NAG Library Function Document

nag_dsterf (f08jfc)

1 Purpose

nag_dsterf (f08jfc) computes all the eigenvalues of a real symmetric tridiagonal matrix.

2 Specification

```
#include <nag.h>
#include <nagf08.h>
void nag_dsterf (Integer n, double d[], double e[], NagError *fail)
```

3 Description

nag_dsterf (f08jfc) computes all the eigenvalues of a real symmetric tridiagonal matrix, using a square-root-free variant of the QR algorithm.

The function uses an explicit shift, and, like nag_dsteqr (f08jec), switches between the QR and QL variants in order to handle graded matrices effectively (see Greenbaum and Dongarra (1980)).

4 References

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: **n** – Integer *Input*

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

Constraint: $\mathbf{n} \geq 0$.

2: **d**[dim] – double *Input/Output*

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

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).

3: **e**[dim] – double *Input/Output*

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

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

On exit: **e** is overwritten.

4: **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_CONVERGENCE

The algorithm has failed to find all the eigenvalues after a total of $30 \times \mathbf{n}$ iterations; ⟨*value*⟩ elements of **e** have not converged to zero.

NE_INT

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

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.

7 Accuracy

The computed eigenvalues 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

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

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

8 Parallelism and Performance

Not applicable.

9 Further Comments

The total number of floating-point operations is typically about $14n^2$, but depends on how rapidly the algorithm converges. The operations are all performed in scalar mode.

There is no complex analogue of this function.

10 Example

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

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

10.1 Program Text

```
/* nag_dsterf (f08jfc) Example Program.
 *
 * Copyright 2014 Numerical Algorithms Group.
 *
 * Mark 7, 2001.
 */

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

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

  INIT_FAIL(fail);

  printf("nag_dsterf (f08jfc) 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
  d_len = n;
  e_len = n - 1;

  /* Allocate memory */
  if (!(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 of T*/
  /* nag_dsterf (f08jfc).
   * All eigenvalues of real symmetric tridiagonal matrix,
   * root-free variant of QL or QR
   */
  nag_dsterf(n, d, e, &fail);
  if (fail.code != NE_NOERROR)
    {
      printf("Error from nag_dsterf (f08jfc).\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }
  /* Print eigenvalues */
  printf(" Eigenvalues\n");
  for (i = 0; i < n; ++i)
    printf("  %7.4lf", d[i]);
  printf("\n");
 END:
  NAG_FREE(d);
  NAG_FREE(e);
  return exit_status;
}
```

10.2 Program Data

```
nag_dsterf (f08jfc) 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_dsterf (f08jfc) Example Program Results

 Eigenvalues
  -7.0037  -0.4059   2.0028   8.9968
```