

## nag\_real\_eigenvalues (f02afc)

### 1. Purpose

**nag\_real\_eigenvalues (f02afc)** calculates all the eigenvalues of a real unsymmetric matrix.

### 2. Specification

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

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

### 3. Description

The matrix  $A$  is first balanced and then reduced to upper Hessenberg form using stabilised elementary similarity transformations. The eigenvalues are then found using the  $QR$  algorithm for real Hessenberg matrices.

### 4. Parameters

**n**

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

**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\_eigenvalues** is called.  
Constraint: **tda**  $\geq$  **n**.

**r[n]**

Output: the eigenvalues.

**iter[n]**

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

**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  $\langle value \rangle$  iterations are required to isolate all the eigenvalues.

#### NE\_INT\_ARG\_LT

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

#### NE\_2\_INT\_ARG\_LT

On entry, **tda** =  $\langle value \rangle$  while **n** =  $\langle value \rangle$ . These parameters must satisfy **tda**  $\geq$  **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 367.

### 6.2. References

Wilkinson J H and Reinsch C (1971) *Handbook for Automatic Computation (Vol II, Linear Algebra)* Springer-Verlag pp 339–371.

### 7. See Also

None.

### 8. Example

To calculate all the eigenvalues 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_eigenvalues(f02afc) 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 4
#define TDA NMAX
#define COMPLEX(A) A.re, A.im

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

    Vprintf("f02afc 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]);
    f02afc(n, (double *)a, (Integer)TDA, r, iter, NAGERR_DEFAULT);
    Vprintf("Eigenvalues\n");
    for (i=0; i<n; i++)
        Vprintf("( %7.3f , %7.3f ) \n", COMPLEX(r[i]));
    exit(EXIT_SUCCESS);
}
```

**8.2. Program Data**

f02afc 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**

f02afc Example Program Results

Eigenvalues

|   |       |   |        |   |
|---|-------|---|--------|---|
| ( | 3.000 | , | 4.000  | ) |
| ( | 3.000 | , | -4.000 | ) |
| ( | 4.000 | , | 0.000  | ) |
| ( | 2.000 | , | 0.000  | ) |

---