nag_real_symm_eigensystem (f02abc)

1. Purpose

nag_real_symm_eigensystem (f02abc) calculates all the eigenvalues and eigenvectors of a real symmetric matrix.

2. Specification

```
#include <nag.h>
#include <nagf02.h>

void nag_real_symm_eigensystem(Integer n, double a[], Integer tda,
     double r[], double v[], Integer tdv, NagError *fail)
```

3. Description

This function reduces the real symmetric matrix A to a real symmetric tridiagonal matrix by Householder's method. The eigenvalues and eigenvectors are calculated using the QL algorithm.

4. Parameters

n

Input: n, the order of the matrix A.
Constraint: $\mathbf{n} \geq 1$.

a[n][tda]

Input: the lower triangle of the n by n symmetric matrix A. The elements of the array above the diagonal need not be set. See also Section 6.

tda

Input: the second dimension of the array **a** as declared in the function from which nag_real_symm_eigensystem is called.
Constraint: $\mathbf{tda} \geq \mathbf{n}$.

r[n]

Output: the eigenvalues in ascending order.

v[n][tdv]

Output: the normalised eigenvectors, stored by columns; the ith column corresponds to the ith eigenvalue. The eigenvectors are normalised so that the sum of squares of the elements is equal to 1.

tdv

Input: the second dimension of the array **v** as declared in the function from which nag_real_symm_eigensystem is called.
Constraint: $\mathbf{tdv} \geq \mathbf{n}$.

fail

The NAG error parameter, see the Essential Introduction to the NAG C Library.

5. Error Indications and Warnings

NE_TOO_MANY_ITERATIONS

More than ⟨*value*⟩ iterations are required to isolate all the eigenvalues.

NE_INT_ARG_LT

On entry, **n** must not be less than 1: $\mathbf{n} = \langle value \rangle$.

NE_2_INT_ARG_LT

On entry, $\mathbf{tda} = \langle value \rangle$ while $\mathbf{n} = \langle value \rangle$. These parameters must satisfy $\mathbf{tda} \geq \mathbf{n}$.
On entry, $\mathbf{tdv} = \langle value \rangle$ while $\mathbf{n} = \langle value \rangle$. These parameters must satisfy $\mathbf{tdv} \geq \mathbf{n}$.

NE_ALLOC_FAIL

Memory allocation failed.

6. Further Comments

The time taken by the function is approximately proportional to n^3.

The function may be called with the same actual array supplied for parameters **a** and **v**, in which case the eigenvectors will overwrite the original matrix.

6.1. Accuracy

The eigenvectors are always accurately orthogonal but the accuracy of the individual eigenvectors is dependent on their inherent sensitivity to changes in the original matrix. For a detailed error analysis see Wilkinson and Reinsch (1971) pp 222 and 235.

6.2. References

Wilkinson J H and Reinsch C (1971) *Handbook for Automatic Computation (Vol II, Linear Algebra)* Springer-Verlag pp 212–226 and 227–240.

7. See Also

None.

8. Example

To calculate all the eigenvalues and eigenvectors of the real symmetric matrix

$$\begin{pmatrix} 0.5 & 0.0 & 2.3 & -2.6 \\ 0.0 & 0.5 & -1.4 & -0.7 \\ 2.3 & -1.4 & 0.5 & 0.0 \\ -2.6 & -0.7 & 0.0 & 0.5 \end{pmatrix}.$$

8.1. Program Text

```
/* nag_real_symm_eigensystem(f02abc) Example Program
 *
 * Copyright 1990 Numerical Algorithms Group.
 *
 * Mark 2 revised, 1992.
 */

#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagf02.h>

#define NMAX 8
#define TDA NMAX
#define TDV NMAX



main()
{
  Integer  i, j, n;
  double a[NMAX][TDA], r[NMAX], v[NMAX][TDV];

  Vprintf("f02abc Example Program Results\n");
  /* Skip heading in data file */
  Vscanf("%*[^\n]");
  Vscanf("%ld", &n);
  if (n<1 || n>NMAX)
    {
      Vfprintf(stderr, "n is out of range: n = %5ld\n", n);
      exit(EXIT_FAILURE);
    }
  for (i=0; i<n; i++)
    for (j=0; j<n; j++)
      Vscanf("%lf", &a[i][j]);
```

```
  f02abc(n, (double *)a, (Integer)TDA, r, (double *)v, (Integer)TDV,
         NAGERR_DEFAULT);
  Vprintf("Eigenvalues\n");
  for (i=0; i<n; i++)
    Vprintf("%9.4f%s", r[i], (i%8==7 || i==n-1) ? "\n" : " ");
  Vprintf("Eigenvectors\n");
  for ( i=0; i<n; i++)
    for (j=0; j<n; j++)
      printf("%9.4f%s", v[i][j], (j%8==7 || j==n-1) ? "\n" : " ");
  exit(EXIT_SUCCESS);
}
```

8.2. Program Data

```
f02abc Example Program Data
  4
  0.5  0.0  2.3 -2.6
  0.0  0.5 -1.4 -0.7
  2.3 -1.4  0.5  0.0
 -2.6 -0.7  0.0  0.5
```

8.3. Program Results

```
f02abc Example Program Results
Eigenvalues
  -3.0000   -1.0000    2.0000    4.0000
Eigenvectors
   0.7000   -0.1000    0.1000    0.7000
  -0.1000   -0.7000    0.7000   -0.1000
  -0.5000   -0.5000   -0.5000    0.5000
   0.5000   -0.5000   -0.5000   -0.5000
```