

mrgsolve

Generated by Doxygen 1.8.13

Contents

1	Main Page	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Class Documentation	9
5.1	CompRec Struct Reference	9
5.1.1	Detailed Description	9
5.2	databox Struct Reference	10
5.3	dataobject Class Reference	11
5.4	datarecord Class Reference	12
5.4.1	Member Function Documentation	14
5.4.1.1	schedule()	14
5.5	odepack_dlsoda Class Reference	14
5.6	odeproblem Class Reference	16
5.6.1	Constructor & Destructor Documentation	18
5.6.1.1	~odeproblem()	18
5.6.2	Member Function Documentation	18
5.6.2.1	init_call()	18
5.6.2.2	init_call_record()	19
5.6.2.3	y_init()	19
5.7	resim Struct Reference	19
5.7.1	Detailed Description	20

6 File Documentation	21
6.1 inst/include/dataobject.h File Reference	21
6.2 inst/include/datarecord.h File Reference	21
6.3 inst/include/mrgsolv.h File Reference	22
6.4 inst/include/mrgsolve.h File Reference	23
6.4.1 Function Documentation	23
6.4.1.1 digits()	23
6.4.1.2 find_position()	24
6.4.1.3 MVGAUSS()	24
6.5 inst/include/odepack_dlsoda.h File Reference	24
6.6 inst/include/odeproblem.h File Reference	25
6.6.1 Function Documentation	26
6.6.1.1 as_config_func()	26
6.6.1.2 as_deriv_func()	26
6.6.1.3 as_init_func()	27
6.6.1.4 as_table_func()	27
6.6.1.5 PolyExp()	27
6.7 inst/include/tofunptr.h File Reference	27
6.8 inst/mrgx/mrgx.h File Reference	28
6.8.1 Function Documentation	28
6.8.1.1 get() [1/3]	28
6.8.1.2 get() [2/3]	29
6.8.1.3 get() [3/3]	29
6.8.1.4 get_envir()	29
6.8.1.5 mt_fun()	30
6.8.1.6 readRDS()	30
6.8.1.7 rlognorm()	30
6.8.1.8 rnorm()	31
6.9 src/dataobject.cpp File Reference	31
6.10 src/datarecord.cpp File Reference	32

6.11	src/devtran.cpp File Reference	32
6.11.1	Function Documentation	33
6.11.1.1	DEVTRAN()	33
6.12	src/mrgsolve.cpp File Reference	33
6.12.1	Function Documentation	34
6.12.1.1	digits()	34
6.12.1.2	find_position()	34
6.12.1.3	MVGAUSS()	35
6.13	src/odepack_dlsoda.cpp File Reference	35
6.14	src/odeproblem.cpp File Reference	35
6.14.1	Function Documentation	36
6.14.1.1	as_config_func()	36
6.14.1.2	as_deriv_func()	36
6.14.1.3	as_init_func()	37
6.14.1.4	as_table_func()	37
6.14.1.5	main_derivs()	37
6.14.1.6	PolyExp()	38
6.14.1.7	TOUCH_FUNS()	38
6.15	src/quick.cpp File Reference	38

Chapter 1

Main Page

Documentation for mrgsolve C++ code.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

CompRec	9
databox	10
dataobject	11
datarecord	12
odepack_dlsoda	14
odeproblem	16
resim	19

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

CompRec		
	Functor for sorting data records in <code>reclist</code>	9
databox	10
dataobject	11
datarecord	12
odepack_dlsoda	14
odeproblem	16
resim		
	Resim functor	19

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

inst/include/ dataobject.h	21
inst/include/ datarecord.h	21
inst/include/ mrgsolv.h	22
inst/include/ mrgsolve.h	23
inst/include/ odepack_dlsoda.h	24
inst/include/ odeproblem.h	25
inst/include/ RcppInclude.h	??
inst/include/ tofunptr.h	27
inst/mrgx/ mrgx.h	28
src/ dataobject.cpp	31
src/ datarecord.cpp	32
src/ devtran.cpp	32
src/ mrgsolve.cpp	33
src/ odepack_dlsoda.cpp	35
src/ odeproblem.cpp	35
src/ quick.cpp	38

Chapter 5

Class Documentation

5.1 CompRec Struct Reference

Functor for sorting data records in `reclist`.

```
#include <datarecord.h>
```

Public Member Functions

- `bool operator()` (`const rec_ptr &a`, `const rec_ptr &b`)

5.1.1 Detailed Description

Functor for sorting data records in `reclist`.

Records are first sorted by time, then by position.

Parameters

<i>a</i>	first record
<i>b</i>	second record

Returns

boolean

The documentation for this struct was generated from the following file:

- `inst/include/datarecord.h`

5.2 databox Struct Reference

Public Attributes

- [dvec ETA](#)
vector of ETA values
- [dvec EPS](#)
vector of EPS values
- unsigned int [newind](#)
new individual flag
- double [time](#)
current simulation time
- int [evid](#)
event ID flag
- bool [SYSTEMOFF](#)
flag to stop advancing system for current ID
- [dvec mtime](#)
model time values
- double [id](#)
current ID
- double [amt](#)
current dosing amount value
- short int [cmt](#)
current compartment value
- int [nid](#)
number of IDs in the data set
- int [idn](#)
current ID number
- int [nrow](#)
number of rows in output data set
- int [rown](#)
current output row number
- bool [CFONSTOP](#)
carry forward on stop indicator
- void * [envir](#)
model environment

The documentation for this struct was generated from the following file:

- inst/include/[odeproblem.h](#)

5.3 dataobject Class Reference

Public Member Functions

- [dataobject](#) (Rcpp::NumericMatrix _data, Rcpp::CharacterVector _parnames)
constructor
- [dataobject](#) (Rcpp::NumericMatrix _data, Rcpp::CharacterVector _parnames, Rcpp::CharacterVector _initnames)
constructor
- unsigned int **nrow** () const
- unsigned int **ncol** () const
- unsigned int **nid** () const
- unsigned int **idcol** () const
- int **start** (int i) const
- int **end** (int i) const
- void **map_uid** ()
- double **get_uid** (int i) const
- uidtype **return_uid** () const
- void **copy_parameters** (int this_row, [odeproblem](#) *prob)
- void **copy_inits** (int this_row, [odeproblem](#) *prob)
- void **reload_parameters** (const Rcpp::NumericVector ¶m, [odeproblem](#) *prob)
- void **idata_row** ()
- unsigned int **get_idata_row** (const double ID)
- void **locate_tran** ()
- void **get_records** ([recstack](#) &a, int NID, int neq, unsigned int &obscount, unsigned int &evcount, bool obsonly, bool debug)
- void **check_idcol** ([dataobject](#) &data)
- double **get_value** (const int row, const int col) const
- double **get_id_value** (const int row) const
- void **get_ids** (uidtype *ids)
- Rcpp::IntegerVector **get_col_n** (const Rcpp::CharacterVector &what)
- void **carry_out** (const [recstack](#) &a, Rcpp::NumericMatrix &ans, [dataobject](#) &idat, const Rcpp::IntegerVector &data_carry, const unsigned int data_carry_start, const Rcpp::IntegerVector &idata_carry, const unsigned int idata_carry_start)

Public Attributes

- Rcpp::NumericMatrix **Data**

Protected Attributes

- uidtype **Uid**
unique IDs in the data set
- datarowtype **Startrow**
start row for each ID
- datarowtype **Endrow**
data set end row for each ID
- int **Idcol**
which column holds ID value
- Rcpp::CharacterVector **Data_names**
- std::vector< unsigned int > **col**

- `Rcpp::IntegerVector` [par_from](#)
index for parameters in data set
- `Rcpp::IntegerVector` [par_to](#)
index for parameters in param list
- `Rcpp::CharacterVector` [parnames](#)
names of model parameters
- `idat_map` [idmap](#)
map to get
- `Rcpp::IntegerVector` [cmt_from](#)
index for compartments in data set
- `Rcpp::IntegerVector` [cmt_to](#)
index for compartments in init list
- `Rcpp::CharacterVector` [cmtnames](#)
names of model compartments

The documentation for this class was generated from the following files:

- `inst/include/dataobject.h`
- `src/dataobject.cpp`

5.4 datarecord Class Reference

Public Member Functions

- [datarecord](#) (double time_, int pos_, bool output_)
constructor
- [datarecord](#) (double time_, short int cmt_, int pos_, double id_)
constructor
- [datarecord](#) (short int cmt_, int evid_, double amt_, double time_, double rate_, int pos_, double id_)
constructor
- [datarecord](#) (short int cmt_, int evid_, double amt_, double time_, double rate_)
short event constructor
- double **time** ()
- void **time** (double time_)
- double **id** ()
- void **id** (double id_)
- unsigned int **evid** ()
- void **evid** (unsigned short int evid_)
- int **pos** ()
- void **pos** (int pos_)
- short int **cmt** ()
- void **output** (bool in)
- bool **output** ()
- bool **from_data** ()
- void **from_data** (bool val)
- double **amt** ()
- double **rate** ()
- void **rate** (double value)
- double **dur** (double b)
- void **addl** (unsigned int addl_)

- unsigned int **addl** ()
- void **ss** (unsigned short int ss_)
- unsigned short **ss** ()
- void **ii** (double ii_)
- double **ii** ()
- void **fn** (double value)
- double **fn** ()
- void **schedule** (std::vector< rec_ptr > &thisi, double maxtime, bool put_ev_first)
- void **implement** (odeproblem *prob)
- void **steady_infusion** (odeproblem *prob)
- void **steady_bolus** (odeproblem *prob)
- bool **infusion** ()
- bool **is_event** ()
- bool **needs_sorting** ()
- bool **unarmed** ()
- void **arm** ()
- void **unarm** ()
- void **phantom_rec** ()

Protected Attributes

- double **Time**
record time
- double **Id**
record ID value
- int **Pos**
record position number
- unsigned short int **Evid**
record event ID
- bool **Output**
should this record be included in output?
- bool **Fromdata**
is this record from the original data set?
- short int **Cmt**
record compartment number
- unsigned int **Addl**
number of additional doses
- unsigned short int **Ss**
record steady-state indicator
- double **Amt**
record dosing amount value
- double **Rate**
record infusion rate value
- double **li**
record inter-dose interval value
- double **Fn**
record bioavailability value
- bool **Armed**
only armed records are actually executed

5.4.1 Member Function Documentation

5.4.1.1 `schedule()`

```
void datarecord::schedule (
    std::vector< rec_ptr > & thisi,
    double maxtime,
    bool put_ev_first )
```

Schedule out doses. If the dose was an infusion, schedule the off infusion event. If the dose included additional doses, create those events and add them to the stack. No doses will be scheduled beyond the maximum time for that individual.

Parameters

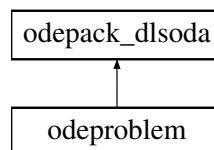
<i>thisi</i>	the record stack for this individual
<i>maxtime</i>	the last time already in the record for the individual
<i>put_ev_first</i>	logical; if true, the position of the event is -600; otherwise, it is beyond the last record of the stack. But records are always sorted first by time, then by position.

The documentation for this class was generated from the following files:

- [inst/include/datarecord.h](#)
- [src/datarecord.cpp](#)

5.5 `odepack_dlsoda` Class Reference

Inheritance diagram for `odepack_dlsoda`:



Public Member Functions

- **`odepack_dlsoda`** (int npar_, int neq_)
- void **`hmax`** (double value)
- void **`hmin`** (double value)
- void **`ixpr`** (int value)
- void **`maxsteps`** (int value)
- void **`mxhnil`** (int value)
- int **`istate`** ()
- void **`istate`** (int value)
- void **`lsoda_init`** ()

- int **itask** ()
- void **itask** (int itask)
- void **tol** (double atol, double rtol)
- double * **rwork** ()
- void **rwork** (int pos, double value)
- int * **iwork** ()
- void **iwork** (int pos, int value)
- void **tcrit** (double value)
- double * **y** ()
- void **y** (const int pos, const double value)
- double **y** (const int pos)
- double * **ydot** ()
- int **npar** ()
- int **neq** ()

Protected Attributes

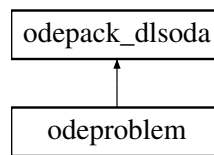
- int [xliwork](#)
length of iwork array
- int [xlrwork](#)
length of rwork array
- int [xistate](#)
istate value
- int [xitask](#)
itask value
- int [xiopt](#)
iopt value
- int [xitol](#)
itol value
- int [Neq](#)
number of state variables
- int [Npar](#)
number of model parameters
- int [xjt](#)
jacobian indicator
- double [xatol](#)
absolute tolerance
- double [xrtol](#)
relative tolerance
- double * [xrwork](#)
rwork array
- int * [xiwork](#)
iwork array
- double * [Y](#)
current value of state variables
- double * [Ydot](#)
current value of ODEs

The documentation for this class was generated from the following files:

- inst/include/odepack_dlsoda.h
- src/odepack_dlsoda.cpp

5.6 odeproblem Class Reference

Inheritance diagram for odeproblem:



Public Member Functions

- **odeproblem** (Rcpp::NumericVector param, Rcpp::NumericVector init, Rcpp::List funs, int n_capture_)
- virtual [~odeproblem](#) ()
Destructor for odeproblem object.
- void **advance** (double tfrom, double tto)
- void **call_derivs** (int *neq, double *t, double *y, double *ydot)
- void **init** (int pos, double value)
- double **init** (int pos)
- void **init_call** (const double &time)
- void **init_call_record** (const double &time)
- void **y_init** (int pos, double value)
- void **y_init** (Rcpp::NumericVector x)
- void **y_add** (const unsigned int pos, const double &value)
add value to compartment pos
- void **table_call** ()
Call \$TABLE function.
- void **table_init_call** ()
- void **config_call** ()
Call \$PREAMBLE function.
- void **set_d** (rec_ptr this_rec)
- void **omega** (Rcpp::NumericMatrix &x)
- void **sigma** (Rcpp::NumericMatrix &x)
- arma::mat **mv_omega** (int n)
- arma::mat **mv_sigma** (int n)
- void **pass_envir** (Rcpp::Environment *x)
- bool **CFONSTOP** ()
- const double * **param** () const
- void **param** (int pos, double value)
- void **rate** (unsigned int pos, double value)
- double **rate** (unsigned int pos)
- void **rate0** (unsigned int pos, double value)
- double **rate0** (unsigned int pos)
- int **rate_count** (unsigned int pos)
- void **rate_add** (unsigned int pos, const double &value)
- void **rate_rm** (unsigned int pos, const double &value)
- void **rate_bump** (const unsigned int pos, const double &value)
- void **rate_reset** ()
Reset all infusion rates.
- void **dur** (unsigned int pos, double value)
- double **dur** (unsigned int pos)
- void **fbio** (unsigned int pos, double value)

- double **fbio** (unsigned int pos)
- double **alag** (int cmt)
- void **reset_newid** (const double id_)
Reset odeproblem object for new individual.
- void **eta** (int pos, double value)
- void **eps** (int pos, double value)
- bool **systemoff** ()
- void **on** (unsigned short int cmt)
- void **off** (unsigned short int cmt)
- int **is_on** (unsigned int eq_n)
- void **time** (double time_)
- void **newind** (unsigned int newind_)
- unsigned int **newind** ()
- void **advan** (int x)
- int **advan** ()
- void **advan2** (const double &tfrom, const double &tto)
- void **advan4** (const double &tfrom, const double &tto)
- void **neta** (int n)
set number of ETAs in the model
- void **neps** (int n)
set number of EPSs in the model
- void **nid** (int n)
- void **nrow** (int n)
- void **idn** (int n)
- void **rown** (int n)
- **dvec** & **mtime** ()
- **dvec** & **get_capture** ()
- double **capture** (int i)
- void **copy_parin** (const Rcpp::List &parin)
- void **copy_funs** (const Rcpp::List &funs)

Protected Attributes

- double * **Param**
model parameters
- **dvec** **R0**
acutal current infusion rate
- std::vector< unsigned int > **infusion_count**
number of active infusions
- **dvec** **R**
receive user input for infusion rate
- **dvec** **D**
receive user input for infusion duration
- **dvec** **Init_value**
initial conditions
- **dvec** **Init_dummy**
initial conditions for user input
- **dvec** **F**
bioavailability
- **dvec** **Alag**
dosing lag time
- **deriv_func** * **Derivs**

- *\$ODE function*
- [init_func](#) * [Inits](#)
- *\$MAIN function*
- [table_func](#) * [Table](#)
- *\$TABLE function*
- [config_func](#) * [Config](#)
- *\$PREAMBLE function*
- `std::vector< int >` [On](#)
- *compartment on/off indicator*
- [databox d](#)
- *various data passed to model functions*
- `int` [Advan](#)
- *simulation mode: 1/2/3/4 (PK models) or 13 (odes)*
- [dvec a](#)
- *used for advan 1/2/3/4 calculations*
- [dvec alpha](#)
- *used for advan 1/2/3/4 calculation*
- [resim simeta](#)
- *functor for resimulating etas*
- [resim simeps](#)
- *functor for resimulating epsilons*
- `arma::mat` [Omega](#)
- *variance/covariance matrix for between-subject variability*
- `arma::mat` [Sigma](#)
- *variance/covariance matrix for within-subject variability*
- [dvec pred](#)
- *brings clearances, volumes, and kas for advan 1/2/3/4 calculations*
- [dvec Capture](#)
- *captured data items*

5.6.1 Constructor & Destructor Documentation

5.6.1.1 `~odeproblem()`

```
odeproblem::~odeproblem ( ) [virtual]
```

Destructor for odeproblem object.

Upon object construction, odeproblem dynamically allocates the Param array.

5.6.2 Member Function Documentation

5.6.2.1 `init_call()`

```
void odeproblem::init_call (
    const double & time )
```

Call \$MAIN to get the initial conditions.

Parameters

<i>time</i>	the time to assume for the calculation
-------------	--

5.6.2.2 init_call_record()

```
void odeproblem::init_call_record (
    const double & time )
```

Call \$MAIN with the dummy initial condition vector.

Parameters

<i>time</i>	the time to assume when making the call.
-------------	--

5.6.2.3 y_init()

```
void odeproblem::y_init (
    int pos,
    double value )
```

Assigns values to both the compartment and the vector of initial conditions.

Parameters

<i>pos</i>	the compartment number (C++ indexing)
<i>value</i>	the value for the compartment

The documentation for this class was generated from the following files:

- inst/include/[odeproblem.h](#)
- src/[odeproblem.cpp](#)

5.7 resim Struct Reference

Resim functor.

```
#include <mrgsolv.h>
```

Public Member Functions

- [resim](#) (refun *x, void *y)
resim constructor
- void **operator()** ()

Protected Attributes

- `refun * fun`
function to call to re-simulate
- `void * prob`
object to pass to re-simulated function

5.7.1 Detailed Description

Resim functor.

These functors are used to re-simulate `ETA` and `EPS` values.

The documentation for this struct was generated from the following file:

- `inst/include/mrgsolv.h`

Chapter 6

File Documentation

6.1 inst/include/dataobject.h File Reference

```
#include <vector>
#include <boost/shared_ptr.hpp>
#include <boost/make_shared.hpp>
#include "odeproblem.h"
#include "RcppInclude.h"
```

Classes

- class [dataobject](#)

Typedefs

- typedef std::map< double, int > **idat_map**
- typedef std::deque< double > **uidtype**
- typedef std::deque< int > **datarowtype**

6.2 inst/include/datarecord.h File Reference

```
#include <boost/shared_ptr.hpp>
#include "mrgsolv.h"
```

Classes

- class [datarecord](#)
- struct [CompRec](#)

Functor for sorting data records in reclist.

Typedefs

- typedef boost::shared_ptr< [datarecord](#) > **rec_ptr**
- typedef std::vector< **rec_ptr** > **reclist**

Functions

- void **add_mtime** ([reclist](#) &thisi, [dvec](#) &b, [dvec](#) &c, bool debug)
- bool **CompByTimePosRec** (const **rec_ptr** &a, const **rec_ptr** &b)

6.3 inst/include/mrgsolv.h File Reference

```
#include <vector>
#include <map>
#include <string>
```

Classes

- struct [resim](#)
Resim functor.

Macros

- #define [MRGSOLVE_INIT_SIGNATURE](#) [dvec](#)& _A_0_,const double* _A_, const double* _THETA_, [dvec](#)& _F_, [dvec](#)& _ALAG_, [dvec](#)& _R_, [dvec](#)& _D_, [databox](#)& self, [dvec](#)& _pred_, [resim](#)& simeta
signature for \$MAIN
- #define [MRGSOLVE_TABLE_SIGNATURE](#) const double* _A_, const [dvec](#)& _A_0_, const double* _THETA_, const [dvec](#)& _F_, const [dvec](#)& _R_, [databox](#)& self, const [dvec](#)& _pred_, [dvec](#)& _capture_, [resim](#)& simeps
signature for \$TABLE
- #define [MRGSOLVE_ODE_SIGNATURE](#) const double* _ODETIME_, const double* _A_, double* _DAD_, const [dvec](#)& _A_0_, const double* _THETA_
signature for \$ODE
- #define [MRGSOLVE_CONFIG_SIGNATURE](#) [databox](#)& self, const double* _THETA_, const double neq, const double npar
signature for \$PREAMBLE

Typedefs

- typedef std::vector< double > [dvec](#)
vector of doubles
- typedef std::vector< std::string > [svec](#)
vector of strings
- typedef std::vector< int > [ivec](#)
vector of integers
- typedef void **refun**(void *)

6.4 inst/include/mrgsolve.h File Reference

```
#include "RcppInclude.h"
#include <R_ext/Rdynload.h>
```

Typedefs

- typedef std::map< std::string, int > [si_map](#)
map key: string, value: integer
- typedef std::map< std::string, double > [sd_map](#)
map key: string, value: double
- typedef std::vector< std::string > [svec](#)
vector of strings
- typedef std::vector< int > [ivec](#)
vector of integers
- typedef std::map< std::string, [ivec](#) > [sivec_map](#)
map key: string, value: integer vector

Functions

- void **neg_istate** (int)
- DL_FUNC **tofun** (SEXP a)
- arma::mat [MVGAUSS](#) (Rcpp::NumericMatrix &OMEGA_, int n)
- arma::mat **MVGAUSS** (arma::mat &OMEGA_, int n)
- Rcpp::List **SIMRE** (int n1, Rcpp::NumericMatrix &OMEGA, int n2, Rcpp::NumericMatrix &SIGMA, int seed)
- template<class T >
void **sort_unique** (T &a)
- int [find_position](#) (const Rcpp::CharacterVector &what, const Rcpp::CharacterVector &table)
- double [digits](#) (const double &a, const double &b)
- void **decorr** (const Rcpp::NumericMatrix &x)
- Rcpp::NumericMatrix **SUPERMATRIX** (const Rcpp::List &a)
- void **from_to** (const Rcpp::CharacterVector &a, const Rcpp::CharacterVector &b, Rcpp::IntegerVector &ai, Rcpp::IntegerVector &bi)
- Rcpp::List **get_tokens** (const Rcpp::CharacterVector &code)
- void **set_omega** (SEXP loc, Rcpp::NumericMatrix &omega_)
- Rcpp::NumericMatrix **EXPAND_EVENTS** (const Rcpp::IntegerVector &idcol_, const Rcpp::NumericMatrix &events, const Rcpp::NumericVector &id)
- Rcpp::NumericMatrix **recdata** (Rcpp::NumericMatrix &dose, Rcpp::NumericMatrix &obs, Rcpp::IntegerVector &cols, const int n_out_col, const int n_out_row, const Rcpp::NumericVector &addl_, const Rcpp::NumericVector &ii_, const int nid, const int ntime, const int namt, const int nevid, const int ncmt, const int nrate)

6.4.1 Function Documentation

6.4.1.1 digits()

```
double digits (
    const double & a,
    const double & b )
```

Limit a number to a specific number of significant digits.

Parameters

<i>a</i>	the number to limit
<i>b</i>	the number of digits

6.4.1.2 find_position()

```
int find_position (
    const Rcpp::CharacterVector & what,
    const Rcpp::CharacterVector & table )
```

Find the position of a string in a character vector.

Parameters

<i>what</i>	the string to look for
<i>table</i>	where to look for the string

Returns

the position of the string with 0-based indexing if the string is found; -1 otherwise

6.4.1.3 MVGAUSS()

```
arma::mat MVGAUSS (
    Rcpp::NumericMatrix & OMEGA_,
    int n )
```

Simulate from a multivariate normal distribution with mean 0.

Parameters

<i>OMEGA_</i> <i>A_</i>	the covariance matrix
<i>n</i>	the number of variates to simulate

Returns

matrix of simulated variates

6.5 inst/include/odepack_dlsoda.h File Reference

```
#include <math.h>
```

Classes

- class [odepack_dlsoda](#)

6.6 inst/include/odeproblem.h File Reference

```
#include <math.h>
#include <vector>
#include "odepack_dlsoda.h"
#include "mrgsolv.h"
#include "RcppInclude.h"
#include "datarecord.h"
```

Classes

- struct [databox](#)
- class [odeproblem](#)

Macros

- `#define MRGSOLVE_GET_PRED_CL (pred[0])`
map CL to pred position 0 for \$PKMODEL
- `#define MRGSOLVE_GET_PRED_VC (pred[1])`
map VC to pred position 1 for \$PKMODEL
- `#define MRGSOLVE_GET_PRED_KA (pred[2])`
map KA to pred position 2 for \$PKMODEL
- `#define MRGSOLVE_GET_PRED_Q (pred[3])`
map Q to pred position 3 for \$PKMODEL
- `#define MRGSOLVE_GET_PRED_VP (pred[4])`
map VP to pred position 4 for \$PKMODEL
- `#define MRGSOLVE_GET_PRED_K10 (pred[0]/pred[1])`
rate constants for \$PKMODEL
- `#define MRGSOLVE_GET_PRED_K12 (pred[3]/pred[1])`
rate constants for \$PKMODEL
- `#define MRGSOLVE_GET_PRED_K21 (pred[3]/pred[4])`
rate constants for \$PKMODEL

Typedefs

- `typedef std::vector< rec_ptr > reclist`
vector of [datarecord](#) objects for one ID
- `typedef std::vector< reclist > recstack`
vector of [reclist](#) vectors comprising a data set
- `typedef void init_func(MRGSOLVE_INIT_SIGNATURE)`
\$MAIN function
- `typedef void table_func(MRGSOLVE_TABLE_SIGNATURE)`
\$TABLE function
- `typedef void deriv_func(MRGSOLVE_ODE_SIGNATURE)`
\$ODE function
- `typedef void config_func(MRGSOLVE_CONFIG_SIGNATURE)`
\$PREAMBLE function
- `typedef void main_deriv_func(int *neq, double *t, double *y, double *ydot, odeproblem *prob)`
function to hand off to DLSODA

Functions

- `deriv_func` * `as_deriv_func` (SEXP `derivs`)
- `init_func` * `as_init_func` (SEXP `inits`)
- `table_func` * `as_table_func` (SEXP `table`)
- `config_func` * `as_config_func` (SEXP `config`)
- DL_FUNC **tofunptr** (SEXP `a`)
- void **neg_istate** (int `istate`)
- template<typename T , typename type2 >
void **tofunptr** (T `b`, type2 `a`)
- void **dosimeta** (void *)
- void **dosimeps** (void *)
- double **PolyExp** (const double &`x`, const double &`dose`, const double &`rate`, const double &`xinf`, const double &`tau`, const bool `ss`, const **dvec** &`a`, const **dvec** &`alpha`, const int `n`)
- Rcpp::List **TOUCH_FUNS** (const Rcpp::NumericVector &`lparam`, const Rcpp::NumericVector &`linit`, const Rcpp::CharacterVector &`capture`, const Rcpp::List &`funcs`)

Variables

- `main_deriv_func` **main_derivs**

6.6.1 Function Documentation

6.6.1.1 `as_config_func()`

```
config_func* as_config_func (
    SEXP config )
```

Get pointer for \$PREAMBLE function.

Parameters

<i>config</i>	address for \$PREAMBLE function
---------------	---------------------------------

6.6.1.2 `as_deriv_func()`

```
deriv_func* as_deriv_func (
    SEXP derivs )
```

Get pointer for \$ODE function.

Parameters

<i>derivs</i>	address for \$ODE function
---------------	----------------------------

6.6.1.3 as_init_func()

```
init_func* as_init_func (
    SEXP inits )
```

Get pointer for \$MAIN function.

Parameters

<i>inits</i>	address for \$MAIN function
--------------	-----------------------------

6.6.1.4 as_table_func()

```
table_func* as_table_func (
    SEXP table )
```

Get pointer for \$TABLE function.

Parameters

<i>table</i>	address for \$TABLE function
--------------	------------------------------

6.6.1.5 PolyExp()

```
double PolyExp (
    const double & x,
    const double & dose,
    const double & rate,
    const double & xinf,
    const double & tau,
    const bool ss,
    const dvec & a,
    const dvec & alpha,
    const int n )
```

Calculate PK model polyexponentials.

6.7 inst/include/tofunptr.h File Reference

```
#include <Rinternals.h>
```

Typedefs

- typedef void *(* **DL_FUNC**) ()

Functions

- **DL_FUNC tofunptr** (SEXP a)

6.8 inst/mrgx/mrgx.h File Reference

```
#include "modelheader.h"
```

Functions

- Rcpp::Environment **mrgx::get_envir** (databox &self)
Return the model environment.
- double **mrgx::rnorm** (const double mean, const double sd, const double lower, const double upper)
- double **mrgx::rlognorm** (const double mean, const double sd, const double lower, const double upper)
- template<typename T >
T **mrgx::get** (const std::string name, const databox &self)
- template<typename T >
T **mrgx::get** (const std::string name)
- template<typename T >
T **mrgx::get** (const std::string package, const std::string name)
- template<typename T >
T **mrgx::readRDS** (const std::string filename)
- Rcpp::Function **mrgx::mt_fun** ()

6.8.1 Function Documentation

6.8.1.1 **get()** [1/3]

```
template<typename T >
T mrgx::get (
    const std::string name,
    const databox & self )
```

Get an R object from the model environment.

Parameters

<i>name</i>	name of the R object to get
<i>self</i>	the model data object

Returns

an object from the model environment

6.8.1.2 get() [2/3]

```
template<typename T >
T mrgx::get (
    const std::string name )
```

Get an R object from the global environment.

Parameters

<i>name</i>	name of the R object to get
-------------	-----------------------------

Returns

an object from the global environment

6.8.1.3 get() [3/3]

```
template<typename T >
T mrgx::get (
    const std::string package,
    const std::string name )
```

Get an R object from a package namespace. This is typically used to get a function from a specific package.

Parameters

<i>package</i>	name of the package
<i>name</i>	name of the object to get

Returns

an object from the package namespace

6.8.1.4 get_envir()

```
Rcpp::Environment mrgx::get_envir (
    databox & self )
```

Return the model environment.

With each mrgsolve model object, there is an R environment that can be used to maintain arbitrary R objects, potentially for use in the model.

Parameters

<i>self</i>	the model databox object
-------------	--------------------------

Returns

the model environment

6.8.1.5 mt_fun()

```
Rcpp::Function mrgx::mt_fun ( )
```

An empty R function. This is typically used as a placeholder when declaring an `Rcpp::Function` object.

Returns

the function `mt_fun` from the mrgsolve namespace

6.8.1.6 readRDS()

```
template<typename T >
T mrgx::readRDS (
    const std::string filename )
```

Read an RDS file.

Parameters

<i>filename</i>	the name of the RDS file to read
-----------------	----------------------------------

Returns

an object saved in the RDS file

6.8.1.7 rlognorm()

```
double mrgx::rlognorm (
    const double mean,
```

```

    const double sd,
    const double lower,
    const double upper )

```

Simulate random lognormal variate between lower and upper boundaries. An error is generated if a variate between lower and upper bounds cannot be generated in 50 tries.

Parameters

<i>mean</i>	normal distribution mean
<i>sd</i>	normal distribution standard deviation
<i>lower</i>	lower bound for variates
<i>upper</i>	upper bound for variates

Returns

the simulated variate

6.8.1.8 rnorm()

```

double mrgx::rnorm (
    const double mean,
    const double sd,
    const double lower,
    const double upper )

```

Simulate random normal variate between lower and upper boundaries. An error is generated if a variate between lower and upper bounds cannot be generated in 50 tries.

Parameters

<i>mean</i>	normal distribution mean
<i>sd</i>	normal distribution standard deviation
<i>lower</i>	lower bound for variates
<i>upper</i>	upper bound for variates

Returns

the simulated variate

6.9 src/dataobject.cpp File Reference

```

#include "RcppInclude.h"
#include "dataobject.h"
#include "mrgsolve.h"
#include "mrgsolv.h"

```

Macros

- `#define _COL_amt_ 0u`
- `#define _COL_ii_ 1u`
- `#define _COL_addl_ 2u`
- `#define _COL_ss_ 3u`
- `#define _COL_rate_ 4u`
- `#define _COL_evid_ 5u`
- `#define _COL_cmt_ 6u`
- `#define _COL_time_ 7u`

6.10 src/datarecord.cpp File Reference

```
#include "RcppInclude.h"
#include "datarecord.h"
#include "odeproblem.h"
#include <boost/make_shared.hpp>
#include <functional>
#include <algorithm>
```

Macros

- `#define N_SS 1000`
- `#define CRIT_DIFF_SS 1E-10`

Functions

- `bool CompByTimePosRec (const rec_ptr &a, const rec_ptr &b)`
- `void add_mtime (reclist &thisi, dvec &b, dvec &c, bool debug)`

6.11 src/devtran.cpp File Reference

```
#include <boost/shared_ptr.hpp>
#include <boost/pointer_cast.hpp>
#include <string>
#include "mrgsolve.h"
#include "odeproblem.h"
#include "dataobject.h"
#include "RcppInclude.h"
```

Macros

- `#define CRUMP(a) Rcpp::stop(a)`
- `#define REP(a) Rcpp::Rcout << #a << std::endl;`
- `#define nREP(a) Rcpp::Rcout << a << std::endl;`
- `#define say(a) Rcpp::Rcout << a << std::endl;`
- `#define __ALAG_POS -1200`

Functions

- Rcpp::List **DEVTRAN** (const Rcpp::List *parin*, const Rcpp::NumericVector &*inpar*, const Rcpp::CharacterVector &*parnames*, const Rcpp::NumericVector &*init*, Rcpp::CharacterVector &*cmtnames*, const Rcpp::IntegerVector &*capture*, const Rcpp::List &*funcs*, const Rcpp::NumericMatrix &*data*, const Rcpp::NumericMatrix &*idata*, Rcpp::NumericMatrix &*OMEGA*, Rcpp::NumericMatrix &*SIGMA*, Rcpp::Environment *envir*)

6.11.1 Function Documentation

6.11.1.1 DEVTRAN()

```
Rcpp::List DEVTRAN (
    const Rcpp::List parin,
    const Rcpp::NumericVector & inpar,
    const Rcpp::CharacterVector & parnames,
    const Rcpp::NumericVector & init,
    Rcpp::CharacterVector & cmtnames,
    const Rcpp::IntegerVector & capture,
    const Rcpp::List & funcs,
    const Rcpp::NumericMatrix & data,
    const Rcpp::NumericMatrix & idata,
    Rcpp::NumericMatrix & OMEGA,
    Rcpp::NumericMatrix & SIGMA,
    Rcpp::Environment envir )
```

Perform a simulation run.

Parameters

<i>parin</i>	list of data and options for the simulation
<i>inpar</i>	numeric parameter values
<i>parnames</i>	parameter names
<i>init</i>	numeric initial values
<i>cmtnames</i>	compartment names
<i>capture</i>	indices in capture vector to actually get
<i>funcs</i>	list of pointer addresses to model functions generated by getNativeSymbolInfo()
<i>data</i>	the main data set
<i>idata</i>	the idata data aset
<i>OMEGA</i>	between-ID normal random effects
<i>SIGMA</i>	within-ID normal random effects

Returns

list containing matrix of simulated data and a character vector of tran names that may have been carried into the output

6.12 src/mrgsolve.cpp File Reference

```
#include "RcppInclude.h"
```

```
#include "mrgsolve.h"
#include <vector>
#include <string>
#include "boost/tokenizer.hpp"
```

Functions

- double **digits** (const double &a, const double &b)
- int **find_position** (const Rcpp::CharacterVector &what, const Rcpp::CharacterVector &table)
- void **neg_istate** (int istate)
- arma::mat **MVGAUSS** (Rcpp::NumericMatrix &OMEGA_, int n)
- arma::mat **MVGAUSS** (arma::mat &OMEGA, int n)
- void **decorr** (Rcpp::NumericMatrix &x)
- Rcpp::NumericMatrix **ZERO** (Rcpp::NumericMatrix &x)
- Rcpp::NumericMatrix **SUPERMATRIX** (const Rcpp::List &a, bool keep_names)
- Rcpp::List **get_tokens** (const Rcpp::CharacterVector &code)
- void **from_to** (const Rcpp::CharacterVector &a, const Rcpp::CharacterVector &b, Rcpp::IntegerVector &ai, Rcpp::IntegerVector &bi)
- Rcpp::NumericMatrix **EXPAND_EVENTS** (const Rcpp::IntegerVector &idcol_, const Rcpp::NumericMatrix &events, const Rcpp::NumericVector &id)

6.12.1 Function Documentation

6.12.1.1 digits()

```
double digits (
    const double & a,
    const double & b )
```

Limit a number to a specific number of significant digits.

Parameters

<i>a</i>	the number to limit
<i>b</i>	the number of digits

6.12.1.2 find_position()

```
int find_position (
    const Rcpp::CharacterVector & what,
    const Rcpp::CharacterVector & table )
```

Find the position of a string in a character vector.

Parameters

<i>what</i>	the string to look for
<i>table</i>	where to look for the string

Returns

the position of the string with 0-based indexing if the string is found; -1 otherwise

6.12.1.3 MVGAUSS()

```
arma::mat MVGAUSS (
    Rcpp::NumericMatrix & OMEGA_,
    int n )
```

Simulate from a multivariate normal distribution with mean 0.

Parameters

$OMEGA_{A_}$	the covariance matrix
n	the number of variates to simulate

Returns

matrix of simulated variates

6.13 src/odepack_dlsoda.cpp File Reference

```
#include "odepack_dlsoda.h"
#include <algorithm>
```

6.14 src/odeproblem.cpp File Reference

```
#include <cmath>
#include <vector>
#include "RcppInclude.h"
#include "odeproblem.h"
#include "mrgsolve.h"
```

Macros

- `#define MARGSOLVE_MAX_SS_ITER 1000`
the maximum number of iterations for steady-state calculation

Functions

- void **dosimeta** (void *prob_)
- void **dosimeps** (void *prob_)
- void **main_derivs** (int *neq, double *t, double *y, double *ydot, **odeproblem** *prob)
- void F77_NAME() **dlsoda** (**main_deriv_func** *derivs, int *neq, double *y, const double *tfrom, const double *tto, int *itol, double *rtol, double *atol, int *itask, int *istate, int *iopt, double *rwork, int *lrwork, int *iwork, int *liwork, int *dum, int *jt, **odeproblem** *prob)
- double **PolyExp** (const double &x, const double &dose, const double &rate, const double &xinf, const double &tau, const bool ss, const **dvec** &a, const **dvec** &alpha, const int n)
- **init_func** * **as_init_func** (SEXP inits)
- **deriv_func** * **as_deriv_func** (SEXP derivs)
- **table_func** * **as_table_func** (SEXP table)
- **config_func** * **as_config_func** (SEXP config)
- Rcpp::List **TOUCH_FUNS** (const Rcpp::NumericVector &lparam, const Rcpp::NumericVector &lini, int Neta, int Neps, const Rcpp::CharacterVector &capture, const Rcpp::List &funs, Rcpp::Environment envir)

6.14.1 Function Documentation

6.14.1.1 as_config_func()

```
config_func* as_config_func (
    SEXP config )
```

Get pointer for \$PREAMBLE function.

Parameters

<i>config</i>	address for \$PREAMBLE function
---------------	---------------------------------

6.14.1.2 as_deriv_func()

```
deriv_func* as_deriv_func (
    SEXP derivs )
```

Get pointer for \$ODE function.

Parameters

<i>derivs</i>	address for \$ODE function
---------------	----------------------------

6.14.1.3 as_init_func()

```
init_func* as_init_func (
    SEXP inits )
```

Get pointer for \$MAIN function.

Parameters

<i>inits</i>	address for \$MAIN function
--------------	-----------------------------

6.14.1.4 as_table_func()

```
table_func* as_table_func (
    SEXP table )
```

Get pointer for \$TABLE function.

Parameters

<i>table</i>	address for \$TABLE function
--------------	------------------------------

6.14.1.5 main_derivs()

```
void main_derivs (
    int * neq,
    double * t,
    double * y,
    double * ydot,
    odeproblem * prob )
```

Derivative function that gets called by the solver.

Parameters

<i>neq</i>	number of equations
<i>t</i>	solver time
<i>y</i>	current state
<i>ydot</i>	left hand side of differential equations
<i>prob</i>	an odeproblem object

6.14.1.6 PolyExp()

```
double PolyExp (
    const double & x,
    const double & dose,
    const double & rate,
    const double & xinf,
    const double & tau,
    const bool ss,
    const dvec & a,
    const dvec & alpha,
    const int n )
```

Calculate PK model polyexponentials.

6.14.1.7 TOUCH_FUNS()

```
Rcpp::List TOUCH_FUNS (
    const Rcpp::NumericVector & lparam,
    const Rcpp::NumericVector & linit,
    int Neta,
    int Neps,
    const Rcpp::CharacterVector & capture,
    const Rcpp::List & funs,
    Rcpp::Environment envir )
```

Call the \$MAIN function from a model object.

Parameters

<i>lparam</i>	model parameters
<i>linit</i>	model initial contitions
<i>Neta</i>	number of rows in OMEGA
<i>Neps</i>	number of rows in SIGMA
<i>capture</i>	vector of capture names
<i>funs</i>	a list of model function pointers

Returns

list with updated initial conditions, number of paramerters, and number of equations

6.15 src/quick.cpp File Reference

```
#include "odeproblem.h"
#include "RcppInclude.h"
#include "dataobject.h"
#include "mrgsolve.h"
```

Typedefs

- typedef Rcpp::NumericMatrix::Column **mcol**

Functions

- `Rcpp::NumericMatrix QUICKSIM (const Rcpp::List &parin, const Rcpp::NumericVector ¶m, const Rcpp::NumericVector &init, Rcpp::CharacterVector &parnames, Rcpp::NumericMatrix &data, Rcpp::IntegerVector &n, const Rcpp::NumericMatrix &idata, const Rcpp::IntegerVector &req, const Rcpp::IntegerVector &capturei, const Rcpp::List &funs, const Rcpp::IntegerVector &nre)`
- `Rcpp::NumericMatrix PREDSIM (const Rcpp::List &parin, const Rcpp::NumericVector ¶m, const Rcpp::NumericVector &init, Rcpp::CharacterVector &parnames, Rcpp::CharacterVector &cmtnames, const Rcpp::NumericMatrix &idata, const Rcpp::IntegerVector &capturei, const Rcpp::List &funs)`

Index

- ~odeproblem
 - odeproblem, 18
- as_config_func
 - odeproblem.cpp, 36
 - odeproblem.h, 26
- as_deriv_func
 - odeproblem.cpp, 36
 - odeproblem.h, 26
- as_init_func
 - odeproblem.cpp, 36
 - odeproblem.h, 27
- as_table_func
 - odeproblem.cpp, 37
 - odeproblem.h, 27
- CompRec, 9
- DEVTRAN
 - devtran.cpp, 33
- databox, 10
- dataobject, 11
- datarecord, 12
 - schedule, 14
- devtran.cpp
 - DEVTRAN, 33
- digits
 - mrgsolve.cpp, 34
 - mrgsolve.h, 23
- find_position
 - mrgsolve.cpp, 34
 - mrgsolve.h, 24
- get
 - mrgx.h, 28, 29
- get_envir
 - mrgx.h, 29
- init_call
 - odeproblem, 18
- init_call_record
 - odeproblem, 19
- inst/include/dataobject.h, 21
- inst/include/datarecord.h, 21
- inst/include/mrgsolv.h, 22
- inst/include/mrgsolve.h, 23
- inst/include/odepack_dlsoda.h, 24
- inst/include/odeproblem.h, 25
- inst/include/tofunptr.h, 27
- inst/mrgx/mrgx.h, 28
- MVGAUSS
 - mrgsolve.cpp, 35
 - mrgsolve.h, 24
- main_derivs
 - odeproblem.cpp, 37
- mrgsolve.cpp
 - digits, 34
 - find_position, 34
 - MVGAUSS, 35
- mrgsolve.h
 - digits, 23
 - find_position, 24
 - MVGAUSS, 24
- mrgx.h
 - get, 28, 29
 - get_envir, 29
 - mt_fun, 30
 - readRDS, 30
 - rlognorm, 30
 - rnorm, 31
- mt_fun
 - mrgx.h, 30
- odepack_dlsoda, 14
- odeproblem, 16
 - ~odeproblem, 18
 - init_call, 18
 - init_call_record, 19
 - y_init, 19
- odeproblem.cpp
 - as_config_func, 36
 - as_deriv_func, 36
 - as_init_func, 36
 - as_table_func, 37
 - main_derivs, 37
 - PolyExp, 37
 - TOUCH_FUNS, 38
- odeproblem.h
 - as_config_func, 26
 - as_deriv_func, 26
 - as_init_func, 27
 - as_table_func, 27
 - PolyExp, 27
- PolyExp
 - odeproblem.cpp, 37
 - odeproblem.h, 27
- readRDS
 - mrgx.h, 30

- resim, [19](#)
- rlognorm
 - mrgh.h, [30](#)
- rnorm
 - mrgh.h, [31](#)
- schedule
 - datarecord, [14](#)
- src/dataobject.cpp, [31](#)
- src/datarecord.cpp, [32](#)
- src/devtran.cpp, [32](#)
- src/mrghsolve.cpp, [33](#)
- src/odepack_dlsoda.cpp, [35](#)
- src/odeproblem.cpp, [35](#)
- src/quick.cpp, [38](#)
- TOUCH_FUNS
 - odeproblem.cpp, [38](#)
- y_init
 - odeproblem, [19](#)