Package ‘RLumCarlo’

October 9, 2019

Type Package

Title Monte-Carlo Methods for Simulating Luminescence Phenomena

Version 0.1.0.9000-63

Date 2019-10-09

Author Johannes Friedrich [aut, trl] (<https://orcid.org/0000-0002-0805-9547>),
Sebastian Kreutzer [aut, trl, cre] (<https://orcid.org/0000-0002-0734-2199>),
Vasilis Pagonis [aut] (<https://orcid.org/0000-0002-4852-9312>),
Christoph Schmidt [aut] (<https://orcid.org/0000-0002-2309-3209>),
Ena Rajovic [ctb],
Alex Roy Duncan [ctb],
Christian Laag [ctb]

Maintainer Sebastian Kreutzer <sebastian.kreutzer@u-bordeaux-montaigne.fr>

Description A Collection of Functions to Simulate Luminescence Production in Minerals using Monte-Carlo methods.

Contact Package Developer Team <sebastian.kreutzer@u-bordeaux-montaigne.fr>

License GPL-3

BugReports https://github.com/R-Lum/RLumCarlo/issues

Depends R (>= 3.3.0),
utils,
magrittr

URL https://CRAN.R-project.org/package=RLumCarlo

LinkingTo Rcpp (>= 1.0.2),
RcppArmadillo (>= 0.9.700.2.0)

Imports abind (>= 1.4-5),
doParallel (>= 1.0.15),
foreach (>= 1.4.7),
parallel,
methods,
Rcpp (>= 1.0.2)

Suggests R.rsp (>= 0.43.1),
testthat (>= 2.0.0)

Encoding UTF-8

VignetteBuilder R.rsp

RoxygenNote 6.1.1

R topics documented:

plot_RLumCarlo *Plot results from Monte-Carlo simulations with RLumCarlo*

Description

Plot results from Monte-Carlo simulations with RLumCarlo

Usage

```
plot_RLumCarlo(object, times = NULL, plot_uncertainty = "range",
  norm = FALSE, add = FALSE, ...)
```

Arguments

`object`	data.frame (**required**)
`times`	numeric (*optinal*): Optional vector for the x-axis
`plot_uncertainty`	logical (*with default*): Enable/disable uncertainty polygon plot
`norm`	logical (*with default*): Normalise curve to the highest intensity
`add`	logical (*with default*): allow overplotting of results
`...`	further arguments that can be passed to control the plot output. Currently supported are: `xlab`, `xlim`, `ylim`, `main`, `lwd`, `type`, `pch`, `lty`,`col`, `grid`, `legend`

Value

This function returns a graphical output

Function version

0.1.0

How to cite

Friedrich, J., Kreutzer, S., 2019. plot_RLumCarlo(): Plot results from Monte-Carlo simulations with RLumCarlo. Function version 0.1.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-63.

Author(s)

Johannes Friedrich, University of Bayreuth (Germany), Sebastian Kreutzer, IRAMAT-CRP2A, Université Bordeaux Montaigne (France)

`run_MC_CW_IRSL_DELOC` *Run Monte-Carlo simulation for CW-IRSL for delocalized transition*

Description

Runs a Monte-Carlo (MC) simulation of constant wave infrared stimulated luminesence (CW-IRSL) using the one trap one recombination center (OTOR) model.

Usage

```
run_MC_CW_IRSL_DELOC(A, times, clusters = 10, N_e = 200,
  n_filled = N_e, R, method = "par", output = "signal", ...)
```

Arguments

`A`	numeric (**required**): The transition probability (cm^3/s).
`times`	numeric (*with default*): The number of MC runs.
`clusters`	numeric (*with default*): The number of clusters.
`N_e`	integer (*with default*): The number of electrons.
`n_filled`	integer (*with default*): The number of electron traps that are filled at the beginning of the simulation.
`R`	numeric (*with default*): The retrapping ratio.
`method`	character (*with default*):
`output`	character (*with default*):
`...`	further arguments

Details

$$I_{DELOC}(t) = -dn/dt = p(t) * (n^2/(NR + n(1-R)))$$

Value

This function returns an array with dimension length(times) x length(r) x clusters

Function version

0.1.0

How to cite

Kreutzer, S., 2019. run_MC_CW_IRSL_DELOC(): Run Monte-Carlo simulation for CW-IRSL for delocalized transition. Function version 0.1.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-63.

Author(s)

Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Université Bordeaux Montaigne (France)

References

Pagonis, V. and Kulp, C., 2017. Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars, Journal of Luminescence 181, 114–120. doi: 10.1016/j.jlumin.2016.09.014

Examples

```
##=========================================================================##
## Example 1: Simulate CW-IRSL
##=========================================================================##
## Not run:
run_MC_CW_IRSL_DELOC(
 A = 0.12,
 R = 1,
 times = 0:100) %>%
   plot_RLumCarlo(legend = T)


## End(Not run)

#' @examples
##=========================================================================##
## Example 2: Simulate CW-IRSL DELOC with several parameter changes
##=========================================================================##
## Not run:

# define your parameters
A=c(0.1,0.3,0.5,1)
times=seq(0,60,1)
s=1e12
E=1
R<-c(1e-7,1e-6,0.01,0.1) # sequence of different R values
clusters=1000 # number of Monte Carlo simulations
N_e =c(200, 500, 700, 400) # number of free electrons
n_filled =c(200, 500, 100, 70) # number of filled traps
method="par"
output ="signal"
col=c(1,2,3,4) # ifferent colours for the individual curves
plot_uncertainty <- c(T,F,T,F)  # do you want to see the uncertainty?
add_TF <- c(F,rep(T, (length(R)-1)))
for (u in 1:length(R)){
 results <-run_MC_CW_IRSL_DELOC(A=A[u], times, clusters =clusters, N_e = N_e[u],
                         n_filled = n_filled[u], R=R[u], method = method, output = output)
 plot_RLumCarlo(results,add=add_TF[u],legend = F, col=col[u], main=" your plot")
}
```

```
legend("topright",ncol=4,cex=0.55,title = "parameters" ,legend=c(paste0("A = ", A),
                                                        paste0("n_filled = ", n_filled),
                                                                paste0("N_e = ", N_e),
                                                      paste0("R = ", R)),  text.col=col)


## End(Not run)
```

`run_MC_CW_IRSL_LOC` *Run Monte-Carlo simulation for CW-IRSL for localised transition*

Description

Runs a Monte-Carlo (MC) simulation of constant wave infrared stimulated luminesence (CW-IRSL) using the generalized one trap (GOT) model.

Usage

```
run_MC_CW_IRSL_LOC(A, times, clusters = 10, n_filled = 100, r,
  method = "par", output = "signal", ...)
```

Arguments

`A` numeric (**required**): The transition probability (cm^3/s).

`times` numeric (*with default*): The number of MC runs.

`clusters` numeric (*with default*): The number of clusters.

`n_filled` integer (*with default*): The number of electron traps that are filled at the beginning of the simulation.

`r` numeric (*with default*): The retrapping ratio.

`method` character (*with default*):

`output` character (*with default*):

`...` further arguments

Details

$$I_{LOC}(t) = -dn/dt = A * (n^2/(r + n))$$

Value

This function returns an array with dimension length(times) x length(r) x clusters

Function version

0.1.0

How to cite

Kreutzer, S., 2019. run_MC_CW_IRSL_LOC(): Run Monte-Carlo simulation for CW-IRSL for localised transition. Function version 0.1.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-63.

Author(s)

Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Université Bordeaux Montaigne (France)

References

Pagonis, V. and Kulp, C., 2017. Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars. Journal of Luminescence 181, 114–120. doi: 10.1016/j.jlumin.2016.09.014.

Examples

```
##============================================================================##
## Example 1: Simulate CW-IRSL
##============================================================================##
## Not run:
run_MC_CW_IRSL_LOC(
 A = 0.12,
 r = 1,
 times = 0:100) %>%
   plot_RLumCarlo(legend = T)


## End(Not run)
```

run_MC_CW_IRSL_TUN *Run Monte-Carlo simulation for CW-IRSL*

Description

Runs a Monte-Carlo (MC) simulation of constant wave infrared stimulated luminesence (CW-IRSL) using the model.

Usage

```
run_MC_CW_IRSL_TUN(A, rho, times, clusters = 10, r = NULL, N_e = 200,
  method = "seq", output = "signal", ...)
```

Arguments

A	numeric (**required**): The transition probability (cm^3/s).
rho	numeric (**required**): The calculated dimesionless Charge density (normally written Rho').
times	numeric (*with default*): The number of MC runs.
clusters	numeric (*with default*): The number of clusters.

`r` numeric (*with default*): The retrapping ratio.

`N_e` numeric (*with default*): The number of electrons

`method` character (*with default*):

`output` character (*with default*):

`...` further arguments

Details

####equation here please####

Value

This function returns a list.

Function version

0.2.0

How to cite

Friedrich, J., Kreutzer, S., 2019. run_MC_CW_IRSL_TUN(): Run Monte-Carlo simulation for CW-IRSL. Function version 0.2.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-63.

Author(s)

Johannes Friedrich, University of Bayreuth (Germany), Sebastian Kreutzer, IRAMAT-CRP2A, Université Bordeaux Montaigne (France)

References

Pagonis, V. and Kulp, C. (2017) ‘Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars’, Journal of Luminescence. Elsevier, 181, pp. 114–120. doi: 10.1016/j.jlumin.2016.09.014.

Examples

```
## Not run:

##============================================================================##
## Example 1: Simulate CW-IRSL measurement
##============================================================================##

run_MC_CW_IRS_TUNL(A = 0.12, rho = 0.003, times = 0:1000) %>%
    plot_RLumCarlo(norm = T, legend = T)

## End(Not run)
```

run_MC_ISO_DELOC *Run Monte-Carlo simulation for ISO for delocalized transition*

Description

Runs a Monte-Carlo (MC) simulation of isothermally stimulated luminesence (ISO-TL or ITL) using the one trap one recombination center (OTOR) model.

Usage

```
run_MC_ISO_DELOC(s, E, T = 20, times, clusters = 10, N_e = 200,
  n_filled = N_e, R, method = "par", output = "signal", ...)
```

Arguments

s	numeric (**required**): Escape frequency of the trap (s^-1).
E	numeric (**required**): Thermal activation energy of the trap (eV).
T	numeric (*with default*): Temperature (deg. C).
times	numeric (*with default*): The number of MC runs.
clusters	numeric (*with default*): The number of clusters.
N_e	integer (*with default*): The number of electrons.
n_filled	integer (*with default*): The number of electron traps that are filled at the beginning of the simulation.
R	numeric (*with default*): The retrapping ratio.
method	character (*with default*):
output	character (*with default*):
...	further arguments

Details

$$I_{DELOC}(t) = -dn/dt = p(t) * (n^2/(NR + n(1-R)))$$

Value

This function returns an array with dimension length(times) x length(r) x clusters

Function version

0.0.1

How to cite

Kreutzer, S., 2019. run_MC_ISO_DELOC(): Run Monte-Carlo simulation for ISO for delocalized transition. Function version 0.0.1. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-63.

Author(s)

Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Université Bordeaux Montaigne (France)

References

Pagonis, V. and Kulp, C. (2017) 'Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars', Journal of Luminescence. Elsevier, 181, pp. 114–120. doi: 10.1016/j.jlumin.2016.09.014.

Examples

```
##============================================================================##
## Example 1: Simulate ITL
##============================================================================##
## Not run:
run_MC_ISO_DELOC(
 s = 3.5e12,
 E = 1.45,
 T = 200,
 R = 1,
 times = 0:10000) %>%
   plot_RLumCarlo(legend = T)


## End(Not run)
```

---

run_MC_ISO_LOC *Run Monte-Carlo simulation for ITL for localised transition*

Description

Runs a Monte-Carlo (MC) simulation of isothermally stimulated luminesence (ISO-TL or ITL) using the model.

Usage

```
run_MC_ISO_LOC(s, E, T = 20, times, clusters = 10, n_filled = 100, r,
  method = "par", output = "signal", ...)
```

Arguments

`s`	numeric (**required**): Escape frequency of the trap (s^-1).
`E`	numeric (**required**): Thermal activation energy of the trap (eV).
`T`	numeric (*with default*): Temperature (deg. C).
`times`	numeric (*with default*): The number of MC runs.
`clusters`	numeric (*with default*): The number of clusters.
`n_filled`	integer (*with default*): The number of electron traps that are filled at the beginning of the simulation.
`r`	numeric (*with default*): The retrapping ratio.

`method`	character (*with default*):
`output`	character (*with default*):
`...`	further arguments

Details

$$I_{LOC}(t) = -dn/dt = p(t) * (n^2/(r + n))$$

Value

This function returns an array with dimension length(times) x length(r) x clusters

Function version

0.0.1

How to cite

Kreutzer, S., 2019. run_MC_ISO_LOC(): Run Monte-Carlo simulation for ITL for localised transition. Function version 0.0.1. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-63.

Author(s)

Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Université Bordeaux Montaigne (France)

References

Pagonis, V. and Kulp, C. (2017) ‘Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars’, Journal of Luminescence. Elsevier, 181, pp. 114–120. doi: 10.1016/j.jlumin.2016.09.014.

Examples

```
##=============================================================================##
## Example 1: Simulate ITL
##=============================================================================##
## Not run:
run_MC_ISO_LOC(
 s = 3.5e12,
 E = 1.45,
 T = 200,
 r = 1,
 times = 0:10000) %>%
   plot_RLumCarlo(legend = T)


## End(Not run)
```

`run_MC_ISO_TUN` *Run Monte-Carlo simulation for isothermal measurements*

Description

Runs a Monte-Carlo (MC) simulation of isothermally stimulated luminesence (ISO-TL or ITL) using the .

Usage

```
run_MC_ISO_TUN(E, s, T = 200, rho, times, clusters = 10, r = NULL,
  N_e = 200, method = "par", output = "signal", ...)
```

Arguments

`E`	numeric (**required**): Thermal activation energy of the trap (eV).
`s`	numeric (**required**): Escape frequency of the trap (s^-1).
`T`	numeric (**required**): Temperature (deg. C).
`rho`	numeric (**required**): The calculated dimesionless Charge density (normally written Rho').
`times`	numeric (*with default*): The number of MC runs.
`clusters`	numeric (*with default*): The number of clusters.
`r`	numeric (*with default*): The retrapping ratio.
`N_e`	numeric (*with default*): The number of electrons.
`method`	character (*with default*)
`output`	character (*with default*)
`...`	further arguments

Details

$$I_{LOC}(t) = -dn/dt = A * (n^2/(r + n))$$

Value

This function returns a list.

Function version

0.1.0

How to cite

Friedrich, J., Kreutzer, S., 2019. run_MC_ISO_TUN(): Run Monte-Carlo simulation for isothermal measurements. Function version 0.1.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-63.

Author(s)

Johannes Friedrich, University of Bayreuth (Germany), Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Univerité Bordeaux Montaigne (France)

References

Pagonis, V. and Kulp, C., 2017. ‘Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars’, Journal of Luminescence 181, 114–120. doi: 10.1016/j.jlumin.2016.09.014

Examples

```
## Not run:
##===============================================================================##
## Example 1: Simulate isothermal measurement
##===============================================================================##

times <- seq(0, 5000)
run_MC_ISO_TUN(
 E = 1.2,
 s = 1e10,
 T = 200,
 rho = 0.007,
 times = times) %>%
  plot_RLumCarlo(legend = T)

## End(Not run)
```

---

run_MC_LM_OSL_DELOC *Run Monte-Carlo simulation for LM-OSL for delocalized transition*

Description

Runs a Monte-Carlo (MC) simulation of linearly modulated optically stimulated luminesence (LM-OSL) using the one trap one recombination center (OTOR) model.

Usage

```
run_MC_LM_OSL_DELOC(A, times, clusters = 10, N_e = 200,
  n_filled = N_e, R, method = "par", output = "signal", ...)
```

Arguments

A	numeric (**required**): The transition probability (cm^3/s).
times	numeric (*with default*): The number of MC runs.
clusters	numeric (*with default*): The number of clusters.
N_e	integer (*with default*): The number of electrons.
n_filled	integer (*with default*): The number of electron traps that are filled at the beginning of the simulation.
R	numeric (*with default*): The retrapping ratio.

`method`	character (*with default*):
`output`	character (*with default*):
`...`	further arguments

Details

$$I_{DELOC}(t) = -dn/dt = p(t) * (n^2/(NR + n(1 - R)))$$

Value

This function returns an array with dimension length(times) x length(r) x clusters

Function version

0.1.0

How to cite

Kreutzer, S., 2019. run_MC_LM_OSL_DELOC(): Run Monte-Carlo simulation for LM-OSL for delocalized transition. Function version 0.1.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-63.

Author(s)

Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Université Bordeaux Montaigne (France)

References

Pagonis, V. and Kulp, C. (2017) 'Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars', Journal of Luminescence. Elsevier, 181, pp. 114–120. doi: 10.1016/j.jlumin.2016.09.014.

Examples

```
##============================================================================##
## Example 1: Simulate LM-OSL
##============================================================================##
## Not run:
run_MC_LM_OSL_DELOC(
 A = 0.12,
 R = 1,
 times = 0:100) %>%
   plot_RLumCarlo(legend = T)


## End(Not run)
```

run_MC_LM_OSL_LOC *Run Monte-Carlo simulation for LM-OSL for localized transition*

Description

Runs a Monte-Carlo (MC) simulation of linearly modulated optically stimulated luminesence (LM-OSL) using the generalized one trap (GOT) model.

Usage

```
run_MC_LM_OSL_LOC(A, times, clusters = 10, n_filled = 100, r,
  method = "par", output = "signal", ...)
```

Arguments

A	numeric (**required**): The transition probability (cm^3/s).
times	numeric (*with default*): The number of MC runs.
clusters	numeric (*with default*): The number of clusters.
n_filled	integer (*with default*): The number of electron traps that are filled at the beginning of the simulation.
r	numeric (*with default*):
method	character (*with default*):
output	character (*with default*):
...	further arguments

Details

$$I_{DELOC}(t) = -dn/dt = p(t) * (n^2/(NR + n(1 - R)))$$

Value

This function returns an array with dimension length(times) x length(r) x clusters

Function version

0.0.1

How to cite

Kreutzer, S., 2019. run_MC_LM_OSL_LOC(): Run Monte-Carlo simulation for LM-OSL for localized transition. Function version 0.0.1. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-63.

Author(s)

Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Université Bordeaux Montaigne (France)

References

Pagonis, V. and Kulp, C. (2017) ‘Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars’, Journal of Luminescence. Elsevier, 181, pp. 114–120. doi: 10.1016/j.jlumin.2016.09.014.

Examples

```
##============================================================================##
## Example 1: Simulate LM-OSL
##============================================================================##
## Not run:
run_MC_LM_OSL_LOC(
 A = 0.12,
 r = 1,
 times = 0:100) %>%
   plot_RLumCarlo(legend = T)


## End(Not run)
```

`run_MC_LM_OSL_TUN` *Run Monte-Carlo simulation for LM-OSL*

Description

Run Monte-Carlo simulation for LM-OSL

Usage

```
run_MC_LM_OSL_TUN(A, rho, times, clusters = 10, r = NULL,
  delta.r = 0.1, N_e = 200, method = "par", output = "signal", ...)
```

Arguments

`A` numeric (**required**): The transition probaility (cm^3/s).

`rho` numeric (**required**): The calculated dimesionless Charge density (normally written Rho’).

`times` vector (*with default*): The number of MC runs.

`clusters` numeric (*with default*): The number of clusters.

`r` numeric (*with default*): The retrapping ratio.

`delta.r` numeric (*with default*):

`N_e` numeric (*with default*): The number of electrons.

`method` character (*with default*):

`output` character (*with default*):

`...` further arguments

Details

ADD EQUATION

Value

This function returns a list.

Function version

0.1.0

How to cite

Friedrich, J., 2019. run_MC_LM_OSL_TUN(): Run Monte-Carlo simulation for LM-OSL. Function version 0.1.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-63.

Author(s)

Johannes Friedrich, University of Bayreuth (Germany)

References

Pagonis, V. and Kulp, C. (2017) ‘Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars’, Journal of Luminescence. Elsevier, 181, pp. 114–120. doi: 10.1016/j.jlumin.2016.09.014.

Examples

```
## Not run:

##TODO: Primary example, should be verified
run_MC_LM_OSL_TUN(A = 10000, rho = 0.0001, times = 1:100, clusters = 10, r = NULL,
 delta.r = 0.1,
 N_e = 200, method = "par", output = "signal") %>%
 plot_RLumCarlo(norm = T)


## End(Not run)
```

run_MC_TL_DELOC *Run Monte-Carlo simulation for TL for delocalized transition*

Description

Runs a Monte-Carlo (MC) simulation of Thermo-luminesence (LM-OSL) using the one trap one recombination center (OTOR) model.

Usage

```
run_MC_TL_DELOC(s, E, times, clusters = 10, N_e = 200,
  n_filled = N_e, R, method = "par", output = "signal", ...)
```

Arguments

`s` numeric (**required**): Escape frequency of the trap (s^-1).

`E` numeric (**required**): Thermal activation energy of the trap (eV).

`times` numeric (*with default*): the number of MC runs.

`clusters` numeric (*with default*): the number of clusters.

`N_e` integer (*with default*): The number of electrons.

`n_filled` integer (*with default*): The number of electron traps that are filled at the beginning of the simulation.

`R` numeric (*with default*): The retrapping ratio.

`method` character (*with default*):

`output` character (*with default*):

`...` further arguments

Details

$$I_{DELOC}(t) = -dn/dt = p(t) * (n^2/(NR + n(1 - R)))$$

Value

This function returns an array with dimension length(times) x length(r) x clusters

Function version

0.0.1

How to cite

Kreutzer, S., 2019. run_MC_TL_DELOC(): Run Monte-Carlo simulation for TL for delocalized transition. Function version 0.0.1. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-63.

Author(s)

Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Université Bordeaux Montaigne (France)

References

Pagonis, V. and Kulp, C. (2017) ‘Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars’, Journal of Luminescence. Elsevier, 181, pp. 114–120. doi: 10.1016/j.jlumin.2016.09.014.

Examples

```
##=========================================================================##
## Example 1: Simulate TL
##=========================================================================##
## Not run:
run_MC_TL_DELOC(
 s = 3.5e12,
 E = 1.45,
 R = 1,
 times = 100:450) %>%
   plot_RLumCarlo(legend = T)


## End(Not run)
```

run_MC_TL_LOC *Run Monte-Carlo simulation for TL for localised transition*

Description

Runs a Monte-Carlo (MC) simulation of thermo-luminesence (LM-OSL) using the generalized one trap (GOT) model.

Usage

```
run_MC_TL_LOC(s, E, times, clusters = 10, n_filled = 100, r,
  method = "par", output = "signal", ...)
```

Arguments

s	numeric (**required**): Escape frequency of the trap (s^-1).
E	numeric (**required**): Thermal activation energy of the trap (eV).
times	numeric (*with default*): The number of Mc runs.
clusters	numeric (*with default*): The number of clusters.
n_filled	integer (*with default*): The number of electron traps that are filled at the beginning of the simulation.
r	numeric (*with default*): The retrapping ratio.
method	character (*with default*):
output	character (*with default*):
...	further arguments

Details

$$I_{LOC}(t) = -dn/dt = p(t) * (n^2/(r + n))$$

Value

This function returns an array with dimension length(times) x length(r) x clusters

Function version

0.1.0

How to cite

Kreutzer, S., 2019. run_MC_TL_LOC(): Run Monte-Carlo simulation for TL for localised transition. Function version 0.1.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-63.

Author(s)

Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Université Bordeaux Montaigne (France)

References

Pagonis, V. and Kulp, C. (2017) ‘Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars’, Journal of Luminescence. Elsevier, 181, pp. 114–120. doi: 10.1016/j.jlumin.2016.09.014.

Examples

```
##============================================================================##
## Example 1: Simulate TL
##============================================================================##
## Not run:
run_MC_TL_LOC(
 s = 3.5e12,
 E = 1.45,
 r = 1,
 times = 100:450) %>%
   plot_RLumCarlo(legend = T)


## End(Not run)
```

`run_MC_TL_TUN` *Run Monte-Carlo simulation for TL*

Description

Run Monte-Carlo simulation for TL

Usage

```
run_MC_TL_TUN(s, E, rho, r_c, times, clusters = 10, N_e = 200,
  delta.r = 0.1, method = "par", output = "signal", ...)
```

Arguments

`s`	list (**required**): Escape frequency of the trap (s^-1).
`E`	numeric (**required**): Thermal activation energy of the trap (eV).
`rho`	numeric (**required**): The calculated dimesionless Charge density (normally written Rho’).

`r_c`	numeric (*with default*):
`times`	vector (*with default*): The number of MC runs.
`clusters`	numeric (*with default*): The number of clusters.
`N_e`	numeric (*with default*): The number of electrons
`delta.r`	numeric (*with default*):
`method`	character (*with default*):
`output`	character (*with default*):
`...`	further arguments

Details

ADD EQUATION

Value

This function returns an `array` with dimension length(times) x length(r) x clusters

Function version

0.1.0

How to cite

Friedrich, J., 2019. run_MC_TL_TUN(): Run Monte-Carlo simulation for TL. Function version 0.1.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-63.

Author(s)

Johannes Friedrich, University of Bayreuth (Germany)

References

Pagonis, V. and Kulp, C. (2017) ‘Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars’, Journal of Luminescence. Elsevier, 181, pp. 114–120. doi: 10.1016/j.jlumin.2016.09.014.

Examples

```
## Not run:
##=========================================================================##
## Example 1: Simulate TL measurement
##=========================================================================##

times <- seq(200, 500) # time = temperature

run_MC_TL_TUN(s = 3.5e12,
          E = 1.45,
          rho = 0.015,
          r_c = 0.85,
          times = times) %>%
    plot_RLumCarlo(legend = T)

## End(Not run)
```

Index