NAG Toolbox for MATLAB

f11bs

1 Purpose

f11bs is an iterative solver for a complex general (non-Hermitian) system of simultaneous linear equations; f11bs is the second in a suite of three functions, where the first function, f11br, must be called prior to f11bs to set up the suite, and the third function in the suite, f11bt, can be used to return additional information about the computation.

These three functions are suitable for the solution of large sparse general (non-Hermitian) systems of equations.

2 Syntax

```
[irevcm, u, v, work, ifail] = f11bs(irevcm, u, v, wgt, work, 'lwork', lwork)
```

3 Description

f11bs solves the general (non-Hermitian) system of linear simultaneous equations $Ax = b$ using one of four available methods: RGMRES, the preconditioned restarted generalized minimum residual method (see Saad and Schultz 1986); CGS, the preconditioned conjugate gradient squared method (see Sonneveld 1989); Bi-CGSTAB(ℓ), the bi-conjugate gradient stabilized method of order ℓ (see Van der Vorst 1989 and Sleijpen and Fokkema 1993); or TFQMR, the quasi-minimal transpose-free quasi-minimum residual method (see Freund and Nachtigal 1991 and Freund 1993).

For a general description of the methods employed you are referred to Section 3 of the document for f11br.

f11bs can solve the system after the first function in the suite, f11br, has been called to initialize the computation and specify the method of solution. The third function in the suite, f11bt, can be used to return additional information generated by the computation during monitoring steps and after f11bs has completed its tasks.

f11bs uses **reverse communication**, i.e., it returns repeatedly to the calling program with the parameter **irevcm** (see Section 5) set to specified values which require the calling program to carry out one of the following tasks:

compute the matrix-vector product $v = Au$ or $v = A^{\mathrm{H}}u$ (the four methods require the matrix transpose-vector product only if $\|A\|_1$ or $\|A\|_\infty$ is estimated internally by Higham's method (see Higham 1988));

solve the preconditioning equation $Mv = u$;

notify the completion of the computation;

allow the calling program to monitor the solution.

Through the parameter **irevcm** the calling program can cause immediate or tidy termination of the execution. On final exit, the last iterates of the solution and of the residual vectors of the original system of equations are returned.

Reverse communication has the following advantages.

1. Maximum flexibility in the representation and storage of sparse matrices: all matrix operations are performed outside the solver function, thereby avoiding the need for a complicated interface with enough flexibility to cope with all types of storage schemes and sparsity patterns. This applies also to preconditioners.
2. Enhanced user interaction: the progress of the solution can be closely monitored by you and tidy or immediate termination can be requested. This is useful, for example, when alternative termination

criteria are to be employed or in case of failure of the external functions used to perform matrix operations.

4 References

Freund R W 1993 A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems *SIAM J. Sci. Comput.* **14** 470–482

Freund R W and Nachtigal N 1991 QMR: a Quasi-Minimal Residual Method for Non-Hermitian Linear Systems *Numer. Math.* **60** 315–339

Higham N J 1988 FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation *ACM Trans. Math. Software* **14** 381–396

Saad Y and Schultz M 1986 GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems *SIAM J. Sci. Statist. Comput.* **7** 856–869

Sleijpen G L G and Fokkema D R 1993 BiCGSTAB(ℓ) for linear equations involving matrices with complex spectrum *ETNA* **1** 11–32

Sonneveld P 1989 CGS, a fast Lanczos-type solver for nonsymmetric linear systems *SIAM J. Sci. Statist. Comput.* **10** 36–52

Van der Vorst H 1989 Bi-CGSTAB, a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems *SIAM J. Sci. Statist. Comput.* **13** 631–644

5 Parameters

Note: this function uses **reverse communication.** Its use involves an initial entry, intermediate exits and re-entries, and a final exit, as indicated by the **parameter IREVCM**. Between intermediate exits and re-entries, **all parameters other than irevcm and v must remain unchanged**.

5.1 Compulsory Input Parameters

1: **irevcm – int32 scalar**

On initial entry: **irevcm** $= 0$, otherwise an error condition will be raised.

On intermediate re-entry: must either be unchanged from its previous exit value, or can have one of the following values.

5 Tidy termination: the computation will terminate at the end of the current iteration. Further reverse communication exits may occur depending on when the termination request is issued. f11bs will then return with the termination code **irevcm** $= 4$. Note that before calling f11bs with **irevcm** $= 5$ the calling program must have performed the tasks required by the value of **irevcm** returned by the previous call to f11bs, otherwise subsequently returned values may be invalid.

6 Immediate termination: f11bs will return immediately with termination code **irevcm** $= 4$ and with any useful information available. This includes the last iterate of the solution. The residual vector is generally not available. f11bs will then return with the termination code **irevcm** $= 4$.
Immediate termination may be useful, for example, when errors are detected during matrix-vector multiplication or during the solution of the preconditioning equation.

Changing **irevcm** to any other value between calls will result in an error.

Constraint: on initial entry, **irevcm** $= 0$; on re-entry, either **irevcm** must either remain unchanged or be reset to 5 or 6.

2: **u**$(*)$ **– complex array**

Note: the dimension of the array **u** must be at least n.

On initial entry: an initial estimate, x_0, of the solution of the system of equations $Ax = b$.

On intermediate re-entry: must remain unchanged.

3: **v**$(*)$ **– complex array**

Note: the dimension of the array **v** must be at least n.

On initial entry: the right-hand side b of the system of equations $Ax = b$.

On intermediate re-entry: the returned value of **irevcm** determines the contents of **v** in the following way:

irevcm $= -1$, 1 or 2

v must store the vector v, the result of the operation specified by the value of **irevcm** returned by the previous call to f11bs.

irevcm $= 3$

v must remain unchanged.

4: **wgt**$(*)$ **– double array**

Note: the dimension of the array **wgt** must be at least $\max(1, n)$.

The user-supplied weights, if these are to be used in the computation of the vector norms in the termination criterion (see Sections 3 and 5 of the document for f11br).

5: **work**(**lwork**) **– complex array**

On initial entry: the workspace **work** as returned by f11br (see also Section 5 of the document for f11br).

On intermediate re-entry: must remain unchanged.

5.2 Optional Input Parameters

1: **lwork – int32 scalar**

Default: The dimension of the array **work**.

On initial entry: The required amount of workspace is as follows:

Method	**Requirements**
RGMRES	**lwork** $= 120 + n(m+3) + m(m+5) + 1$, where m is the dimension of the basis
CGS	**lwork** $= 120 + 7n$
Bi-CGSTAB(ℓ)	**lwork** $= 120 + (2n + \ell)(\ell + 2) + p$, where ℓ is the order of the method
TFQMR	**lwork** $= 120 + 10n$,

where

$p = 2n$, if $\ell > 1$ and **iterm** $= 2$ was supplied;

$p = n$, if $\ell > 1$ and a preconditioner is used or **iterm** $= 2$ was supplied;

$p = 0$, otherwise.

Constraint: **lwork** $\geq$ **lwreq**, where **lwreq** is returned by f11br.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **irevcm – int32 scalar**

On intermediate exit: has the following meanings.

-1 The calling program must compute the matrix-vector product $v = A^{\mathrm{H}}u$, where u and v are stored in **u** and **v**, respectively; RGMRES, CGS and Bi-CGSTAB(ℓ) methods return **irevcm** $= -1$ only if the matrix norm $\|A\|_1$ or $\|A\|_\infty$ is estimated internally using Higham's method. This can only happen if **iterm** $= 1$ in f11br.

1 The calling program must compute the matrix-vector product $v = Au$, where u and v are stored in **u** and **v**, respectively.

2 The calling program must solve the preconditioning equation $Mv = u$, where u and v are stored in **u** and **v**, respectively.

3 Monitoring step: the solution and residual at the current iteration are returned in the arrays **u** and **v**, respectively. No action by the calling program is required. f11bt can be called at this step to return additional information.

On final exit: **irevcm** $= 4$: f11bs has completed its tasks. The value of **ifail** determines whether the iteration has been successfully completed, errors have been detected or the calling program has requested termination.

2: **u**$(*)$ **– complex array**

Note: the dimension of the array **u** must be at least n.

On intermediate exit: the returned value of **irevcm** determines the contents of **u** in the following way:

irevcm $= -1$, 1 or 2

u holds the vector u on which the operation specified by **irevcm** is to be carried out.

irevcm $= 3$

u holds the current iterate of the solution vector.

On final exit: if **ifail** $= 3$ or $-i$, the array **u** is unchanged from the initial entry to f11bs. If **ifail** $= 1$, the array **u** is unchanged from the last entry to f11bs. Otherwise, **u** holds the last available iterate of the solution of the system of equations, for all returned values of **ifail**.

3: **v**$(*)$ **– complex array**

Note: the dimension of the array **v** must be at least n.

On intermediate exit: if **irevcm** $= 3$, **v** holds the current iterate of the residual vector. Note that this is an approximation to the true residual vector. Otherwise, it does not contain any useful information.

On final exit: if **ifail** $= 3$ or $-i$, the array **v** is unchanged from the initial entry to f11bs. If **ifail** $= 1$, the array **v** is unchanged from the last entry to f11bs. If **ifail** $= 0$ or 2, the array **v** contains the true residual vector of the system of equations (see also Section 6). Otherwise, **v** stores the last available iterate of the residual vector unless **ifail** $= 8$ is returned on last entry, in which case **v** is set to 0.0.

4: **work**(**lwork**) **– complex array**

5: **ifail – int32 scalar**

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= -i$

If **ifail** $= -i$, parameter i had an illegal value on entry. The parameters are numbered as follows: 1: **irevcm**, 2: **u**, 3: **v**, 4: **wgt**, 5: **work**, 6: **lwork**, 7: **ifail**.

ifail $= 1$

f11bs has been called again after returning the termination code **irevcm** $= 4$. No further computation has been carried out and all input data and data stored for access by f11bt have remained unchanged.

ifail $= 2$

The required accuracy could not be obtained. However, f11bs has terminated with reasonable accuracy: the last iterate of the residual satisfied the termination criterion but the exact residual $r = b - Ax$, did not. After the first occurrence of this situation, the iteration was restarted once, but f11bs could not improve on the accuracy. This error code usually implies that your problem has been fully and satisfactorily solved to within or close to the accuracy available on your system. Further iterations are unlikely to improve on this situation. You should call f11bt to check the values of the left- and right-hand sides of the termination condition.

ifail $= 3$

f11br was either not called before calling f11bs or it returned an error. The arguments **u** and **v** remain unchanged.

ifail $= 4$

The calling program requested a tidy termination before the solution had converged. The arrays **u** and **v** return the last iterates available of the solution and of the residual vector, respectively.

ifail $= 5$

The solution did not converge within the maximum number of iterations allowed. The arrays **u** and **v** return the last iterates available of the solution and of the residual vector, respectively.

ifail $= 6$

Algorithmic breakdown. The last available iterates of the solution and residuals are returned, although it is possible that they are completely inaccurate.

ifail $= 8$

The calling program requested an immediate termination. However, the array **u** returns the last iterate of the solution, the array **v** returns the last iterate of the residual vector, for the CGS and TFQMR methods only.

7 Accuracy

On completion, i.e., **irevcm** $= 4$ on exit, the arrays **u** and **v** will return the solution and residual vectors, x_k and $r_k = b - Ax_k$, respectively, at the kth iteration, the last iteration performed, unless an immediate termination was requested.

On successful completion, the termination criterion is satisfied to within the user-specified tolerance, as described in f11br. The computed values of the left- and right-hand sides of the termination criterion selected can be obtained by a call to f11bt.

8 Further Comments

The number of operations carried out by f11bs for each iteration is likely to be principally determined by the computation of the matrix-vector products $v = Au$ and by the solution of the preconditioning equation $Mv = u$ in the calling program. Each of these operations is carried out once every iteration.

The number of the remaining operations in f11bs for each iteration is approximately proportional to n.

The number of iterations required to achieve a prescribed accuracy cannot be easily determined at the onset, as it can depend dramatically on the conditioning and spectrum of the preconditioned matrix of the coefficients $\bar{A} = M^{-1}A$ (RGMRES, CGS and TFQMR methods) or $\bar{A} = AM^{-1}$ (Bi-CGSTAB(ℓ) method).

Additional matrix-vector products are required for the computation of $\|A\|_1$ or $\|A\|_\infty$, when this has not been supplied to f11br and is required by the termination criterion employed.

If the termination criterion $\|r_k\|_p \leq \tau\left(\|b\|_p + \|A\|_p, \|x_k\|_p\right)$ is used (see Section 3 of the document for f11br) and $\|x_0\| \gg \|x_k\|$, then the required accuracy cannot be obtained due to loss of significant digits. The iteration is restarted automatically at some suitable point: f11bs sets $x_0 = x_k$ and the computation begins again. For particularly badly scaled problems, more than one restart may be necessary. This does not apply to the RGMRES method which, by its own nature, self-restarts every super-iteration. Naturally, restarting adds to computational costs: it is recommended that the iteration should start from a value x_0 which is as close to the true solution $\tilde{x}$ as can be estimated. Otherwise, the iteration should start from $x_0 = 0$.

9 Example

```
method = 'TFQMR   ';
precon = 'P';
n = int32(8);
m = int32(1);
tol = 1e-08;
maxitn = int32(20);
anorm = 0;
sigmax = 0;
monit = int32(2);
lwork = int32(6000);
nnz=int32(24);
a=complex(zeros(10000,1));
a(1:24)=[complex( 2., 1.),
         complex(-1., 1.),
         complex( 1.,-3.),
         complex( 4., 7.),
         complex(-3., 0.),
         complex( 2., 4.),
         complex(-7.,-5.),
         complex( 2., 1.),
         complex( 3., 2.),
         complex(-4., 2.),
         complex( 0., 1.),
         complex( 5.,-3.),
         complex(-1., 2.),
         complex( 8., 6.),
         complex(-3.,-4.),
         complex(-6.,-2.),
         complex( 5.,-2.),
         complex( 2., 0.),
         complex( 0.,-5.),
         complex(-1., 5.),
         complex( 6., 2.),
         complex(-1., 4.),
         complex( 2., 0.),
         complex( 3., 3.)];
irow=zeros(10000,1,'int32');
irow(1:24) = [int32(1),
```

```
              int32(1),
              int32(1),
              int32(2),
              int32(2),
              int32(2),
              int32(3),
              int32(3),
              int32(4),
              int32(4),
              int32(4),
              int32(4),
              int32(5),
              int32(5),
              int32(5),
              int32(6),
              int32(6),
              int32(6),
              int32(7),
              int32(7),
              int32(7),
              int32(8),
              int32(8),
              int32(8)];
icol=zeros(10000,1,'int32');
icol(1:24) = [int32(1),
              int32(4),
              int32(8),
              int32(1),
              int32(2),
              int32(5),
              int32(3),
              int32(6),
              int32(1),
              int32(3),
              int32(4),
              int32(7),
              int32(2),
              int32(5),
              int32(7),
              int32(1),
              int32(3),
              int32(6),
              int32(3),
              int32(5),
              int32(7),
              int32(2),
              int32(6),
              int32(8)];
ipivp = zeros(n,1,'int32');
ipivq = zeros(n,1,'int32');
irevcm = int32(0);
lfill = int32(0);
dtol = 0;
milu = 'N';
wgt = zeros(8,1);
u = complex(zeros(8,1));
v = [complex(  7., 11.);
     complex(  1., 24.);
     complex(-13.,-18.);
     complex(-10.,  3.);
     complex( 23., 14.);
     complex( 17., -7.);
     complex( 15., -3.);
     complex( -3., 20.)];
[a, irow, icol, ipivp, ipivq, istr, idiag, nnzc, npivm, ifail] = ...
    f11dn(nnz, a, irow, icol, lfill, dtol, milu, ipivp, ipivq);

[lwreq, work, ifail] = ...
       f11br(method, precon, n, m, tol, maxitn, anorm, sigmax, monit,
lwork, 'norm_p', '1');
```

```
while (irevcm ~= 4)
  [irevcm, u, v, work, ifail] = f11bs(irevcm, u, v, wgt, work);

  if (irevcm == -1)
    [v, ifail] = f11xn('T', a(1:nnz), irow(1:nnz), icol(1:nnz), 'N', u);
  elseif (irevcm == 1)
    [v, ifail] = f11xn('N', a(1:nnz), irow(1:nnz), icol(1:nnz), 'N', u);
  elseif (irevcm == 2)
     [v, ifail] = f11dp('N', a, irow, icol, ipivp, ipivq, istr, idiag,
'N', u);
  elseif (irevcm == 3)
    [itn, stplhs, stprhs, anorm, sigmax, work, ifail] = f11bt(work);
        fprintf('\nMonitoring  at  iteration  number  %d\nresidual  norm:
%14.4e\n', itn, stplhs);
    fprintf('\n   Solution Vector\n');
    for i = 1:n
      fprintf('%+16.4e + %+16.4eI\n', real(u(i)), imag(u(i)));
    end
    fprintf('\n   Residual Vector\n');
    for i = 1:n
      fprintf('%+16.4e + %+16.4eI\n', real(v(i)), imag(v(i)));
    end
  end
end
% Get information about the computation
[itn, stplhs, stprhs, anorm, sigmax, work, ifail] = f11bt(work);
fprintf('\nNumber of iterations for convergence: %d\n', itn);
fprintf('Residual norm:                          %14.4e\n', stplhs);
fprintf('Right-hand side of termination criteria: %14.4e\n', stprhs);
fprintf('i-norm of matrix a:                     %14.4e\n', anorm);
fprintf('\n   Solution Vector\n');
for i = 1:n
  fprintf('%+16.4e + %+16.4eI\n', real(u(i)), imag(u(i)));
end
fprintf('\n   Residual Vector\n');
for i = 1:n
  fprintf('%+16.4e + %+16.4eI\n', real(v(i)), imag(v(i)));
end
```

```
Monitoring at iteration number 2
residual norm:     8.2345e+01

   Solution Vector
     +6.9055e-01 +      +1.4236e+00I
     +7.3931e-02 +      -1.1880e+00I
     +1.4778e+00 +      +4.7846e-01I
     +5.6572e+00 +      -3.0786e+00I
     +1.4243e+00 +      -1.1246e+00I
     +1.0374e-01 +      +1.9740e+00I
     +4.4985e-01 +      -1.2715e+00I
     +2.5704e+00 +      +1.7578e+00I

   Residual Vector
     +1.7772e+00 +      +4.6797e+00I
     +1.0774e+00 +      +6.4600e+00I
     -3.2812e+00 +      -1.1314e+01I
     -3.8698e+00 +      -1.6438e+00I
     +8.9912e+00 +      +1.1100e+01I
     +9.7428e+00 +      -4.6218e-01I
     +3.1668e+00 +      +2.8721e+00I
     -1.0323e+01 +      +1.5837e+00I
Number of iterations for convergence: 4
Residual norm:                                  1.3386e-11
Right-hand side of termination criteria:        8.9100e-06
i-norm of matrix a:                             2.7000e+01
   Solution Vector
+1.0000e+00 +      +1.0000e+00I
     +2.0000e+00 +      -1.0000e-00I
     +3.0000e+00 +      +1.0000e+00I
```

```
     +4.0000e+00 +      -1.0000e+00I
     +3.0000e+00 +      -1.0000e+00I
     +2.0000e+00 +      +1.0000e+00I
     +1.0000e-00 +      -1.0000e-00I
     -4.4216e-13 +      +3.0000e+00I
   Residual Vector
-5.8620e-14 +      -7.8160e-13I
     +1.5508e-12 +      -1.3607e-12I
     +8.1712e-13 +      +6.9633e-13I
     +9.8588e-13 +      -2.2027e-13I
     -1.2541e-12 +      -4.1744e-13I
     -7.3186e-13 +      +5.2935e-13I
     +2.8777e-13 +      +1.5721e-13I
     +1.1173e-12 +      +2.4194e-12I
```