NAG Library Function Document

nag_dstevr (f08jdc)

1 Purpose

nag_dstevr (f08jdc) computes selected eigenvalues and, optionally, eigenvectors of a real \( n \) by \( n \) symmetric tridiagonal matrix \( T \). Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

2 Specification

```c
#include <nag.h>
#include <nagf08.h>
void nag_dstevr (Nag_OrderType order, Nag_JobType job, Nag_RangeType range,
                 Integer n, double d[], double e[], double vl, double vu, Integer il,
                 Integer iu, double abstol, Integer *m, double w[], double z[],
                 Integer pdz, Integer isuppz[], NagError *fail)
```

3 Description

Whenever possible nag_dstevr (f08jdc) computes the eigenspectrum using Relatively Robust Representations. nag_dstevr (f08jdc) computes eigenvalues by the dqds algorithm, while orthogonal eigenvectors are computed from various ‘good’ \( LDL^T \) representations (also known as Relatively Robust Representations). Gram–Schmidt orthogonalization is avoided as far as possible. More specifically, the various steps of the algorithm are as follows. For the \( i \)th unreduced block of \( T \):

(a) compute \( T - \sigma_i I = L_i D_i L_i^T \), such that \( L_i D_i L_i^T \) is a relatively robust representation,

(b) compute the eigenvalues, \( \lambda_j \), of \( L_i D_i L_i^T \) to high relative accuracy by the dqds algorithm,

(c) if there is a cluster of close eigenvalues, ‘choose’ \( \sigma_i \) close to the cluster, and go to (a),

(d) given the approximate eigenvalue \( \lambda_j \) of \( L_i D_i L_i^T \), compute the corresponding eigenvector by forming a rank-revealing twisted factorization.

The desired accuracy of the output can be specified by the argument \texttt{abstol}. For more details, see Dhillon (1997) and Parlett and Dhillon (2000).

4 References


5 Arguments

1: \textit{order} \hspace{1em} \text{Nag\_OrderType} \hspace{1em} \text{Input}

\textit{On entry:} the \textit{order} argument specifies the two-dimensional storage scheme being used, i.e., row-major ordering or column-major ordering. C language defined storage is specified by \textit{order} = Nag\_RowMajor. See Section 3.2.1.3 in the Essential Introduction for a more detailed explanation of the use of this argument.

\textit{Constraint:} \textit{order} = Nag\_RowMajor or Nag\_ColMajor.

2: \textit{job} \hspace{1em} \text{Nag\_JobType} \hspace{1em} \text{Input}

\textit{On entry:} indicates whether eigenvectors are computed.

\texttt{job} = Nag\_EigVals

Only eigenvalues are computed.

\texttt{job} = Nag\_DoBoth

Eigenvalues and eigenvectors are computed.

\textit{Constraint:} \texttt{job} = Nag\_EigVals or Nag\_DoBoth.

3: \textit{range} \hspace{1em} \text{Nag\_RangeType} \hspace{1em} \text{Input}

\textit{On entry:} if \textit{range} = Nag\_AllValues, all eigenvalues will be found.

If \textit{range} = Nag\_Interval, all eigenvalues in the half-open interval \([vl, vu]\) will be found.

If \textit{range} = Nag\_Indices, the \(il^{th}\) to \(iu^{th}\) eigenvalues will be found.

\textit{Constraint:} \textit{range} = Nag\_AllValues, Nag\_Interval or Nag\_Indices.

4: \textit{n} \hspace{1em} \text{Integer} \hspace{1em} \text{Input}

\textit{On entry:} \(n\), the order of the matrix.

\textit{Constraint:} \(n \geq 0\).

5: \textit{d}[	extit{dim}] \hspace{1em} \text{double} \hspace{1em} \text{Input/Output}

\texttt{Note:} the dimension, \textit{dim}, of the array \textit{d} must be at least max(1\(, n\)).

\textit{On entry:} the \textit{n} diagonal elements of the tridiagonal matrix \(T\).

\textit{On exit:} may be multiplied by a constant factor chosen to avoid over/underflow in computing the eigenvalues.

6: \textit{e}[	extit{dim}] \hspace{1em} \text{double} \hspace{1em} \text{Input/Output}

\texttt{Note:} the dimension, \textit{dim}, of the array \textit{e} must be at least max(1\(, n - 1\)).

\textit{On entry:} the \((n - 1)\) subdiagonal elements of the tridiagonal matrix \(T\).

\textit{On exit:} may be multiplied by a constant factor chosen to avoid over/underflow in computing the eigenvalues.

7: \textit{vl} \hspace{1em} \text{double} \hspace{1em} \text{Input}

8: \textit{vu} \hspace{1em} \text{double} \hspace{1em} \text{Input}

\textit{On entry:} if \textit{range} = Nag\_Interval, the lower and upper bounds of the interval to be searched for eigenvalues.

If \textit{range} = Nag\_AllValues or Nag\_Indices, \textit{vl} and \textit{vu} are not referenced.

\textit{Constraint:} if \textit{range} = Nag\_Interval, \(vl < vu\).
9: il – Integer

10: iu – Integer

On entry: if range = Nag_Indices, the indices (in ascending order) of the smallest and largest eigenvalues to be returned.

If range = Nag_AllValues or Nag_Interval, il and iu are not referenced.

Constraints:

if range = Nag_Indices and n = 0, il = 1 and iu = 0;
if range = Nag_Indices and n > 0, 1 ≤ il ≤ iu ≤ n.

11: abstol – double

On entry: the absolute error tolerance for the eigenvalues. An approximate eigenvalue is accepted as converged when it is determined to lie in an interval [a, b] of width less than or equal to

\[ \text{abstol} + \epsilon \max(|a|, |b|), \]

where \( \epsilon \) is the machine precision. If abstol is less than or equal to zero, then \( \epsilon \|T\|_1 \) will be used in its place. See Demmel and Kahan (1990).

If high relative accuracy is important, set abstol to nag_real_safe_small_number() , although doing so does not currently guarantee that eigenvalues are computed to high relative accuracy. See Barlow and Demmel (1990) for a discussion of which matrices can define their eigenvalues to high relative accuracy.

12: m – Integer

On exit: the total number of eigenvalues found. 0 ≤ m ≤ n.

If range = Nag_AllValues, m = n.
If range = Nag_Indices, m = iu – il + 1.

13: w[dim] – double

Note: the dimension, dim, of the array w must be at least max(1, n).

On exit: the first m elements contain the selected eigenvalues in ascending order.

14: z[dim] – double

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

\[ \max(1, \text{pdz} \times n) \] when job = Nag_DoBoth;
\[ 1 \] otherwise.

The (i, j)th element of the matrix Z is stored in

\[ z[(j - 1) \times \text{pdz} + i - 1] \] when order = Nag_ColMajor;
\[ z[(i - 1) \times \text{pdz} + j - 1] \] when order = Nag_RowMajor.

On exit: if job = Nag_DoBoth, the first m columns of Z contain the orthonormal eigenvectors of the matrix A corresponding to the selected eigenvalues, with the i-th column of Z holding the eigenvector associated with w[i - 1].
If job = Nag_EigVals, z is not referenced.

15: pdz – Integer

On entry: the stride separating row or column elements (depending on the value of order) in the array z.

Constraints:

if job = Nag_DoBoth, pdz ≥ max(1, n);
otherwise pdz ≥ 1.
isuppz[\text{dim}] \quad \text{– Integer} \quad \text{Output}

Note: the dimension, \text{dim}, of the array isuppz must be at least $\max(1, 2 \times m)$.

On exit: the support of the eigenvectors in $z$, i.e., the indices indicating the nonzero elements in $z$. The $i$th eigenvector is nonzero only in elements isuppz[2 \times i - 2] through isuppz[2 \times i - 1]. Implemented only for range = Nag_AllValues or Nag_Indices and $i u - i l = n - 1$.

\begin{verbatim}
fail \quad \text{– NagError * Input/Output}
\end{verbatim}

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

6 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_2}

On entry, $\text{job} = \langle\text{value}\rangle$, $\text{pdz} = \langle\text{value}\rangle$ and $\text{n} = \langle\text{value}\rangle$.
Constraint: if $\text{job} = \text{Nag_DoBoth}$, $\text{pdz} \geq \max(1, \text{n})$;
otherwise $\text{pdz} \geq 1$.

\textbf{NE_ENUM_INT_3}

On entry, $\text{range} = \langle\text{value}\rangle$, $\text{il} = \langle\text{value}\rangle$, $\text{iu} = \langle\text{value}\rangle$ and $\text{n} = \langle\text{value}\rangle$.
Constraint: if $\text{range} = \text{Nag_Indices}$ and $\text{n} = 0$, $\text{il} = 1$ and $\text{iu} = 0$;
if $\text{range} = \text{Nag_Indices}$ and $\text{n} > 0$, $1 \leq \text{il} \leq \text{iu} \leq \text{n}$.

\textbf{NE_ENUM_REAL_2}

On entry, $\text{range} = \langle\text{value}\rangle$, $\text{vl} = \langle\text{value}\rangle$ and $\text{vu} = \langle\text{value}\rangle$.
Constraint: if $\text{range} = \text{Nag_Interval}$, $\text{vl} < \text{vu}$.

\textbf{NE_INT}

On entry, $\text{n} = \langle\text{value}\rangle$.
Constraint: $\text{n} \geq 0$.

On entry, $\text{pdz} = \langle\text{value}\rangle$.
Constraint: $\text{pdz} > 0$.

\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 internal error has occurred in this function. Please refer to fail in nag_dstebz (f08jkc).
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.
7 Accuracy

The computed eigenvalues and eigenvectors are exact for a nearby matrix \( A + E \), where

\[ \|E\|_2 = O(\epsilon)\|A\|_2, \]

and \( \epsilon \) is the machine precision. See Section 4.7 of Anderson et al. (1999) for further details.

8 Parallelism and Performance

`nag_dstevr (f08jdc)` is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

`nag_dstevr (f08jdc)` makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this function. Please also consult the Users’ Note for your implementation for any additional implementation-specific information.

9 Further Comments

The total number of floating-point operations is proportional to \( n^2 \) if `job = Nag_EigVals` and is proportional to \( n^3 \) if `job = Nag_DoBoth` and `range = Nag_AllValues`, otherwise the number of floating-point operations will depend upon the number of computed eigenvectors.

10 Example

This example finds the eigenvalues with indices in the range \([2, 3]\), and the corresponding eigenvectors, of the symmetric tridiagonal matrix

\[
T = \begin{pmatrix}
1 & 1 & 0 & 0 \\
1 & 4 & 2 & 0 \\
0 & 2 & 9 & 3 \\
0 & 0 & 3 & 16
\end{pmatrix}.
\]

10.1 Program Text

```c
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagf08.h>
#include <nagx04.h>

int main(void) {
    /* Scalars */
    double abstol, vl = 0.0, vu = 0.0;
    Integer exit_status = 0, i, il, iu, j, m, n, pdz;
    /* Arrays */
    double *d = 0, *e = 0, *w = 0, *z = 0;
    Integer *isuppz = 0;
    /* Nag Types */
    Nag_OrderType order;
    NagError fail;

    // Code goes here...
}
```

Mark 25
order = Nag_ColMajor;
#else
order = Nag_RowMajor;
#endif

INIT_FAIL(fail);

printf("nag_dstevr (f08jdc) Example Program Results\n\n");

/* Skip heading in data file and read n and the lower and upper
* indices of the eigenvalues to be found.
*/
#endif
scanf_s("%*[\n"]);
#else
scanf("%*[\n"]);
#endif
#endif
scanf_s("%"NAG_IFMT"%"NAG_IFMT"%"NAG_IFMT"%*[\n"]", &n, &il, &iu);
#else
scanf("%"NAG_IFMT"%"NAG_IFMT"%"NAG_IFMT"%*[\n"]", &n, &il, &iu);
#endif

m = iu - il +1;
pdz = n;

/* Allocate memory */
if (!(d = NAG_ALLOC(n, double)) ||
 !(e = NAG_ALLOC(n-1, double)) ||
 !(w = NAG_ALLOC(n, double)) ||
 !(z = NAG_ALLOC(n * n, double)) ||
 !(isuppz = NAG_ALLOC(n, Integer)))
{
 printf("Allocation failure\n");
 exit_status = -1;
 goto END;
}

/* Read the diagonal and off-diagonal elements of the matrix A
* from data file.
*/
#endif
scanf_s("%lf", &d[i]);
#else
scanf("%lf", &d[i]);
#endif
#endif
scanf_s("%*[\n"]);
#else
scanf("%*[\n"]);
#endif

for (i = 0; i < n; ++i)
#endif
scanf_s("%lf", &e[i]);
#else
scanf("%lf", &e[i]);
#endif
#endif
scanf_s("%*[\n"]);
#else
scanf("%*[\n"]);
#endif

/* Set the absolute error tolerance for eigenvalues.
* With abstol set to zero, the default value is used instead.
*/
abstol = 0.0;

/* nag_dstevr (f08jdc)."
* Solve the symmetric tridiagonal eigenvalue problem. *

```c
nag_dstevr(order, Nag_DoBoth, Nag_Indices, n, d, e, vl, vu, il, iu, abstol, &m, w, z, pdz, isuppz, &fail);
if (fail.code != NE_NOERROR)
{
    printf("Error from nag_dstevr (f08jdc).\n", fail.message);
    exit_status = 1;
    goto END;
}

/* Print solution */
printf("Selected eigenvalues\n ");
for (j = 0; j < m; ++j)
    printf("%8.4f%s", w[j], (j+1)%8 == 0?"\n": ");
printf("\n\n");
/* nag_gen_real_mat_print (x04cac).
* Print selected eigenvectors. */
fflush(stdout);
nag_gen_real_mat_print(order, Nag_GeneralMatrix, Nag_NonUnitDiag, n, m, z, pdz, "Selected eigenvectors", 0, &fail);
if (fail.code != NE_NOERROR)
{
    printf("Error from nag_gen_real_mat_print (x04cac).\n", fail.message);
    exit_status = 1;
    goto END;
}
END:
NAG_FREE(d);
NAG_FREE(e);
NAG_FREE(w);
NAG_FREE(z);
NAG_FREE(isuppz);
return exit_status;
```

### 10.2 Program Data

nag_dstevr (f08jdc) Example Program Data

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

:Values of n, il and iu

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>4.0</td>
<td>9.0</td>
<td>16.0</td>
</tr>
</tbody>
</table>

:End of diagonal elements

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2.0</td>
<td>3.0</td>
<td></td>
</tr>
</tbody>
</table>

:End of off-diagonal elements

### 10.3 Program Results

nag_dstevr (f08jdc) Example Program Results

**Selected eigenvalues**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5470</td>
<td>8.6578</td>
</tr>
</tbody>
</table>

**Selected eigenvectors**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>0.3388</td>
</tr>
<tr>
<td>2</td>
<td>0.8628</td>
</tr>
<tr>
<td>3</td>
<td>-0.3648</td>
</tr>
<tr>
<td>4</td>
<td>0.0879</td>
</tr>
</tbody>
</table>