# NAG Toolbox for MATLAB

# f11gs

## 1 Purpose

f11gs is an iterative solver for a complex Hermitian system of simultaneous linear equations; f11gs is the second in a suite of three functions, where the first function, f11gr, must be called prior to f11gs to set up the suite, and the third function in the suite, f11gt, can be used to return additional information about the computation.

These three functions are suitable for the solution of large sparse complex Hermitian systems of equations.

## 2 Syntax

```
[irevcm, u, v, work, ifail] = f11gs(irevcm, u, v, wgt, work, 'lwork',
lwork)
```

## 3 Description

f11gs solves the complex Hermitian system of linear simultaneous equations $Ax = b$ using either the preconditioned conjugate gradient method (see Hestenes and Stiefel 1952, Golub and Van Loan 1996, Barrett *et al.* 1994 and Dias da Cunha and Hopkins 1994) or a preconditioned Lanczos method based upon the algorithm SYMMLQ (see Paige and Saunders 1975 and Barrett *et al.* 1994).

For a general description of the methods employed you are referred to Section 3 of the document for f11gr.

f11gs can solve the system after the first function in the suite, f11gr, has been called to initialize the computation and specify the method of solution. The third function in the suite, f11gt, can be used to return additional information generated by the computation during monitoring steps and after f11gs has completed its tasks.

f11gs uses **reverse communication**, i.e., f11gs 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 a specific task: either to compute the matrix-vector product $v = Au$; to solve the preconditioning equation $Mv = u$; to notify the completion of the computation; or, to 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

Barrett R, Berry M, Chan T F, Demmel J, Donato J, Dongarra J, Eijkhout V, Pozo R, Romine C and Van der Vorst H 1994 *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods* SIAM, Philadelphia

Dias da Cunha R and Hopkins T 1994 PIM 1.1 — the parallel iterative method package for systems of linear equations user's guide — Fortran 77 version *Technical Report* Computing Laboratory, University of Kent at Canterbury, Kent, UK

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

Hestenes M and Stiefel E 1952 Methods of conjugate gradients for solving linear systems *J. Res. Nat. Bur. Stand.* **49** 409–436

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

Paige C C and Saunders M A 1975 Solution of sparse indefinite systems of linear equations *SIAM J. Numer. Anal.* **12** 617–629

## 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. f11gs will then return with the termination code **irevcm** $= 4$. Note that before calling f11gs with **irevcm** $= 5$ the calling program must have performed the tasks required by the value of **irevcm** returned by the previous call to f11gs, otherwise subsequently returned values may be invalid.

6 Immediate termination: f11gs will return immediately with termination code **irevcm** $= 4$ and with any useful information available. This includes the last iterate of the solution and, for conjugate gradient only, the last iterate of the residual vector. The residual vector is generally not available when the Lanczos method (SYMMLQ) is used. f11gs 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 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** $= 1$ 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 f11gs

**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 Section 3 of the document for f11gr).

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

*On initial entry*: the workspace **work** as returned by f11gr (see also Section 5 of the document for f11gr).

*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** |
|---|---|
| CG | **lwork** $= 120 + 5n + p$. |
| SYMMLQ | **lwork** $= 120 + 6n + p$. |

where

$p = 2 * (\textbf{maxits} + 1)$, when an estimate of $\sigma_1(A)$ (**sigmax**) is computed;

$p = 0$, otherwise.

*Constraint*: **lwork** $\geq$ **lwreq**, where **lwreq** is returned by f11gr.

## 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 $v = Au$, 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. f11gt can be called at this step to return additional information.

*On final exit*: if **irevcm** $= 4$, f11gs 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** $= 1$ 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 f11gs. If **ifail** $= 1$, the array **u** is unchanged from the last entry to f11gs. Otherwise, **u** holds the last 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 last entry to f11gs. If **ifail** $= 1$, the array **v** is unchanged from the initial entry to f11gs. 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 iterate of the residual vector unless the Lanczos method (SYMMLQ) was used and **ifail** $\geq 5$, 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** $= 1$

f11gs 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 f11gt have remained unchanged.

**ifail** $= 2$

The required accuracy could not be obtained. However, f11gs has terminated with reasonable accuracy: the last iterate of the residual satisfied the termination criterion but the exact residual $r = b - Ax$, did not. A small number of iterations have been carried out after the iterated residual satisfied the termination criterion, but were unable to improve on the accuracy. This error code usually implies that your problem has been fully and satisfactory solved to within or close to the accuracy available on your system. Further iterations are unlikely to improve on this situation. You should call f11gt to check the values of the left-and right-hand side of the termination condition.

**ifail** = 3

f11gr was either not called before calling f11gs or it returned an error. The arrays **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

The preconditioner appears not to be positive-definite. It is likely that your results are meaningless: both methods require a positive-definite preconditioner (see also Section 3). However, the array **u** returns the last iterate of the solution, the array **v** returns the last iterate of the residual vector, for the conjugate gradient method only.

**ifail** = 7

The matrix of the coefficients appears not to be positive-definite (conjugate gradient method only). The arrays **u** and **v** return the last iterates of the solution and residual vector, respectively. However, you should be warned that the results returned can be be in error.

**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 conjugate gradient method 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 $k$th iteration, the last iteration performed, unless an immediate termination was requested and the Lanczos method (SYMMLQ) was used.

On successful completion, the termination criterion is satisfied to within the user-specified tolerance, as described in Section 3 of the document for f11gr. The computed values of the left- and right-hand sides of the termination criterion selected can be obtained by a call to f11gt.

## 8 Further Comments

The number of operations carried out by f11gs 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 f11gs for each iteration is approximately proportional to $n$. Note that the Lanczos method (SYMMLQ) requires a slightly larger number of operations than the conjugate gradient method.

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 $\bar{A} = E^{-1}AE^{-\mathrm{H}}$.

Additional matrix-vector products are required for the computation of $\|A\|_1 = \|A\|_\infty$, when this has not been supplied to f11gr and is required by the termination criterion employed.

The number of operations required to compute $\sigma_1(\bar{A})$ is negligible for reasonable values of **sigtol** and **maxits** (see Sections 5 and 8 of the document for f11gr).

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 f11gr) and $\|x_0\| \gg \|x_k\|$, so that because of loss of significant digits the required accuracy could not be obtained, the iteration is restarted automatically at some suitable point: f11gs sets $x_0 = x_k$ and the computation begins again. For particularly badly scaled problems, more than one restart may be necessary. 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

```
nnz = int32(23);
a = zeros(10000,1);
a(1:nnz) = [complex( 6., 0.);
            complex(-1., 1.);
            complex( 6., 0.);
            complex( 0., 1.);
            complex( 5., 0.);
            complex( 5., 0.);
            complex( 2.,-2.);
            complex( 4., 0.);
            complex( 1., 1.);
            complex( 2., 0.);
            complex( 6., 0.);
            complex(-4., 3.);
            complex( 0., 1.);
            complex(-1., 0.);
            complex( 6., 0.);
            complex(-1.,-1.);
            complex( 0.,-1.);
            complex( 9., 0.);
            complex( 1., 3.);
            complex( 1., 2.);
            complex(-1., 0.);
            complex( 1., 4.);
            complex( 9., 0.)];
irow = zeros(10000, 1, 'int32');
irow(1:nnz) = [int32(1);
               int32(2);
               int32(2);
               int32(3);
               int32(3);
               int32(4);
               int32(5);
               int32(5);
               int32(6);
               int32(6);
               int32(6);
               int32(7);
               int32(7);
               int32(7);
               int32(7);
               int32(8);
               int32(8);
               int32(8);
               int32(9);
               int32(9);
               int32(9);
               int32(9);
               int32(9)];
icol = zeros(10000, 1, 'int32');
icol(1:nnz) = [int32(1);
               int32(1);
               int32(2);
               int32(2);
               int32(3);
               int32(4);
               int32(1);
               int32(5);
               int32(3);
               int32(4);
               int32(6);
               int32(2);
               int32(5);
               int32(6);
               int32(7);
               int32(4);
               int32(6);
               int32(8);
               int32(1);
               int32(5);
               int32(6);
               int32(8);
               int32(9)];
lfill = int32(0);
dtol = 0;
mic = ’N’;
dscale = 0;
ipiv = [int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0)];
method = ’CG    ’;
precon = ’P’;
n = int32(9);
tol = 1e-06;
maxitn = int32(20);
anorm = 0;
sigmax = 0;
maxits = int32(9);
monit = int32(2);

irevcm = int32(0);
u = [complex(0, +0);
     complex(0, +0);
     complex(0, +0);
     complex(0, +0);
     complex(0, +0);
     complex(0, +0);
     complex(0, +0);
     complex(0, +0);
     complex(0, +0)];
v = [complex(8, +54);
     complex(-10, -92);
     complex(25, +27);
     complex(26, -28);
     complex(54, +12);
     complex(26, -22);
     complex(47, +65);
     complex(71, -57);
     complex(60, +70)];
wgt = [0; 0; 0; 0; 0; 0; 0; 0; 0];


[a, irow, icol, ipiv, istr, nnzc, npivm, ifail] = ...
    f11jn(nnz, a, irow, icol, lfill, dtol, mic, dscale, ipiv);

[lwreq, work, ifail] = ...
      f11gr(method, precon, n, tol, maxitn, anorm, sigmax, maxits, monit,
’sigcmp’, ’s’, ’norm_p’, ’1’);
```

```
while (irevcm ~= 4)
  [irevcm, u, v, work, ifail] = f11gs(irevcm, u, v, wgt, work);

  if (irevcm == 1)
    [v, ifail] = f11xs(a(1:nnz), irow(1:nnz), icol(1:nnz), 'N', u);
    if (ifail ~= 0)
      irecvm = 6;
    end
  elseif (irevcm == 2)
    [v, ifail] = f11jp(a, irow, icol, ipiv, istr, 'N', u);
    if (ifail ~= 0)
      irecvm = 6;
    end
  elseif (irevcm == 3)
        [itn, stplhs, stprhs, anorm, sigmax, its, sigerr, ifail] =
f11gt(work);
    if (ifail ~= 0)
      irecvm = 6;
    end
        fprintf('\nMonitoring at iteration number %d\nresidual norm:
%14.4e\n', itn, stplhs);
    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, its, sigerr, ifail] = f11gt(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:     1.4937e+01
     +2.1423e-01 +      +4.5333e+00I
     -1.6589e+00 +      -1.2672e+01I
     +2.4101e+00 +      +7.4551e+00I
     +4.4400e+00 +      -6.4174e+00I
     +9.1135e+00 +      +3.7812e+00I
     +4.4419e+00 +      -4.0382e+00I
     +1.4757e+00 +      +1.2662e+00I
     +8.4872e+00 +      -3.5347e+00I
     +5.9948e+00 +      +9.6851e-01I

   Residual Vector
     -1.8370e+00 +      +3.6956e+00I
     -6.5005e-01 +      +2.5458e-01I
     -1.2616e-01 +      -1.3625e-01I
     -1.3120e-01 +      +1.4130e-01I
     -1.1471e+00 +      +7.3386e-01I
     -5.5054e-01 +      -1.0535e+00I
     +1.7165e+00 +      -1.4614e+00I
     -3.5829e-01 +      +2.8764e-01I
     -3.0278e-01 +      -3.5324e-01I

Monitoring at iteration number 4
```

```
residual norm:     1.4602e+00
     +1.0061e+00 +      +8.9847e+00I
     +1.9637e+00 +      -7.9768e+00I
     +3.0067e+00 +      +7.0285e+00I
     +3.9830e+00 +      -5.9636e+00I
     +5.0390e+00 +      +5.0432e+00I
     +6.0488e+00 +      -4.0771e+00I
     +6.9710e+00 +      +3.0168e+00I
     +8.0118e+00 +      -1.9806e+00I
     +9.0074e+00 +      +9.6458e-01I

   Residual Vector
     +1.1524e-02 +      -2.8188e-02I
     +1.3513e-02 +      -1.7345e-01I
     +1.8173e-02 +      +1.9627e-02I
     +1.8900e-02 +      -2.0354e-02I
     -9.0877e-02 +      -1.0895e-01I
     -2.3890e-01 +      +3.2440e-01I
     +1.9031e-01 +      -1.5499e-02I
     +5.1611e-02 +      -4.1435e-02I
     +4.3615e-02 +      +5.0884e-02I
Number of iterations for convergence: 5
Residual norm:                               9.0594e-14
Right-hand side of termination criteria:     2.7340e-03
i-norm of matrix a:                          2.2000e+01
   Solution Vector
+1.0000e-00 +      +9.0000e+00I
     +2.0000e+00 +      -8.0000e+00I
     +3.0000e+00 +      +7.0000e+00I
     +4.0000e+00 +      -6.0000e+00I
     +5.0000e+00 +      +5.0000e+00I
     +6.0000e+00 +      -4.0000e+00I
     +7.0000e+00 +      +3.0000e+00I
     +8.0000e+00 +      -2.0000e+00I
     +9.0000e+00 +      +1.0000e+00I
   Residual Vector
-1.7764e-15 +      +0.0000e+00I
     +3.5527e-15 +      -2.8422e-14I
     -3.5527e-15 +      +3.5527e-15I
     +3.5527e-15 +      -7.1054e-15I
     -7.1054e-15 +      +3.5527e-15I
     -7.1054e-15 +      +0.0000e+00I
     +0.0000e+00 +      +0.0000e+00I
     +0.0000e+00 +      -7.1054e-15I
     +0.0000e+00 +      -1.4211e-14I
```