NAG Library Function Document

nag_sparse_herm_basic_diagnostic (f11gtc)

1 Purpose

nag_sparse_herm_basic_diagnostic (f11gtc) is the third in a suite of three functions for the iterative solution of a complex Hermitian system of simultaneous linear equations (see Golub and Van Loan (1996)). nag_sparse_herm_basic_diagnostic (f11gtc) returns information about the computations during an iteration and/or after this has been completed. The first function of the suite, nag_sparse_herm_basic_setup (f11grc), is a setup function, the second function, nag_sparse_herm_basic_solver (f11gsc) is the proper iterative solver.

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

2 Specification

```
#include <nag.h>
#include <nagf11.h>
void nag_sparse_herm_basic_diagnostic (Integer *itn, double *stplhs,
     double *stprhs, double *anorm, double *sigmax, Integer *its,
     double *sigerr, const Complex work[], Integer lwork, NagError *fail)
```

3 Description

nag_sparse_herm_basic_diagnostic (f11gtc) returns information about the solution process. It can be called both during a monitoring step of the solver nag_sparse_herm_basic_solver (f11gsc) or after this solver has completed its tasks. Calling nag_sparse_herm_basic_diagnostic (f11gtc) at any other time will result in an error condition being raised.

For further information you should read the documentation for nag_sparse_herm_basic_setup (f11grc) and nag_sparse_herm_basic_solver (f11gsc).

4 References

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

5 Arguments

1: **itn** – Integer * *Output*

On exit: the number of iterations carried out by nag_sparse_herm_basic_solver (f11gsc).

2: **stplhs** – double * *Output*

On exit: the current value of the left-hand side of the termination criterion used by nag_sparse_herm_basic_solver (f11gsc).

3: **stprhs** – double * *Output*

On exit: the current value of the right-hand side of the termination criterion used by nag_sparse_herm_basic_solver (f11gsc).

4: **anorm** – double * *Output*

On exit: the norm $\|A\|_1 = \|A\|_\infty$ when either it has been supplied to nag_sparse_herm_basic_set up (f11grc) or it has been estimated by nag_sparse_herm_basic_solver (f11gsc) (see also Sections 3 and 5 in nag_sparse_herm_basic_setup (f11grc)).
Otherwise, **anorm** $= 0.0$ is returned.

5: **sigmax** – double * *Output*

On exit: the current estimate of the largest singular value $\sigma_1(\bar{A})$ of the preconditioned iteration matrix $\bar{A} = E^{-1}AE^{-\mathrm{H}}$, when either it has been supplied to nag_sparse_herm_basic_setup (f11grc) or it has been estimated by nag_sparse_herm_basic_solver (f11gsc) (see also Sections 3 and 5 in nag_sparse_herm_basic_setup (f11grc)). Note that if **its** $<$ **itn** then **sigmax** contains the final estimate. If, on final exit from nag_sparse_herm_basic_solver (f11gsc), **its** $=$ **itn**, then the estimation of $\sigma_1(\bar{A})$ may have not converged: in this case you should look at the value returned in **sigerr**. Otherwise, **sigmax** $= 0.0$ is returned.

6: **its** – Integer * *Output*

On exit: the number of iterations employed so far in the computation of the estimate of $\sigma_1(\bar{A})$, the largest singular value of the preconditioned matrix $\bar{A} = E^{-1}AE^{-\mathrm{H}}$, when $\sigma_1(\bar{A})$ has been estimated by nag_sparse_herm_basic_solver (f11gsc) using the bisection method (see also Sections 3, 5 and 9 in nag_sparse_herm_basic_setup (f11grc)). Otherwise, **its** $= 0$ is returned.

7: **sigerr** – double * *Output*

On exit: if $\sigma_1(\bar{A})$ has been estimated by nag_sparse_herm_basic_solver (f11gsc) using bisection,

$$\textbf{sigerr} = \max\left(\frac{\left|\sigma_1^{(k)} - \sigma_1^{(k-1)}\right|}{\sigma_1^{(k)}}, \frac{\left|\sigma_1^{(k)} - \sigma_1^{(k-2)}\right|}{\sigma_1^{(k)}}\right),$$

where $k =$ **its** denotes the iteration number. The estimation has converged if **sigerr** $\leq$ **sigtol** where **sigtol** is an input argument to nag_sparse_herm_basic_setup (f11grc).
Otherwise, **sigerr** $= 0.0$ is returned.

8: **work**[**lwork**] – const Complex *Communication Array*

On entry: the array **work** as returned by nag_sparse_herm_basic_solver (f11gsc) (see also Section 3 in nag_sparse_herm_basic_solver (f11gsc)).

9: **lwork** – Integer *Input*

On entry: the dimension of the array **work** (see also Section 5 in nag_sparse_herm_basic_setup (f11grc)).

Constraint: **lwork** ≥ 120.

Note: although the minimum value of **lwork** ensures the correct functioning of nag_sparse_herm_basic_diagnostic (f11gtc), a larger value is required by the iterative solver nag_sparse_herm_basic_solver (f11gsc) (see also Section 5 in nag_sparse_herm_basic_setup (f11grc)).

10: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in How to Use the NAG Library and its Documentation for further information.

NE_BAD_PARAM

On entry, argument ⟨*value*⟩ had an illegal value.

NE_INT

On entry, **lwork** = ⟨*value*⟩.
Constraint: **lwork** ≥ 120.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in How to Use the NAG Library and its Documentation for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in How to Use the NAG Library and its Documentation for further information.

NE_OUT_OF_SEQUENCE

nag_sparse_herm_basic_diagnostic (f11gtc) has been called out of sequence.

7 Accuracy

Not applicable.

8 Parallelism and Performance

nag_sparse_herm_basic_diagnostic (f11gtc) is not threaded in any implementation.

9 Further Comments

None.

10 Example

See Section 10 in nag_sparse_herm_basic_setup (f11grc).