Package 'EigenR'

November 25, 2020

Type Package

Title Complex Matrix Algebra with 'Eigen'

Version 1.0.0

Author Stéphane Laurent

Maintainer Stéphane Laurent <laurent_step@outlook.fr>

Description Matrix algebra using the 'Eigen' C++ library: determinant, rank, inversion, kernel and image, QR decomposition, Cholesky decomposition, linear least-squares problems. Complex matrices are supported.

License GPL-3

Imports Rcpp (>= 1.0.5)

LinkingTo Rcpp, RcppEigen

RoxygenNote 7.1.1

Encoding UTF-8

URL https://github.com/stla/EigenR

BugReports https://github.com/stla/EigenR/issues

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-11-25 10:20:05 UTC

R topics documented:

Eigen_chol 2
Eigen_det 3
Eigen_inverse 3
Eigen_kernel 4
Eigen_lsSolve 4
Eigen_QR 5
Eigen_range 6
Eigen_rank 6
Eigen_UtDU 7
SparseMatrix 7

Index **9**

`Eigen_chol` *Cholesky decomposition of a matrix*

Description

Cholesky decomposition of a symmetric or Hermitian matrix.

Usage

```
Eigen_chol(M)
```

Arguments

`M`	a square symmetric/Hermitian positive-definite matrix or `SparseMatrix`, real/complex

Details

Symmetry is not checked; only the lower triangular part of `M` is used.

Value

The upper triangular factor of the Cholesky decomposition of `M`.

Examples

```
M <- rbind(c(5,1), c(1,3))
U <- Eigen_chol(M)
t(U) %*% U # this is `M`
# a Hermitian example:
A <- rbind(c(1,1i), c(1i,2))
( M <- A %*% t(Conj(A)) )
try(chol(M)) # fails
U <- Eigen_chol(M)
t(Conj(U)) %*% U # this is `M`
# a sparse example
M <- asSparseMatrix(diag(1:5))
Eigen_chol(M)
```

Eigen_det *Determinant of a matrix*

Description

Determinant of a real or complex matrix.

Usage

```
Eigen_det(M)
```

Arguments

M	a square matrix or SparseMatrix, real or complex

Value

The determinant of M.

Examples

```
set.seed(666)
M <- matrix(rpois(25, 1), 5L, 5L)
Eigen_det(M)
# determinants of complex matrices are supported:
Eigen_det(M + 1i * M)
# as well as determinants of sparse matrices:
Eigen_det(asSparseMatrix(M))
Eigen_det(asSparseMatrix(M + 1i * M))
```

Eigen_inverse *Inverse of a matrix*

Description

Inverse of a real or complex matrix.

Usage

```
Eigen_inverse(M)
```

Arguments

M	an invertible square matrix, real or complex

Value

The inverse matrix of M.

Eigen_kernel *Kernel of a matrix*

Description

Kernel (null-space) of a real or complex matrix.

Usage

```
Eigen_kernel(M, method = "COD")
```

Arguments

M	a matrix, real or complex
method	one of "COD" or "LU"; the faster method depends on the size of the matrix

Value

A basis of the kernel of M. With method = "COD", the basis is orthonormal, while it is not with method = "LU".

Examples

```
set.seed(666)
M <- matrix(rgamma(30L, 12, 1), 10L, 3L)
M <- cbind(M, M[,1]+M[,2], M[,2]+2*M[,3])
# basis of the kernel of `M`:
Eigen_kernel(M, method = "LU")
# orthonormal basis of the kernel of `M`:
Eigen_kernel(M, method = "COD")
```

Eigen_lsSolve *Linear least-squares problems*

Description

Solves a linear least-squares problem.

Usage

```
Eigen_lsSolve(A, b)
```

Arguments

A	a n*p matrix, real or complex
b	a vector of length n or a matrix with n rows, real or complex

Value

The solution `X` of the least-squares problem `AX ~= b` (similar to `lm.fit(A,b)$coefficients`). This is a matrix if `b` is a matrix, or a vector if `b` is a vector.

Examples

```
set.seed(129)
n <- 7; p <- 2
A <- matrix(rnorm(n * p), n, p)
b <- rnorm(n)
lsfit <- Eigen_lsSolve(A, b)
b - A %*% lsfit # residuals
```

`Eigen_QR` *QR decomposition of a matrix*

Description

QR decomposition of a real or complex matrix.

Usage

```
Eigen_QR(M)
```

Arguments

`M`	a matrix, real or complex

Value

A list with the `Q` matrix and the `R` matrix.

Examples

```
M <- cbind(c(1,2,3), c(4,5,6))
x <- Eigen_QR(M)
x$Q %*% x$R
```

Eigen_range *Range of a matrix*

Description

Range (column-space, image, span) of a real or complex matrix.

Usage

```
Eigen_range(M, method = "QR")
```

Arguments

M	a matrix, real or complex
method	one of "LU", "QR", or "COD"; the "LU" method is faster

Value

A basis of the range of M. With method = "LU", the basis is not orthonormal, while it is with method = "QR" and method = "COD".

Eigen_rank *Rank of a matrix*

Description

Rank of a real or complex matrix.

Usage

```
Eigen_rank(M)
```

Arguments

M	a matrix, real or complex

Value

The rank of M.

Eigen_UtDU *'UtDU' decomposition of a matrix*

Description

Cholesky-'UtDU' decomposition of a symmetric or Hermitian matrix.

Usage

```
Eigen_UtDU(M)
```

Arguments

M	a square symmetric/Hermitian positive or negative semidefinite matrix, real/complex

Details

Symmetry is not checked; only the lower triangular part of M is used.

Value

The Cholesky-'UtDU' decomposition of M in a list (see example).

Examples

```
x <- matrix(c(1:5, (1:5)^2), 5, 2)
x <- cbind(x, x[, 1] + 3*x[, 2])
M <- crossprod(x)
UtDU <- Eigen_UtDU(M)
U <- UtDU$U
D <- UtDU$D
perm <- UtDU$perm
UP <- U[, perm]
t(UP) %*% diag(D) %*% UP # this is `M`
```

SparseMatrix *Sparse matrix*

Description

Constructs a sparse matrix, real or complex.

Usage

```
SparseMatrix(i, j, Mij, nrows, ncols)

## S3 method for class 'SparseMatrix'
print(x, ...)

asSparseMatrix(M)
```

Arguments

`i, j`	indices of the non-zero coefficients
`Mij`	values of the non-zero coefficients; must be a vector of the same length as `i` and `j` or a single number which will be recycled
`nrows, ncols`	dimensions of the matrix
`x`	a `SparseMatrix` object
`...`	ignored
`M`	a matrix, real or complex

Value

A list with the class `SparseMatrix`.

Examples

```
set.seed(666)
( M <- matrix(rpois(50L, 1), 10L, 5L) )
asSparseMatrix(M)
```

Index

asSparseMatrix *(*SparseMatrix*)*, 7

Eigen_chol, 2
Eigen_det, 3
Eigen_inverse, 3
Eigen_kernel, 4
Eigen_lsSolve, 4
Eigen_QR, 5
Eigen_range, 6
Eigen_rank, 6
Eigen_UtDU, 7

print.SparseMatrix *(*SparseMatrix*)*, 7

SparseMatrix, *2*, *3*, 7