Package ‘lsoda’

April 24, 2025

Type Package

Title 'C++' Header Library for Ordinary Differential Equations

Version 1.1

Description A 'C++' header library for using the 'libsoda-cxx' library with R. The 'C++' header reimplements the 'lsoda' function from the 'ODEPACK' library for solving initial value problems for first order ordinary differential equations (Hindmarsh, 1982; <https://computing.llnl.gov/sites/default/files/ODEPACK_pub1_u88007.pdf>). The 'C++' header can be used by other R packages by linking against this package. The 'C++' functions can be called inline using 'Rcpp'. Finally, the package provides an 'ode' function to call from R.

License MIT + file LICENSE

URL https://github.com/mclements/lsoda

BugReports https://github.com/mclements/lsoda/issues

Imports Rcpp (>= 1.0.12)

Suggests deSolve, RcppArmadillo, RcppEigen, microbenchmark

LinkingTo Rcpp

RoxygenNote 7.3.2

Encoding UTF-8

NeedsCompilation yes

Author Mark Clements [aut, cre],
Dilawar Singh [ctb],
Heng Li [ctb],
Peter N. Brown [ctb],
George D. Byrne [ctb],
Alan C. Hindmarsh [ctb],
Cleve Moler [ctb],
Linda R. Petzold [ctb]

Maintainer Mark Clements <mark.clements@ki.se>

Repository CRAN

Date/Publication 2025-04-24 11:40:10 UTC

Contents

`ode` *Ordinary differential equation solver using lsoda*

Description

Ordinary differential equation solver using lsoda

Usage

```
ode(y, times, func, rtol = 1e-06, atol = 1e-06)
```

Arguments

`y` vector of initial state values

`times` vector of times – including the start time

`func` R function with signature function(t,y) that returns a list. The first list element is a vector for dy/dt. The second list elements, if it exists, is a vector of result calculations to be retained.

`rtol` double for the relative tolerance

`atol` double for the absolute tolerance

Value

a matrix for times in the first column and the state andd results values in the other columns.

Examples

```
 times = c(0,0.4*10^(0:10))
y = c(1,0,0)
func = function(t,y,...) {
    ydot = rep(0,3)
    ydot[1] = 1.0E4 * y[2] * y[3] - .04E0 * y[1]
    ydot[3] = 3.0E7 * y[2] * y[2]
    ydot[2] = -1.0 * (ydot[1] + ydot[3])
    list(ydot, sum(y))
}
lsoda::ode(y,times,func, rtol=1e-8, atol=1e-8)
```

Index