1 SUMMARY

This subroutine uses the Lanczos algorithm to **compute the part of the spectrum of a large symmetric matrix A that lies in a specified interval,** that is, it computes eigenvalues without regard to multiplicities. The user need only provide **A** in the form of code that computes **u**+**Av** for any given vectors **u** and **v**. Auxiliary calls allow corresponding eigenvectors to be found.

ATTRIBUTES — **Version:** 1.1.0. (17 February 2005) **Types:** Real (single, double). **Calls:** FA14, FD15. **Original date:** May 2001. **Remark:** EA25 is a threadsafe version of EA15. **Origin:** J.K.Reid, Harwell.

2 HOW TO USE THE PACKAGE

Although there are both single and double precision versions of the routine available, the user is **strongly advised** to use the double precision version unless single precision on his or her machine actually means 8-byte arithmetic.

2.1 Overall control

Rather than ask the user to furnish a subroutine to compute **u**+**Av** inside the Lanczos process, control is returned at each iteration so that code for computing **u**+**Av** can be written by the user in line. On such returns, those eigenvalues that have been so far computed will be available. Following successful termination for one interval, a new interval of the spectrum of the same matrix may be processed more economically by a first call with the ends of the new interval held in the arguments EL and ER and all other arguments unchanged. Auxiliary calls to EA25E/ED may be made to find eigenvectors corresponding to computed eigenvalues.

Simple examples of suitable calls are given in §5.

2.2 The argument list and calling sequence (initialization)

The EA25I/ID entry must be called prior to the first call to the EA25A/AD entry to initialize private workspace.

The single precision version

```
CALL EA25I(KEEP,RKEEP)
```

The double precision version

```
CALL  EA25ID(KEEP,RKEEP)
```

KEEP is an INTEGER array of length 10 used by EA25 as private workspace and should not be altered by the user. The element KEEP(8) holds the random number seed used by FA14, which is used by EA25 to generate a random start vector. In exceptional circumstances the user may wish to reset it after the EA25I/ID call to to obtain a different random sequence.

RKEEP is a REAL (DOUBLE PRECISION in the D version) array of length 7 used by EA25 as private workspace and must not be altered by the user.

2.3 The argument list and calling sequence (main entry)

The single precision version

```
CALL EA25A(N,EL,ER,ACC,LEIG,LX,LALFA,LP,ITAPE,IFLAG,U,V,
*           EIG,JEIG,NEIG,X,DEL,NU,ALFA,BETA,KEEP,RKEEP)
```

The double precision version

```
CALL EA25AD(N,EL,ER,ACC,LEIG,LX,LALFA,LP,ITAPE,IFLAG,U,V,
```

```
     *              EIG,JEIG,NEIG,X,DEL,NU,ALFA,BETA,KEEP,RKEEP)
```

N is an INTEGER variable which must be set to the order n of the matrix **A**. N is not altered. **Restriction:** N > 0.

EL,ER are REAL (DOUBLE PRECISION in the D version) variables which must be set to indicate the range [EL,ER] within which the eigenvalues are required. EL may be set to a very large negative number if no lower bound is required and ER may be set to a very large positive number if no upper bound is required. If EL ≥ ER, it is assumed that all eigenvalues are wanted. EL,ER are not altered.

ACC is a REAL (DOUBLE PRECISION in the D version) variable which must be set to the required relative precision (relative to the largest eigenvalue of **A**). If ACC is negative or very small, as much accuracy as the precision reasonably allows will be obtained. ACC is not altered.

LEIG is an INTEGER variable which must be set to the length of array EIG. It must be as least as large as the number of distinct eigenvalues in [EL,ER]. LEIG is not altered.

LX is an INTEGER variable which must be set to the length of arrays X, DEL and NU. A value three times the number of distinct eigenvalues in [EL,ER] usually suffices. LX is not altered. **Restriction:** LX > 5.

LALFA is an INTEGER variable which must be set to the length of arrays ALFA and BETA. It limits the number of Lanczos steps possible. The number needed varies widely from problem to problem. In our experimental use of the code we have never needed more than $5n$, and when looking at a small interval near the end of the spectrum have sometimes needed less than $n/4$. LALFA is not altered.

LP is an INTEGER variable which must be set to the unit number for diagnostic messages. If LP ≤ 0, no messages are printed. LP is not altered.

ITAPE is an INTEGER variable which must be set to the unit number for a sequential file which is needed if eigenvectors are to be calculated by EA25E/ED. This file must have room for LALFA records, each of length n. It must not be accessed or altered by the user. If eigenvectors are not wanted, ITAPE should be set non-positive and no file is used. ITAPE is not altered.

IFLAG is an INTEGER variable which must be set prior to the first call for a particular matrix to one of the following values:

–1 The user does not want to specify a start vector.

–2 The user has placed a start vector in V.

It should not otherwise be changed by the user. In particular it should be left at the value

0 if the user wishes to follow a successful call for one interval of the spectrum of a matrix by a call for another interval of the spectrum of the same matrix.

On output it has one of the following values:

0 Successful completion of calculation.

1 Normal intermediate return.

2 LEIG is too small. This error return permits subsequent recall with a reduced range [EL,ER], IFLAG reset to zero, and other arguments unchanged.

3 LX is too small. This error return permits subsequent recall with a reduced range [EL,ER], IFLAG reset to zero, and other arguments unchanged.

4 LALFA is too small.

5 N ≤ 0, LX ≤ 5, or LALFA ≤ 0.

6 Premature termination, probably caused by the initial vector **v** being an eigenvector. EIG will hold NEIG eigenvalues but there may be more in the given interval. If the user began with IFLAG=–2, he or she is recommended to restart with IFLAG=–1.

7 The input value of IFLAG is outside the range [–2,1].

U,V are REAL (DOUBLE PRECISION in the D version) arrays of length *n*. On a return with IFLAG=1, the user must add to U the vector **Av**, where **v** is the vector stored in V, without altering V. If the user wishes to specify the Lanczos starting vector, it should be placed in V before the first (IFLAG=–2) call for the matrix **A**. Otherwise the user must never set or alter U or V.

EIG is a REAL (DOUBLE PRECISION in the D version) array of length LEIG. It should never be set or altered by the user. On any return it contains NEIG computed eigenvalues, stored in increasing order. If IFLAG > 0, there may be more in the requested interval.

JEIG is an INTEGER array of size (2,LEIG). It should never be set or altered by the user. On any return JEIG(1,I) contains the Lanczos step at which EIG(I) was accepted and JEIG(2,I) contains a value needed for the recurrences to get the eigenvector, for I = 1, 2,..., NEIG.

NEIG is an INTEGER variable which should never be set or altered by the user. On return it holds the number of eigenvalues in EIG.

X,DEL are REAL (DOUBLE PRECISION in the D version) arrays of length LX, which should never be set or altered by the user.

NU is an INTEGER array of length LX which should never be set or altered by the user.

ALFA,BETA are REAL (DOUBLE PRECISION in the D version) arrays of length LALFA used to hold the tridiagonal matrix generated by the Lanczos process. The user should never set or alter these arguments.

KEEP is an INTEGER array of length 10 used by EA25 as private workspace, see Section 2.1.

RKEEP is a REAL (DOUBLE PRECISION in the D version) array of length 7 used by EA25 as private workspace, see Section 2.1.

2.4 The argument list and calling sequence (eigenvector entry)

The single precision version

```
      CALL EA25E(N,LALFA,LP,ITAPE,EIG,JEIG,NEIG,
     *           ALFA,BETA,LY,LZ,JFLAG,Y,W,Z,KEEP)
```

The double precision version

```
      CALL EA25ED(N,LALFA,LP,ITAPE,EIG,JEIG,NEIG,
     *           ALFA,BETA,LY,LZ,JFLAG,Y,W,Z,KEEP)
```

N is an INTEGER variable which must be set to the order *n* of the matrix **A**. N is not altered. **Restriction:** N > 0.

LALFA is an INTEGER variable which must be set to the length of arrays ALFA and BETA. LALFA is not altered.

LP is an INTEGER variable which must be set to the unit number for diagnostic messages. If LP ≤ 0, no messages are printed. LP is not altered.

ITAPE is an INTEGER variable which must be set to the unit number for the sequential file which was used by EA25A/AD. ITAPE is not altered.

EIG is a REAL (DOUBLE PRECISION in the D version) array of length NEIG. It must hold some or all of the eigenvalues computed on a call of EA25A/AD. Note that where EA25A/AD finds a large number of eigenvalues, the corresponding eigenvectors may be found by a sequence of calls of EA25E/ED each of which finds some of them, so avoiding massive storage demands in arrays Y and Z. EIG is not altered.

JEIG is an INTEGER array of size (2,NEIG). It must hold the information on the eigenvalues whose eigenvectors are wanted, as returned by EA25A/AD, namely JEIG(1,I) must contain the Lanczos step at which EIG(I) was accepted and JEIG(2,I) must contain the value needed for the recurrences to get the eigenvector, for I = 1, 2,..., NEIG. JEIG is not altered.

NEIG is an INTEGER variable which must be set to the number of eigenvalues in EIG. NEIG is not altered.

ALFA,BETA are REAL (DOUBLE PRECISION in the D version) arrays of length LALFA which must be as left by EA25A/AD. The user should never set or alter these arguments.

LY is an INTEGER variable which must be set to the first dimension of array Y. It is not altered. **Restriction:** LY $\geq n$.

LZ is an INTEGER variable which must be set to the first dimension of array Z. It is not altered. **Restriction:** LZ $\geq$ max(JEIG(1,I),I=1,NEIG).

JFLAG is an INTEGER variable which need not be set. On return it has one of the following values:

0 Successful completion of calculation.

1 N ≤ 0, LALFA ≤ 0, ITAPE ≤ 0 or NEIG ≤ 0.

2 LY $< n$.

3 LZ $<$ max(JEIG(1,I),I=1,NEIG).

Y is a REAL (DOUBLE PRECISION in the D version) array of size (LY,NEIG) used to return eigenvectors. The eigenvector corresponding to EIG(I) is placed in Y(K,I),K=1,N.

W is a REAL (DOUBLE PRECISION in the D version) array of length n used for workspace.

Z is a REAL (DOUBLE PRECISION in the D version) array of size (LZ,NEIG) used for workspace.

KEEP is an INTEGER array of length 10 used by EA25 as private workspace. and must not be altered by the user.

2.5 Sequential file for Lanczos vectors

The Lanczos vectors are stored and recovered from a sequential file by subroutine EA25F/FD. If another method of storage is desired, this subroutine may be replaced. EA25A/AD calls EA25F/FD with a storage request for each vector in natural order and EA25E/ED calls EA25F/FD with a recovery request for each vector in natural order. The argument list is as follows.

The single precision version

```
CALL EA25F(N,ITAPE,NLAN,IO,V,LAST)
```

The double precision version

```
CALL EA25FD(N,ITAPE,NLAN,IO,V,LAST)
```

N is an INTEGER variable that holds the order n of the matrix $\mathbf{A}$. N must not be altered by the subroutine.

ITAPE is an INTEGER variable that holds the unit number of the sequential file. ITAPE must not be altered by the subroutine.

NLAN is an INTEGER variable that holds the Lanczos iteration number. The vectors are always recovered in order (NLAN=1,2,...), but further vectors may be sent for storage after such a sequence of recoveries. NLAN must not be altered by the subroutine.

IO is an INTEGER variable that has the value 1 if a vector is to be stored and to 2 if one is to be recovered. If NLAN and IO both have the value 1, this indicates that a new problem is in hand and any stored vectors may be discarded. If NLAN>1 and IO=1, it can be assumed that the previous vectors have already been sent. IO must not be altered by the subroutine.

V is a REAL (DOUBLE PRECISION in the D version) array of length N. If IO has the value 1, V holds the vector to be stored and must not be altered by the subroutine. If IO has the value 2, V must be set by the subroutine to the vector previously sent to it for iteration NLAN.

LAST is an INTEGER variable which on entry has the value it had on the previous return. It is used by the HSL version of EA25F/FD to remember the previous NLAN value.

3 GENERAL INFORMATION

Use of common: None.

Workspace: Provided in the arguments X, DEL, NU, ALFA, BETA, W and Z.

Other routines called directly: FA14A/AD, FD15A/AD, EA25C/CD, EA25D/DD, EA25F/FD, EA25G/GD.

Input/output: Diagnostic messages on unit LP (see §2.3) and Lanczos vectors stored on unit ITAPE (see §2.3).

Restrictions: N > 0, LX > 5, LY ≥ *n*, LZ ≥ max(JEIG(1,I),I=1,NEIG).

4 METHOD

The Lanczos algorithm generates a sequence of symmetric tridiagonal matrices $\mathbf{T}_j$, each of which is the *j*-th order leading principal submatrix of all subsequent tridiagonal matrices. Each eigenvalue of **A** is the limit of a sequence constructed by taking a particular eigenvalue from each $\mathbf{T}_j$. The subroutine aims to follow the convergence of these sequences by constructing intervals containing the eigenvalues of $\mathbf{T}_j$ and having lengths comparable with the separation from corresponding eigenvalues of $\mathbf{T}_{j-1}$, continuing until it is convinced that all eigenvalues have been found. For further details see 'Tracking the progress of the Lanczos algorithm for large symmetric eigenproblems', by B. N. Parlett and J. K. Reid, IMA Journal of Numerical Analysis (1981) **1**, 135-155.

5 EXAMPLE OF USE

Suppose the spectrum of the matrix

$$\mathbf{A} = \begin{pmatrix} \mathbf{T} & -\mathbf{I} & \\ -\mathbf{I} & \mathbf{T} & -\mathbf{I} \\ & -\mathbf{I} & \mathbf{T} \end{pmatrix}, \quad \text{where} \quad \mathbf{T} = \begin{pmatrix} 4 & -1 & \\ -1 & 4 & -1 \\ & -1 & 4 & -1 \\ & & -1 & 4 \end{pmatrix}$$

is required. The following code finds it in two parts and finds the eigenvectors corresponding to the two smallest eigenvalues in the two parts.

```
      DOUBLE PRECISION EL,ER,ACC,EIG(10),X(50),DEL(50),ALFA(40),
     +  BETA(40),U(12),V(12),Y(12,6),W(12),Z(40,6),RKEEP(7)
      INTEGER NU(50),JEIG(2,10),N,LEIG,LX,LALFA,LP,ITAPE,IFLAG,
     +  LY,LZ,IRANGE,ITER,NEIG,K,I,J,JFLAG,KEEP(10)

      CALL EA25ID(KEEP,RKEEP)
      EL=0.0D0
      ER=4.0D0
      ACC=1.0D-9
      OPEN(3,STATUS='UNKNOWN',FORM='UNFORMATTED')
      N=12
      LEIG=10
      LX=50
      LALFA=40
      LP=6
      ITAPE=3
      IFLAG=-1
      LY=12
      LZ=40
      DO 60 IRANGE=1,2
        DO 30 ITER=1,40
          CALL EA25AD(N,EL,ER,ACC,LEIG,LX,LALFA,LP,ITAPE,IFLAG,U,V,
     *                EIG,JEIG,NEIG,X,DEL,NU,ALFA,BETA,KEEP,RKEEP)
          IF(IFLAG.EQ.0)GO TO 40
          IF(IFLAG.GT.1)GO TO 70
```

```
C IFLAG=1. Form U=U+A*V
          K=1
          DO 20 I=1,3
            DO 10 J=1,4
              U(K)=U(K)+4.0D0*V(K)
              IF(I.GT.1)U(K)=U(K)-V(K-4)
              IF(I.LT.3)U(K)=U(K)-V(K+4)
              IF(J.GT.1)U(K)=U(K)-V(K-1)
              IF(J.LT.4)U(K)=U(K)-V(K+1)
              K=K+1
   10       CONTINUE
   20     CONTINUE
   30   CONTINUE
        GO TO 70

C Write out spectrum found
   40   WRITE (LP,FMT=50) EL,ER, (EIG(I),I=1,NEIG)
   50   FORMAT (/,' Spectrum in interval (',F5.1,',',F5.1,') IS',/,
     +         10F7.4)

C Find two eigenvectors
        NEIG=2
        CALL EA25ED(N,LALFA,LP,ITAPE,EIG,JEIG,NEIG,
     *            ALFA,BETA,LY,LZ,JFLAG,Y,W,Z,KEEP)
        DO 57 I=1,2
          WRITE(LP,55)EIG(I),(Y(K,I),K=1,N)
   55     FORMAT (' Eigenvector corresponding to eigenvalue',F7.4,
     +           ' is',/, (4F10.4))
   57   CONTINUE

C Reset EL,ER for second interval
        EL=4.0D0
        ER=8.0D0
   60 CONTINUE
      GO TO 100

C EA25A/AD is signalling failure
   70 WRITE(LP,80)IFLAG
   80 FORMAT (' EA25AD has failed. IFLAG=',I2)

  100 STOP
      END
```

This produces the following output

```
 Spectrum in interval (  0.0,  4.0) IS
 0.9678 1.9678 2.3820 3.2038 3.3820 3.7962
 Eigenvector corresponding to eigenvalue 0.9678 is
   -0.1859   -0.3008   -0.3008   -0.1859
   -0.2629   -0.4253   -0.4253   -0.2629
   -0.1859   -0.3008   -0.3008   -0.1859
 Eigenvector corresponding to eigenvalue 1.9678 is
   -0.3008   -0.1859    0.1859    0.3008
   -0.4253   -0.2629    0.2629    0.4253
   -0.3008   -0.1859    0.1859    0.3008

 Spectrum in interval (  4.0,  8.0) IS
 4.2038 4.6180 4.7962 5.6180 6.0322 7.0322
 Eigenvector corresponding to eigenvalue 4.2038 is
   -0.1859    0.3008   -0.3008    0.1859
```

```
    -0.2629    0.4253   -0.4253    0.2629
    -0.1859    0.3008   -0.3008    0.1859
  Eigenvector corresponding to eigenvalue 4.6180 is
    -0.4253    0.2629    0.2629   -0.4253
     0.0000    0.0000    0.0000    0.0000
     0.4253   -0.2629   -0.2629    0.4253
```