

NAG C 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

```
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

$$T = Z\Lambda Z^T,$$

where Λ is a diagonal matrix whose diagonal elements are the eigenvalues λ_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 :

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

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) + nag_dorgtr (f08ffc)
full matrix, packed storage	nag_dsprtd (f08gec) + nag_dopgtr (f08gfc)
band matrix	nag_dsbtrd (f08hec) with vect = 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 ± 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

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

5 Parameters

- 1: **order** – Nag_OrderType *Input*
On entry: the **order** parameter 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 2.2.1.4 of the Essential Introduction for a more detailed explanation of the use of this parameter.
Constraint: **order = Nag_RowMajor** or **Nag_ColMajor**.
- 2: **compz** – Nag_ComputeZType *Input*
On entry: indicates whether the eigenvectors are to be computed as follows:
 if **compz = Nag_NotZ**, only the eigenvalues are computed (and the array **z** is not referenced);
 if **compz = Nag_InitZ**, the eigenvalues and eigenvectors of T are computed (and the array **z** is initialised by the routine);
 if **compz = Nag_UpdateZ**, the eigenvalues and eigenvectors of A are computed (and the array **z** must contain the matrix Q on entry).
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** > 0 (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: the array is overwritten.
- 6: **z**[*dim*] – double *Input/Output*
Note: the dimension, *dim*, of the array **z** must be at least $\max(1, \mathbf{pdz} \times n)$ when **compz = Nag_UpdateZ** or **Nag_InitZ**.
 If **order = Nag_ColMajor**, the (i, j) th element of the matrix Z is stored in $\mathbf{z}[(j - 1) \times \mathbf{pdz} + i - 1]$ and if **order = Nag_RowMajor**, the (i, j) th element of the matrix Z is stored in $\mathbf{z}[(i - 1) \times \mathbf{pdz} + j - 1]$.
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 a **NULL** pointer, i.e., (double *) 0.
On exit: if **compz = Nag_InitZ** or **Nag_UpdateZ**, 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** > 0.
z is not changed if **compz = Nag_NotZ**.

- 7: **pdz** – Integer *Input*
On entry: the first dimension of the array **z** as declared in the function from which nag_dsteqr (f08jec) is called. When **z** is a NULL pointer then **pdz** should be set to 0.
Constraints:
 if **compz** = Nag_NotZ, **pdz** ≥ 0;
 if **compz** = Nag_UpdateZ or Nag_InitZ, **pdz** ≥ max(1, **n**).
- 8: **fail** – NagError * *Output*
 The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_INT

On entry, **n** = *⟨value⟩*.
 Constraint: **n** ≥ 0.

On entry, **pdz** = *⟨value⟩*.
 Constraint: **pdz** > 0.

NE_ENUM_INT_2

On entry, **compz** = *⟨value⟩*, **n** = *⟨value⟩*, **pdz** = *⟨value⟩*.
 Constraint: if **compz** = Nag_UpdateZ or Nag_InitZ, **pdz** ≥ max(1, **n**);
 if **compz** = Nag_NotZ, **pdz** ≥ 1.

On entry, **compz** = *⟨value⟩*, **n** = *⟨value⟩*, **pdz** = *⟨value⟩*.
 Constraint: if **compz** = Nag_NotZ, **pdz** ≥ 0;
 if **compz** = Nag_UpdateZ or Nag_InitZ, **pdz** ≥ max(1, **n**).

On entry, **n** = *⟨value⟩*, **compz** = *⟨value⟩*, **pdz** = *⟨value⟩*.
 Constraint: if **compz** = Nag_UpdateZ or Nag_InitZ, **pdz** ≥ max(1, **n**);
 if **compz** = Nag_NotZ, **pdz** ≥ 1.

NE_CONVERGENCE

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

NE_ALLOC_FAIL

Memory allocation failed.

NE_BAD_PARAM

On entry, parameter *⟨value⟩* had an illegal value.

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 consult NAG for assistance.

7 Accuracy

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

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

and ϵ is the *machine precision*.

If λ_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 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).

9 Example

To compute 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.

9.1 Program Text

```

/* nag_dsteqr (f08jec) Example Program.
 *
 * Copyright 2001 Numerical Algorithms Group.
 *
 * Mark 7, 2001.
 */

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

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

#ifdef NAG_COLUMN_MAJOR
    order = Nag_ColMajor;
#else

```

```

    order = Nag_RowMajor;
#endif

    INIT_FAIL(fail);
    Vprintf("f08jec Example Program Results\n\n");

    /* Skip heading in data file */
    Vscanf("%*[^\\n] ");
    Vscanf("%ld%*[^\\n] ", &n);
    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)) )
    {
        Vprintf("Allocation failure\n");
        exit_status = -1;
        goto END;
    }
    /* Read T from data file */
    for (i = 0; i < d_len; ++i)
        Vscanf("%lf", &d[i]);
    for (i = 0; i < e_len; ++i)
        Vscanf("%lf", &e[i]);
    /* Calculate all the eigenvalues and eigenvectors of T */
    f08jec(order, Nag_InitZ, n, d, e, z, pdz, &fail);
    if (fail.code != NE_NOERROR)
    {
        Vprintf("Error from f08jec.\n%s\n", fail.message);
        exit_status = 1;
        goto END;
    }
    /* Print eigenvalues and eigenvectors */
    Vprintf(" Eigenvalues\n");
    for (i = 0; i < n; ++i)
        Vprintf("  %7.4lf", d[i]);
    Vprintf("\n\n");
    x04cac(order, Nag_GeneralMatrix, Nag_NonUnitDiag, n, n,
           z, pdz, "Eigenvectors", 0, &fail);
    if (fail.code != NE_NOERROR)
    {
        Vprintf("Error from x04cac.\n%s\n", fail.message);
        exit_status = 1;
        goto END;
    }
END:
    if (d) NAG_FREE(d);
    if (e) NAG_FREE(e);
    if (z) NAG_FREE(z);
    return exit_status;
}

```

9.2 Program Data

```

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

```

9.3 Program Results

```

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
