

Package ‘RcppTrust’

September 7, 2026

Type Package

Title Thread-Safe C++ Port of the 'trust' Trust Region Optimizer

Version 0.1.0

Description Does local optimization using two derivatives and trust regions. Guaranteed to converge to a local minimum of the objective function. This is a thread-safe C++ port of the algorithm in the 'trust' package by Charles J. Geyer, exposed both as a drop-in R function and as a C-callable, thread-safe entry point (with function pointers registered for other packages to call without linking against this package's shared library) so it can be used from parallel C++ code such as in 'nlmixr2est'.

License MIT + file LICENSE

URL <https://github.com/nlmixr2/RcppTrust/>,
<https://nlmixr2.github.io/RcppTrust/>,
<http://www.stat.umn.edu/geyer/trust/>

Encoding UTF-8

Depends R (>= 4.2.0)

Imports Rcpp

LinkingTo Rcpp, RcppArmadillo, BH

Suggests testthat (>= 3.0.0), stats, knitr, rmarkdown, trust,
microbenchmark

Config/testthat/edition 3

SystemRequirements C++17

VignetteBuilder knitr

Config/roxygen2/version 8.1.0

NeedsCompilation yes

Author Charles J. Geyer [aut] (ORCID: <<https://orcid.org/0000-0003-1471-1703>>),
Matthew Fidler [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-8538-6691>>)

Maintainer Matthew Fidler <matthew.fidler@gmail.com>

Repository CRAN
Date/Publication 2026-09-07 12:20:02 UTC

Contents

.RcppTrustPtr	2
trust	2
Index	4

.RcppTrustPtr	<i>Positionally-indexed function pointer table for downstream packages</i>
---------------	--

Description

Returns the list of native function pointers a consuming package (e.g. `nlmixr2est`) resolves via the header-only registration pattern in `inst/include/RcppTrust.h` – the same pattern `rxode2/nlqn1c/lbfgsb3c` use, so no linker dependency on this package’s shared library is required. Not intended to be called directly by end users.

Usage

`.RcppTrustPtr()`

Value

An unclassified list of external pointers.

trust	<i>Non-Linear Optimization by a Trust Region Algorithm</i>
-------	--

Description

Thread-safe C++ port of `trust::trust()` (Geyer). Carries out a minimization or maximization of a function using a trust region algorithm; see the original package’s documentation for the algorithm and argument semantics, which this function reproduces exactly.

Usage

```
trust(  
  objfun,  
  parinit,  
  rinit,  
  rmax,  
  parscale,  
  iterlim = 100,  
  fterm = sqrt(.Machine$double.eps),  
  mterm = sqrt(.Machine$double.eps),  
  minimize = TRUE,  
  blather = FALSE,  
  ...  
)
```

Arguments

objfun	an R function computing value, gradient, and Hessian; see <code>trust::trust</code> for the exact contract.
parinit	starting parameter values.
rinit	starting trust region radius.
rmax	maximum allowed trust region radius.
parscale	optional parameter scale vector.
iterlim	maximum number of iterations.
fterm	function-value termination tolerance.
mterm	model-decrease termination tolerance.
minimize	if TRUE minimize, if FALSE maximize.
blather	if TRUE return extra per-iteration information.
...	additional arguments passed to <code>objfun</code> .

Value

A list with the same components as `trust::trust()`.

Index

.RcppTrustPtr, [2](#)

trust, [2](#)