

# NAG Library Function Document

## nag\_ddisna (f08flc)

### 1 Purpose

nag\_ddisna (f08flc) computes the reciprocal condition numbers for the eigenvectors of a real symmetric or complex Hermitian  $m$  by  $m$  matrix  $A$ , or for the left or right singular vectors of a general  $m$  by  $n$  matrix  $A$ .

### 2 Specification

```
#include <nag.h>
#include <nagf08.h>
void nag_ddisna (Nag_JobType job, Integer m, Integer n, const double d[],
                double sep[], NagError *fail)
```

### 3 Description

The bound on the error, measured by the angle in radians, for the  $i$ th computed vector is given by  $\epsilon \|A\|_2 / \text{sep}_i$ , where  $\epsilon$  is the *machine precision* and  $\text{sep}_i$  is the reciprocal condition number for the vectors, returned in the array element **sep**[ $i - 1$ ]. **sep**[ $i - 1$ ] is restricted to be at least  $\epsilon \|A\|_2$  in order to limit the size of the error bound.

### 4 References

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

### 5 Arguments

- 1: **job** – Nag\_JobType *Input*  
*On entry:* specifies for which problem the reciprocal condition number should be computed.  
**job** = Nag\_EigVecs  
 The eigenvectors of a symmetric or Hermitian matrix.  
**job** = Nag\_LeftSingVecs  
 The left singular vectors of a general matrix.  
**job** = Nag\_RightSingVecs  
 The right singular vectors of a general matrix.  
*Constraint:* **job** = Nag\_EigVecs, Nag\_LeftSingVecs or Nag\_RightSingVecs.
- 2: **m** – Integer *Input*  
*On entry:*  $m$ , the number of rows of the matrix  $A$ .  
*Constraint:* **m**  $\geq 0$ .
- 3: **n** – Integer *Input*  
*On entry:*  $n$ , the number of columns of the matrix when **job** = Nag\_LeftSingVecs or Nag\_RightSingVecs.  
 If **job** = Nag\_EigVecs, **n** is not referenced.  
*Constraint:* if **job** = Nag\_LeftSingVecs or Nag\_RightSingVecs, **n**  $\geq 0$ .

4: **d**[*dim*] – const double *Input*

**Note:** the dimension, *dim*, of the array **d** must be at least

$\max(1, \mathbf{m})$  when **job** = Nag\_EigVecs;  
 $\max(1, \min(\mathbf{m}, \mathbf{n}))$  when **job** = Nag\_LeftSingVecs or Nag\_RightSingVecs.

*On entry:* the eigenvalues if **job** = Nag\_EigVecs, or singular values if **job** = Nag\_LeftSingVecs or Nag\_RightSingVecs of the matrix *A*.

*Constraints:*

the elements of the array **d** must be in either increasing or decreasing order;  
 if **job** = Nag\_LeftSingVecs or Nag\_RightSingVecs the elements of **d** must be non-negative.

5: **sep**[*dim*] – double *Output*

**Note:** the dimension, *dim*, of the array **sep** must be at least

$\max(1, \mathbf{m})$  when **job** = Nag\_EigVecs;  
 $\max(1, \min(\mathbf{m}, \mathbf{n}))$  when **job** = Nag\_LeftSingVecs or Nag\_RightSingVecs.

*On exit:* the reciprocal condition numbers of the vectors.

6: **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\_ENUM\_INT

On entry, **job** = *⟨value⟩* and **n** = *⟨value⟩*.

Constraint: if **job** = Nag\_LeftSingVecs or Nag\_RightSingVecs, **n** ≥ 0.

### NE\_INT

On entry, **m** = *⟨value⟩*.

Constraint: **m** ≥ 0.

### 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\_NOT\_MONOTONIC**

Constraint: the elements of the array **d** must be in either increasing or decreasing order.  
if **job** = Nag\_LeftSingVecs or Nag\_RightSingVecs the elements of **d** must be non-negative.

**7 Accuracy**

The reciprocal condition numbers are computed to *machine precision* relative to the size of the eigenvalues, or singular values.

**8 Parallelism and Performance**

nag\_ddisna (f08flc) is not threaded in any implementation.

**9 Further Comments**

nag\_ddisna (f08flc) may also be used towards computing error bounds for the eigenvectors of the generalized symmetric or Hermitian definite eigenproblem. See Golub and Van Loan (1996) for further details on the error bounds.

**10 Example**

The use of nag\_ddisna (f08flc) in computing error bounds for eigenvectors of the symmetric eigenvalue problem is illustrated in Section 10 in nag\_dsyev (f08fac); its use in computing error bounds for singular vectors is illustrated in Section 10 in nag\_dgesvd (f08kbc); and its use in computing error bounds for eigenvectors of the generalized symmetric definite eigenvalue problem is illustrated in Section 10 in nag\_dsygv (f08sac).

---