Package: geigen (via r-universe)

October 28, 2026

Version 2.4

Title Calculate Generalized Eigenvalues, the Generalized Schur Decomposition and the Generalized Singular Value Decomposition of a Matrix Pair with Lapack

Date 2026-07-20

Depends R (>= 3.5.0)

Author Berend Hasselman [aut], Jonathan M. Lees [cre], Lapack authors [aut, cph]

Maintainer Jonathan M. Lees <jonathan.lees@unc.edu>

Description Functions to compute generalized eigenvalues and eigenvectors, the generalized Schur decomposition and the generalized Singular Value Decomposition of a matrix pair, using Lapack routines.

License GPL (>= 2)

Encoding UTF-8

Copyright see inst/COPYRIGHTS for the files in src/.

NeedsCompilation yes

Repository https://cran.r-universe.dev

Date/Publication 2026-07-22 06:46:07 UTC

RemoteUrl https://github.com/cran/geigen

RemoteRef HEAD

RemoteSha a610855e3ac425c079c3ac4eba382cdea279ed76

Contents

geigen-package 2
geigen 2
gevalues 4
gqz 5
gsvd 8
gsvd.Auxiliaries 10

Index 12

`geigen-package` *Calculate Generalized Eigenvalues, the Generalized Schur Decomposition and the Generalized Singular Value Decomposition of a Matrix Pair with Lapack*

Description

Functions to compute generalized eigenvalues and eigenvectors, the generalized Schur decomposition and the generalized Singular Value Decomposition of a matrix pair, using Lapack routines.

Details

The **geigen** package provides functions to compute the generalized eigenvalues of a pair of matrices and optionally provides the generalized eigenvectors. Both numeric and complex matrices are allowed. The package also provides a function for computing the generalized Schur decomposition of a pair of matrices, either numeric or complex. Finally the package provides a function for computing the generalized singular value decomposition for a pair of matrices, either numeric or complex. The package uses routines provided by the Lapack linear algebra package.

`geigen` *Generalized Eigenvalues*

Description

Computes generalized eigenvalues and eigenvectors of a pair of matrices.

Usage

```
geigen(A, B, symmetric, only.values=FALSE)
```

Arguments

`A`	left hand side matrix.
`B`	right hand side matrix.
`symmetric`	if `TRUE`, both matrices are assumed to be symmetric (or Hermitian if complex) and only their lower triangle (diagonal included) is used. If `symmetric` is not specified, the matrices are inspected for symmetry.
`only.values`	if `TRUE` only eigenvalues are computed otherwise both eigenvalues and eigenvctors are returned.

Details

If the argument `symmetric` is missing, the function will try to determine if the matrices are symmetric with the function `isSymmetric` from the **base** package. It is faster to specify the argument.

Both matrices must be square. This function provides the solution to the generalized eigenvalue problem defined by

$$Ax = \lambda Bx$$

If either one of the matrices is complex the other matrix is coerced to be complex.

If the matrices are symmetric then the matrix `B` must be positive definite; if it is not an error message will be issued. If the matrix `B` is known to be symmetric but not positive definite then the argument `symmetric` should be set to `FALSE` explicitly.

If the matrix `B` is not positive definite when it should be an error message of the form

`Leading minor of order ... of B is not positive definite`

will be issued. In that case set the argument `symmetric` to `FALSE` if not set and try again.

For general matrices the generalized eigenvalues λ are calculated as the ratio α/β where β may be zero or very small leading to non finite or very large values for the eigenvalues. Therefore the values for α and β are also included in the return value of the function. When both matrices are complex (or coerced to be so) the generalized eigenvalues, α and β are complex. When both matrices are numeric α may be numeric or complex and β is numeric.

When both matrices are symmetric (or Hermitian) the generalized eigenvalues are numeric and no components α and β are available.

Value

A list containing components

`values`	a vector containing the n generalized eigenvalues.
`vectors`	an $n \times n$ matrix containing the generalized eigenvectors or NULL if `only.values` is `TRUE`.
`alpha`	the numerator of the generalized eigenvalues and may be NULL if not applicable.
`beta`	the denominator of the generalized eigenvalues and may be NULL if not applicable.

Source

`geigen` uses the LAPACK routines `DGGEV`, `DSYGV`, `ZHEGV` and `ZGGEV`. LAPACK is from https://www.netlib.org/lapack/. The complex routines used by the package come from LAPACK 3.8.0.

References

Anderson. E. and ten others (1999) *LAPACK Users' Guide*. Third Edition. SIAM.
Available on-line at https://www.netlib.org/lapack/lug/lapack_lug.html. See the section *Generalized Eigenvalue and Singular Value Problems* (https://www.netlib.org/lapack/lug/node33.html).

See Also

eigen

Examples

```
A <- matrix(c(14, 10, 12,
              10, 12, 13,
              12, 13, 14), nrow=3, byrow=TRUE)

B <- matrix(c(48, 17, 26,
              17, 33, 32,
              26, 32, 34), nrow=3, byrow=TRUE)

z1 <- geigen(A, B, symmetric=FALSE, only.values=TRUE)
z2 <- geigen(A, B, symmetric=FALSE, only.values=FALSE )
z2

# geigen(A, B)
z1 <- geigen(A, B, only.values=TRUE)
z2 <- geigen(A, B, only.values=FALSE)
z1;z2

A.c <- A + 1i
B.c <- B + 1i

A[upper.tri(A)] <- A[upper.tri(A)] + 1i
A[lower.tri(A)] <- Conj(t(A[upper.tri(A)]))

B[upper.tri(B)] <- B[upper.tri(B)] + 1i
B[lower.tri(B)] <- Conj(t(B[upper.tri(B)]))

isSymmetric(A)
isSymmetric(B)

z1 <- geigen(A, B, only.values=TRUE)
z2 <- geigen(A, B, only.values=FALSE)
z1;z2
```

---

gevalues *Calculate Generalized Eigenvalues from a Generalized Schur decomposition*

Description

Computes the generalized eigenvalues from an object constructed with gqz.

Usage

```
gevalues(x)
```

Arguments

x	an object created with gqz.

Details

The function calculates the generalized eigenvalues from elements of the object returned by the function gqz. The generalized eigenvalues are computed from a ratio where the denominator (the β component of the argument) may be zero. The function attempts to guard against nonsensical complex NaN values when dividing by zero which may happen on some systems.

Value

A vector containing the generalized eigenvalues. The vector is numeric if the imaginary parts of the eigenvalues are all zero and complex otherwise.

See Also

geigen, gqz

Examples

```
# Real matrices
# example from NAG: https://www.nag.com/lapack-ex/node116.html
# Find the generalized Schur decomposition with the real eigenvalues ordered to come first

A <- matrix(c(  3.9, 12.5,-34.5,-0.5,
                4.3, 21.5,-47.5, 7.5,
                4.3, 21.5,-43.5, 3.5,
                4.4, 26.0,-46.0, 6.0), nrow=4, byrow=TRUE)

B <- matrix(c( 1.0, 2.0, -3.0, 1.0,
               1.0, 3.0, -5.0, 4.0,
               1.0, 3.0, -4.0, 3.0,
               1.0, 3.0, -4.0, 4.0), nrow=4, byrow=TRUE)

z <- gqz(A, B,"R")
z
# compute the generalized eigenvalues
ger <- gevalues(z)
ger
```

---

gqz *Generalized Schur decomposition*

Description

Computes the generalized eigenvalues and Schur form of a pair of matrices.

Usage

```
gqz(A, B, sort=c("N","-","+","S","B","R"))
```

Arguments

A	left hand side matrix.
B	right hand side matrix.
sort	how to sort the generalized eigenvalues. See 'Details'.

Details

Both matrices must be square. This function provides the solution to the generalized eigenvalue problem defined by

$$Ax = \lambda Bx$$

If either one of the matrices is complex the other matrix is coerced to be complex.

The `sort` argument specifies how to order the eigenvalues on the diagonal of the generalized Schur form, where it is noted that non-finite eigenvalues never satisfy any ordering condition (even in the case of a complex infinity). Eigenvalues that are placed in the leading block of the Schur form satisfy

`N` unordered.

`-` negative real part.

`+` positive real part.

`S` absolute value < 1.

`B` absolute value > 1.

`R` imaginary part identical to 0 with a tolerance of 100*machine_precision as determined by Lapack.

Value

The generalized Schur form for *numeric* matrices is

$$(A, B) = (QSZ^T, QTZ^T)$$

The matrices Q and Z are orthogonal. The matrix S is quasi-upper triangular and the matrix T is upper triangular. The return value is a list containing the following components

`S` generalized Schur form of A.

`T` generalized Schur form of B.

`sdim` the number of eigenvalues (after sorting) for which the sorting condition is true.

`alphar` numerator of the real parts of the eigenvalues (numeric).

`alphai` numerator of the imaginary parts of the eigenvalues (numeric).

`beta` denominator of the expression for the eigenvalues (numeric).

`Q` matrix of left Schur vectors (matrix Q).

`Z` matrix of right Schur vectors (matrix Z).

The generalized Schur form for *complex* matrices is

$$(A, B) = (QSZ^H, QTZ^H)$$

The matrices Q and Z are unitary and the matrices S and T are upper triangular. The return value is a list containing the following components

`S` generalized Schur form of A.

`T` generalized Schur form of B.

`sdim` the number of eigenvalues. (after sorting) for which the sorting condition is true.

`alpha` numerator of the eigenvalues (complex).

`beta` denominator of the eigenvalues (complex).

`Q` matrix of left Schur vectors (matrix Q).

`Z` matrix of right Schur vectors (matrix Z).

The generalized eigenvalues can be computed by calling function `gevalues`.

Source

`gqz` uses the LAPACK routines `DGGES` and `ZGGES`. LAPACK is from https://www.netlib.org/lapack/. The complex routines used by the package come from LAPACK 3.8.0.

References

Anderson. E. and ten others (1999) *LAPACK Users' Guide*. Third Edition. SIAM.
Available on-line at https://www.netlib.org/lapack/lug/lapack_lug.html. See the section *Eigenvalues, Eigenvectors and Generalized Schur Decomposition* (https://www.netlib.org/lapack/lug/node56.html).

See Also

`geigen`, `gevalues`

Examples

```
# Real matrices
# example from NAG: https://www.nag.com/lapack-ex/node116.html
# Find the generalized Schur decomposition with the real eigenvalues ordered to come first

A <- matrix(c(  3.9, 12.5,-34.5,-0.5,
                4.3, 21.5,-47.5, 7.5,
                4.3, 21.5,-43.5, 3.5,
                4.4, 26.0,-46.0, 6.0), nrow=4, byrow=TRUE)

B <- matrix(c( 1.0, 2.0, -3.0, 1.0,
               1.0, 3.0, -5.0, 4.0,
               1.0, 3.0, -4.0, 3.0,
               1.0, 3.0, -4.0, 4.0), nrow=4, byrow=TRUE)

z <- gqz(A, B,"R")
```

```
z

# complexify
A <- A+0i
B <- B+0i
z <- gqz(A, B,"R")
z
```

---

gsvd *Generalized Singular Value Decomposition*

Description

Computes the generalized singular value decomposition of a pair of matrices.

Usage

```
gsvd(A,B)
```

Arguments

A	a matrix with m rows and n columns.
B	a matrix with p rows and n columns.

Details

The matrix A is a m-by-n matrix and the matrix B is a p-by-n matrix. This function decomposes both matrices; if either one is complex than the other matrix is coerced to be complex.

The Generalized Singular Value Decomposition of *numeric* matrices A and B is given as

$$A = U D_1 [0\, R] Q^T$$

and

$$B = V D_2 [0\, R] Q^T$$

where

U an $m \times m$ orthogonal matrix.

V a $p \times p$ orthogonal matrix.

Q an $n \times n$ orthogonal matrix.

R an r-by-r upper triangular non singular matrix and the matrix $[0\, R]$ is an r-by-n matrix. The quantity r is the rank of the matrix $\begin{pmatrix} A \\ B \end{pmatrix}$ with $r \leq n$.

D_1,D_2 are quasi diagonal matrices and nonnegative and satisfy $D_1^T D_1 + D_2^T D_2 = I$. D_1 is an m-by-r matrix and D_2 is a p-by-r matrix.

The Generalized Singular Value Decomposition of *complex* matrices A and B is given as

$$A = U D_1 [0\, R] Q^H$$

and

$$B = V D_2 [0\, R] Q^H$$

where

U an $m \times m$ unitary matrix.

V a $p \times p$ unitary matrix.

Q an $n \times n$ unitary matrix.

R an r-by-r upper triangular non singular matrix and the matrix $[0\, R]$ is an r-by-n matrix. The quantity r is the rank of the matrix $\begin{pmatrix} A \\ B \end{pmatrix}$ with $r \leq n$.

D_1,D_2 are quasi diagonal matrices and nonnegative and satisfy $D_1^T D_1 + D_2^T D_2 = I$. D_1 is an m-by-r matrix and D_2 is a p-by-r matrix.

For details on this decomposition and the structure of the matrices D_1 and D_2 see https://www.netlib.org/lapack/lug/node36.html.

Value

The return value is a list containing the following components

`A` the upper triangular matrix or a part of R.

`B` lower part of the triangular matrix R if $k + l > m$ (see below).

`m` number of rows of A.

`k` $r-l$. The number of rows of the matrix R is $k+l$. The first k generalized singular values are infinite.

`l` effective rank of the input matrix B. The number of finite generalized singular values after the first k infinite ones.

`alpha` a numeric vector with length n containing the numerators of the generalized singular values in the first $(k+l)$ entries.

`beta` a numeric vector with length n containing the denominators of the generalized singular value in the first $(k+l)$ entries.

`U` the matrix U.

`V` the matrix V.

`Q` the matrix Q.

For a detailed description of these items see https://www.netlib.org/lapack/lug/node36.html. Auxiliary functions are provided for extraction and manipulation of the various items.

Source

`gsvd` uses the LAPACK routines `DGGSVD3` and `ZGGSVD3` from Lapack 3.8.0. LAPACK is from https://www.netlib.org/lapack/. The decomposition is fully explained in https://www.netlib.org/lapack/lug/node36.html.

References

Anderson. E. and ten others (1999) *LAPACK Users' Guide*. Third Edition. SIAM.
Available on-line at https://www.netlib.org/lapack/lug/lapack_lug.html. See the section *Generalized Eigenvalue and Singular Value Problems* (https://www.netlib.org/lapack/lug/node33.html) and the section *Generalized Singular Value Decomposition (GSVD)* (https://www.netlib.org/lapack/lug/node36.html).

See Also

gsvd.aux

Examples

```
A <- matrix(c(1,2,3,3,2,1,4,5,6,7,8,8), nrow=2, byrow=TRUE)
B <- matrix(1:18,byrow=TRUE, ncol=6)
A
B

z <- gsvd(A,B)
z
```

---

gsvd.Auxiliaries *Extract the R, D1, D2 matrices from a gsvd object*

Description

Returns a component of the object as a matrix

Usage

```
gsvd.R(z)
gsvd.oR(z)
gsvd.D1(z)
gsvd.D2(z)
```

Arguments

z	an object created with gsvd

Value

gsvd.R returns the R matrix implied by the GSVD.

gsvd.oR returns the matrix $[0\ R]$ implied by the GSVD.

gsvd.D1 returns the matrix D1 implied by the GSVD.

gsvd.D2 returns the matrix D2 implied by the GSVD.

See Also

gsvd

Examples

```
A <- matrix(c(1,2,3,3,2,1,4,5,6,7,8,8), nrow=2, byrow=TRUE)
B <- matrix(1:18,byrow=TRUE, ncol=6)
A
B

z <- gsvd(A,B)
z

R <- gsvd.R(z)
oR <- gsvd.oR(z)
D1 <- gsvd.D1(z); D2 <- gsvd.D2(z)
R;oR
D1;D2
```

Index

∗ **algebra**
  geigen, 2
  gevalues, 4
  gqz, 5
  gsvd, 8
  gsvd.Auxiliaries, 10
∗ **array**
  geigen, 2
  gevalues, 4
  gqz, 5
  gsvd, 8
  gsvd.Auxiliaries, 10
∗ **package**
  geigen-package, 2

geigen, 2
geigen-package, 2
geigen.Intro *(*geigen-package*)*, 2
gevalues, 4
gqz, 5
gsvd, 8, *11*
gsvd.aux, *10*
gsvd.aux *(*gsvd.Auxiliaries*)*, 10
gsvd.Auxiliaries, 10
gsvd.D1 *(*gsvd.Auxiliaries*)*, 10
gsvd.D2 *(*gsvd.Auxiliaries*)*, 10
gsvd.oR *(*gsvd.Auxiliaries*)*, 10
gsvd.R *(*gsvd.Auxiliaries*)*, 10