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 ith computed vector is given by $\epsilon\|A\|_2/\text{sep}_i$, where ϵ is the ***machine precision*** and 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** ≥ 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** ≥ 0.

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

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

max(1, **m**) when **job** = Nag_EigVecs;
max(1, min(**m**, **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, **m**) when **job** = Nag_EigVecs;
max(1, min(**m**, **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 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction 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 the Essential Introduction 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 the Essential Introduction 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

Not applicable.

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).