

## NAG Library Function Document

### nag\_dsteqr (f08jec)

#### 1 Purpose

nag\_dsteqr (f08jec) computes all the eigenvalues and, optionally, all the eigenvectors of a real symmetric tridiagonal matrix, or of a real symmetric matrix which has been reduced to tridiagonal form.

#### 2 Specification

```
#include <nag.h>
#include <nagf08.h>

void nag_dsteqr (Nag_OrderType order, Nag_ComputeZType compz, Integer n,
                double d[], double e[], double z[], Integer pdz, NagError *fail)
```

#### 3 Description

nag\_dsteqr (f08jec) computes all the eigenvalues and, optionally, all the eigenvectors of a real symmetric tridiagonal matrix  $T$ . In other words, it can compute the spectral factorization of  $T$  as

$$T = ZAZ^T,$$

where  $A$  is a diagonal matrix whose diagonal elements are the eigenvalues  $\lambda_i$ , and  $Z$  is the orthogonal matrix whose columns are the eigenvectors  $z_i$ . Thus

$$Tz_i = \lambda_i z_i, \quad i = 1, 2, \dots, n.$$

The function may also be used to compute all the eigenvalues and eigenvectors of a real symmetric matrix  $A$  which has been reduced to tridiagonal form  $T$ :

$$\begin{aligned} A &= QTQ^T, \text{ where } Q \text{ is orthogonal} \\ &= (QZ)A(QZ)^T. \end{aligned}$$

In this case, the matrix  $Q$  must be formed explicitly and passed to nag\_dsteqr (f08jec), which must be called with **compz** = Nag\_UpdateZ. The functions which must be called to perform the reduction to tridiagonal form and form  $Q$  are:

full matrix	nag_dsytrd (f08fec) and nag_dorgtr (f08ffc)
full matrix, packed storage	nag_dsprtd (f08gec) and nag_dopgtr (f08gfc)
band matrix	nag_dsbtrd (f08hec) with <b>vect</b> = Nag_FormQ.

nag\_dsteqr (f08jec) uses the implicitly shifted  $QR$  algorithm, switching between the  $QR$  and  $QL$  variants in order to handle graded matrices effectively (see Greenbaum and Dongarra (1980)). The eigenvectors are normalized so that  $\|z_i\|_2 = 1$ , but are determined only to within a factor  $\pm 1$ .

If only the eigenvalues of  $T$  are required, it is more efficient to call nag\_dsterf (f08jfc) instead. If  $T$  is positive definite, small eigenvalues can be computed more accurately by nag\_dpsteqr (f08jgc).

## 4 References

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

Greenbaum A and Dongarra J J (1980) Experiments with QR/QL methods for the symmetric triangular eigenproblem *LAPACK Working Note No. 17 (Technical Report CS-89-92)* University of Tennessee, Knoxville <http://www.netlib.org/lapack/lawnspdf/lawn17.pdf>

Parlett B N (1998) *The Symmetric Eigenvalue Problem* SIAM, Philadelphia

## 5 Arguments

1: **order** – Nag\_OrderType *Input*

*On entry:* the **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 **order** = Nag\_RowMajor. See Section 2.3.1.3 in How to Use the NAG Library and its Documentation for a more detailed explanation of the use of this argument.

*Constraint:* **order** = Nag\_RowMajor or Nag\_ColMajor.

2: **compz** – Nag\_ComputeZType *Input*

*On entry:* indicates whether the eigenvectors are to be computed.

**compz** = Nag\_NotZ

Only the eigenvalues are computed (and the array **z** is not referenced).

**compz** = Nag\_UpdateZ

The eigenvalues and eigenvectors of  $A$  are computed (and the array **z** must contain the matrix  $Q$  on entry).

**compz** = Nag\_InitZ

The eigenvalues and eigenvectors of  $T$  are computed (and the array **z** is initialized by the function).

*Constraint:* **compz** = Nag\_NotZ, Nag\_UpdateZ or Nag\_InitZ.

3: **n** – Integer *Input*

*On entry:*  $n$ , the order of the matrix  $T$ .

*Constraint:*  $n \geq 0$ .

4: **d**[*dim*] – double *Input/Output*

**Note:** the dimension, *dim*, of the array **d** must be at least  $\max(1, n)$ .

*On entry:* the diagonal elements of the tridiagonal matrix  $T$ .

*On exit:* the  $n$  eigenvalues in ascending order, unless **fail.code** = NE\_CONVERGENCE (in which case see Section 6).

5: **e**[*dim*] – double *Input/Output*

**Note:** the dimension, *dim*, of the array **e** must be at least  $\max(1, n - 1)$ .

*On entry:* the off-diagonal elements of the tridiagonal matrix  $T$ .

*On exit:* **e** is overwritten.

6: **z**[*dim*] – double *Input/Output*

**Note:** the dimension, *dim*, of the array **z** must be at least  $\mathbf{pdz} \times n$  when **compz** = Nag\_UpdateZ or Nag\_InitZ.

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

$$\begin{aligned} & \mathbf{z}[(j-1) \times \mathbf{pdz} + i - 1] \text{ when } \mathbf{order} = \text{Nag\_ColMajor}; \\ & \mathbf{z}[(i-1) \times \mathbf{pdz} + j - 1] \text{ when } \mathbf{order} = \text{Nag\_RowMajor}. \end{aligned}$$

*On entry:* if **compz** = Nag\_UpdateZ, **z** must contain the orthogonal matrix  $Q$  from the reduction to tridiagonal form.

If **compz** = Nag\_InitZ, **z** must be allocated, but its contents need not be set.

If **compz** = Nag\_NotZ, **z** is not referenced and may be **NULL**.

*On exit:* if **compz** = Nag\_UpdateZ or Nag\_InitZ, the  $n$  required orthonormal eigenvectors stored as columns of  $Z$ ; the  $i$ th column corresponds to the  $i$ th eigenvalue, where  $i = 1, 2, \dots, n$ , unless **fail.errnum** > 0.

**z** is not changed if **compz** = Nag\_NotZ.

7: **pdz** – Integer *Input*

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

*Constraints:*

if **compz** = Nag\_UpdateZ or Nag\_InitZ, **pdz** ≥ **n**;  
if **compz** = Nag\_NotZ, **z** may be **NULL**.

8: **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  $\langle value \rangle$  had an illegal value.

### NE\_CONVERGENCE

The algorithm has failed to find all the eigenvalues after a total of  $30 \times \mathbf{n}$  iterations. In this case, **d** and **e** contain on exit the diagonal and off-diagonal elements, respectively, of a tridiagonal matrix orthogonally similar to  $T$ .  $\langle value \rangle$  off-diagonal elements have not converged to zero.

### NE\_ENUM\_INT\_2

On entry, **compz** =  $\langle value \rangle$ , **pdz** =  $\langle value \rangle$  and **n** =  $\langle value \rangle$ .

Constraint: if **compz** = Nag\_UpdateZ or Nag\_InitZ, **pdz** ≥ **n**.

### NE\_INT

On entry, **n** =  $\langle value \rangle$ .

Constraint: **n** ≥ 0.

On entry, **pdz** =  $\langle value \rangle$ .

Constraint: **pdz** > 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.

**7 Accuracy**

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

$$\|E\|_2 = O(\epsilon)\|T\|_2,$$

and  $\epsilon$  is the *machine precision*.

If  $\lambda_i$  is an exact eigenvalue and  $\tilde{\lambda}_i$  is the corresponding computed value, then

$$|\tilde{\lambda}_i - \lambda_i| \leq c(n)\epsilon\|T\|_2,$$

where  $c(n)$  is a modestly increasing function of  $n$ .

If  $z_i$  is the corresponding exact eigenvector, and  $\tilde{z}_i$  is the corresponding computed eigenvector, then the angle  $\theta(\tilde{z}_i, z_i)$  between them is bounded as follows:

$$\theta(\tilde{z}_i, z_i) \leq \frac{c(n)\epsilon\|T\|_2}{\min_{i \neq j} |\lambda_i - \lambda_j|}.$$

Thus the accuracy of a computed eigenvector depends on the gap between its eigenvalue and all the other eigenvalues.

**8 Parallelism and Performance**

nag\_dsteqr (f08jec) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

nag\_dsteqr (f08jec) 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 typically about  $24n^2$  if **compz** = Nag\_NotZ and about  $7n^3$  if **compz** = Nag\_UpdateZ or Nag\_InitZ, but depends on how rapidly the algorithm converges. When **compz** = Nag\_NotZ, the operations are all performed in scalar mode; the additional operations to compute the eigenvectors when **compz** = Nag\_UpdateZ or Nag\_InitZ can be vectorized and on some machines may be performed much faster.

The complex analogue of this function is nag\_zsteqr (f08jsc).

## 10 Example

This example computes all the eigenvalues and eigenvectors of the symmetric tridiagonal matrix  $T$ , where

$$T = \begin{pmatrix} -6.99 & -0.44 & 0.00 & 0.00 \\ -0.44 & 7.92 & -2.63 & 0.00 \\ 0.00 & -2.63 & 2.34 & -1.18 \\ 0.00 & 0.00 & -1.18 & 0.32 \end{pmatrix}.$$

See also the examples for `nag_dorgtr` (f08ffc), `nag_dopgtr` (f08gfc) or `nag_dsbtrd` (f08hec), which illustrate the use of this function to compute the eigenvalues and eigenvectors of a full or band symmetric matrix.

### 10.1 Program Text

```

/* nag_dsteqr (f08jec) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagf08.h>
#include <nagf16.h>
#include <nagx04.h>

int main(void)
{
    /* Scalars */
    Integer i, j, n, pdz, d_len, e_len;
    Integer exit_status = 0;
    /* Arrays */
    double *z = 0, *d = 0, *e = 0;
    /* Nag Types */
    NagError fail;
    Nag_OrderType order;

#ifdef NAG_COLUMN_MAJOR
#define Z(I, J) z[(J - 1) * pdz + I - 1]
    order = Nag_ColMajor;
#else
#define Z(I, J) z[(I - 1) * pdz + J - 1]
    order = Nag_RowMajor;
#endif

    INIT_FAIL(fail);

    printf("nag_dsteqr (f08jec) Example Program Results\n\n");

    /* Skip heading in data file */
#ifdef _WIN32
    scanf_s("%*[\n] ");
#else
    scanf("%*[\n] ");
#endif
#ifdef _WIN32
    scanf_s("%" NAG_IFMT "%*[\n] ", &n);
#else
    scanf("%" NAG_IFMT "%*[\n] ", &n);
#endif
    pdz = n;
    d_len = n;
    e_len = n - 1;

```

```

/* Allocate memory */
if (!(z = NAG_ALLOC(n * n, double)) ||
    !(d = NAG_ALLOC(d_len, double)) || !(e = NAG_ALLOC(e_len, double)))
{
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
}
/* Read T from data file */
for (i = 0; i < d_len; ++i)
#ifdef _WIN32
    scanf_s("%lf", &d[i]);
#else
    scanf("%lf", &d[i]);
#endif
for (i = 0; i < e_len; ++i)
#ifdef _WIN32
    scanf_s("%lf", &e[i]);
#else
    scanf("%lf", &e[i]);
#endif
/* Calculate all the eigenvalues and eigenvectors of tridiagonal matrix T,
 * reduced from real symmetric matrix using nag_dsteqr (f08jec).
 */
nag_dsteqr(order, Nag_InitZ, n, d, e, z, pdz, &fail);
if (fail.code != NE_NOERROR) {
    printf("Error from nag_dsteqr (f08jec).\n%s\n", fail.message);
    exit_status = 1;
    goto END;
}

/* Standardize the eigenvectors so that first elements are non-negative. */
for (j = 1; j <= n; j++) {
    if (z(1,j) < 0.0) {
        for (i = 1; i <= n; i++)
            z(i, j) = -z(i, j);
    }
}

/* Print eigenvalues and eigenvectors */
printf(" Eigenvalues\n");
for (i = 0; i < n; ++i)
    printf("  %7.4lf", d[i]);
printf("\n\n");

/* Print real general matrix Z using easy-to-use
 * nag_gen_real_mat_print (x04cac).
 */
fflush(stdout);
nag_gen_real_mat_print(order, Nag_GeneralMatrix, Nag_NonUnitDiag, n, n, z,
    pdz, "Eigenvectors", 0, &fail);
if (fail.code != NE_NOERROR) {
    printf("Error from nag_gen_real_mat_print (x04cac).\n%s\n", fail.message);
    exit_status = 1;
}

END:
    NAG_FREE(d);
    NAG_FREE(e);
    NAG_FREE(z);
    return exit_status;
}

```

## 10.2 Program Data

```

nag_dsteqr (f08jec) Example Program Data
4                               :Value of N
-6.99   7.92   2.34   0.32
-0.44  -2.63  -1.18           :End of matrix T

```

### **10.3 Program Results**

nag\_dsteqr (f08jec) Example Program Results

Eigenvalues

-7.0037 -0.4059 2.0028 8.9968

Eigenvectors

	1	2	3	4
1	0.9995	0.0109	0.0167	0.0255
2	0.0310	-0.1627	-0.3408	-0.9254
3	0.0089	-0.5170	-0.7696	0.3746
4	0.0014	-0.8403	0.5397	-0.0509

---