

# Package ‘rminqa’

May 9, 2022

**Type** Package

**Title** Derivative-Free Optimization in R using C++

**Version** 0.1.1

**Date** 2022-05-05

**Maintainer** Yi Pan <ypan1988@gmail.com>

**Description** Perform derivative-free optimization algorithms in R using C++.

A wrapper interface is provided to call C function of the 'bobyqa' implementation  
(See <<https://github.com/emmt/Algorithms/tree/master/bobyqa>>).

**License** GPL (>= 2)

**Encoding** UTF-8

**SystemRequirements** C++11

**Imports** Rcpp (>= 1.0.7)

**LinkingTo** Rcpp, RcppArmadillo

**RoxygenNote** 7.1.2

**Suggests** minqa

**NeedsCompilation** yes

**Author** Sam Watson [aut],  
Yi Pan [aut, cre],  
Éric Thiébaud [aut],  
Mike Powell [aut]

**Repository** CRAN

**Date/Publication** 2022-05-09 12:10:02 UTC

## R topics documented:

bobyqa_rosen_x1 . . . . .	2
bobyqa_rosen_x1e . . . . .	2
rminqa . . . . .	3

<b>Index</b>	<b>4</b>
--------------	----------

---

bobyqa\_rosen\_x1

*Example 1a: Minimize Rosenbrock function using bobyqa*


---

**Description**

Minimize Rosenbrock function using bobyqa and expect a normal exit from bobyqa.

**Usage**

```
bobyqa_rosen_x1()
```

**Value**

No return value, called for side effects.

**Examples**

```
fr <- function(x) { ## Rosenbrock Banana function
  x1 <- x[1]
  x2 <- x[2]
  100 * (x2 - x1 * x1)^2 + (1 - x1)^2
}
(x1 <- minqa::bobyqa(c(1, 2), fr, lower = c(0, 0), upper = c(4, 4)))
## => optimum at c(1, 1) with fval = 0
str(x1) # see that the error code and msg are returned

## corresponding C++ implementation:
bobyqa_rosen_x1()
```

---

bobyqa\_rosen\_x1e

*Example 1b: Minimize Rosenbrock function using bobyqa*


---

**Description**

Minimize Rosenbrock function using bobyqa and expect a normal exit from bobyqa.

**Usage**

```
bobyqa_rosen_x1e()
```

**Value**

No return value, called for side effects.

**Examples**

```
fr <- function(x) { ## Rosenbrock Banana function
  x1 <- x[1]
  x2 <- x[2]
  100 * (x2 - x1 * x1)^2 + (1 - x1)^2
}
# check the error exits
# too many iterations
x1e <- minqa::bobyqa(c(1, 2), fr, lower = c(0, 0), upper = c(4, 4), control = list(maxfun=50))
str(x1e)

## corresponding C++ implementation:
bobyqa_rosen_x1e()
```

---

rminqa

*rminqa*

---

**Description**

Perform derivative-free optimization algorithms in R using C++. A wrapper interface is provided to call C function of the bobyqa implementation.

**Author(s)**

Yi Pan, Samuel Watson

# Index

bobyqa\_rosen\_x1, [2](#)

bobyqa\_rosen\_x1e, [2](#)

rminqa, [3](#)