

NAG Library Routine Document

F11GSF

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

F11GSF is an iterative solver for a complex Hermitian system of simultaneous linear equations; F11GSF is the second in a suite of three routines, where the first routine, F11GRF, must be called prior to F11GSF to set up the suite, and the third routine in the suite, F11GTF, can be used to return additional information about the computation.

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

2 Specification

```
SUBROUTINE F11GSF (IREVCM, U, V, WGT, WORK, LWORK, IFAIL)
  INTEGER          IREVCM, LWORK, IFAIL
  REAL (KIND=nag_wp) WGT(*)
  COMPLEX (KIND=nag_wp) U(*), V(*), WORK(LWORK)
```

3 Description

F11GSF 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 in F11GRF.

F11GSF can solve the system after the first routine in the suite, F11GRF, has been called to initialize the computation and specify the method of solution. The third routine in the suite, F11GTF, can be used to return additional information generated by the computation during monitoring steps and after F11GSF has completed its tasks.

F11GSF uses **reverse communication**, i.e., F11GSF returns repeatedly to the calling program with the argument 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 argument 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 routine, 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: you can closely monitor the progress of the solution 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 routines 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 Arguments

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

1: IREVCM – INTEGER *Input/Output*

On initial entry: IREVCM = 0, otherwise an error condition will be raised.

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

IREVCM = 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. F11GSF will then return with the termination code IREVCM = 4. Note that before calling F11GSF with IREVCM = 5 the calling program must have performed the tasks required by the value of IREVCM returned by the previous call to F11GSF, otherwise subsequently returned values may be invalid.

IREVCM = 6

Immediate termination: F11GSF 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. F11GSF 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.

On intermediate exit: has the following meanings.

IREVCM = 1

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

IREVCM = 2

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

IREVCM = 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. To return additional information F11GTF can be called at this step.

On final exit: if IREVCM = 4, F11GSF 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.

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

2: U(*) – COMPLEX (KIND=nag_wp) array *Input/Output*

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.

On intermediate exit: the returned value of IREVCM determines the contents of U in the following way.

If IREVCM = 1 or 2, U holds the vector u on which the operation specified by IREVCM is to be carried out.

If 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 F11GSF. If IFAIL = 1, the array U is unchanged from the last entry to F11GSF. Otherwise, U holds the last iterate of the solution of the system of equations, for all returned values of IFAIL.

3: V(*) – COMPLEX (KIND=nag_wp) array *Input/Output*

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.

If 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 F11GSF

If IREVCM = 3, V must remain unchanged.

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 0, the array V is unchanged from the last entry to F11GSF. If IFAIL = 1, the array V is unchanged from the initial entry to F11GSF. 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 ≥ 5 , in which case V is set to 0.0.

4: WGT(*) – REAL (KIND=nag_wp) array *Input*

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

On entry: 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 in F11GRF).

Constraint: if weights are to be used, at least one element of WGT must be nonzero.

5: WORK(LWORK) – COMPLEX (KIND=nag_wp) array *Communication Array*

On initial entry: the array WORK as returned by F11GRF (see also Section 5 in F11GRF).

On intermediate re-entry: must remain unchanged.

6: LWORK – INTEGER *Input*

On initial entry: the dimension of the array WORK as declared in the (sub)program from which F11GSF is called (see also Section 3 in F11GRF). The required amount of workspace is as follows:

Method	Requirements
CG	$LWORK = 120 + 5n + p.$
SYMMMLQ	$LWORK = 120 + 6n + p.$

where

$p = 2 \times (\text{MAXITS} + 1)$, when an estimate of $\sigma_1(A)$ (SIGMAX) is computed;

$p = 0$, otherwise.

Constraint: $LWORK \geq LWREQ$, where LWREQ is returned by F11GRF.

7: IFAIL – INTEGER *Input/Output*

On initial entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

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, because for this routine the values of the output arguments may be useful even if $IFAIL \neq 0$ on exit, the recommended value is -1. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On intermediate exit: the value of IFAIL is meaningless and should be ignored.

On final exit: (i.e., when IREVCM = 4) $IFAIL = 0$, unless the routine detects an error or a warning has been flagged (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:

$IFAIL = -i$

On entry, the i th argument had an illegal value.

$IFAIL = 1$

F11GSF 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 F11GTF have remained unchanged.

$IFAIL = 2$

The required accuracy could not be obtained. However, F11GSF 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 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 F11GTF to check the values of the left- and right-hand side of the termination condition.

IFAIL = 3

F11GRF was either not called before calling F11GSF 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 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.

IFAIL = 9

The matrix appears to be singular.

IFAIL = 10

User-supplied weights are to be used, but all elements of the array WGT are zero.

IFAIL = -99

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in *How to Use the NAG Library and its Documentation* for further information.

IFAIL = -399

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in *How to Use the NAG Library and its Documentation* for further information.

IFAIL = -999

Dynamic memory allocation failed.

See Section 3.7 in *How to Use the NAG Library and its Documentation* for further information.

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 in F11GRF. The computed values of the left- and right-hand sides of the termination criterion selected can be obtained by a call to F11GTF.

8 Parallelism and Performance

F11GSF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F11GSF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The number of operations carried out by F11GSF 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 F11GSF 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 $\bar{A} = E^{-1}AE^{-H}$.

Additional matrix-vector products are required for the computation of $\|A\|_1 = \|A\|_\infty$, when this has not been supplied to F11GRF 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 9 in F11GRF).

If the termination criterion $\|r_k\|_p \leq \tau(\|b\|_p + \|A\|_p \times \|x_k\|_p)$ is used (see Section 3 in F11GRF) 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: F11GSF 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 $x_0 = 0$.

10 Example

See Section 10 in F11GRF.
