

# Package ‘blockmatrix’

May 7, 2026

**Maintainer** Emanuele Cordano <emanuele.cordano@gmail.com>

**License** GPL (>= 2)

**Title** blockmatrix: Tools to solve algebraic systems with partitioned matrices

**Type** Package

**Author** Emanuele Cordano

**Description** Some elementary matrix algebra tools are implemented to manage block matrices or partitioned matrix, i.e. ``matrix of matrices" ([http://en.wikipedia.org/wiki/Block\\_matrix](http://en.wikipedia.org/wiki/Block_matrix)). The block matrix is here defined as a new S3 object. In this package, some methods for ``matrix" object are rewritten for ``blockmatrix" object. New methods are implemented. This package was created to solve equation systems with block matrices for the analysis of environmental vector time series .  
Bugs/comments/questions/collaboration of any kind are warmly welcomed.

**Version** 1.0

**Repository** CRAN

**Date** 2014-01-20

**Depends** R (>= 2.13)

**URL** <http://cri.gmpf.eu/Research/Sustainable-Agro-Ecosystems-and-Bioresources/Dynamics-in-the-agro-ecosystems/people/Emanuele-Cordano>

**Collate** 'as.blockmatrix.matrix.R' 'as.matrix.blockmatrix.R'  
'blockmatrix.R' 'dim.blockmatrix.R' 'is.zero.blockmatrix.R'  
'length.blockmatrix.R' 'matmult.blockmatrix.R'  
'methods.blockmatrix.R' 'names.blockmatrix.R'  
'ncol.blockmatrix.R' 'ncol\_elements.blockmatrix.R'  
'nrow.blockmatrix.R' 'nrow\_elements.blockmatrix.R'  
'solve.blockmatrix.R' 't.blockmatrix.R' 'value.blockmatrix.R'  
'value.replacement.blockmatrix.R' 'zbracket[.blockmatrix.R'  
'zbracket[.replacement.blockmatrix.R'

**NeedsCompilation** no

**Date/Publication** 2014-01-19 18:10:34

## Contents

as.blockmatrix . . . . .	2
as.matrix.blockmatrix . . . . .	3
blockmatmult . . . . .	4
blockmatrix . . . . .	4
dim.blockmatrix . . . . .	6
is.zero.blockmatrix . . . . .	6
length.blockmatrix . . . . .	7
Math.blockmatrix . . . . .	7
names.blockmatrix . . . . .	8
ncol.blockmatrix . . . . .	8
ncol_elements . . . . .	9
nrow.blockmatrix . . . . .	9
nrow_elements . . . . .	10
solve.blockmatrix . . . . .	11
t.blockmatrix . . . . .	11
value . . . . .	12
value<- . . . . .	13
[.blockmatrix . . . . .	13
[<-.blockmatrix . . . . .	14

<b>Index</b>	<b>16</b>
--------------	-----------

---

as.blockmatrix	as.blockmatrix S3 method for blockmatrix, matrix and NULL object
----------------	--

---

### Description

as.blockmatrix S3 method for blockmatrix, matrix and NULL object

### Usage

```
as.blockmatrix(M = NULL, ...)

## Default S3 method:
as.blockmatrix(M, adjust_zero = TRUE,
  zero_element = "0", ...)

## S3 method for class 'blockmatrix'
as.blockmatrix(M,
  adjust_zero = TRUE, add_zero_matrix = FALSE,
  zero_element = "0", ...)

## S3 method for class 'matrix'
as.blockmatrix(M, nrow = 2, ncol = 2,
  nrow = NULL, ncol = NULL, adjust_zero = TRUE,
  zero_element = "0", ...)
```

**Arguments**

M	a matrix or blockmatrix object
nrowe	number of rows for each block (element of the blockmatrix)
ncole	number of columns for each block (element of the blockmatrix)
nrow	number of rows for block-matrix
ncol	number of columns of blockmatrix
adjust_zero	logical value. If TRUE (Default) it replaces the zero matrices with zero_element.
add_zero_matrix	logical value. If TRUE it adds a zero-element element matrix as an object called zero_element in the blockmatrix
zero_element	see <a href="#">ncol_elements</a> or <a href="#">nrow_elements</a>
...	further arguments

**Author(s)**

Emanuele Cordano

---

as.matrix.blockmatrix *as.matrix S3 method for blockmatrix object*


---

**Description**

as.matrix S3 method for blockmatrix object

**Usage**

```
## S3 method for class 'blockmatrix'
as.matrix(x, zero_element = "0",
  ...)
```

**Arguments**

x	a blockmatrix object
zero_element	(see <a href="#">ncol_elements</a> or <a href="#">nrow_elements</a> )
...	further arguments (see <a href="#">ncol_elements</a> or <a href="#">nrow_elements</a> )

**Author(s)**

Emanuele Cordano

---

blockmatmult	<i>blockmatmult implements the implents betwwen two blockmatrix ( see <a href="#">matmult</a> for matrix objects)</i>
--------------	---

---

**Description**

blockmatmult implements the implents betwwen two blockmatrix ( see [matmult](#) for matrix objects)

**Usage**

```
blockmatmult(x, y, ...)
```

**Arguments**

x, y	blockmatrix objects
...	further arguments

**Value**

The inner product between x and y as a blockmatrix object

**Author(s)**

Emanuele Cordano

---

blockmatrix	<i>This function builds a blockmatrix</i>
-------------	---

---

**Description**

This function builds a blockmatrix

**Usage**

```
blockmatrix(dim, value = NULL, names = NULL, list = NULL,
  use.as.blockmatrix = TRUE, adjust_zero = TRUE,
  add_zero_matrix = FALSE, zero_element = "0", ...)
```

**Arguments**

<code>dim</code>	dimension of a block-matrix
<code>value</code>	matrix containing the indices (names) of blockmatrix element. If missing, it is NULL (Default). (sse <a href="#">value</a> )
<code>names</code>	character vector containing the names for each matrix-type element of the block-matrix
<code>list</code>	list containing the matrices to be inserted into the block-matrix. If NULL (Default) the matrix are taken from ...
<code>use.as.blockmatrix</code>	logical value. If TRUE (Default) the method <a href="#">as.blockmatrix</a> for blockmatrix object is applied to the output blockmatrix before being returned.
<code>adjust_zero, add_zero_matrix, zero_element</code>	arguments passed to <a href="#">as.blockmatrix</a>
<code>...</code>	elements of the block-matrix.

**Author(s)**

Emanuele Cordano

**See Also**

[as.blockmatrix](#)

**Examples**

```
rm(list=ls())
library(blockmatrix)

A <- array(rnorm(9,mean=1),c(3,3))
B <- 0 #array(rnorm(9,mean=2),c(3,3))
C <- 0
D <- array(rnorm(9,mean=4),c(3,3))
F <- array(rnorm(9,mean=10),c(3,3))

M <- blockmatrix(names=c("A","0","D","0"),A=A,D=D,dim=c(2,2))
E <- blockmatrix(names=c("0","F","D","0"),F=F,D=D,dim=c(2,2))

R <- M+E
S <- solve(R)
P <- blockmatmult(R,E)

l <- list(A=A,B=B,C=C,D=D,F=F)
mv <- array(c("A","B","C","D","F","F"),c(3,2))
BB <- blockmatrix(value=mv,list=l)
```

---

dim.blockmatrix      *dim S3 method for blockmatrix object*

---

**Description**

dim S3 method for blockmatrix object

**Usage**

```
## S3 method for class 'blockmatrix'
dim(x)
```

**Arguments**

x                    a blockmatrix object

**Author(s)**

Emanuele Cordano

---

is.zero.blockmatrix      *is.zero.bolockmatrix*

---

**Description**

is.zero.bolockmatrix

**Usage**

```
is.zero.blockmatrix(M, not.a.blockmatrix = FALSE)
```

**Arguments**

M                    a blockmatrix object  
not.a.blockmatrix      value to be returned in case M is not a a blockmatrix object

**Value**

logical value in case M is a zero blockmatrix

**Author(s)**

Emanuele Cordano

---

length.blockmatrix      length *S3 method for blockmatrix object*

---

**Description**

length *S3 method for blockmatrix object*

**Usage**

```
## S3 method for class 'blockmatrix'  
length(x)
```

**Arguments**

x                      a blockmatrix object

**Author(s)**

Emanuele Cordano

---

Math.blockmatrix      Math *and Ops group of S3 methods for blockmatrix object*

---

**Description**

Math and Ops group of *S3 methods for blockmatrix object*

**Usage**

```
## S3 method for class 'blockmatrix'  
Math(x, as.blockmatrix = TRUE,  
      ...)  
  
## S3 method for class 'blockmatrix'  
Ops(e1, e2)
```

**Arguments**

x, e1, e2              blockmatrix objects  
as.blockmatrix      logical value. If TRUE (Default), the output is a blockmatrix object  
...                    further arguments

**Author(s)**

Emanuele Cordano

names.blockmatrix      names *S3 method for blockmatrix object*

---

**Description**

names S3 method for blockmatrix object

**Usage**

```
## S3 method for class 'blockmatrix'  
names(x)
```

**Arguments**

x                      a blockmatrix object

**Author(s)**

Emanuele Cordano

---

ncol.blockmatrix      ncol *S3 method for blockmatrix object*

---

**Description**

ncol S3 method for blockmatrix object

**Usage**

```
## S3 method for class 'blockmatrix'  
ncol(M)
```

**Arguments**

M                      a blockmatrix object

**Value**

Number of columns of blockmatrix M

**Author(s)**

Emanuele Cordano

---

ncol_elements	<i>ncol_elements S3 method for blockmatrix object</i>
---------------	---

---

**Description**

ncol\_elements S3 method for blockmatrix object

**Usage**

```
ncol_elements(M, zero_element = "0", ...)

## Default S3 method:
ncol_elements(M, zero_element = "0",
              ...)

## S3 method for class 'blockmatrix'
ncol_elements(M,
              zero_element = "0", ...)
```

**Arguments**

M	a blockmatrix object
zero_element	character value indicating a zero matrix. Default is "0"
...	further arguments

**Value**

The number of columns of a matrix-type element of M. It is NA if the elements has different number of columns.

**Author(s)**

Emanuele Cordano

---

nrow.blockmatrix	<i>nrow S3 method for blockmatrix object</i>
------------------	--

---

**Description**

nrow S3 method for blockmatrix object

**Usage**

```
## S3 method for class 'blockmatrix'
nrow(M)
```

**Arguments**

M                    a blockmatrix object

**Value**

Number of rows of blockmatrix M

**Author(s)**

Emanuele Cordano

---

nrow\_elements            nrow\_elements S3 method for blockmatrix object

---

**Description**

nrow\_elements S3 method for blockmatrix object

**Usage**

```
nrow_elements(M, zero_element = "0", ...)  
  
## Default S3 method:  
nrow_elements(M, zero_element = "0",  
              ...)  
  
## S3 method for class 'blockmatrix'  
nrow_elements(M,  
              zero_element = "0", ...)
```

**Arguments**

M                    a blockmatrix object  
zero\_element        character value indicating a zero matrix. Default is "0"  
...                  further arguments

**Value**

The number of rows of a matrix-type element of M. It is NA if the elements has different number of rows.

**Author(s)**

Emanuele Cordano

---

solve.blockmatrix	<i>solve</i>
-------------------	--------------

---

**Description**

dim S3 solve for blockmatrix object

**Usage**

```
## S3 method for class 'blockmatrix'
solve(a, b = NULL,
      as.blockmatrix = TRUE, ...)
```

**Arguments**

a	a blockmatrix or numeric object
b	a blockmatrix or numeric object. If omitted, it is NULL. See Details.
as.blockmatrix	logical value. If TRUE (Default), the output is a blockmatrix object
...	further arguments for method solve

**Value**

the object  $x$  such that  $a * x = b$  where  $*$  is the matrix product.

**Note**

If  $b$  is missing, i.e. NULL, it will be replaced by the corresponding identity matrix. So  $x$  is calculated as the right inverse of  $a$ . The matrix system must be nonsingular and nonhomogeneous.

**Author(s)**

Emanuele Cordano

---

t.blockmatrix	<i>t 'transpose' S3 method for blockmatrix object</i>
---------------	---

---

**Description**

*t 'transpose'* S3 method for blockmatrix object

**Usage**

```
## S3 method for class 'blockmatrix'
t(x)
```

**Arguments**

x                    a blockmatrix object

**Author(s)**

Emanuele Cordano

---

value                    value *S3 method for blockmatrix object*

---

**Description**

value S3 method for blockmatrix object

**Usage**

```
value(M)

## Default S3 method:
value(M)

## S3 method for class 'blockmatrix'
value(M)
```

**Arguments**

M                    a blockmatrix object

**Value**

The character matrix without numerical values (e.g. only the matrix M\$value)

**Author(s)**

Emanuele Cordano

---

value<-	value<- <i>S3 Replacement method for blockmatrix object</i>
---------	---

---

**Description**

value<- S3 Replacement method for blockmatrix object

**Usage**

```
value(M) <- value

## Default S3 replacement method:
value(M) <- value

## S3 replacement method for class 'blockmatrix'
value(M) <- value
```

**Arguments**

M	a blockmatrix object
value	object replaced matrix

**Value**

Replaces M\$value with a new matrix value

**Author(s)**

Emanuele Cordano

---

[.blockmatrix	[ <i>S3 method for</i> <a href="#">blockmatrix</a> <i>object</i>
---------------	--

---

**Description**

[ S3 method for [blockmatrix](#) object

**Usage**

```
## S3 method for class 'blockmatrix'
M[i = 1:nrow(M), j =
  1:ncol(M), numeric_value=TRUE, blockmatrix=FALSE, ...]
```

**Arguments**

M	a <code>blockmatrix</code> object
i, j	matrix indices (numerical or character)
numeric_value	logical value . If TRUE (Default if i, j have both length 1) and i, j have both length 1, a i, j numeric matrix is returned.
blockmatrix	logical value. If TRUE (Default if i or j have length greater than 1) a <code>blockmatrix</code> is returned.
...	further argument for <code>[]</code> method

**Value**

The i, j matrix as a numerical matrix if `blockmatrix` is FALSE, otherwise the return object is a `blockmatrix` object. In case i is a character vector, the method returns a list of objects with name containing in i and taken from M.

**Author(s)**

Emanuele Cordano

---

[<- .blockmatrix      ' [<- ' S3 Replacement method for `blockmatrix` object

---

**Description**

' [<- ' S3 Replacement method for `blockmatrix` object

**Usage**

```
## S3 replacement method for class 'blockmatrix'
M[i = 1:nrow(M), j = 1:ncol(M)] <- value
```

**Arguments**

M	a <code>blockmatrix</code> object
i, j	matrix indices (numerical or character)
value	a <code>blockmatrix</code> object to be replaced

**Value**

The "replaced" `blockmatrix` object.

**Note**

In case i is a character vector, the elements whose names is in value is replaced.

**Author(s)**

Emanuele Cordano

**Examples**

```
rm(list=ls())
library(blockmatrix)
A <- array(rnorm(9,mean=1),c(3,3))
B <- 0 #array(rnorm(9,mean=2),c(3,3))
C <- 0
D <- array(rnorm(9,mean=4),c(3,3))
F <- array(rnorm(9,mean=10),c(3,3))
M <- blockmatrix(names=c("A","0","D","0"),A=A,D=D,dim=c(2,2))
E <- blockmatrix(names=c("0","F","D","0"),F=F,D=D,dim=c(2,2))
E[,1] <- M[,1]
```

# Index

[, [14](#)  
[ ([.blockmatrix), [13](#)  
[.blockmatrix, [13](#)  
[<-.blockmatrix, [14](#)  
[<- , extract\_replacemethod  
    ([<-.blockmatrix), [14](#)

as.blockmatrix, [2, 5](#)  
as.matrix (as.matrix.blockmatrix), [3](#)  
as.matrix.blockmatrix, [3](#)

blockmatmult, [4](#)  
blockmatrix, [4, 13, 14](#)

dim (dim.blockmatrix), [6](#)  
dim.blockmatrix, [6](#)

Extract ([.blockmatrix), [13](#)

is.zero.blockmatrix, [6](#)

length (length.blockmatrix), [7](#)  
length.blockmatrix, [7](#)

Math (Math.blockmatrix), [7](#)  
Math.blockmatrix, [7](#)  
matmult, [4](#)

names (names.blockmatrix), [8](#)  
names.blockmatrix, [8](#)  
ncol (ncol.blockmatrix), [8](#)  
ncol.blockmatrix, [8](#)  
ncol\_elements, [3, 9](#)  
nrow (nrow.blockmatrix), [9](#)  
nrow.blockmatrix, [9](#)  
nrow\_elements, [3, 10](#)

Ops (Math.blockmatrix), [7](#)

solve (solve.blockmatrix), [11](#)  
solve.blockmatrix, [11](#)

t (t.blockmatrix), [11](#)  
t.blockmatrix, [11](#)

value, [5, 12](#)  
value<-, [13](#)