

NAG Toolbox

nag_sparse_real_symm_basic_solver (f11ge)

1 Purpose

nag_sparse_real_symm_basic_solver (f11ge) is an iterative solver for a symmetric system of simultaneous linear equations; nag_sparse_real_symm_basic_solver (f11ge) is the second in a suite of three functions, where the first function, nag_sparse_real_symm_basic_setup (f11gd), must be called prior to nag_sparse_real_symm_basic_solver (f11ge) to set up the suite, and the third function in the suite, nag_sparse_real_symm_basic_diag (f11gf), can be used to return additional information about the computation.

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

2 Syntax

```
[irevcn, u, v, work, ifail] = nag_sparse_real_symm_basic_solver(irevcn, u, v,
wgt, work)
[irevcn, u, v, work, ifail] = f11ge(irevcn, u, v, wgt, work)
```

Note: the interface to this routine has changed since earlier releases of the toolbox:

At Mark 22: *lwork* was removed from the interface.

3 Description

nag_sparse_real_symm_basic_solver (f11ge) solves the symmetric system of linear simultaneous equations $Ax = b$ using 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)), a preconditioned Lanczos method based upon the algorithm SYMMLQ (see Paige and Saunders (1975) and Barrett *et al.* (1994)), or the MINRES algorithm (see Paige and Saunders (1975)).

For a general description of the methods employed you are referred to Section 3 in nag_sparse_real_symm_basic_setup (f11gd).

nag_sparse_real_symm_basic_solver (f11ge) can solve the system after the first function in the suite, nag_sparse_real_symm_basic_setup (f11gd), has been called to initialize the computation and specify the method of solution. The third function in the suite, nag_sparse_real_symm_basic_diag (f11gf), can be used to return additional information generated by the computation during monitoring steps and after nag_sparse_real_symm_basic_solver (f11ge) has completed its tasks.

nag_sparse_real_symm_basic_solver (f11ge) uses **reverse communication**, i.e., nag_sparse_real_symm_basic_solver (f11ge) returns repeatedly to the calling program with the argument **irevcn** (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 **irevcn** 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.

- Enhanced user interaction: you can closely monitor 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 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 argument **irevcn**. Between intermediate exits and re-entries, **all arguments other than irevcn and v must remain unchanged**.

5.1 Compulsory Input Parameters

- irevcn** – INTEGER

On initial entry: **irevcn** = 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.

irevcn = 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. `nag_sparse_real_symm_basic_solver (f11ge)` will then return with the termination code **irevcn** = 4. Note that before calling `nag_sparse_real_symm_basic_solver (f11ge)` with **irevcn** = 5 the calling program must have performed the tasks required by the value of **irevcn** returned by the previous call to `nag_sparse_real_symm_basic_solver (f11ge)`, otherwise subsequently returned values may be invalid.

irevcn = 6

Immediate termination: `nag_sparse_real_symm_basic_solver (f11ge)` will return immediately with termination code **irevcn** = 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. `nag_sparse_real_symm_basic_solver (f11ge)` will then return with the termination code **irevcn** = 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 **irevcn** to any other value between calls will result in an error.

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

- 2: **u**(:) – REAL (KIND=nag_wp) array

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**(:) – REAL (KIND=nag_wp) array

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 **irevcn** determines the contents of **v** in the following way.

If **irevcn** = 1 or 2, **v** must store the vector v , the result of the operation specified by the value of **irevcn** returned by the previous call to `nag_sparse_real_symm_basic_solver` (f11ge).

If **irevcn** = 3, **v** must remain unchanged.

- 4: **wgt**(:) – REAL (KIND=nag_wp) array

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 and Section 5 in `nag_sparse_real_symm_basic_setup` (f11gd)).

Weights are NOT used in the MINRES algorithm.

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

- 5: **work**(*lwork*) – REAL (KIND=nag_wp) array

lwork, the dimension of the array, must satisfy the constraint $lwork \geq \mathbf{lwreq}$, where **lwreq** is returned by `nag_sparse_real_symm_basic_setup` (f11gd).

On initial entry: the array **work** as returned by `nag_sparse_real_symm_basic_setup` (f11gd) (see also Section 5 in `nag_sparse_real_symm_basic_setup` (f11gd)).

lwork, the dimension of the array, must satisfy the constraint $lwork \geq \mathbf{lwreq}$, where **lwreq** is returned by `nag_sparse_real_symm_basic_setup` (f11gd).

On intermediate re-entry: must remain unchanged.

5.2 Optional Input Parameters

None.

5.3 Output Parameters

- 1: **irevcn** – INTEGER

On intermediate exit: has the following meanings.

irevcn = 1

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

irevcn = 2

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

irevcn = 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 nag_sparse_real_symm_basic_diag (f11gf) can be called at this step.

On final exit: if **irevcn** = 4, nag_sparse_real_symm_basic_solver (f11ge) 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**(:) – REAL (KIND=nag_wp) array

The dimension of the array **u** will be n

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

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

If **irevcn** = 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 nag_sparse_real_symm_basic_solver (f11ge). If **ifail** = 1, the array **u** is unchanged from the last entry to nag_sparse_real_symm_basic_solver (f11ge). Otherwise, **u** holds the last iterate of the solution of the system of equations, for all returned values of **ifail**.

3: **v**(:) – REAL (KIND=nag_wp) array

The dimension of the array **v** will be n

On intermediate exit: if **irevcn** = 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 nag_sparse_real_symm_basic_solver (f11ge). If **ifail** = 1, the array **v** is unchanged from the initial entry to nag_sparse_real_symm_basic_solver (f11ge). 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*) – REAL (KIND=nag_wp) array

5: **ifail** – INTEGER

On final exit: **ifail** = 0 unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1 (*warning*)

nag_sparse_real_symm_basic_solver (f11ge) has already completed its tasks. You need to set a new problem.

ifail = 2 (*warning*)

The required accuracy could not be obtained. However, a reasonable accuracy may have been achieved.

User-requested termination: the required accuracy could not be obtained. However, a reasonable accuracy may have been achieved.

ifail = 3

Either `nag_sparse_real_symm_basic_setup` (f11gd) was not called before calling this function or it has returned an error.

ifail = 4 (*warning*)

User-requested tidy termination.

ifail = 5 (*warning*)

The solution has not converged after $\langle value \rangle$ iterations.

ifail = 6

The preconditioner appears not to be positive definite. The computation cannot continue.

ifail = 7 (*warning*)

The matrix of the coefficients A appears not to be positive definite. The computation cannot continue.

ifail = 8 (*warning*)

User-requested immediate termination.

ifail = 9

The matrix of the coefficients A appears to be singular. The computation cannot continue.

ifail = 10

The weights in array `wgt` are all zero.

ifail = -1

On initial entry, `irevcm` = $\langle value \rangle$.

Constraint: `irevcm` = 0.

On intermediate re-entry, `irevcm` = $\langle value \rangle$.

Constraint: either `irevcm` must be unchanged from its previous exit value or `irevcm` = 5 or 6.

ifail = -6

Constraint: $lwork \geq lwreq$, where `lwreq` is returned by `nag_sparse_real_symm_basic_setup` (f11gd).

ifail = -99

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

ifail = -399

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

ifail = -999

Dynamic memory allocation failed.

7 Accuracy

On completion, i.e., `irevcn = 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 `nag_sparse_real_symm_basic_setup` (f11gd). The computed values of the left- and right-hand sides of the termination criterion selected can be obtained by a call to `nag_sparse_real_symm_basic_diag` (f11gf).

8 Further Comments

The number of operations carried out by `nag_sparse_real_symm_basic_solver` (f11ge) 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 `nag_sparse_real_symm_basic_solver` (f11ge) 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 $\bar{A} = E^{-1}AE^{-T}$.

Additional matrix-vector products are required for the computation of $\|A\|_1 = \|A\|_\infty$, when this has not been supplied to `nag_sparse_real_symm_basic_setup` (f11gd) 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 Section 5 and Section 9 in `nag_sparse_real_symm_basic_setup` (f11gd)).

If the termination criterion $\|r_k\|_p \leq \tau(\|b\|_p + \|A\|_p \times \|x_k\|_p)$ is used (see Section 3 in `nag_sparse_real_symm_basic_setup` (f11gd)) 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: `nag_sparse_real_symm_basic_solver` (f11ge) 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

See Section 10 in `nag_sparse_real_symm_basic_setup` (f11gd).

9.1 Program Text

```
function f11ge_example

fprintf('f11ge example results\n\n');

% Sparse matrix A
n   = nag_int(7);
nz  = nag_int(16);
a   = zeros(1000,1);
irow = zeros(1000,1,nag_int_name);
icol = irow;

a(1:16)   = [4  1  5  2  2  3 -1  1  4  1 -2  3  2 -1 -2  5];
irow(1:16) = [1  2  2  3  4  4  5  5  5  6  6  6  7  7  7  7];
icol(1:16) = [1  1  2  3  2  4  1  4  5  2  5  6  1  2  3  7];

% Incomplete Cholesky factorization, zero fill
```

```

lfill = nag_int(0);
dtol = 0;
mic = 'N';
dscale = 0;
ipiv = zeros(n, 1, nag_int_name);

[a, irow, icol, ipiv, istr, nnzc, npivm, ifail] = ...
f11ja( ...
    nz, a, irow, icol, lfill, dtol, mic, dscale, ipiv);

% Solver setup
method = 'CG';
precon = 'P';
tol = 1e-6;
maxitn = nag_int(20);
anorm = 0;
sigmax = 0;
maxits = nag_int(7);
monit = nag_int(2);

[lwreq, work, ifail] = ...
    f11gd(method, precon, n, tol, maxitn, anorm, sigmax, ...
        maxits, monit, 'sigcmp', 'S', 'norm_p', '1');

% Solver inputs
irevcm = nag_int(0);
u = zeros(n,1);
v = [15; 18; -8; 21; 11; 10; 29];
wgt = zeros(n,1);

% Reverse communication loop
while (irevcm ~= 4)
    [irevcm, u, v, work, ifail] = f11ge( ...
        irevcm, u, v, wgt, work);

    if (irevcm == 1)
        % v = Au
        [v, ifail] = f11xe( ...
            a, irow, icol, 'N', u, 'nz', nz);
    elseif (irevcm == 2)
        % preconditioner
        [v, ifail] = f11jb( ...
            a, irow, icol, ipiv, istr, 'N', u);
    elseif (irevcm == 3)
        [itn, stplhs, stprhs, anorm, sigmax, its, sigerr, ifail] = ...
            f11gf(work);

        fprintf('\nMonitoring at iteration number %d\n', itn);
        fprintf('residual norm: %14.4e\n\n', stplhs);
        fprintf('  Solution Vector  Residual Vector\n');
        fprintf('%16.4f  %16.4e\n', [u'; v']);
    end
end

% Get information about the computation
[itn, stplhs, stprhs, anorm, sigmax, its, sigerr, ifail] = f11gf(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  Residual Vector\n');
fprintf('%16.4f  %12.2e\n', [u'; v']);

```

9.2 Program Results

f11ge example results

```

Monitoring at iteration number 2
residual norm: 1.9938e+00

```

```

  Solution Vector  Residual Vector

```

0.9632	-2.2960e-01
1.9934	2.2254e-01
3.0583	9.5827e-02
4.1453	-2.5155e-01
4.8289	-1.7160e-01
5.6630	6.7533e-01
7.1062	-3.4737e-01

Monitoring at iteration number 4
residual norm: 6.6574e-03

Solution Vector	Residual Vector
0.9994	-1.0551e-03
2.0011	-2.4675e-03
3.0008	-1.7116e-05
3.9996	4.4929e-05
4.9991	2.1359e-03
5.9993	-8.7482e-04
7.0007	6.2045e-05

Number of iterations for convergence: 5
Residual norm: 2.0428e-14
Right-hand side of termination criteria: 3.9200e-04
i-norm of matrix a: 1.0000e+01

Solution Vector	Residual Vector
1.0000	0.00e+00
2.0000	0.00e+00
3.0000	-2.66e-15
4.0000	-3.55e-15
5.0000	-5.33e-15
6.0000	1.78e-15
7.0000	7.11e-15
