A User's Guide

A.1 Summary

HLZPACK is a Fortran 77 implementation of a Lanczos based method to compute some of the extreme eigenvalues λ and associated eigenvectors x of a complex Hermitian matrix $\mathcal{H}$, i.e., of the problem $\mathcal{H}x = \lambda x$. The method is implemented in a one-vector form in combination with a modified partial reorthogonalization strategy.

A.2 How to use the routine

The user interface for HLZPACK is the subroutine HLZDRS/D, which sets values for control parameters, computes some eigenvalues of $\mathcal{H}$ (usually the extreme ones) and, if required, the associated eigenvectors. The algorithm used by HLZPACK requires the multiplication of a vector by the matrix $\mathcal{H}$, or alternatively a solution of a system of equations, until the convergence for the number of solutions required is reached. However, every time a matrix-vector computation has to be performed the control is returned to the user (reverse communication strategy). Thus, the matrix $\mathcal{H}$ does not have to be passed as an argument. A simple example of the utilization of HLZDRS/D is given in Section A.5.

A.2.1 Argument list

The single precision version,

CALL HLZDRS (LFLAG,LPSET,INFO,HNRM,EIG,RNRM,WORK,Q,R,BASIS,X)

The double precision version,

CALL HLZDRD (LFLAG,LPSET,INFO,HNRM,EIG,RNRM,WORK,Q,R,BASIS,X)

LFLAG is an INTEGER variable used to control the code as follows:

- LFLAG<0, an error has been detected (output, the computation is interrupted and the control is returned to the user).
- LFLAG=0, must be used on the first call to HLZDRS/D (input).
- LFLAG=1, the user must compute the product $\mathcal{H}$Q, assign it to R (or assign to R the solution of $\mathcal{H}$R=Q), and call HLZDRS/D again (output).
- LFLAG=2, the convergence for the solutions required, or the maximum number of steps, or an invariant subspace, has been reached (output, the control is returned to the user).
- LFLAG=3, must be used for the computation of the eigenvectors corresponding to the positive entries in RNRM (input).

LPSET is an INTEGER array of length 5 holding basic parameters which must be set by the user as follows. It is not altered by the subroutine.

LPSET(1)=N, where N is the dimension of the Hermitian matrix $\mathcal{H}$.
LPSET(2)=NREIG, where NREIG is the the number of eigenpairs required.
LPSET(3)=NSTEPS, where NSTEPS is the maximum number of steps to be performed, i.e., the maximum dimension of the basis.
LPSET(4)=LPRNT, where LPRNT is the information to be printed. See Section A.2.2 for more information.
LPSET(5)=LN, where LPRNT is the first dimension of the arrays X and BASIS.

INFO is an INTEGER array of size 4 used to hold information about the execution of the code:
INFO(1)=J, where J is the current dimension of the basis (step counter).
INFO(2)=NCEIG, where NCEIG is the number of eigenpairs converged with the current basis.
INFO(3)=LWARN, where LWARN is a code for warning messages. A value of zero indicates a normal execution. For nonzero values, see Section A.2.3.
INFO(4)=LRERR, where LRERR is a code for error messages (associated with LFLAG<0). A value of zero indicates a normal execution. For nonzero values, see Section A.2.4.

HNRM is a REAL (DOUBLE PRECISION in the D version) variable which can be set by the user to the norm of the Hermitian matrix. If it is set to zero, an application dependent value will be assigned (see Section A.4).

EIG is a REAL (DOUBLE PRECISION in the D version) array of length NSTEPS which need not be set by the user. On each exit with LFLAG=1 or LFLAG=2, it holds the eigenvalue approximations.

RNRM is a REAL (DOUBLE PRECISION in the D version) array of length NSTEPS which need not be set by the user. On each exit with LFLAG=1 or LFLAG=2, it holds the residuals of the eigenpair approximations: the positive ones indicate the pairs that satisfy the convergence criterion (see Section A.4).

WORK is a REAL (DOUBLE PRECISION in the D version) array of length at least (NSTEPS+10)*NSTEPS which need not be set by the user. It is used to store the reduced eigenproblem, the associated solutions and other information. This array must not be altered by the user between calls to HLZDRS/D.

Q, R are COMPLEX*8 (COMPLEX*16 in the D version) arrays of length N. On each exit with LFLAG=1 the user must multiply $\mathcal{H}$ by Q, place the result in R, and recall HLZDRS/D (maintaining LFLAG=1). As an option, one can solve $\mathcal{H}$R=Q, placing the result in R. On the first call with LFLAG=0 the user can specify the starting vector for the basis generation process by defining the components of Q, $\|Q\| \neq 0$, otherwise random entries are generated.

BASIS is a COMPLEX*8 (COMPLEX*16 in the D version) two-dimensional array with dimensions (LN,NSTEPS) which need not be set by the user. It is used to store the basis generated by the algorithm (the Lanczos vectors). This array must not be altered by the user between calls to HLZDRS/D.

X is a COMPLEX*8 (COMPLEX*16 in the D version) two-dimensional array with dimensions (LN,NSTEPS) which need not be set by the user. On a call with LFLAG=3 the subroutine HLZDRS/D will use the information available in RNRM and WORK to compute the eigenvectors. Only the eigenvectors corresponding to positive entries in RNRM are computed and stored in X (and printed, if required). However, such entries can be redefined by the user.

A.2.2 Print options

The input parameter LPSET(4) is used as a flag for printing information, as indicated in Table 1. Therefore, if one wants to print only the header, the exit values and the eigenvalues, for example, LPSET(4) should be set to 6, i.e., the sum of the corresponding codes.

A.2.3 Warning and error messages

The output parameter INFO(3) is used as a flag for warning messages, as indicated in Table 2. Therefore, a value of INFO(3) equal to 24 (the sum of the codes 8 and 16), for instance, indicates that no solution converged with NSTEPS and the eigenvectors can not be computed.

The output parameter INFO(4) is used as a flag for error messages whenever LFLAG<0, as indicated in Table 3. Therefore, a value of INFO(4) equal to 24 (the sum of the codes 8 and 16), for instance, indicates an invalid specification of the input parameters LPSET(2) and LPSET(3).

A.2.4 Choosing a value for NSTEPS

The value of NSTEPS required for the convergence of NREIG solutions depends on the starting vector and mainly on the eigenvalue distribution of the associated problem. However, a value for NSTEPS between 2×NREIG and 5×NREIG is recommended if no information on the application is available.

A.3 General information

Invariant subspace. After some steps the norm or the vector r_j (see Section A.4) may drop to zero (actually the machine precision), which means that an invariant subspace has been computed. If this is the case, all the eigenpairs of the tridiagonal matrix lead to solutions of the original problem and the algorithm can not be continued (this situation is clearly reached for J=N). However, it can happen that the number of solutions converged is less than the number of solutions required. Then, one possible remedy is to run the algorithm again (with LFLAG set to 0) using a different starting vector, preferably orthogonal to the previously converged eingenvectors.

Table 1: Options for printing.

code	meaning
0	nothing is printed
2	prints header and exit parameters
4	prints eigenvalues
8	prints eigenvectors
16	prints eigenvalue approximations at each step
32	prints orthogonality control warning
64	prints warning messages
128	prints error messages

Table 2: Table of warnings.

code	meaning
0	normal execution
2	nothing to be done (N=1)
4	invariant subspace found at j-th step
8	no solution converged with NSTEPS
16	eigenvectors can not be computed
32	not enough solutions converged

Table 3: Table of errors.

code	meaning
0	normal execution
2	invalid specification of LFLAG
4	invalid specification of LPSET(1), N
8	invalid specification of LPSET(2), NREIG
16	invalid specification of LPSET(3), NSTEPS
32	invalid specification of LPSET(4), LPRNT
64	invalid specification of LPSET(5), LN
128	reduced eigenproblem could not be solved

The converged solutions. All converged solutions are printed when the printing code is properly set. Similarly, all eigenvectors corresponding to positive entries in RNRM are computed when HLZDRS/D is called with LFLAG=3. Even if no solution has satisfied the convergence criterion, the user can "force" the computation of approximate eigenvectors by changing the signs of the entries in RNRM.

Intrinsic functions. The package uses intrinsic functions to generate random numbers and to perform bit tests. However, the Makefile provided with the package can generate executables on many different platforms. If the Makefile does not fit on a particular machine the user should perform modifications accordingly.

BLAS. The following BLAS kernels are called by the package: IDAMAX, ZAXPY, ZCOPY, ZDOTC, ZGEMM, ZGERU, ZSCAL, ZSYMM and ZTRMM.

A.4 Method

The algorithm used by HLZPACK is based on the Lanczos method, in combination with a modified partial reorthogonalization strategy, as summarized below. It should be noted that $\mathcal{H}$ corresponds to the complex Hermitian matrix, $t_{i,j}$ are the entries of T_j, ϵ is the machine precision, $s_j^{(l)}$ is the bottom element of the eigenvector s_l, q_j is the j-th column of the matrix $\mathcal{Q}_j$, NCEIG is the number of eigenpairs converged, NREIG is the number of eigenpairs required and NSTEPS is the maximum number of steps to be performed. If $\|\mathcal{H}\|$ is not given (i.e., HNRM is set to zero on input), the largest absolute eigenvalue of T_j is used in the convergence criterion (step 2.j).

1. *Initialization*:

set $q_0 = 0$ and $\beta_1 = 0$
set $q_1 \neq 0$ so that $q_1^* q_1 = 1$

2. *Lanczos steps*:

for j=1,2,... NSTEPS

a) $r_j \leftarrow \mathcal{H} q_j$
b) $r_j \leftarrow r_j - q_{j-1}\beta_j$
c) $\alpha_j \leftarrow q_j^* r_j$
d) $r_j \leftarrow r_j - q_j \alpha_j$
e) $\beta_{j+1} \leftarrow \sqrt{r_j^* r_j}$
f) $q_{j+1} \leftarrow \frac{1}{\beta_{j+1}} r_j$
g) if required orthogonalize q_j and q_{j+1} against the vectors of $\mathcal{Q}_{j-1}$

h) insert q_j into $\mathcal{Q}_j$, $t_{j,j-1} \leftarrow \beta_j$, $t_{j-1,j} \leftarrow \beta_j$, $t_{j,j} \leftarrow \alpha_j$
i) solve the reduced problem $T_j s_l = s_l \theta_l$
j) check NCEIG, the number of eigenpairs for which
$\|\mathcal{H}\hat{x} - \hat{\lambda}\hat{x}\| = \|\beta_{j+1} s_j^{(l)}\| \leq \sqrt{\epsilon}\,\|\mathcal{H}\|$

If NCEIG≥NREIG, exit.

3. *Compute the eingenvector approximations* (related with NCEIG):

$\hat{x} = \mathcal{Q}_j s_l$

A.5 Direct and Invert Operators

The idea of the method is the generation of a basis of vectors, $\mathcal{Q}_j = [q_1 \;\; q_2 \;\; ... \;\; q_j]$, so that the projection of $\mathcal{H}$ onto such a basis leads to a smaller problem involving the symmetric tridiagonal matrix T_j ($\mathcal{Q}_j^* \mathcal{H} \mathcal{Q}_j = T_j$). An eigensolution (θ, s) of the reduced problem leads to an approximate solution $(\hat{\lambda}, \hat{x})$ through the relations $\hat{\lambda} \approx \theta_l$ and $\hat{x} \approx \mathcal{Q}_j s_l$.

As presented in Section A.4, the basis is generated by means of products $\mathcal{H}q_j$. With such a strategy the largest eigenvalues are likely to converge first. If only the smallest eigenvalues are sought, one can invert the spectrum, replacing $\mathcal{H}q_j$ by $\mathcal{H}^{-1}q_j$. However, the matrix $\mathcal{H}$ does not require an inversion because the basis can be generated by solving $\mathcal{H}r_j = q_j$.

A.6 Example of use

The following program illustrates the use of HLZDRS/D. We want to compute the eigenvalues (and eigenvectors) of the matrix

$$\mathcal{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 3+i & 0 & 2-3i & 0 & 0 \\ 0 & -3 & 0 & 0 & 0 & 0 & 0 & 0 & -5i \\ 0 & 0 & 2 & 0 & 0 & 20+3i & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 & 0 & -5-i & 0 \\ 3-i & 0 & 0 & 0 & -5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20-3i & 0 & 0 & 4 & 0 & 0 & 0 \\ 2+3i & 0 & 0 & 0 & 0 & 0 & 13 & 0 & 0 \\ 0 & 0 & 0 & -5+i & 0 & 0 & 0 & -1 & 20i \\ 0 & 5i & 0 & 0 & 0 & 0 & 0 & -20i & 4 \end{bmatrix}$$

For this small matrix all eigenvectors converge at the same time (after N steps), using a random starting vector on a Convex C220. Therefore, all N converged eigenpairs have been printed. It should be noted that different starting vectors might lead to "different" eigenvectors: if (λ, x) corresponds to an eigenpair with $x^*x = 1$, so does $(\lambda, \xi x)$ providing ξ is a complex number satisfying $\bar{\xi}\xi = 1$.

Considering the driver model,

```
      PROGRAM MODEL
C
C**** This is a driver model to illustrate the use of HLDRVS/D *********
C
      INTEGER         N,NREIG,NSTEPS,LPRNT
C
      PARAMETER       (N=9,NREIG=2,NSTEPS=9,LPRNT=14)
C
C***********************************************************************
C
C-----------------------------------------------------------------------
      INTEGER          I,J
      INTEGER          INFO(4),LFLAG,LPSET(5)
      COMPLEX*16       CMPLX0,CMPLX1
      COMPLEX*16       BASIS(N,N),H(N,N),Q(N),R(N),X(N,N)
      DOUBLE PRECISION EIG(N),HNRM,RNRM(N),W((NSTEPS+10)*NSTEPS)
C-----------------------------------------------------------------------
C
      CMPLX0 = ( 0.0D0, 0.0D0 )
      CMPLX1 = ( 1.0D0, 0.0D0 )
C
C.... initialize H ..................................................
C
      DO 10 I = 1,N
C        Q(I) = CMPLX0
         DO 20 J = 1,N
            H(I,J) = CMPLX0
   20    CONTINUE
   10 CONTINUE
C
C.... non zero entries ..............................................
C
      H(1,1) = ( 1.0D0, 0.0D0)
      H(5,1) = ( 3.0D0,-1.0D0)
      H(7,1) = ( 2.0D0, 3.0D0)
      H(2,2) = (-3.0D0, 0.0D0)
      H(9,2) = ( 0.0D0, 5.0D0)
      H(3,3) = ( 2.0D0, 0.0D0)
      H(6,3) = ( 2.0D1,-3.0D0)
      H(4,4) = ( 1.0D1, 0.0D0)
      H(8,4) = (-5.0D0, 1.0D0)
      H(1,5) = ( 3.0D0, 1.0D0)
      H(5,5) = (-5.0D0, 0.0D0)
      H(3,6) = ( 2.0D1, 3.0D0)
      H(6,6) = ( 4.0D0, 0.0D0)
      H(1,7) = ( 2.0D0,-3.0D0)
      H(7,7) = ( 1.3D1, 0.0D0)
      H(4,8) = (-5.0D0,-1.0D0)
```

```
      H(8,8) = (-1.0D0, 0.0D0)
      H(9,8) = ( 0.0D0,-2.0D1)
      H(2,9) = ( 0.0D0,-5.0D0)
      H(8,9) = ( 0.0D0, 2.0D1)
      H(9,9) = ( 4.0D0, 0.0D0)
C
C.... input parameters ..................................................
C
      LFLAG = 0
      HNRM  = 0.0D0
C
      LPSET(1) = N
      LPSET(2) = NREIG
      LPSET(3) = NSTEPS
      LPSET(4) = LPRNT
      LPSET(5) = N
C
C.... look for the solutions required ..................................
C
   30 CONTINUE
C
      CALL HLZDRD (LFLAG,LPSET,INFO,HNRM,EIG,RNRM,W,Q,R,BASIS,X)
C
      IF      ( LFLAG .LT. 0 ) THEN
C
C............ error condition ........................................
C
              WRITE (*,2000) INFO(3),INFO(4)
              STOP
C
      ELSE IF ( LFLAG .EQ. 1 ) THEN
C
C............ R=H*Q, since H is small a BLAS kernel has been used ......
C
              CALL ZGEMV  ('N',N,N,CMPLX1,H,N,Q,1,CMPLX0,R,1)
              GO TO 30
C
      END IF
C
C.... compute the eigenvectors .........................................
C
      CALL HLZDRD (3,LPSET,INFO,HNRM,EIG,RNRM,W,Q,R,BASIS,X)
C
      WRITE (*,2100) INFO(3),INFO(4)
C
      STOP
C
 2000 FORMAT (/'abnormal exit - execution finished',7x,
     &         'warning code:',i4,3x,'error code:',i4)
 2100 FORMAT (/'standard exit - execution finished',7x,
```

```
     &         'warning code:',i4,3x,'error code:',i4)
C
C**** end of MODEL ****************************************************
C
      END
```

the following output is produced:

```
*********************************************
*                                           *
*        Lanczos Algorithm Eigensolver      *
*        - Complex Hermitian Version -      *
*                                           *
*            CERFACS -- april 1994          *
*                                           *
*********************************************

dimension of the problem          :        9
number of solutions required      :        2
maximum number of steps           :        9
printing code                     :       14
roundoff unit                     :  2.2204E-16


* exit parameters *
number of steps performed    :    9
reorthogonalizations         :    0
solutions converged          :    9
warning code                 :    0
error code                   :    0
exit flag                    :    2

eigenvalues and residual norms

   value       norm
 -1.9800E+01  9.5263E-16
 -1.7248E+01  2.6557E-15
 -6.4701E+00  2.7910E-14
 -3.0000E+00  2.1630E-14
 +1.4312E+00  1.1718E-14
 +9.7279E+00  4.4349E-14
 +1.4039E+01  8.9104E-15
 +2.3072E+01  2.4418E-14
 +2.3248E+01  2.1644E-14

eigenvector related to the eigenvalue: -1.9800E+01 -------------------------
  6.6787E-17  1.1493E-16   -1.8678E-01 -4.9921E-02    5.3028E-16  7.2555E-16
  1.1120E-01  5.4892E-02    1.9949E-17  1.1276E-17   -6.2407E-16 -5.9458E-16
  3.6429E-17  1.1818E-17    7.0015E-01  1.8713E-01   -1.6774E-01  6.2760E-01

eigenvector related to the eigenvalue: -1.7248E+01 -------------------------
  3.8164E-17 -1.0799E-16   -1.4745E-16 -1.6740E-16   -4.2237E-01 -5.8847E-01
  6.9389E-17  1.7347E-17   -9.8879E-17  1.9082E-17    4.8064E-01  4.9426E-01
  9.9313E-17  2.1467E-16    1.6653E-16  3.0358E-16   -2.6194E-16  2.7062E-16

eigenvector related to the eigenvalue: -6.4701E+00 -------------------------
```

```
  3.7732E-01 -1.8509E-01    4.9960E-16 -2.0817E-17    1.0755E-16 -4.4409E-16
 -3.0531E-16  5.5511E-17   -6.4410E-01  6.3439E-01   -1.3878E-17  4.5797E-16
 -6.7279E-02 -3.9126E-02   -1.3392E-15  8.3267E-17    1.3878E-16 -1.0547E-15

eigenvector related to the eigenvalue: -3.0000E+00 -------------------------
 -7.0777E-16 -9.7491E-16   -7.1372E-01 -6.5065E-01    1.2837E-16 -2.0123E-16
 -5.6114E-02 -7.6288E-02   -1.6653E-16 -7.4940E-16   -5.5511E-17  2.4980E-16
 -1.0061E-16  5.7072E-16   -1.7843E-01 -1.6266E-01    1.2490E-16 -6.2450E-16

eigenvector related to the eigenvalue:  1.4312E+00 -------------------------
 -4.4158E-01 -7.4289E-01    5.2736E-16  7.1124E-16    2.0296E-16  4.5797E-16
  4.6491E-16 -2.3592E-16   -3.2150E-01 -2.7788E-01   -1.2143E-16 -2.0123E-16
 -1.1630E-01  2.4294E-01    6.8001E-16  2.5674E-16   -6.2450E-17  5.6899E-16

eigenvector related to the eigenvalue:  9.7279E+00 -------------------------
 -1.3323E-15 -9.7145E-16    5.1001E-02 -9.3408E-02   -1.2490E-16  0.0000E+00
 -6.1338E-01  7.3243E-01   -4.9960E-16 -3.6082E-16   -6.9389E-17 -5.5511E-16
 -1.2629E-15 -2.3696E-15   -2.4432E-02  4.4746E-02    2.3778E-01  1.2983E-01

eigenvector related to the eigenvalue:  1.4039E+01 -------------------------
  2.7657E-01 -1.8106E-03   -1.8041E-16 -2.2551E-17   -1.1709E-16 -4.0593E-16
 -8.3267E-17 -7.6328E-17    4.3484E-02 -1.4812E-02   -2.1858E-16 -3.4348E-16
  5.3767E-01  7.9518E-01   -1.0408E-17 -3.3307E-16   -2.7756E-16 -5.7593E-16

eigenvector related to the eigenvalue:  2.3072E+01 -------------------------
  1.1102E-16  4.1633E-17    1.0265E-01 -8.9556E-02   -8.6736E-16 -1.4988E-15
  2.1778E-01 -1.2468E-01    1.6653E-16 -1.6653E-16   -1.0686E-15 -1.5821E-15
  3.8164E-16  7.7022E-16   -4.8478E-01  4.2293E-01    4.6698E-01  5.3527E-01

eigenvector related to the eigenvalue:  2.3248E+01 -------------------------
  6.9389E-17  6.9389E-17    7.6328E-16 -9.0206E-17    2.2386E-01  6.5207E-01
  1.0131E-15  5.8287E-16   -8.3267E-17  1.9429E-16    3.3423E-01  6.4264E-01
 -1.3531E-16 -4.8746E-16   -4.0107E-15 -2.9143E-16   -1.1102E-16  3.4278E-15

standard exit - execution finished       warning code:   0   error code:   0
```