

# Package ‘rmumps’

December 22, 2016

**Type** Package

**Title** Wrapper for MUMPS Library

**Version** 5.0.1.11-1

**Date** 2016-12-21

**Author** Serguei Sokol

**Maintainer** Serguei Sokol <sokol@insa-toulouse.fr>

**Description** Some basic features of MUMPS (MULTifrontal Massively Parallel sparse direct Solver) are wrapped in a Rmumps class. It can be used for storing a sparse matrix and sequentially solving corresponding linear system with one or many right hand sides (in their turn dense or sparse). There is a possibility to do separately symbolic analysis, LU factorization and system solving.

**License** GPL (>= 2)

**Imports** methods, Rcpp (>= 0.12.0)

**LinkingTo** Rcpp

**SystemRequirements** C++11, GNU Make

**NeedsCompilation** yes

**Biarch** yes

**Suggests** testthat, Matrix, slam

**BugReports** <https://github.com/sgsokol/rmumps/issues>

**URL** <http://mumps.enseeiht.fr/>, <https://github.com/sgsokol/rmumps/>

**Repository** CRAN

**Date/Publication** 2016-12-22 23:42:21

## R topics documented:

rmumps-package . . . . .	2
Rcpp_Rmumps-class . . . . .	2

<b>Index</b>	<b>5</b>
--------------	----------

---

 rmumps-package

*Rcpp port of MUMPS library for LU factorization of sparse matrices*


---

### Description

Creates a MUMPS compatible object storing a sparse matrix. Gives a possibility to do separately symbolic analysis, LU factorization and system solving.

### Details

The DESCRIPTION file: Create a new Rmumps object with `A <- Rmumps$new(asparsed)` then solve a linear system with one or many right hand sides `x <- solve(A, b)`

### Author(s)

Serguei Sokol, INRA

Maintainer: Serguei Sokol (sokol at insa-toulouse.fr)

### References

MUMPS official site <http://mumps.enseeiht.fr>

Sokol S (2016). `_Rmumps: Rcpp port of MUMPS_`. rmumps package version 5.0.1.11, <URL: <http://CRAN.R-project.org/package=rmumps>>.

### Examples

```
## Not run:
A <- Rmumps$new(asparsed)
x <- solve(A, b)

## End(Not run)
```

---

 Rcpp\_Rmumps-class

*Rcpp Exported Class Wrapping MUMPS library*


---

### Description

This class can be used for storing sparse matrix and solving corresponding linear system with one or many right hand sides. There is a possibility to do separately symbolic analysis, LU factorization and system solving.

### Fields

`copy`: logical, copy or not rhs and matrix values

`mrhs`: numeric matrix, multiple rhs (always overwritten with solution)

`rhs`: numeric vector, single rhs (always overwritten with solution)

**Methods**

`new(asp, copy=TRUE)`: constructor from `Matrix::dgTMatrix` class (or from convertible to it) and `slam::simple_triplet_matrix` class

`new(i, j, x, n, copy=TRUE)`: constructor from triade rows, cols, vals

`symbolic()`: do symbolic analysis (stored internally)

`numeric()`: do LU factorization (stored internally)

`solve(b)`: solve single rhs (if b is a vector) or multiple rhs if b is a matrix (can be dense or sparse).  
Return the solution(s).

`inv()`: Return inverse of the matrix)

`set_mat_data(x)`: updates matrix entries (x must be in same order as in previous calls)

`set_icntl(iv, ii)`: set ICNTL parameter vector

`get_icntl()`: get ICNTL parameter vector

`set_cntl(v, iv)`: set CNTL parameter vector

`get_cntl()`: get CNTL parameter vector

`get_infos()`: get a named list of information vectors: info, rinfo, infog and rinfog

`dim()`: Return a dimension vector of the matrix)

`nrow()`: Return a row number of the matrix)

`ncol()`: Return a column number of the matrix)

`print()`: Print summary information on the matrix)

`show()`: Print summary information on the matrix)

**Note**

For meaning of entries in MUMPS vectors `cntl`, `icntl`, `info`, `rinfo`, `infog` and `rinfog` cf. original documentation of MUMPS project.

No need to call `symbolic()` and `numeric()` methods before a `solve()` call.

If in constructor, parameter `copy` is set to `FALSE`, no rhs neither matrix copying is done. The solution is written "in place" thus overwriting rhs (watch out side effects)

For a detailed error diagnostic (e.g. when factorizing a singular matrix), use method `get_infos()` and cf. MUMPS documentation on the official MUMPS site).

**Author(s)**

Serguei Sokol, INRA

**References**

MUMPS official site <http://mumps.enseiht.fr>

Sokol S (2015). `_Rmumps: Rcpp port of MUMPS_`. `rmumps` package version 5.0.1\_1, <URL: <http://CRAN.R-project.org/package=rmumps>>.

**Examples**

```
## Not run:
# prepare random sparse matrix
library(Matrix)
library(Rmumps)
n=2000
a=Matrix(0, n, n)
set.seed(7)
ij=sample(1:(n*n), 15*n)
a[ij]=runif(ij)
diag(a)=0
diag(a)=-rowSums(a)
a[1,1]=a[1,1]-1
am=Rmumps$new(a)
b=as.double(a%%(1:n)) # rhs for an exact solution vector 1:n
# following time includes symbolic analysis, LU factorization and system solving
system.time(x<-solve(am, b))
bb=2*b
# this second time should be much shorter
# as symbolic analysis and LU factorization are already done
system.time(xx<-solve(am, bb))
# compare to Matrix corresponding times
system.time(xm<-solve(a, b))
system.time(xxm<-solve(a, bb))
# compare to Matrix precision
range(x-1:n) # mumps
range(xm-1:n) # Matrix

# matrix inversion
system.time(aminv <- solve(am))
system.time(ainv <- solve(a)) # the same in Matrix

# clean up by hand to avoid possible interference between gc() and
# Rcpp object destructor after unloading this namespace
rm(am)

## End(Not run)
```

# Index

## \*Topic **classes**

`Rcpp_Rmumps-class`, [2](#)

`dim.Rcpp_Rmumps` (`Rcpp_Rmumps-class`), [2](#)

`ncol.Rcpp_Rmumps` (`Rcpp_Rmumps-class`), [2](#)

`nrow.Rcpp_Rmumps` (`Rcpp_Rmumps-class`), [2](#)

`print.Rcpp_Rmumps` (`Rcpp_Rmumps-class`), [2](#)

`Rcpp_Rmumps-class`, [2](#)

`Rmumps` (`Rcpp_Rmumps-class`), [2](#)

`rmumps` (`rmumps-package`), [2](#)

`rmumps-package`, [2](#)

`show.Rcpp_Rmumps` (`Rcpp_Rmumps-class`), [2](#)

`solve.Rcpp_Rmumps` (`Rcpp_Rmumps-class`), [2](#)