Algorithm xxx: A Testing Infrastructure for Symmetric Tridiagonal Eigensolvers. Usage of `stetester`

OSNI A. MARQUES and CHRISTOF VÖMEL
Lawrence Berkeley National Laboratory
and
JAMES W. DEMMEL and BERESFORD N. PARLETT
University of California, Berkeley

LAPACK is often mentioned as a positive example of a software library that encapsulates complex, robust, and widely used numerical algorithms for a wide range of applications. At installation time, the user has the option of running a (limited) number of test cases to verify the integrity of the installation process. On the algorithm developer's side, however, more exhaustive tests are usually performed to study algorithm behavior on a variety of problem settings and also computer architectures. In this process, difficult test cases need to be found that reflect particular challenges of an application or push algorithms to extreme behavior. These tests are then assembled into a comprehensive collection, therefore making it possible for any new or competing algorithm to be stressed in a similar way.

This document describes the supported macros of the testing infrastrucure and gives examples of input and output files.

Categories and Subject Descriptors: G.1.3 [**Numerical Analysis**]: Numerical Linear Algebra

General Terms: Algorithms, Design

Additional Key Words and Phrases: Eigenvalues, eigenvectors, symmetric matrix, LAPACK, accuracy, performance, test matrices, numerical software, design, implementation, testing.

1. USAGE OF THE TESTING INFRASTRUCTURE STETESTER

This document serves as a reference for use of `stetester`. Section 1.1 contains the supported macros. Section 1.2 shows some of the available test matrices. In

Osni A. Marques and Christof Vömel: Lawrence Berkeley National Laboratory, Computational Research Division, 1 Cyclotron Road, MS 50F-1650, Berkeley, CA 94720, USA, OAMarques,CVoemel@lbl.gov.
James W. Demmel and Beresford N. Parlett: Mathematics Department and Computer Science Division, University of California, Berkeley, CA 94720, USA. demmel@cs.berkeley.edu;parlett@math.berkeley.edu
This work was partly supported by a grant from the National Science Foundation (Cooperative Agreement no. ACI-9619020), and by the Director, Office of Computational and Technology Research, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy under contract No. DE-AC03-76SF00098. (©)

Section 1.3, a sample input file is given that illustrates how to use `stetester`. Section 1.4 shows a generated output file in Matlab format that allows for easy post-processing and plotting of test data.

1.1 Supported macros

Tables I, II, and III contain all currently supported macros in alphabetical order.

Table I. Key words for `stetester`, part 1.

Key word	argument	purpose
`CALLST`	*list*	Defines the subroutines to be tested. Possible entries in *list* are: `STEQRV` (calls `steqr` with COMPZ='V') `STEVXA` (calls `stevx` with RANGE='A') `STEVXI` (calls `stevx` with RANGE='I') `STEVXV` (calls `stevx` with RANGE='V') `STEDCI` (calls `stedc` with COMPZ='I') `STEGRA` (calls `stegr` with RANGE='A') `STEGRI` (calls `stegr` with RANGE='I') `STEGRV` (calls `stegr` with RANGE='V') `ALL` (performs all tests above)
`DUMP`	*list*	Defines data to be written into files. Possible entries in *list* are: `T` (writes the tridiagonal matrix as triplets $i, t_{i,i}, t_{i,i+1}$ in file *stetester.out.T*) `W` (writes the eigenvalues in file *stetester.out.W*) `Z` (writes the eigenvectors in file *stetester.out.Z*) `LOG` (writes timings, residuals and orthogonality level in file *stetester.out.log*) `T.M` (writes the tridiagonal matrix in Matlab format in file *stetester.out.m*) `W.M` (writes the eigenvalues in Matlab format in file *stetester.out.m*) `Z.M` (writes the eigenvectors in Matlab format in file *stetester.out.m*)
`ECOND`	*int*	Sets the condition number for types 1 to 4 in Table V. Possible values of *int* are: 1, then $k = \frac{1}{\sqrt{ulp}}$, default 2, then $k = \frac{1}{n \times \sqrt{ulp}}$ 3, then $k = \frac{1}{10 \times n \times \sqrt{ulp}}$ 4, then $k = \frac{1}{ulp}$ 5, then $k = \frac{1}{n \times ulp}$ 6, then $k = \frac{1}{10 \times n \times ulp}$
`EDIST`	*int*	Sets the random distribution to be used in type 6 in Table V. Possible values of *int* are: 1, for uniform distribution (-1,1), default 2, for uniform distribution (0,1) 3, for normal distribution (0,1)
`ESIGN`	*int*	Assigns (random) signs to the eigenvalues defined in Table V. Possible values of *int* are: 0, then the eigenvalues will not be negative, default 1, then the eigenvalues can be positive, negative or zero

Table II. Key words for `stetester`, part 2.

Key word	argument	purpose
`EIGVAL`		Defines the built-in eigenvalue distributions to be used in the generation of test matrices. The next two lines must set integers $etype_1 \quad etype_2 \quad etype_3 \; \ldots$ $esize_1 \quad esize_2 \quad esize_3 \; \ldots$ where *etype* is a list of types (see Table V) and *esize* is a list of dimensions. A negative *etype* reverses the eigenvalue distribution. For example, *etype*$= -1$ results in $\lambda_i = \frac{1}{k},\ i = 1, 2, \ldots n-1,\ \lambda_n = 1$. *esize* can also be defined with `NMIN[:NINC]:NMAX`, where `NMIN` (> 0) is the minimum dimension, `NMAX` ($\geq$`NMIN`) is the maximum dimension, and `NINC` (> 0) is the increment.
`EIGVALF`	*string*	Defines a file containing an eigenvalue distribution to be used in the generation of a tridiagonal matrix. The file defined by *string* should contain only one entry per line as follows n λ_1 $\vdots$ λ_n
`EIGVI`		Defines indices of the smallest and largest eigenvalues to be computed. The next two lines must define pairs of integers $\mathtt{IL}_1 \; \mathtt{IL}_2 \; \ldots$ $\mathtt{IU}_1 \; \mathtt{IU}_2 \; \ldots$ with $1 \leq \mathtt{IL}_i \leq \mathtt{IU}_i$. These indices are used only in the tests where RANGE='I'.
`EIGVV`		Defines lower and upper bounds of intervals to be searched for eigenvalues. The next two lines must define pairs of values $\mathtt{VL}_1 \; \mathtt{VL}_2 \; \ldots$ $\mathtt{VU}_1 \; \mathtt{VU}_2 \; \ldots$ with $\mathrm{VL}_i \leq \mathrm{VU}_i$. These indices are used only in the tests where RANGE='V'.
`GLUED`		Defines glued matrices. The next fours lines must set $gform_1 \quad gform_2 \quad \ldots \quad gform_{k-1} \quad gform_k$ $gtype_1 \quad gtype_2 \quad \ldots \quad gtype_{k-1} \quad gtype_k$ $gsize_1 \quad gsize_2 \quad \ldots \quad gsize_{k-1} \quad gsize_k$ $\gamma_1 \quad \gamma_2 \quad \cdots \quad \gamma_{k-1}$ where the integers *gform*, *gtype* and *gsize* define, respectively, how the matrix is generated (1 for built-in eigenvalue distribution, 2 for built-in tridiagonal matrix), its type (accordingly to Tables V and IV) and its dimension. The real value γ (real) is the glue factor.

Table III. Key words for `stetester`, part 3.

Key word	argument	purpose
`HBANDA`	k	Sets the halfbandwidth of the symmetric matrix A to be generated and then tridiagonalized; k must be an integer between 1 and 100, which corresponds to $\max(1,(kn)/100))$ subdiagonals, where n is the dimension of the matrix. By default $k = 100$.
`HBANDR`	k	Sets the halfbandwidth of the matrices used in the tests for numerical orthogonality and residual norm; k must be an integer between 0 and 100, then $\max(1,(kn)/100))$ subdiagonals of those matrices are computed. If $k = 0$ the tests are not performed and the corresponding results are simply set to 0. By default, $k = 100$.
`ISEED`	k_1 k_2 k_3 k_4	Sets the (initial) seed of the random number generator. Each (integer) k should lie between 0 and 4095 inclusive and k_4 should be odd. The default is $k_i = 5 - i$.
`MATRIX`		Defines built-in tridiagonal matrices to be used in the tests. The next two lines must set integers $mtype_1$ $mtype_2$ $mtype_3$ ... $msize_1$ $msize_2$ $msize_3$ ... where *mtype* (integer) is a list of built-in tridiagonal matrices (see Table IV), and *msize* (integer) is a list of dimensions. *msize* can also be defined with `NMIN[:NINC]:NMAX`, where `NMIN` (> 0) is the minimum dimension, `NMAX` ($\geq$`NMIN`) is the maximum dimension, and `NINC` (> 0) is the increment.
`MATRIXF`	*string*	Defines a file containing a tridiagonal matrix, where *string* is a file name. This file should contain n 1 d_1 e_1 2 d_2 e_2 $\vdots$ n d_n 0.0 which will then be used to generate a tridiagonal matrix with diagonal entries set to d_i and offdiagonals set to e_i.
`NRILIU`	k	Defines the number of k random indices of the smallest and largest eigenvalues to be computed. These indices are used only in the tests where RANGE='I'.
`NRVLVU`	k	Defines the number of k random lower and upper bounds of intervals to be searched for eigenvalues. These i intervals are used only in the tests where RANGE='V'.
`END`		End of data (subsequent lines are ignored).

1.2 Examples of available test matrices

Tables IV and V list some of the test matrices available as part of `stetester`.

Table IV. Built-in matrices with distinguishing performance-relevant features.

type	description
0	zero matrix
1	identity matrix
2	(1,2,1) tridiagonal matrix
3	Wilkinson-type tridiagonal matrix
4	Clement-type tridiagonal matrix
5	Legendre-type tridiagonal matrix
6	Laguerre-type tridiagonal matrix
7	Hermite-type tridiagonal matrix

Table V. LAPACK-style test matrices with a given eigenvalue distribution. For distributions 1-5, the parameter k can be chosen as ulp^{-1} like in the LAPACK tester but other choices are also possible, see the options for parameter `ECOND` in Table I. For distribution 6, see parameter `EDIST` in Table I.

type	description
1	$\lambda_1 = 1,\ \lambda_i = \frac{1}{k},\ i = 2, 3, \dots n$
2	$\lambda_i = 1,\ i = 1, 2, \dots n-1,\ \lambda_n = \frac{1}{k}$
3	$\lambda_i = k^{-(\frac{i-1}{n-1})},\ i = 1, 2, \dots n$
4	$\lambda_i = 1 - (\frac{i-1}{n-1})(1 - \frac{1}{k}),\ i = 1, 2, \dots n$
5	n random numbers in the range $(\frac{1}{k}, 1)$, their logarithms are uniformly distributed
6	n random numbers from a specified distribution
7	$\lambda_i = ulp \times i,\ i = 1, 2, \dots n-1,\ \lambda_n = 1$
8	$\lambda_1 = ulp,\ \lambda_i = 1 + \sqrt{ulp} \times i,\ i = 2, 3, \dots n-1, \lambda_n = 2$
9	$\lambda_1 = 1,\ \lambda_i = \lambda_{i-1} + 100 \times ulp, i = 2, \dots n$

1.3 A sample input file

Table VI contains a sample input file. After the test matrices have been specified, the algorithms to be tested ('ALL') and the output format are selected.

1.4 A sample output file in Matlab format

Table VII contains a sample output file in Matlab format of a tridiagonal matrix with diagonal D and offdiagonal E. Printed are the computed eigenpairs W, Z from two different computations with the same matrix, first using QR and then MRRR.

Table VI. A sample input file for stetester.

```
%-------------------------------------------------------------------
% This is a simple input file for STETESTER.
%-------------------------------------------------------------------
%
EIGVAL               % Sets built-in eigenvalue distributions
        3            % Distribution 3, EIG(i)=COND**(-(i-1)/(N-1))
       10  15        % Dimensions of the matrices to be generated
%
MATRIX               % Sets built-in matrices
        2   3        % Matrix type 2 and 3
       20:2:25       % Dimension of the matrices to be generated
%
GLUED                % Sets glued matrices
     1     2     1   % If 1, set eigenvalues; if 2, set matrix
     1     2     3   % Eigenvalue distribution or matrix type
    10    11    12   % Dimensions
 0.001 0.002         % Glue factors
%
EIGVALF DATA/T_0010.eig % Eigenvalues read from file 'T10.eig'
%
MATRIXF DATA/T_0010.dat % Matrix read from file 'T10.dat'
%
% Tests to be performed. Note that 'ALL' is equivalent to
%
%   "STEQRV"  (calls STEQR with COMPZ='V'),
%   "STEVXA"  (calls STEVX with RANGE='A'),
%   "STEVXI"  (calls STEVX with RANGE='I'),
%   "STEVXV"  (calls STEVX with RANGE='V'),
%   "STEDCI"  (calls STEDC with COMPZ='I'),
%   "STEGRA"  (calls STEGR with RANGE='A'),
%   "STEGRI"  (calls STEGR with RANGE='I'),
%   "STEGRV"  (calls STEGR with RANGE='V'),
%
% Also note that no interval was specified (by means of EIGVI,
% EIGVV, NRILIU or NRVLVU) so in spite of 'ALL' some tests
% will be skipped.
%
CALLST  ALL
%
% Halfbandwidth of the symmetric matrix to be generated and then
% tridiagonalized. This can save time for big matrices.
%
HBANDA  100
%
% Dump results in different formats (including Matlab)
%
DUMP    LOG  T  W  Z  T.M  W.M  Z.M
%
END
```

Table VII. A sample output file generated by `stetester`. The data is printed in Matlab format and stored with a name whose trailing part identifies the test that has been executed

```
% Case:    1 ################################################################
N =    5;
N_001 = N;
D = zeros(N,1); E = zeros(N,1);
D(    1)= 6.364984420732012E-002; E(    1)=-2.438638589637976E-001;
D(    2)= 9.364644735979822E-001; E(    2)=-4.811682261688812E-003;
D(    3)= 1.093556433149412E-002; E(    3)= 3.729709837873370E-005;
D(    4)= 1.218230276935389E-004; E(    4)= 5.319506124657539E-006;
D(    5)= 2.722043635334067E-007; E(    5)= 0.000000000000000E+000;
D_001 = D; E_001 = E; clear D E;
% QR algorithm STEQR(COMPZ=I) ==============================================
M =    5;
W = zeros(M,1);
W(    1)= 1.490116120469489E-008; W(    2)= 1.348699152530776E-006;
W(    3)= 1.220703124999231E-004; W(    4)= 1.104854345603982E-002;
W(    5)= 1.000000000000000E+000;
W_001_1 = W; clear W;
Z = zeros(N,M);
Z(    1,    1)= 1.307984066973555E-001; Z(    2,    1)= 3.413911473056844E-002;
Z(    3,    1)= 1.518458390297948E-002; Z(    4,    1)=-4.786418109812126E-002;
Z(    5,    1)= 9.895477483327149E-001; Z(    1,    2)= 9.521524057428332E-001;
Z(    2,    2)= 2.485118884672883E-001; Z(    3,    2)= 1.094543724694096E-001;
Z(    4,    2)=-2.781623695716281E-002; Z(    5,    2)=-1.374541190267801E-001;
Z(    1,    3)= 3.316346817614305E-002; Z(    2,    3)= 8.639251903968444E-003;
Z(    3,    3)= 4.003930825830011E-004; Z(    4,    3)= 9.984606995074429E-001;
Z(    5,    3)= 4.360755587691128E-002; Z(    1,    4)= 1.080661771201757E-001;
Z(    2,    4)= 2.330981518906979E-002; Z(    3,    4)=-9.938646010435163E-001;
Z(    4,    4)=-3.392442840664115E-003; Z(    5,    4)=-1.633388614144078E-006;
Z(    1,    5)=-2.520306703255168E-001; Z(    2,    5)= 9.677077957618335E-001;
Z(    3,    5)=-4.707784724629880E-003; Z(    4,    5)=-1.756081031362012E-007;
Z(    5,    5)=-9.341486344518498E-013;
Z_001_1 = Z; clear Z;
M_001_1 = M; clear M;
% MRRR algorithm STEGR(RANGE=A) ============================================
M =    5;
W = zeros(M,1);
W(    1)= 1.490116120299405E-008; W(    2)= 1.348699152440647E-006;
W(    3)= 1.220703124999230E-004; W(    4)= 1.104854345603979E-002;
W(    5)= 9.999999999999978E-001;
W_001_6 = W; clear W;
Z = zeros(N,M);
Z(    1,    1)= 1.307984067061942E-001; Z(    2,    1)= 3.413911473287538E-002;
Z(    3,    1)= 1.518458390399541E-002; Z(    4,    1)=-4.786418109837592E-002;
Z(    5,    1)= 9.895477483314390E-001; Z(    1,    2)= 9.521524057416197E-001;
Z(    2,    2)= 2.485118884669718E-001; Z(    3,    2)= 1.094543724692678E-001;
Z(    4,    2)=-2.781623695669255E-002; Z(    5,    2)=-1.374541190359646E-001;
Z(    1,    3)= 3.316346817611779E-002; Z(    2,    3)= 8.639251903961875E-003;
Z(    3,    3)= 4.003930825800720E-004; Z(    4,    3)= 9.984606995074433E-001;
Z(    5,    3)= 4.360755587691132E-002; Z(    1,    4)=-1.080661771201750E-001;
Z(    2,    4)=-2.330981518906962E-002; Z(    3,    4)= 9.938646010435160E-001;
Z(    4,    4)= 3.392442840664119E-003; Z(    5,    4)= 1.633388614144083E-006;
Z(    1,    5)=-2.520306703255170E-001; Z(    2,    5)= 9.677077957618334E-001;
Z(    3,    5)=-4.707784724629883E-003; Z(    4,    5)=-1.756081031362015E-007;
Z(    5,    5)=-9.341486344518528E-013;
Z_001_6 = Z; clear Z;
M_001_6 = M; clear M;
clear N;
```