Package 'insideRODE'

February 20, 2015

Type Package

Title insideRODE includes buildin functions with deSolve solver and C/FORTRAN interfaces to nlme, together with compiled codes.

Version 2.0

Date 2011-04-19

Author YUZHUO PAN, XIAOYU YAN

Maintainer YUZHUO PAN <yuzhuo.pan@gmail.com>

Depends R (>= 2.13.0), deSolve, nlme, lattice, compiler

Description insideRODE package includes buildin functions from deSolve, compiled functions from compiler, and C/FORTRAN code interfaces to nlme. It includes nlmLSODA, nlmODE, nlmVODE,nlmLSODE for general purpose; cfLSODA,cfLSODE, cfODE, cfVODE call C/FORTRAN compiled dll functions.ver2.0 add sink()function into example it helps to directly combine c/fortran source code in R files. Finally, with new compiler package, we generated compiled functions: nlmODEcp, nlmVODEcp, nlmLSODEcp,nlmLSODAcp and cpODE, cpLSODA, cpLSODE, cpVODE. They will help to increase speed.

License LGPL (> 2.0)

LazyLoad yes

Repository CRAN

Date/Publication 2012-10-29 08:59:01

NeedsCompilation no

R topics documented:

`insideRODE-package` *insideRODE includes buildin functions with deSolve solver and C/FORTRAN interfaces to nlme, together with compiled codes.*

Description

insideRODE build-in function with Ordinary Differential Equation solver and C/FORTRAN interface to nlme, including nlmLSODA, nlmODE, nlmVODE,nlmLSODE for general ODE; cfLSODA, cfLSODE, cfODE, cfVODE solver for C/FORTRAN based ODE. WE USE SEPERATED FILE TO GENERATE FUNCTIONS. V1.0 can read dllname from dynload, sent them to cf FUNCTIONS. insideRODE package also includes buildin functions from deSolve, compiled functions from compiler, and C/FORTRAN code interfaces to nlme. It includes nlmLSODA, nlmODE, nlmVODE,nlmLSODE for general purpose; cfLSODA,cfLSODE, cfODE, cfVODE call C/FORTRAN compiled dll functions. Ver2.0 add sink()function into example it helps to directly combine c/fortran source code in R files. Finally, with new compiler package, we generated compiled functions: nlmODEcp, nlmVODEcp, nlmLSODEcp,nlmLSODAcp and cpODE, cpLSODA, cpLSODE, cpVODE. They will help to increase speed. This package depends on the package of nlmeODE from Christoffer W. Tornoe. This package updated the package from ODESOLVE into deSolve package, and implement the interface to c/fortran code. It will greatly enhance the performance of R and nlme.

Details

Package:	insideRODE
Type:	Package
Version:	2.0
Date:	2011-04-19
License:	LGPL(>2.0)
LazyLoad:	yes

Author(s)

YUZHUO PAN, XIAOYU YAN

Maintainer: YUZHUO PAN <yuzhuo.pan@gmail.com>

See Also

nlme, nlmeODE, deSolve, lattice, compiler

Examples

```
## Not run:
## show examples
example(EXnlmLSODA)
example(EXnlmLSODE)
example(EXnlmODE)
example(EXnlmVODE)
example(EXnlmLSODAcp)#compiled code with compiler
example(EXnlmLSODEcp)
example(EXnlmVODEcp)
example(EXnlmODEcp)
example(EXcfLSODA)# compiled dll function
example(EXcfLSODE)
example(EXcfODE)
example(EXcfVODE)
example(EXcpODE)  #compiled code with compiler
example(EXcpLSODE)
example(EXcpVODE)
example(EXcpLSODA)
## run demos
demo("testfile")     # differential equations


## End(Not run)
```

`cfLSODA` *LSODA Solver for NLME using compiled code(c or fortran)*

Description

Use Solver for Ordinary Differential Equations (ODE), Switching Automatically Between Stiff and Non-stiff Methods and Generate functions to be used in NLME

Usage

```
cfLSODA(model, data, LogParms = TRUE, JAC = FALSE, SEQ = FALSE,rtol = 1e-4, atol = 1e-4, tcrit = NUL
```

Arguments

model	either an R-function that computes the values of the derivatives in the ODE system (the *model definition*) at time t.The return value of model should be a list. See package "nlmeODE" for more details.
data	nlme GroupedData format.
LogParms	transform parameters into log scale
JAC	A JAC set FALSE. This time we can implement this parts.
SEQ	A SEQ set FALSE.
rtol	relative error tolerance, either a scalar or an array as long as y. See details.
atol	absolute error tolerance, either a scalar or an array as long as y. See details.
tcrit	if not NULL, then lsoda cannot integrate past tcrit. The FORTRAN routine lsoda overshoots its targets (times points in the vector times), and interpolates values for the desired time points. If there is a time beyond which integration should not proceed (perhaps because of a singularity), that should be provided in tcrit.
dllname	a string giving the name of the shared library (without extension) that contains all the compiled function or subroutine definitions refered to in func and jacfunc. See package "deSolve".
hmin	an optional minimum value of the integration stepsize. In special situations this parameter may speed up computations with the cost of precision. Don't use hmin if you don't know why!
hmax	an optional maximum value of the integration stepsize. If not specified, hmax is set to the largest difference in times, to avoid that the simulation possibly ignores short-term events. If 0, no maximal size is specified.

Examples

```
#########################################################
#use c code
#cfLSODA SOLVER
#########################################################
rm(list=ls())
require(insideRODE)

data(Theoph)# examples from nlmeODE
TheophODE <- Theoph
TheophODE$Dose[TheophODE$Time!=0] <- 0
TheophODE$Cmt <- rep(1,dim(TheophODE)[1])

# model files
OneComp <- list(DiffEq=list(
                            dy1dt = ~ -ka*y1 ,
                            dy2dt = ~ ka*y1-ke*y2),
                ObsEq=list(
                            c1 = ~ 0,
                            c2 = ~ y2/CL*ke),
                Parms=c("ka","ke","CL"),
```

```
                States=c("y1","y2"),
                Init=list(0,0))

TheophModel <- nlmLSODA(OneComp,TheophODE) #ode solver
```

---

cfLSODE *LSODE Solver for NLME using compiled code(c or fortran)*

Description

Use Solver for Ordinary Differential Equations (ODE), Switching Automatically Between Stiff and Non-stiff Methods and Generate functions to be used in NLME

Usage

```
cfLSODE(model, data, LogParms = TRUE, JAC = FALSE, SEQ = FALSE,rtol = 1e-4, atol = 1e-4, tcrit = NULL,
```

Arguments

model	either an R-function that computes the values of the derivatives in the ODE system (the *model definition*) at time t.The return value of model should be a list. See package "nlmeODE" for more details.
data	nlme GroupedData format.
LogParms	transform parameters into log scale
JAC	A JAC set FALSE. This time we can implement this parts.
SEQ	A SEQ set FALSE.
rtol	relative error tolerance, either a scalar or an array as long as y. See details.
atol	absolute error tolerance, either a scalar or an array as long as y. See details.
tcrit	if not NULL, then lsoda cannot integrate past tcrit. The FORTRAN routine lsoda overshoots its targets (times points in the vector times), and interpolates values for the desired time points. If there is a time beyond which integration should not proceed (perhaps because of a singularity), that should be provided in tcrit.
dllname	a string giving the name of the shared library (without extension) that contains all the compiled function or subroutine definitions refered to in func and jacfunc. See package "deSolve".
hmin	an optional minimum value of the integration stepsize. In special situations this parameter may speed up computations with the cost of precision. Don't use hmin if you don't know why!
hmax	an optional maximum value of the integration stepsize. If not specified, hmax is set to the largest difference in times, to avoid that the simulation possibly ignores short-term events. If 0, no maximal size is specified.

Examples

```
#######################################################
#use c code
#see example cfLSODA
#######################################################
```

`cfODE` *General Solver for Ordinary Differential Equations and compiled functions to be used in NLME, this version just use default function. The future version will provide buildin methods such as "lsoda", "lsode", "lsodes", "lsodar", "vode", "daspk", "euler", "rk4", "ode23", "ode45", "radau", "bdf", "bdf_d", "adams", "impAdams", "impAdams_d".*

Description

Generate functions for NLME Solves using a system of ordinary differential equations; a wrapper around the implemented ODE solvers

Usage

```
cfODE(model, data, LogParms = TRUE, JAC = FALSE, SEQ = FALSE,rtol = 1e-4, atol = 1e-4, tcrit = NULL, dl
```

Arguments

`model`	either an R-function that computes the values of the derivatives in the ODE system (the *model definition*) at time t.The return value of `model` should be a list. See package `"nlmeODE"` for more details.
`data`	nlme GroupedData format.
`LogParms`	transform parameters into log scale
`JAC`	A JAC set FALSE. This time we can implement this parts.
`SEQ`	A SEQ set FALSE.
`rtol`	relative error tolerance, either a scalar or an array as long as `y`. See details.
`atol`	absolute error tolerance, either a scalar or an array as long as `y`. See details.
`tcrit`	if not `NULL`, then `lsoda` cannot integrate past `tcrit`. The FORTRAN routine `lsoda` overshoots its targets (times points in the vector `times`), and interpolates values for the desired time points. If there is a time beyond which integration should not proceed (perhaps because of a singularity), that should be provided in `tcrit`.
`dllname`	a string giving the name of the shared library (without extension) that contains all the compiled function or subroutine definitions refered to in `func` and `jacfunc`. See package `"deSolve"`.

hmin	an optional minimum value of the integration stepsize. In special situations this parameter may speed up computations with the cost of precision. Don't use hmin if you don't know why!
hmax	an optional maximum value of the integration stepsize. If not specified, hmax is set to the largest difference in times, to avoid that the simulation possibly ignores short-term events. If 0, no maximal size is specified.

Examples

```
#####################################################
#use c code
#see example cfLSODA
#####################################################
```

---

cfVODE *Solver for Ordinary Differential Equations (VODE) and compiled functions(c/fortran) to be used in NLME*

Description

Generate functions for NLME Solves using a system of ordinary differential equations; a wrapper around the implemented ODE solvers,The R function vode provides an interface to the FORTRAN ODE solver of the same name, written by Peter N. Brown, Alan C. Hindmarsh and George D. Byrne.

Usage

```
cfVODE(model, data, LogParms = TRUE, JAC = FALSE, SEQ = FALSE,rtol = 1e-4, atol = 1e-4, tcrit = NULL, d
```

Arguments

model	either an R-function that computes the values of the derivatives in the ODE system (the *model definition*) at time t.The return value of model should be a list. See package "nlmeODE" for more details.
data	nlme GroupedData format.
LogParms	transform parameters into log scale
JAC	A JAC set FALSE. This time we can implement this parts.
SEQ	A SEQ set FALSE.
rtol	relative error tolerance, either a scalar or an array as long as y. See details.
atol	absolute error tolerance, either a scalar or an array as long as y. See details.
tcrit	if not NULL, then lsoda cannot integrate past tcrit. The FORTRAN routine lsoda overshoots its targets (times points in the vector times), and interpolates values for the desired time points. If there is a time beyond which integration should not proceed (perhaps because of a singularity), that should be provided in tcrit.

dllname	a string giving the name of the shared library (without extension) that contains all the compiled function or subroutine definitions refered to in func and jacfunc. See package "deSolve".
hmin	an optional minimum value of the integration stepsize. In special situations this parameter may speed up computations with the cost of precision. Don't use hmin if you don't know why!
hmax	an optional maximum value of the integration stepsize. If not specified, hmax is set to the largest difference in times, to avoid that the simulation possibly ignores short-term events. If 0, no maximal size is specified.

Examples

```
######################################################
#use c code
#see example cfLSODA
######################################################
```

---

cpLSODA *LSODA Solver for NLME using compiled code(c or fortran)*

Description

Use Solver for Ordinary Differential Equations (ODE), Switching Automatically Between Stiff and Non-stiff Methods and Generate functions to be used in NLME

Usage

```
cpLSODA(model, data, LogParms = TRUE, JAC = FALSE, SEQ = FALSE,rtol = 1e-4, atol = 1e-4, tcrit = NUL
```

Arguments

model	either an R-function that computes the values of the derivatives in the ODE system (the *model definition*) at time t.The return value of model should be a list. See package "nlmeODE" for more details.
data	nlme GroupedData format.
LogParms	transform parameters into log scale
JAC	A JAC set FALSE. This time we can implement this parts.
SEQ	A SEQ set FALSE.
rtol	relative error tolerance, either a scalar or an array as long as y. See details.
atol	absolute error tolerance, either a scalar or an array as long as y. See details.
tcrit	if not NULL, then lsoda cannot integrate past tcrit. The FORTRAN routine lsoda overshoots its targets (times points in the vector times), and interpolates values for the desired time points. If there is a time beyond which integration should not proceed (perhaps because of a singularity), that should be provided in tcrit.

dllname — a string giving the name of the shared library (without extension) that contains all the compiled function or subroutine definitions refered to in func and jacfunc. See package "deSolve".

hmin — an optional minimum value of the integration stepsize. In special situations this parameter may speed up computations with the cost of precision. Don't use hmin if you don't know why!

hmax — an optional maximum value of the integration stepsize. If not specified, hmax is set to the largest difference in times, to avoid that the simulation possibly ignores short-term events. If 0, no maximal size is specified.

Examples

```
#########################################################
#use c code
#cpLSODA SOLVER
#########################################################
rm(list=ls())
require(insideRODE)

data(Theoph)# examples from nlmeODE
TheophODE <- Theoph
TheophODE$Dose[TheophODE$Time!=0] <- 0
TheophODE$Cmt <- rep(1,dim(TheophODE)[1])

# model files
OneComp <- list(DiffEq=list(
                            dy1dt = ~ -ka*y1 ,
                            dy2dt = ~ ka*y1-ke*y2),
                ObsEq=list(
                            c1 = ~ 0,
                            c2 = ~ y2/CL*ke),
                Parms=c("ka","ke","CL"),
                States=c("y1","y2"),
                Init=list(0,0))

TheophModel <- nlmLSODA(OneComp,TheophODE) #ode solver

#########################################################
#example
#sink functions
#cpLSODE
#########################################################
#sink("mymod.c")
cat("
/* file mymod.c */
#include <R.h>
#include <math.h>
static double parms[3];
#define ka parms[0]
#define ke parms[1]
#define CL parms[2]
```

```
/* initializer */
void initmod(void (* odeparms)(int *, double *))
{
   int N=2;
   odeparms(&N, parms);
}

/* names for states and derivatives */
#define y1 y[0]
#define y2 y[1]
#define dy1 ydot[0]
#define dy2 ydot[1]
#define c1 yout[0]
#define c2 yout[1]

/* Derivatives and 1 output variable */
void derivs (int *neq, double *t, double *y, double *ydot, double *yout, int *ip)
{
    dy1 = -exp(ka)*y1;
    dy2 = exp(ka)*y1-exp(ke)*y2;
    c1 = 0.0;
    c2 = y2/exp(CL)*exp(ke);
}

/* END file mymod1.c */
",fill=TRUE)
#sink()
#system("RCMD SHLIB mymod.c")
#dllname<-dyn.load("mymod.dll")[[1]]


#TheophModelc <- cpLSODA(OneComp,TheophODE,dllname=dllname)

#cpLSODA,cfLSODE, cpLSODA, cfVODE SOLVER
#sink("mymodff.f")
cat("
c file mymodf.f
        subroutine initmod(odeparms)
        external odeparms
        double precision parms(3)
        common /myparms/parms
        call odeparms(2, parms)
        return
        end
        subroutine derivs (neq, t, y, ydot, yout, ip)
        double precision t, y, ydot, ka, ke, CL
        integer neq, ip(*)
        dimension y(2), ydot(2), yout(2)
        common /myparms/ka,ke,CL
        ydot(1) = -exp(ka)*y(1)
        ydot(2) = exp(ka)*y(1)-exp(ke)*y(2)
        yout(1) = 0
        yout(2) = y(2)/exp(CL)*exp(ke)
```

```
        return
        end
",fill=TRUE)
#sink(file = NULL, append = FALSE, type = c("output"),split = FALSE)
```

---

cpLSODE *LSODE Solver for NLME using compiled code(c or fortran)*

Description

Use Solver for Ordinary Differential Equations (ODE), Switching Automatically Between Stiff and Non-stiff Methods and Generate functions to be used in NLME

Usage

```
cpLSODE(model, data, LogParms = TRUE, JAC = FALSE, SEQ = FALSE,rtol = 1e-4, atol = 1e-4, tcrit = NULL,
```

Arguments

model	either an R-function that computes the values of the derivatives in the ODE system (the *model definition*) at time t.The return value of model should be a list. See package "nlmeODE" for more details.
data	nlme GroupedData format.
LogParms	transform parameters into log scale
JAC	A JAC set FALSE. This time we can implement this parts.
SEQ	A SEQ set FALSE.
rtol	relative error tolerance, either a scalar or an array as long as y. See details.
atol	absolute error tolerance, either a scalar or an array as long as y. See details.
tcrit	if not NULL, then lsoda cannot integrate past tcrit. The FORTRAN routine lsoda overshoots its targets (times points in the vector times), and interpolates values for the desired time points. If there is a time beyond which integration should not proceed (perhaps because of a singularity), that should be provided in tcrit.
dllname	a string giving the name of the shared library (without extension) that contains all the compiled function or subroutine definitions refered to in func and jacfunc. See package "deSolve".
hmin	an optional minimum value of the integration stepsize. In special situations this parameter may speed up computations with the cost of precision. Don't use hmin if you don't know why!
hmax	an optional maximum value of the integration stepsize. If not specified, hmax is set to the largest difference in times, to avoid that the simulation possibly ignores short-term events. If 0, no maximal size is specified.

Examples

```
#######################################################
#use c code
#see cpLSODA
#######################################################
```

---

cpODE *General Solver for Ordinary Differential Equations and compiled functions to be used in NLME, this version just use default function. The future version will provide buildin methods such as "lsoda", "lsode", "lsodes", "lsodar", "vode", "daspk", "euler", "rk4", "ode23", "ode45", "radau", "bdf", "bdf_d", "adams", "impAdams", "impAdams_d".*

Description

Generate functions for NLME Solves using a system of ordinary differential equations; a wrapper around the implemented ODE solvers

Usage

```
cpODE(model, data, LogParms = TRUE, JAC = FALSE, SEQ = FALSE,rtol = 1e-4, atol = 1e-4, tcrit = NULL, dl
```

Arguments

model	either an R-function that computes the values of the derivatives in the ODE system (the *model definition*) at time t.The return value of model should be a list. See package "nlmeODE" for more details.
data	nlme GroupedData format.
LogParms	transform parameters into log scale
JAC	A JAC set FALSE. This time we can implement this parts.
SEQ	A SEQ set FALSE.
rtol	relative error tolerance, either a scalar or an array as long as y. See details.
atol	absolute error tolerance, either a scalar or an array as long as y. See details.
tcrit	if not NULL, then lsoda cannot integrate past tcrit. The FORTRAN routine lsoda overshoots its targets (times points in the vector times), and interpolates values for the desired time points. If there is a time beyond which integration should not proceed (perhaps because of a singularity), that should be provided in tcrit.
dllname	a string giving the name of the shared library (without extension) that contains all the compiled function or subroutine definitions refered to in func and jacfunc. See package "deSolve".
hmin	an optional minimum value of the integration stepsize. In special situations this parameter may speed up computations with the cost of precision. Don't use hmin if you don't know why!

hmax	an optional maximum value of the integration stepsize. If not specified, hmax is set to the largest difference in times, to avoid that the simulation possibly ignores short-term events. If 0, no maximal size is specified.

Examples

```
#########################################################
#use c code
#see cpLSODA
#########################################################
```

---

cpVODE *Solver for Ordinary Differential Equations (VODE) and compiled functions(c/fortran) to be used in NLME*

Description

Generate functions for NLME Solves using a system of ordinary differential equations; a wrapper around the implemented ODE solvers,The R function vode provides an interface to the FORTRAN ODE solver of the same name, written by Peter N. Brown, Alan C. Hindmarsh and George D. Byrne.

Usage

```
cpVODE(model, data, LogParms = TRUE, JAC = FALSE, SEQ = FALSE,rtol = 1e-4, atol = 1e-4, tcrit = NULL, d
```

Arguments

model	either an R-function that computes the values of the derivatives in the ODE system (the *model definition*) at time t.The return value of model should be a list. See package "nlmeODE" for more details.
data	nlme GroupedData format.
LogParms	transform parameters into log scale
JAC	A JAC set FALSE. This time we can implement this parts.
SEQ	A SEQ set FALSE.
rtol	relative error tolerance, either a scalar or an array as long as y. See details.
atol	absolute error tolerance, either a scalar or an array as long as y. See details.
tcrit	if not NULL, then lsoda cannot integrate past tcrit. The FORTRAN routine lsoda overshoots its targets (times points in the vector times), and interpolates values for the desired time points. If there is a time beyond which integration should not proceed (perhaps because of a singularity), that should be provided in tcrit.
dllname	a string giving the name of the shared library (without extension) that contains all the compiled function or subroutine definitions refered to in func and jacfunc. See package "deSolve".

hmin	an optional minimum value of the integration stepsize. In special situations this parameter may speed up computations with the cost of precision. Don't use hmin if you don't know why!
hmax	an optional maximum value of the integration stepsize. If not specified, hmax is set to the largest difference in times, to avoid that the simulation possibly ignores short-term events. If 0, no maximal size is specified.

Examples

```
#####################################################
#use c code
#see cpLSODA
#####################################################
```

`nlmLSODA` *LSODA Solver for NLME*

Description

Use Solver for Ordinary Differential Equations (ODE), Switching Automatically Between Stiff and Non-stiff Methods and Generate functions to be used in NLME

Usage

```
nlmLSODA(model, data, LogParms = TRUE, JAC = FALSE, SEQ = FALSE,rtol = 1e-4, atol = 1e-4, tcrit = NULL,
```

Arguments

model	either an R-function that computes the values of the derivatives in the ODE system (the *model definition*) at time t.The return value of model should be a list. See package "nlmeODE" for more details.
data	nlme GroupedData format.
LogParms	transform parameters into log scale
JAC	A JAC set FALSE. This time we can implement this parts.
SEQ	A SEQ set FALSE.
rtol	relative error tolerance, either a scalar or an array as long as y. See details.
atol	absolute error tolerance, either a scalar or an array as long as y. See details.
tcrit	if not NULL, then lsoda cannot integrate past tcrit. The FORTRAN routine lsoda overshoots its targets (times points in the vector times), and interpolates values for the desired time points. If there is a time beyond which integration should not proceed (perhaps because of a singularity), that should be provided in tcrit.
hmin	an optional minimum value of the integration stepsize. In special situations this parameter may speed up computations with the cost of precision. Don't use hmin if you don't know why!

hmax — an optional maximum value of the integration stepsize. If not specified, hmax is set to the largest difference in times, to avoid that the simulation possibly ignores short-term events. If 0, no maximal size is specified.

Examples

```
######################################################################
#general model from nlmeODE package
#nlmLSODA USE ACCORDING FUNCTIONS
######################################################################
rm(list=ls())
require(insideRODE)

data(Theoph)# examples from nlmeODE
TheophODE <- Theoph
TheophODE$Dose[TheophODE$Time!=0] <- 0
TheophODE$Cmt <- rep(1,dim(TheophODE)[1])

# model files
OneComp <- list(DiffEq=list(
                            dy1dt = ~ -ka*y1 ,
                            dy2dt = ~ ka*y1-ke*y2),
                ObsEq=list(
                            c1 = ~ 0,
                            c2 = ~ y2/CL*ke),
                Parms=c("ka","ke","CL"),
                States=c("y1","y2"),
                Init=list(0,0))

TheophModel <- nlmLSODA(OneComp,TheophODE) #ode solver
Theoph.nlme <- nlme(conc ~ TheophModel(ka,ke,CL,Time,Subject),
data = TheophODE, fixed=ka+ke+CL~1, random = pdDiag(ka+CL~1),
start=c(ka=0.5,ke=-2.5,CL=-3.2),
control=list(returnObject=TRUE,msVerbose=TRUE),
verbose=TRUE)

plot(augPred(Theoph.nlme,level=0:1))
```

---

nlmLSODAcp *LSODA Solver for NLME*

Description

Use Solver for Ordinary Differential Equations (ODE), Switching Automatically Between Stiff and Non-stiff Methods and Generate functions to be used in NLME

Usage

```
nlmLSODAcp(model, data, LogParms = TRUE, JAC = FALSE, SEQ = FALSE,rtol = 1e-4, atol = 1e-4, tcrit = NUL
```

Arguments

model	either an R-function that computes the values of the derivatives in the ODE system (the *model definition*) at time t.The return value of model should be a list. See package "nlmeODE" for more details.
data	nlme GroupedData format.
LogParms	transform parameters into log scale
JAC	A JAC set FALSE. This time we can implement this parts.
SEQ	A SEQ set FALSE.
rtol	relative error tolerance, either a scalar or an array as long as y. See details.
atol	absolute error tolerance, either a scalar or an array as long as y. See details.
tcrit	if not NULL, then lsoda cannot integrate past tcrit. The FORTRAN routine lsoda overshoots its targets (times points in the vector times), and interpolates values for the desired time points. If there is a time beyond which integration should not proceed (perhaps because of a singularity), that should be provided in tcrit.
hmin	an optional minimum value of the integration stepsize. In special situations this parameter may speed up computations with the cost of precision. Don't use hmin if you don't know why!
hmax	an optional maximum value of the integration stepsize. If not specified, hmax is set to the largest difference in times, to avoid that the simulation possibly ignores short-term events. If 0, no maximal size is specified.

Examples

```
######################################################################
#general model from nlmeODE package
#nlmLSODAcp USE ACCORDING FUNCTIONS
######################################################################
```

---

nlmLSODE *LSODE Solver for NLME*

Description

Use Solver for Ordinary Differential Equations (ODE), Switching Automatically Between Stiff and Non-stiff Methods and Generate functions to be used in NLME

Usage

```
nlmLSODE(model, data, LogParms = TRUE, JAC = FALSE, SEQ = FALSE,rtol = 1e-4, atol = 1e-4, tcrit = NULL,
```

Arguments

model	either an R-function that computes the values of the derivatives in the ODE system (the *model definition*) at time t.The return value of model should be a list. See package "nlmeODE" for more details.
data	nlme GroupedData format.
LogParms	transform parameters into log scale
JAC	A JAC set FALSE. This time we can implement this parts.
SEQ	A SEQ set FALSE.
rtol	relative error tolerance, either a scalar or an array as long as y. See details.
atol	absolute error tolerance, either a scalar or an array as long as y. See details.
tcrit	if not NULL, then lsoda cannot integrate past tcrit. The FORTRAN routine lsoda overshoots its targets (times points in the vector times), and interpolates values for the desired time points. If there is a time beyond which integration should not proceed (perhaps because of a singularity), that should be provided in tcrit.
hmin	an optional minimum value of the integration stepsize. In special situations this parameter may speed up computations with the cost of precision. Don't use hmin if you don't know why!
hmax	an optional maximum value of the integration stepsize. If not specified, hmax is set to the largest difference in times, to avoid that the simulation possibly ignores short-term events. If 0, no maximal size is specified.

Examples

```
######################################################################
#general model from nlmeODE package
#nlmLSODE SOLVER, USE ACCORDING FUNCTIONS
######################################################################
```

---

nlmLSODEcpcp *LSODE Solver for NLME*

Description

Use Solver for Ordinary Differential Equations (ODE), Switching Automatically Between Stiff and Non-stiff Methods and Generate functions to be used in NLME

Usage

```
nlmLSODEcp(model, data, LogParms = TRUE, JAC = FALSE, SEQ = FALSE,rtol = 1e-4, atol = 1e-4, tcrit = NUL
```

Arguments

model	either an R-function that computes the values of the derivatives in the ODE system (the *model definition*) at time t.The return value of model should be a list. See package "nlmeODE" for more details.
data	nlme GroupedData format.
LogParms	transform parameters into log scale
JAC	A JAC set FALSE. This time we can implement this parts.
SEQ	A SEQ set FALSE.
rtol	relative error tolerance, either a scalar or an array as long as y. See details.
atol	absolute error tolerance, either a scalar or an array as long as y. See details.
tcrit	if not NULL, then lsoda cannot integrate past tcrit. The FORTRAN routine lsoda overshoots its targets (times points in the vector times), and interpolates values for the desired time points. If there is a time beyond which integration should not proceed (perhaps because of a singularity), that should be provided in tcrit.
hmin	an optional minimum value of the integration stepsize. In special situations this parameter may speed up computations with the cost of precision. Don't use hmin if you don't know why!
hmax	an optional maximum value of the integration stepsize. If not specified, hmax is set to the largest difference in times, to avoid that the simulation possibly ignores short-term events. If 0, no maximal size is specified.

Examples

```
######################################################################
#general model from nlmeODE package
#nlmLSODEcp SOLVER, USE ACCORDING FUNCTIONS
######################################################################
```

---

nlmODE — *General Solver for Ordinary Differential Equations and Generate functions to be used in NLME, this version just use default function. The future version will provide buildin methods such as "lsoda", "lsode", "lsodes", "lsodar", "vode", "daspk", "euler", "rk4", "ode23", "ode45", "radau", "bdf", "bdf_d", "adams", "impAdams", "impAdams_d".*

Description

Generate functions for NLME Solves using a system of ordinary differential equations; a wrapper around the implemented ODE solvers

Usage

```
nlmODE(model, data, LogParms = TRUE, JAC = FALSE, SEQ = FALSE,rtol = 1e-4, atol = 1e-4, tcrit = NULL, h
```

Arguments

model	either an R-function that computes the values of the derivatives in the ODE system (the *model definition*) at time t.The return value of model should be a list. See package "nlmeODE" for more details.
data	nlme GroupedData format.
LogParms	transform parameters into log scale
JAC	A JAC set FALSE. This time we can implement this parts.
SEQ	A SEQ set FALSE.
rtol	relative error tolerance, either a scalar or an array as long as y. See details.
atol	absolute error tolerance, either a scalar or an array as long as y. See details.
tcrit	if not NULL, then lsoda cannot integrate past tcrit. The FORTRAN routine lsoda overshoots its targets (times points in the vector times), and interpolates values for the desired time points. If there is a time beyond which integration should not proceed (perhaps because of a singularity), that should be provided in tcrit.
hmin	an optional minimum value of the integration stepsize. In special situations this parameter may speed up computations with the cost of precision. Don't use hmin if you don't know why!
hmax	an optional maximum value of the integration stepsize. If not specified, hmax is set to the largest difference in times, to avoid that the simulation possibly ignores short-term events. If 0, no maximal size is specified.

Examples

```
######################################################################
#general model from nlmeODE package
#nlmODE SOLVER, USE ACCORDING FUNCTIONS
######################################################################
```

---

nlmODEcpcp *General Solver for Ordinary Differential Equations and Generate functions to be used in NLME, this version just use default function. The future version will provide buildin methods such as "lsoda", "lsode", "lsodes", "lsodar", "vode", "daspk", "euler", "rk4", "ode23", "ode45", "radau", "bdf", "bdf_d", "adams", "impAdams", "impAdams_d".*

Description

Generate functions for NLME Solves using a system of ordinary differential equations; a wrapper around the implemented ODE solvers

Usage

```
nlmODEcp(model, data, LogParms = TRUE, JAC = FALSE, SEQ = FALSE,rtol = 1e-4, atol = 1e-4, tcrit = NULL,
```

Arguments

model	either an R-function that computes the values of the derivatives in the ODE system (the *model definition*) at time t.The return value of model should be a list. See package "nlmeODE" for more details.
data	nlme GroupedData format.
LogParms	transform parameters into log scale
JAC	A JAC set FALSE. This time we can implement this parts.
SEQ	A SEQ set FALSE.
rtol	relative error tolerance, either a scalar or an array as long as y. See details.
atol	absolute error tolerance, either a scalar or an array as long as y. See details.
tcrit	if not NULL, then lsoda cannot integrate past tcrit. The FORTRAN routine lsoda overshoots its targets (times points in the vector times), and interpolates values for the desired time points. If there is a time beyond which integration should not proceed (perhaps because of a singularity), that should be provided in tcrit.
hmin	an optional minimum value of the integration stepsize. In special situations this parameter may speed up computations with the cost of precision. Don't use hmin if you don't know why!
hmax	an optional maximum value of the integration stepsize. If not specified, hmax is set to the largest difference in times, to avoid that the simulation possibly ignores short-term events. If 0, no maximal size is specified.

Examples

```
######################################################################
#general model from nlmeODE package
#nlmODEcp SOLVER, USE ACCORDING FUNCTIONS
######################################################################
```

---

nlmVODE *Solver for Ordinary Differential Equations (ODE) and Generate functions to be used in NLME*

Description

Generate functions for NLME Solves using a system of ordinary differential equations; a wrapper around the implemented ODE solvers,The R function vode provides an interface to the FORTRAN ODE solver of the same name, written by Peter N. Brown, Alan C. Hindmarsh and George D. Byrne.

Usage

```
nlmVODE(model, data, LogParms = TRUE, JAC = FALSE, SEQ = FALSE,rtol = 1e-4, atol = 1e-4, tcrit = NULL,
```

Arguments

model	either an R-function that computes the values of the derivatives in the ODE system (the *model definition*) at time t.The return value of model should be a list. See package "nlmeODE" for more details.
data	nlme GroupedData format.
LogParms	transform parameters into log scale
JAC	A JAC set FALSE. This time we can implement this parts.
SEQ	A SEQ set FALSE.
rtol	relative error tolerance, either a scalar or an array as long as y. See details.
atol	absolute error tolerance, either a scalar or an array as long as y. See details.
tcrit	if not NULL, then lsoda cannot integrate past tcrit. The FORTRAN routine lsoda overshoots its targets (times points in the vector times), and interpolates values for the desired time points. If there is a time beyond which integration should not proceed (perhaps because of a singularity), that should be provided in tcrit.
hmin	an optional minimum value of the integration stepsize. In special situations this parameter may speed up computations with the cost of precision. Don't use hmin if you don't know why!
hmax	an optional maximum value of the integration stepsize. If not specified, hmax is set to the largest difference in times, to avoid that the simulation possibly ignores short-term events. If 0, no maximal size is specified.

Examples

```
######################################################################
#general model from nlmeODE package
#nlmVODE SOLVER, USE ACCORDING FUNCTIONS
######################################################################
```

---

nlmVODEcp *Solver for Ordinary Differential Equations (ODE) and Generate functions to be used in NLME*

Description

Generate functions for NLME Solves using a system of ordinary differential equations; a wrapper around the implemented ODE solvers,The R function vode provides an interface to the FORTRAN ODE solver of the same name, written by Peter N. Brown, Alan C. Hindmarsh and George D. Byrne.

Usage

```
nlmVODEcp(model, data, LogParms = TRUE, JAC = FALSE, SEQ = FALSE,rtol = 1e-4, atol = 1e-4, tcrit = NULL
```

Arguments

model	either an R-function that computes the values of the derivatives in the ODE system (the *model definition*) at time t.The return value of model should be a list. See package "nlmeODE" for more details.
data	nlme GroupedData format.
LogParms	transform parameters into log scale
JAC	A JAC set FALSE. This time we can implement this parts.
SEQ	A SEQ set FALSE.
rtol	relative error tolerance, either a scalar or an array as long as y. See details.
atol	absolute error tolerance, either a scalar or an array as long as y. See details.
tcrit	if not NULL, then lsoda cannot integrate past tcrit. The FORTRAN routine lsoda overshoots its targets (times points in the vector times), and interpolates values for the desired time points. If there is a time beyond which integration should not proceed (perhaps because of a singularity), that should be provided in tcrit.
hmin	an optional minimum value of the integration stepsize. In special situations this parameter may speed up computations with the cost of precision. Don't use hmin if you don't know why!
hmax	an optional maximum value of the integration stepsize. If not specified, hmax is set to the largest difference in times, to avoid that the simulation possibly ignores short-term events. If 0, no maximal size is specified.

Examples

```
######################################################################
#general model from nlmeODE package
#nlmVODEcp SOLVER, USE ACCORDING FUNCTIONS
######################################################################
```

Index