nag_real_eigensystem (f02agc)

1. Purpose

nag_real_eigensystem (f02agc) calculates all the eigenvalues and eigenvectors of a real unsymmetric matrix.

2. Specification

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

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

3. Description

The matrix A is first balanced and then reduced to upper Hessenberg form using real stabilised elementary similarity transformations. The eigenvalues and eigenvectors of the Hessenberg matrix are calculated using the QR algorithm. The eigenvectors of the Hessenberg matrix are back-transformed to give the eigenvectors of the original matrix A.

4. Parameters

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

a[n][tda]
Input: the n by n matrix A.
Output: the array is overwritten.

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

r[n]
Output: the eigenvalues.

v[n][tdv]
Output: the eigenvectors, stored by columns. The ith column corresponds to the ith eigenvalue. The eigenvectors are normalised so that the sum of the squares of the moduli of the elements is equal to 1 and the element of largest modulus is real. This ensures that real eigenvalues have real eigenvectors.

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

iter[n]
Output: $\mathbf{iter}[i-1]$ contains the number of iterations used to find the ith eigenvalue. If $\mathbf{iter}[i-1]$ is negative, the ith eigenvalue is the second of a pair found simultaneously.
Note: the eigenvalues are found in reverse order, starting with the nth.

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: **n** = ⟨*value*⟩.

NE_2_INT_ARG_LT
On entry, **tda** = ⟨*value*⟩ while **n** = ⟨*value*⟩. These parameters must satisfy **tda** ≥ **n**.
On entry, **tdv** = ⟨*value*⟩ while **n** = ⟨*value*⟩. These parameters must satisfy **tdv** ≥ **n**.

NE_ALLOC_FAIL
Memory allocation failed.

6. Further Comments

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

6.1. Accuracy

The accuracy of the results depends on the original matrix and the multiplicity of the roots. For a detailed error analysis see Wilkinson and Reinsch (1971) pp 352 and 390.

6.2. References

Wilkinson J H and Reinsch C (1971) *Handbook for Automatic Computation (Vol II, Linear Algebra)* Springer-Verlag pp 339–358 and 372–395.

7. See Also

None.

8. Example

To calculate all the eigenvalues and eigenvectors of the real matrix

$$\begin{pmatrix} 1.5 & 0.1 & 4.5 & -1.5 \\ -22.5 & 3.5 & 12.5 & -2.5 \\ -2.5 & 0.3 & 4.5 & -2.5 \\ -2.5 & 0.1 & 4.5 & 2.5 \end{pmatrix}.$$

8.1. Program Text

```
/* nag_real_eigensystem(f02agc) Example Program
 *
 * Copyright 1989 Numerical Algorithms Group.
 *
 * Mark 1, 1990.
 */

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

#define NMAX 4
#define TDA NMAX
#define TDV NMAX
#define COMPLEX(A) A.re, A.im

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

  Vprintf("f02agc 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]);
  f02agc(n, (double *)a, (Integer)TDA, r, (Complex *)v,
         (Integer)TDV, iter, NAGERR_DEFAULT);
  Vprintf("Eigenvalues\n");
  for (i=0; i<n; i++)
    Vprintf("(%7.3f, %7.3f) \n", COMPLEX(r[i]));
  Vprintf("\nEigenvectors\n");
  for (i=0; i<n; i++)
    for (j=0; j<n; j++)
      Vprintf("(%7.3f, %7.3f) %s", COMPLEX(v[i][j]),
              (j%4==3 || j==n-1)? "\n" : " ");
  exit(EXIT_SUCCESS);
}
```

8.2. Program Data

```
f02agc Example Program Data
  4
   1.5   0.1   4.5  -1.5
 -22.5   3.5  12.5  -2.5
  -2.5   0.3   4.5  -2.5
  -2.5   0.1   4.5   2.5
```

8.3. Program Results

```
f02agc Example Program Results
Eigenvalues
(  3.000,   4.000)
(  3.000,  -4.000)
(  4.000,   0.000)
(  2.000,   0.000)

Eigenvectors
(  0.113,  -0.151)  (  0.113,   0.151)  ( -0.033,   0.000)  (  0.063,   0.000)
(  0.945,   0.000)  (  0.945,   0.000)  (  0.988,   0.000)  (  0.996,   0.000)
(  0.189,   0.000)  (  0.189,   0.000)  (  0.011,   0.000)  (  0.006,   0.000)
(  0.113,  -0.151)  (  0.113,   0.151)  (  0.154,   0.000)  (  0.063,   0.000)
```