NAG Library Function Document

nag_real_symm_eigenvalues (f02aac)

1 Purpose

nag_real_symm_eigenvalues (f02aac) calculates all the eigenvalues of a real symmetric matrix.

2 Specification

```
#include <nag.h>
#include <nagf02.h>
void nag_real_symm_eigenvalues (Integer n, double a[], Integer tda,
     double r[], NagError *fail)
```

3 Description

nag_real_symm_eigenvalues (f02aac) reduces the real symmetric matrix A to a real symmetric tridiagonal matrix using Householder's method. The eigenvalues of the tridiagonal matrix are then determined using the QL algorithm.

4 References

Wilkinson J H and Reinsch C (1971) *Handbook for Automatic Computation II, Linear Algebra* Springer–Verlag

5 Arguments

1: **n** – Integer *Input*

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

Constraint: **n** ≥ 1.

2: **a**[**n** $\times$ **tda**] – double *Input/Output*

Note: the (i,j)th element of the matrix A is stored in **a**[$(i-1) \times$ **tda** $+ j - 1$].

On entry: the lower triangle of the n by n symmetric matrix A. The elements of the array above the diagonal need not be set.

On exit: the elements of A below the diagonal are overwritten, and the rest of the array is unchanged.

3: **tda** – Integer *Input*

On entry: the stride separating matrix column elements in the array **a**.

Constraint: **tda** $\geq$ **n**.

4: **r**[**n**] – double *Output*

On exit: the eigenvalues in ascending order.

5: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_2_INT_ARG_LT

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

NE_ALLOC_FAIL

Dynamic memory allocation failed.

NE_INT_ARG_LT

On entry, **n** $= \langle value \rangle$.
Constraint: **n** ≥ 1.

NE_TOO_MANY_ITERATIONS

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

7 Accuracy

The accuracy of the eigenvalues depends on the sensitivity of the matrix to rounding errors produced in tridiagonalisation. For a detailed error analysis see pages 222 and 235 of Wilkinson and Reinsch (1971).

8 Parallelism and Performance

Not applicable.

9 Further Comments

The time taken by nag_real_symm_eigenvalues (f02aac) is approximately proportional to n^3.

10 Example

To calculate all the eigenvalues of the real symmetric matrix

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

10.1 Program Text

```
/* nag_real_symm_eigenvalues (f02aac) Example Program.
 *
 * Copyright 2014 Numerical Algorithms Group.
 *
 * Mark 1, 1990.
 * Mark 8 revised, 2004.
 */

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

#define A(I, J) a[(I) *tda + J]
int main(void)
{


  Integer  exit_status = 0, i, j, n, tda;
  NagError fail;
```

```
  double   *a = 0, *r = 0;

  INIT_FAIL(fail);

  printf(
          "nag_real_symm_eigenvalues (f02aac) Example Program Results\n");
  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n]");
#else
  scanf("%*[^\n]");
#endif
#ifdef _WIN32
  scanf_s("%"NAG_IFMT"", &n);
#else
  scanf("%"NAG_IFMT"", &n);
#endif

  if (n >= 1)
    {
      if (!(a = NAG_ALLOC(n*n, double)) ||
          !(r = NAG_ALLOC(n, double)))
        {
          printf("Allocation failure\n");
          exit_status = -1;
          goto END;
        }
      tda = n;
    }
  else
    {
      printf("Invalid n.\n");
      exit_status = 1;
      return exit_status;
    }
  for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
#ifdef _WIN32
      scanf_s("%lf", &A(i, j));
#else
      scanf("%lf", &A(i, j));
#endif
  /* nag_real_symm_eigenvalues (f02aac).
   * All eigenvalues of real symmetric matrix
   */
  nag_real_symm_eigenvalues(n, a, tda, r, &fail);
  if (fail.code != NE_NOERROR)
    {
      printf("Error from nag_real_symm_eigenvalues (f02aac).\n%s\n",
              fail.message);
      exit_status = 1;
      goto END;
    }
  printf("Eigenvalues\n");
  for (i = 0; i < n; i++)
    printf("%9.4f%s", r[i], (i%8 == 7 || i == n-1)?"\n":" ");
 END:
  NAG_FREE(a);
  NAG_FREE(r);
  return exit_status;
}
```

10.2 Program Data

```
nag_real_symm_eigenvalues (f02aac) 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
```

10.3 Program Results

```
nag_real_symm_eigenvalues (f02aac) Example Program Results
Eigenvalues
  -3.0000   -1.0000    2.0000    4.0000
```