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

```c
#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 
\[
\frac{1}{\epsilon\|A\|_2/\text{sep}_i},
\]
where \( \epsilon \) is the machine precision and \( \text{sep}_i \) is the reciprocal condition number for the vectors, returned in the array element \( \text{sep}[i-1] \). \( \text{sep}[i-1] \) is restricted to be at least \( \frac{1}{\epsilon\|A\|_2} \) in order to limit the size of the error bound.

4 References


5 Arguments

1: \( \text{job} - \) Nag_JobType \hspace{2cm} \text{Input} \\
\text{On entry:} \text{specifies for which problem the reciprocal condition number should be computed.} \\
\text{job} = \text{Nag_EigVecs} \\
\text{The eigenvectors of a symmetric or Hermitian matrix.} \\
\text{job} = \text{Nag_LeftSingVecs} \\
\text{The left singular vectors of a general matrix.} \\
\text{job} = \text{Nag_RightSingVecs} \\
\text{The right singular vectors of a general matrix.} \\
\text{Constraint: job = Nag_EigVecs, Nag_LeftSingVecs or Nag_RightSingVecs.}

2: \( m - \) Integer \hspace{2cm} \text{Input} \\
\text{On entry:} m, \text{the number of rows of the matrix} A. \\
\text{Constraint:} m \geq 0.

3: \( n - \) Integer \hspace{2cm} \text{Input} \\
\text{On entry:} n, \text{the number of columns of the matrix when} \text{job = Nag_LeftSingVecs or Nag_RightSingVecs.} \\
\text{If} \text{job = Nag_EigVecs,} n \text{is not referenced.} \\
\text{Constraint: if job = Nag_LeftSingVecs or Nag_RightSingVecs,} n \geq 0.
4: \( \text{d}[\text{dim}] \text{ – const double} \)  \(\text{Input}\)

\textbf{Note}: the dimension, \( \text{dim} \), of the array \( \text{d} \) must be at least
\[
\max(1, \text{m}) \text{ when } \text{job} = \text{Nag_EigVecs};
\max(1, \min(\text{m}, \text{n})) \text{ when } \text{job} = \text{Nag_LeftSingVecs or Nag_RightSingVecs}.
\]

\textit{On entry}: the eigenvalues if \( \text{job} = \text{Nag_EigVecs} \), or singular values if \( \text{job} = \text{Nag_LeftSingVecs or Nag_RightSingVecs} \) of the matrix \( A \).

\textbf{Constraints}:
- the elements of the array \( \text{d} \) must be in either increasing or decreasing order;
- if \( \text{job} = \text{Nag_LeftSingVecs or Nag_RightSingVecs} \) the elements of \( \text{d} \) must be non-negative.

5: \( \text{sep}[\text{dim}] \text{ – double} \)  \(\text{Output}\)

\textbf{Note}: the dimension, \( \text{dim} \), of the array \( \text{sep} \) must be at least
\[
\max(1, \text{m}) \text{ when } \text{job} = \text{Nag_EigVecs};
\max(1, \min(\text{m}, \text{n})) \text{ when } \text{job} = \text{Nag_LeftSingVecs or Nag_RightSingVecs}.
\]

\textit{On exit}: the reciprocal condition numbers of the vectors.

6: \( \text{fail} \text{ – NagError *} \)  \(\text{Input/Output}\)

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 \textbf{Error Indicators and Warnings}

\textbf{NE_ALLOC_FAIL}

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

\textbf{NE_BAD_PARAM}

On entry, argument \( \langle \text{value} \rangle \) had an illegal value.

\textbf{NE_ENUM_INT}

On entry, \( \text{job} = \langle \text{value} \rangle \) and \( \text{n} = \langle \text{value} \rangle \).
Constraint: if \( \text{job} = \text{Nag_LeftSingVecs or Nag_RightSingVecs} \), \( \text{n} \geq 0 \).

\textbf{NE_INT}

On entry, \( \text{m} = \langle \text{value} \rangle \).
Constraint: \( \text{m} \geq 0 \).

\textbf{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.

\textbf{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.

\textbf{NE_NOT_MONOTONIC}

Constraint: the elements of the array \( \text{d} \) must be in either increasing or decreasing order.
if \( \text{job} = \text{Nag_LeftSingVecs or Nag_RightSingVecs} \) the elements of \( \text{d} \) 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).