NAG Toolbox

nag_matop_real_symm_posdef_geneig (f01bv)

1 Purpose

nag_matop_real_symm_posdef_geneig (f01bv) transforms the generalized symmetric-definite eigenproblem $Ax = \lambda \mathbf{b}x$ to the equivalent standard eigenproblem $Cy = \lambda y$, where A, **b** and C are symmetric band matrices and **b** is positive definite. **b** must have been decomposed by nag_matop_real_symm_posdef_fac (f01bu).

2 Syntax

```
[a, b, ifail] = nag_matop_real_symm_posdef_geneig(k, a, b, 'n', n, 'ma1', ma1,
'mb1', mb1)
```

```
[a, b, ifail] = f01bv(k, a, b, 'n', n, 'ma1', ma1, 'mb1', mb1)
```

Note: the interface to this routine has changed since earlier releases of the toolbox:

At Mark 22: **ma1** and **mb1** were made optional.

3 Description

A is a symmetric band matrix of order n and bandwidth $2m_A + 1$. The positive definite symmetric band matrix B, of order n and bandwidth $2m_B + 1$, must have been previously decomposed by nag_matop_real_symm_posdef_fac (f01bu) as $ULDL^{\mathrm{T}}U^{\mathrm{T}}$. nag_matop_real_symm_posdef_geneig (f01bv) applies U, L and D to A, m_A rows at a time, restoring the band form of A at each stage by plane rotations. The argument k defines the change-over point in the decomposition of B as used by nag_matop_real_symm_posdef_fac (f01bu) and is also used as a change-over point in the transformations applied by this function. For maximum efficiency, k should be chosen to be the multiple of m_A nearest to $n/2$. The resulting symmetric band matrix C is overwritten on **a**. The eigenvalues of C, and thus of the original problem, may be found using nag_lapack_dsbtrd (f08he) and nag_lapack_dsterf (f08jf). For selected eigenvalues, use nag_lapack_dsbtrd (f08he) and nag_lapack_dstebz (f08jj).

4 References

Crawford C R (1973) Reduction of a band-symmetric generalized eigenvalue problem *Comm. ACM* **16** 41–44

5 Parameters

5.1 Compulsory Input Parameters

1: **k** – INTEGER

Suggested value: the optimum value is the multiple of m_A nearest to $n/2$.

k, the change-over point in the transformations. It must be the same as the value used by nag_matop_real_symm_posdef_fac (f01bu) in the decomposition of B.

Constraint: $\mathbf{mb1} - 1 \leq \mathbf{k} \leq \mathbf{n}$.

2: $\mathbf{a}(lda, \mathbf{n})$ – REAL (KIND=nag_wp) array

lda, the first dimension of the array, must satisfy the constraint $lda \geq \mathbf{ma1}$.

The upper triangle of the n by n symmetric band matrix A, with the diagonal of the matrix stored in the $(m_A + 1)$th row of the array, and the m_A superdiagonals within the band stored in the first m_A rows of the array. Each column of the matrix is stored in the corresponding column of the array. For example, if $n = 6$ and $m_A = 2$, the storage scheme is

$$\begin{array}{cccccc} * & * & a_{13} & a_{24} & a_{35} & a_{46} \\ * & a_{12} & a_{23} & a_{34} & a_{45} & a_{56} \\ a_{11} & a_{22} & a_{33} & a_{44} & a_{55} & a_{66} \end{array}$$

Elements in the top left corner of the array need not be set. The following code assigns the matrix elements within the band to the correct elements of the array:

```
for j=1:n
   for i=max(1,j-ma1+1):j
     a(i-j+ma1,j) = matrix(i,j);
   end
 end
```

3: **b**(ldb, **n**) – REAL (KIND=nag_wp) array

ldb, the first dimension of the array, must satisfy the constraint $ldb \geq$ **mb1**.

The elements of the decomposition of matrix B as returned by nag_matop_real_symm_posdef_fac (f01bu).

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the second dimension of the arrays **a**, **b**. (An error is raised if these dimensions are not equal.)

n, the order of the matrices A, B and C.

2: **ma1** – INTEGER

Default: the first dimension of the array **a**.

$m_A + 1$, where m_A is the number of nonzero superdiagonals in A. Normally **ma1** $\ll$ **n**.

3: **mb1** – INTEGER

Default: the first dimension of the array **b**.

$m_B + 1$, where m_B is the number of nonzero superdiagonals in B.

Constraint: **mb1** $\leq$ **ma1**.

5.3 Output Parameters

1: **a**(lda, **n**) – REAL (KIND=nag_wp) array

Stores the corresponding elements of C.

2: **b**(ldb, **n**) – REAL (KIND=nag_wp) array

The elements of **b** will have been permuted.

3: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **mb1** $>$ **ma1**.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

In general the computed system is exactly congruent to a problem $(A + E)x = \lambda(B + F)x$, where $\|E\|$ and $\|F\|$ are of the order of $\epsilon\kappa(B)\|A\|$ and $\epsilon\kappa(B)\|B\|$ respectively, where $\kappa(B)$ is the condition number of B with respect to inversion and ϵ is the ***machine precision***. This means that when B is positive definite but not well-conditioned with respect to inversion, the method, which effectively involves the inversion of B, may lead to a severe loss of accuracy in well-conditioned eigenvalues.

8 Further Comments

The time taken by nag_matop_real_symm_posdef_geneig (f01bv) is approximately proportional to $n^2m_B^2$ and the distance of k from $n/2$, e.g., $k = n/4$ and $k = 3n/4$ take 502% longer.

When B is positive definite and well-conditioned with respect to inversion, the generalized symmetric eigenproblem can be reduced to the standard symmetric problem $Py = \lambda y$ where $P = L^{-1}AL^{-\mathrm{T}}$ and $B = LL^{\mathrm{T}}$, the Cholesky factorization.

When A and B are of band form, especially if the bandwidth is small compared with the order of the matrices, storage considerations may rule out the possibility of working with P since it will be a full matrix in general. However, for any factorization of the form $B = SS^{\mathrm{T}}$, the generalized symmetric problem reduces to the standard form

$$S^{-1}AS^{-\mathrm{T}}\left(S^{\mathrm{T}}x\right) = \lambda\left(S^{\mathrm{T}}x\right)$$

and there does exist a factorization such that $S^{-1}AS^{-\mathrm{T}}$ is still of band form (see Crawford (1973)). Writing

$$C = S^{-1}AS^{-\mathrm{T}} \quad \text{and} \quad y = S^{\mathrm{T}}x$$

the standard form is $Cy = \lambda y$ and the bandwidth of C is the maximum bandwidth of A and B.

Each stage in the transformation consists of two phases. The first reduces a leading principal sub-matrix of B to the identity matrix and this introduces nonzero elements outside the band of A. In the second, further transformations are applied which leave the reduced part of B unaltered and drive the extra elements upwards and off the top left corner of A. Alternatively, B may be reduced to the identity matrix starting at the bottom right-hand corner and the extra elements introduced in A can be driven downwards.

The advantage of the $ULDL^{\mathrm{T}}U^{\mathrm{T}}$ decomposition of B is that no extra elements have to be pushed over the whole length of A. If k is taken as approximately $n/2$, the shifting is limited to halfway. At each stage the size of the triangular bumps produced in A depends on the number of rows and columns of B which are eliminated in the first phase and on the bandwidth of B. The number of rows and columns over which these triangles are moved at each step in the second phase is equal to the bandwidth of A.

In this function, **a** is defined as being at least as wide as B and must be filled out with zeros if necessary as it is overwritten with C. The number of rows and columns of B which are effectively eliminated at each stage is m_A.

9 Example

This example finds the three smallest eigenvalues of $Ax = \lambda Bx$, where

$$A = \begin{pmatrix} 11 & 12 & & & & & & & \\ 12 & 12 & 13 & & & & & & \\ & 13 & 13 & 14 & & & & & \\ & & 14 & 14 & 15 & & & & \\ & & & 15 & 15 & 16 & & & \\ & & & & 16 & 16 & 17 & & \\ & & & & & 17 & 17 & 18 & \\ & & & & & & 18 & 18 & 19 \\ & & & & & & & 19 & 19 \end{pmatrix}$$

$$B = \begin{pmatrix} 101 & 22 & & & & & & & \\ 22 & 102 & 23 & & & & & & \\ & 23 & 103 & 24 & & & & & \\ & & 24 & 104 & 25 & & & & \\ & & & 25 & 105 & 26 & & & \\ & & & & 26 & 106 & 27 & & \\ & & & & & 27 & 107 & 28 & \\ & & & & & & 28 & 108 & 29 \\ & & & & & & & 29 & 109 \end{pmatrix}.$$

9.1 Program Text

```
     function f01bv_example

fprintf('f01bv example results\n\n');

% The matrices A and B
n = 9;
kd = nag_int(1);
a = [  0,  12,  13,  14,  15,  16,  17,  18,  19;
      11,  12,  13,  14,  15,  16,  17,  18,  19];
b = [  0,  22,  23,  24,  25,  26,  27,  28,  29;
     101, 102, 103, 104, 105, 106, 107, 108, 109];

% Decompose B
k = nag_int(4);
[b, ifail] = f01bu(k, b);

% Use decomposition to transform Ax = lambda Bx to Cy = lambda y
[c, b, ifail] = f01bv(k, a, b);

% Reduce C to tridiagonal form
vect = 'No vectors';
uplo = 'Upper';
q = zeros(n, n);
[~, d, e, ~, info] = f08he( ...
                            vect, uplo, kd, c, q);

% Compute eigenvalues from tridiagonal form
m1 = nag_int(1);
m2 = nag_int(3);
range = 'Indexed';
order = 'Blocked';
vl = 0;
vu = 0;
abstol = 0;
[m, nsplit, w, iblock, isplit, info] = ...
```

```
f08jj( ...
       range, order, vl, vu, m1, m2, abstol, d, e);

disp('Selected eigenvalues');
disp(w(1:m)');
```

9.2 Program Results

```
     f01bv example results

Selected eigenvalues
   -0.2643   -0.1530   -0.0418
```