtrf (f07bd); row i of the matrix was interchanged with row **ipiv**(i).

8: **equed** – CHARACTER(1)

If **fact** = 'N' or 'E', **equed** need not be set.

If **fact** = 'F', **equed** must specify the form of the equilibration that was performed as follows:

if **equed** = 'N', no equilibration;

if **equed** = 'R', row equilibration, i.e., A has been premultiplied by D_R;

if **equed** = 'C', column equilibration, i.e., A has been postmultiplied by D_C;

if **equed** = 'B', both row and column equilibration, i.e., A has been replaced by $D_R A D_C$.

Constraint: if **fact** = 'F', **equed** = 'N', 'R', 'C' or 'B'.

9: **r**$(:)$ – REAL (KIND=nag_wp) array

The dimension of the array **r** must be at least $\max(1,\mathbf{n})$

If **fact** = 'N' or 'E', **r** need not be set.

If **fact** = 'F' and **equed** = 'R' or 'B', **r** must contain the row scale factors for A, D_R; each element of **r** must be positive.

10: **c**(:) – REAL (KIND=nag_wp) array

The dimension of the array **c** must be at least $\max(1, \mathbf{n})$

If **fact** = 'N' or 'E', **c** need not be set.

If **fact** = 'F' or **equed** = 'C' or 'B', **c** must contain the column scale factors for A, D_C; each element of **c** must be positive.

11: **b**(ldb, :) – REAL (KIND=nag_wp) array

The first dimension of the array **b** must be at least $\max(1, \mathbf{n})$.

The second dimension of the array **b** must be at least $\max(1, \mathbf{nrhs\_p})$.

The n by r right-hand side matrix B.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the first dimension of the array **b** and the second dimension of the arrays **ab**, **afb**, **ipiv**, **r**, **c**.

n, the number of linear equations, i.e., the order of the matrix A.

Constraint: $\mathbf{n} \geq 0$.

2: **nrhs_p** – INTEGER

Default: the second dimension of the array **b**.

r, the number of right-hand sides, i.e., the number of columns of the matrix B.

Constraint: $\mathbf{nrhs\_p} \geq 0$.

5.3 Output Parameters

1: **ab**($ldab$, :) – REAL (KIND=nag_wp) array

The first dimension of the array **ab** will be $\mathbf{kl} + \mathbf{ku} + 1$.

The second dimension of the array **ab** will be $\max(1, \mathbf{n})$.

If **fact** = 'F' or 'N', or if **fact** = 'E' and **equed** = 'N', **ab** is not modified.

If **equed** $\neq$ 'N' then, if no constraints are violated, A is scaled as follows:

if **equed** = 'R', $A = D_r A$;

if **equed** = 'C', $A = A D_c$;

if **equed** = 'B', $A = D_r A D_c$.

2: **afb**($ldafb$, :) – REAL (KIND=nag_wp) array

The first dimension of the array **afb** will be $2 \times \mathbf{kl} + \mathbf{ku} + 1$.

The second dimension of the array **afb** will be $\max(1, \mathbf{n})$.

If **fact** = 'F', **afb** is unchanged from entry.

Otherwise, if no constraints are violated, then if **fact** = 'N', **afb** returns details of the LU factorization of the band matrix A, and if **fact** = 'E', **afb** returns details of the LU factorization of the equilibrated band matrix A (see the description of **ab** for the form of the equilibrated matrix).

3: **ipiv**(:) – INTEGER array

The dimension of the array **ipiv** will be $\max(1, \mathbf{n})$

If **fact** = 'F', **ipiv** is unchanged from entry.

Otherwise, if no constraints are violated, **ipiv** contains the pivot indices that define the permutation matrix P; at the ith step row i of the matrix was interchanged with row **ipiv**(i). **ipiv**$(i) = i$ indicates a row interchange was not required.

If **fact** = 'N', the pivot indices are those corresponding to the factorization $A = LU$ of the original matrix A.

If **fact** = 'E', the pivot indices are those corresponding to the factorization of $A = LU$ of the equilibrated matrix A.

4: **equed** – CHARACTER(1)

If **fact** = 'F', **equed** is unchanged from entry.

Otherwise, if no constraints are violated, **equed** specifies the form of equilibration that was performed as specified above.

5: **r**(:) – REAL (KIND=nag_wp) array

The dimension of the array **r** will be $\max(1, \mathbf{n})$

If **fact** = 'F', **r** is unchanged from entry.

Otherwise, if no constraints are violated and **equed** = 'R' or 'B', **r** contains the row scale factors for A, D_R, such that A is multiplied on the left by D_R; each element of **r** is positive.

6: **c**(:) – REAL (KIND=nag_wp) array

The dimension of the array **c** will be $\max(1, \mathbf{n})$

If **fact** = 'F', **c** is unchanged from entry.

Otherwise, if no constraints are violated and **equed** = 'C' or 'B', **c** contains the row scale factors for A, D_C; each element of **c** is positive.

7: **b**$(ldb, :)$ – REAL (KIND=nag_wp) array

The first dimension of the array **b** will be $\max(1, \mathbf{n})$.

The second dimension of the array **b** will be $\max(1, \mathbf{nrhs\_p})$.

If **equed** = 'N', **b** is not modified.

If **trans** = 'N' and **equed** = 'R' or 'B', **b** stores $D_R B$.

If **trans** = 'T' or 'C' and **equed** = 'C' or 'B', **b** stores $D_C B$.

8: **x**$(ldx, :)$ – REAL (KIND=nag_wp) array

The first dimension of the array **x** will be $\max(1, \mathbf{n})$.

The second dimension of the array **x** will be $\max(1, \mathbf{nrhs\_p})$.

If **info** $= 0$ or $\mathbf{n} + 1$, the n by r solution matrix X to the original system of equations. Note that the arrays A and B are modified on exit if **equed** $\neq$ 'N', and the solution to the equilibrated system is $D_C^{-1}X$ if **trans** = 'N' and **equed** = 'C' or 'B', or $D_R^{-1}X$ if **trans** = 'T' or 'C' and **equed** = 'R' or 'B'.

9: **rcond** – REAL (KIND=nag_wp)

If no constraints are violated, an estimate of the reciprocal condition number of the matrix A (after equilibration if that is performed), computed as **rcond** $= 1.0/\left(\|A\|_1 \|A^{-1}\|_1\right)$.

10: **ferr**(**nrhs_p**) – REAL (KIND=nag_wp) array

If **info** $= 0$ or $\mathbf{n} + 1$, an estimate of the forward error bound for each computed solution vector, such that $\|\hat{x}_j - x_j\|_\infty / \|x_j\|_\infty \leq$ **ferr**(j) where $\hat{x}_j$ is the jth column of the computed solution

returned in the array **x** and x_j is the corresponding column of the exact solution X. The estimate is as reliable as the estimate for **rcond**, and is almost always a slight overestimate of the true error.

11: **berr**(**nrhs_p**) – REAL (KIND=nag_wp) array

If **info** $= 0$ or $\mathbf{n}+1$, an estimate of the component-wise relative backward error of each computed solution vector $\hat{x}_j$ (i.e., the smallest relative change in any element of A or B that makes $\hat{x}_j$ an exact solution).

12: **work**(**max**($\mathbf{1}, \mathbf{3} \times \mathbf{n}$)) – REAL (KIND=nag_wp) array

If **info** $= 0$, **work**(1) contains the reciprocal pivot growth factor $\max|a_{ij}|/\max|u_{ij}|$. If **work**$(1)$ is much less than 1, then the stability of the LU factorization of the (equilibrated) matrix A could be poor. This also means that the solution X, condition estimator **rcond**, and forward error bound **ferr** could be unreliable. If the factorization fails with **info** > 0 and **info** $\leq$ **n**, **work**(1) contains the reciprocal pivot growth factor for the leading **info** columns of A.

13: **info** – INTEGER

info $= 0$ unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

info < 0

If **info** $= -i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

info > 0 and **info** $\leq$ **n** (*warning*)

Element $\langle value \rangle$ of the diagonal is exactly zero. The factorization has been completed, but the factor U is exactly singular, so the solution and error bounds could not be computed. **rcond** $= 0.0$ is returned.

info $=$ **n** $+ 1$ (*warning*)

U is nonsingular, but **rcond** is less than ***machine precision***, meaning that the matrix is singular to working precision. Nevertheless, the solution and error bounds are computed because there are a number of situations where the computed solution can be more accurate than the value of **rcond** would suggest.

7 Accuracy

For each right-hand side vector b, the computed solution $\hat{x}$ is the exact solution of a perturbed system of equations $(A+E)\hat{x} = b$, where

$$|E| \leq c(n)\epsilon P|L||U|,$$

$c(n)$ is a modest linear function of n, and ϵ is the ***machine precision***. See Section 9.3 of Higham (2002) for further details.

If x is the true solution, then the computed solution $\hat{x}$ satisfies a forward error bound of the form

$$\frac{\|x-\hat{x}\|_\infty}{\|\hat{x}\|_\infty} \leq w_c \,\mathrm{cond}(A,\hat{x},b)$$

where $\mathrm{cond}(A,\hat{x},b) = \big\||A^{-1}|(|A||\hat{x}|+|b|)\big\|_\infty / \|\hat{x}\|_\infty \leq \mathrm{cond}(A) = \big\||A^{-1}||A|\big\|_\infty \leq \kappa_\infty(A)$. If $\hat{x}$ is the jth column of X, then w_c is returned in **berr**(j) and a bound on $\|x-\hat{x}\|_\infty/\|\hat{x}\|_\infty$ is returned in **ferr**(j). See Section 4.4 of Anderson *et al.* (1999) for further details.

8 Further Comments

The band storage scheme for the array **ab** is illustrated by the following example, when $n = 6$, $k_l = 1$, and $k_u = 2$. Storage of the band matrix A in the array **ab**:

$$\begin{pmatrix} * & * & a_{13} & a_{24} & a_{35} & a_{46} \\ * & a_{12} & a_{23} & a_{34} & a_{45} & a_{56} \\ a_{11} & a_{22} & a_{33} & a_{44} & a_{55} & a_{66} \\ a_{21} & a_{32} & a_{43} & a_{54} & a_{65} & * \end{pmatrix}$$

The total number of floating-point operations required to solve the equations $AX = B$ depends upon the pivoting required, but if $n \gg k_l + k_u$ then it is approximately bounded by $O(nk_l(k_l + k_u))$ for the factorization and $O(n(2k_l + k_u)r)$ for the solution following the factorization. The condition number estimation typically requires between four and five solves and never more than eleven solves, following the factorization. The solution is then refined, and the errors estimated, using iterative refinement; see nag_lapack_dgbrfs (f07bh) for information on the floating-point operations required.

In practice the condition number estimator is very reliable, but it can underestimate the true condition number; see Section 15.3 of Higham (2002) for further details.

The complex analogue of this function is nag_lapack_zgbsvx (f07bp).

9 Example

This example solves the equations

$$AX = B,$$

where A is the band matrix

$$A = \begin{pmatrix} -0.23 & 2.54 & -3.66 & 0 \\ -6.98 & 2.46 & -2.73 & -2.13 \\ 0 & 2.56 & 2.46 & 4.07 \\ 0 & 0 & -4.78 & -3.82 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 4.42 & -36.01 \\ 27.13 & -31.67 \\ -6.14 & -1.16 \\ 10.50 & -25.82 \end{pmatrix}.$$

Estimates for the backward errors, forward errors, condition number and pivot growth are also output, together with information on the equilibration of A.

9.1 Program Text

```
     function f07bb_example

fprintf('f07bb example results\n\n');

kl = nag_int(1);
ku = nag_int(2);
ab = [ 0,      0,    -3.66, -2.13;
       0,      2.54, -2.73,  4.07;
      -0.23,   2.46,  2.46, -3.82;
      -6.98,   2.56, -4.78,  0];
b  = [ 4.42, -36.01;
      27.13, -31.67;
      -6.14,  -1.16;
      10.50, -25.82];
[n,nrhs] = size(b);

% Input parameter initialization
afb   = zeros(2*kl+ku+1, n);
ipiv  = zeros(n,1,nag_int_name);
r     = zeros(n, 1);
c     = zeros(n, 1);
fact  = 'Equilibration';
trans = 'No transpose';
equed = 'No';

% Solve
[ab, afb, ipiv, equed, r, c, b, x, rcond, ferr, berr, work, info] = ...
```

```
  f07bb( ...
         fact, trans, kl, ku, ab, afb, ipiv, equed, r, c, b);

fprintf('Solution is x:\n');
disp(x);
fprintf('\nApproximate condition number = %9.3f\n',1/rcond);
fprintf('Approximate forward  errors  :\n');
fprintf('                               %11.1e\n',ferr);
fprintf('Approximate backward errors  :\n')
fprintf('                               %11.1e\n',berr);
```

9.2 Program Results

```
     f07bb example results

Solution is x:
   -2.0000    1.0000
    3.0000   -4.0000
    1.0000    7.0000
   -4.0000   -2.0000

Approximate condition number =    56.409
Approximate forward  errors  :
                                   1.6e-14
                                   1.9e-14
Approximate backward errors  :
                                   1.1e-16
                                   9.9e-17
```