

# Package ‘RLumCarlo’

October 9, 2019

**Type** Package

**Title** Monte-Carlo Methods for Simulating Luminescence Phenomena

**Version** 0.1.0.9000-73

**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](mailto: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](mailto: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 . . . . .	2
run_MC_CW_IRSL_LOC . . . . .	3
run_MC_CW_IRSL_TUN . . . . .	4
run_MC_CW_OSL_DELOC . . . . .	6
run_MC_ISO_DELOC . . . . .	8
run_MC_ISO_LOC . . . . .	9
run_MC_ISO_TUN . . . . .	11
run_MC_LM_OSL_DELOC . . . . .	12
run_MC_LM_OSL_LOC . . . . .	14
run_MC_LM_OSL_TUN . . . . .	15
run_MC_TL_DELOC . . . . .	16
run_MC_TL_LOC . . . . .	18
run_MC_TL_TUN . . . . .	20
<b>Index</b>	<b>22</b>

---

plot_RLumCarlo	<i>Plot results from Monte-Carlo simulations with RLumCarlo</i>
----------------	---

---

### Description

Plot results from Monte-Carlo simulations with RLumCarlo

### Usage

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

### Arguments

object	<b>data.frame</b> ( <b>required</b> )
times	<b>numeric</b> ( <i>optinal</i> ): Optional vector for the x-axis
plot_uncertainty	<b>logical</b> ( <i>with default</i> ): Enable/disable uncertainty polygon plot
norm	<b>logical</b> ( <i>with default</i> ): Normalise curve to the highest intensity
add	<b>logical</b> ( <i>with default</i> ): 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 Phenomena R package version 0.1.0.9000-73.

**Author(s)**

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

---

run_MC_CW_IRSL_LOC	<i>Run Monte-Carlo simulation for CW-IRSL for localised transition</i>
--------------------	--

---

**Description**

Runs a Monte-Carlo (MC) simulation of constant wave infrared stimulated luminescence (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	<b>numeric (required)</b> : The transition probability (cm <sup>3</sup> /s).
times	<b>numeric (with default)</b> : The number of MC runs.
clusters	<b>numeric (with default)</b> : The number of clusters.
n_filled	<b>integer (with default)</b> : The number of electron traps that are filled at the beginning of the simulation.
r	<b>numeric (with default)</b> : The retrapping ratio.
method	<b>character (with default)</b> : sequential 'seq' or parallel processing 'par'
output	<b>character (with default)</b> : output is either the 'signal' (the default) or 'remaining_e' (the remaining charges, electrons, in the trap)
...	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 Phenomena R package version 0.1.0.9000-73.

## 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	<i>Run Monte-Carlo simulation for CW-IRSL</i>
--------------------	---

---

## Description

Runs a Monte-Carlo (MC) simulation of constant wave infrared stimulated luminescence (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	<b>numeric (required):</b> The transition probability ( $\text{cm}^3/\text{s}$ ).
rho	<b>numeric (required):</b> The calculated dimensionless Charge density (normally written Rho').
times	<b>numeric (with default):</b> The number of MC runs.
clusters	<b>numeric (with default):</b> The number of clusters.

r	<b>numeric</b> ( <i>with default</i> ): The retrapping ratio.
N_e	<b>numeric</b> ( <i>with default</i> ): The number of electrons
method	<b>character</b> ( <i>with default</i> ): sequential 'seq' or parallel processing 'par'
output	<b>character</b> ( <i>with default</i> ): output is either the 'signal' (the default) or 'remaining_e' (the remaining charges, electrons, in the trap)
...	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 0.1.0.9000-73.

## Author(s)

Johannes Friedrich, University of Bayreuth (Germany), Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, 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_IRSL_TUNL(A = 0.12, rho = 0.003, times = 0:1000) %>%
  plot_RLumCarlo(norm = T, legend = T)

## End(Not run)
```

---

run\_MC\_CW\_OSL\_DELOC      *Run Monte-Carlo simulation for CW-OSL for delocalized transition*

---

## Description

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

## Usage

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

## Arguments

A	<b>numeric (required)</b> : The transition probability (cm <sup>3</sup> /s).
times	<b>numeric</b> (with default): The number of MC runs.
clusters	<b>numeric</b> (with default): The number of clusters.
N_e	<b>integer</b> (with default): The number of electrons.
n_filled	<b>integer</b> (with default): The number of electron traps that are filled at the beginning of the simulation.
R	<b>numeric</b> (with default): The retrapping ratio.
method	<b>character</b> (with default): sequential 'seq' or parallel processing 'par'
output	<b>character</b> (with default): output is either the 'signal' (the default) or 'remaining_e' (the remaining charges, electrons, in the trap)
...	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\_OSL\_DELOC(): Run Monte-Carlo simulation for CW-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 Phenomena R package version 0.1.0.9000-73.

## 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](https://doi.org/10.1016/j.jlumin.2016.09.014)

## Examples

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

## End(Not run)

#' @examples
##=====##
## Example 2: Simulate CW-OSL 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_OSL_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\_ISO\_DELOC

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

Runs a Monte-Carlo (MC) simulation of isothermally stimulated luminescence (ISO-TL or ITL) using the one trap one recombination center (OTOR) model. Delocalized refers to involvement of the conduction band.

**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	<b>numeric (required)</b> : Escape frequency of the trap ( $s^{-1}$ ).
E	<b>numeric (required)</b> : Thermal activation energy of the trap (eV).
T	<b>numeric (with default)</b> : Temperature (deg. C).
times	<b>numeric (with default)</b> : The number of MC runs.
clusters	<b>numeric (with default)</b> : The number of clusters.
N_e	<b>integer (with default)</b> : The number of electrons.
n_filled	<b>integer (with default)</b> : The number of electron traps that are filled at the beginning of the simulation.
R	<b>numeric (with default)</b> : The retrapping ratio.
method	<b>character (with default)</b> : sequential 'seq' or parallel processing 'par'
output	<b>character (with default)</b> : output is either the 'signal' (the default) or 'remaining_e' (the remaining charges, electrons, in the trap)
...	further arguments

**Details**

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

Where in the function  $n := n\_filled := N := N\_e$

**Value**

This function returns an **array** with dimension  $\text{length}(\text{times}) \times \text{length}(r) \times \text{clusters}$

**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 Phenomena R package version 0.1.0.9000-73.



**Author(s)**

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

**References**

Pagonis, V., Friedrich, J., Discher, M., Müller-Kirschbaum, A., Schlosser, V., Kreutzer, S., Chen, R. and Schmidt, C., 2019. Excited state luminescence signals from a random distribution of defects: A new Monte Carlo simulation approach for feldspar. *Journal of Luminescence* 207, 266–272. doi: [10.1016/j.jlumin.2018.11.024](https://doi.org/10.1016/j.jlumin.2018.11.024)

Reuven, C. and S. McKeever, 1997. *Theory of thermoluminescence and related phenomena*.

**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 luminescence (ISO-TL or ITL) using the generalized one trap (GOT) model. Localized refers to the lack of involvement of the conduction band.

**Usage**

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

**Arguments**

s	<b>numeric (required)</b> : Escape frequency of the trap ( $s^{-1}$ ).
E	<b>numeric (required)</b> : Thermal activation energy of the trap (eV).
T	<b>numeric (with default)</b> : Temperature (deg. C).
times	<b>numeric (with default)</b> : The number of MC runs.
clusters	<b>numeric (with default)</b> : The number of clusters.

n_filled	<b>integer</b> (with default): The number of electron traps that are filled at the beginning of the simulation.
r	<b>numeric</b> (with default): The retrapping ratio.
method	<b>character</b> (with default): sequential 'seq' or parallel processing 'par'
output	<b>character</b> (with default): output is either the 'signal' (the default) or 'remaining_e' (the remaining charges, electrons, in the trap)
...	further arguments

## Details

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

Where in the function  $n := n\_filled := N := N\_e$

## 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 0.1.0.9000-73.

## Author(s)

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

## References

Pagonis, V., Friedrich, J., Discher, M., Müller-Kirschbaum, A., Schlosser, V., Kreutzer, S., Chen, R. and Schmidt, C., 2019. Excited state luminescence signals from a random distribution of defects: A new Monte Carlo simulation approach for feldspar. Journal of Luminescence 207, 266–272. doi: [10.1016/j.jlumin.2018.11.024](https://doi.org/10.1016/j.jlumin.2018.11.024)

## 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	<i>Run Monte-Carlo Simulation for Isothermal Measurements for Tunneling Transition</i>
----------------	--

---

## Description

Runs a Monte-Carlo (MC) simulation of isothermally stimulated luminescence (ISO-TL or ITL) using the tunneling (TUN) model. Tunneling transitions refers to the direct movement of electrons from a trap directly to the recombination centre.

## Usage

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

## Arguments

E	<b>numeric (required)</b> : Thermal activation energy of the trap (eV).
s	<b>numeric (required)</b> : Escape frequency of the trap (s <sup>-1</sup> ).
T	<b>numeric (required)</b> : Temperature (deg. C).
rho	<b>numeric (required)</b> : The calculated dimensionless Charge density (normally written Rho').
times	<b>numeric (with default)</b> : The number of MC runs.
clusters	<b>numeric (with default)</b> : The number of clusters.
r	<b>numeric (with default)</b> : The retrapping ratio.
N_e	<b>numeric (with default)</b> : The number of electrons.
method	<b>character (with default)</b> : sequential 'seq' or parallel processing 'par'
output	<b>character (with default)</b> : output is either the 'signal' (the default) or 'remaining_e' (the remaining charges, electrons, in the trap)
...	further arguments

## Details

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

Where in the function  $n := n\_filled := N := N\_e := rho := rho'$  } := \code{r\_c} } := \code{rho'\_c}

## 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 for Tunneling 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 0.1.0.9000-73.

## Author(s)

Johannes Friedrich, University of Bayreuth (Germany), 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](https://doi.org/10.1016/j.jlumin.2016.09.014)

Pagonis, V., Friedrich, J., Discher, M., Müller-Kirschbaum, A., Schlosser, V., Kreutzer, S., Chen, R. and Schmidt, C., 2019. Excited state luminescence signals from a random distribution of defects: A new Monte Carlo simulation approach for feldspar. Journal of Luminescence 207, 266–272. doi: [10.1016/j.jlumin.2018.11.024](https://doi.org/10.1016/j.jlumin.2018.11.024)

for a discussion of tunneling see: Aitken, M.J., 1985. Thermoluminescence dating. 276–280. doi: [10.1002/gea.3340020110](https://doi.org/10.1002/gea.3340020110)

## Examples

```
## Not run:
##=====##
## Example 1: Simulate isothermal measurement
##=====##
run_MC_ISO_TUN(
  E = 1.2,
  s = 1e10,
  T = 200,
  rho = 0.007,
  times = 0:5000) %>%
  plot_RLumCarlo(legend = T)

## End(Not run)
```

---

run_MC_LM_OSL_DELOC	<i>Run Monte-Carlo simulation for LM-OSL for delocalized transition</i>
---------------------	---

---

## Description

Runs a Monte-Carlo (MC) simulation of linearly modulated optically stimulated luminescence (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	<b>numeric (required)</b> : The transition probability (cm <sup>3</sup> /s).
times	<b>numeric (with default)</b> : The number of MC runs.
clusters	<b>numeric (with default)</b> : The number of clusters.
N_e	<b>integer (with default)</b> : The number of electrons.
n_filled	<b>integer (with default)</b> : The number of electron traps that are filled at the beginning of the simulation.
R	<b>numeric (with default)</b> : The retrapping ratio.
method	<b>character (with default)</b> : sequential 'seq' or parallel processing 'par'
output	<b>character (with default)</b> : output is either the 'signal' (the default) or 'remaining_e' (the remaining charges, electrons, in the trap)
...	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 Phenomena R package version 0.1.0.9000-73.

**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	<i>Run Monte-Carlo simulation for LM-OSL for localized transition</i>
-------------------	---

---

## Description

Runs a Monte-Carlo (MC) simulation of linearly modulated optically stimulated luminescence (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	<b>numeric (required)</b> : The transition probability (cm <sup>3</sup> /s).
times	<b>numeric</b> (with default): The number of MC runs.
clusters	<b>numeric</b> (with default): The number of clusters.
n_filled	<b>integer</b> (with default): The number of electron traps that are filled at the beginning of the simulation.
r	<b>numeric</b> (with default):
method	<b>character</b> (with default): sequential 'seq' or parallel processing 'par'
output	<b>character</b> (with default): output is either the 'signal' (the default) or 'remaining_e' (the remaining charges, electrons, in the trap)
...	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 Phenomena R package version 0.1.0.9000-73.

**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	<b>numeric (required)</b> : The transition probaility ( $\text{cm}^3/\text{s}$ ).
rho	<b>numeric (required)</b> : The calculated dimesionless Charge density (normally written Rho’).
times	<b>vector (with default)</b> : The number of MC runs.
clusters	<b>numeric (with default)</b> : The number of clusters.
r	<b>numeric (with default)</b> : The retrapping ratio.
delta.r	<b>numeric (with default)</b> :
N_e	<b>numeric (with default)</b> : The number of electrons.
method	<b>character (with default)</b> : sequential 'seq' or parallel processing 'par'
output	<b>character (with default)</b> : output is either the 'signal' (the default) or 'remaining_e' (the remaining charges, electrons, in the trap)
...	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 Phenomena R package version 0.1.0.9000-73.

**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-luminescence (TL) using the one trap one recombination center (OTOR) model. Delocalized refers to involvement of the conduction band.

**Usage**

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



**Arguments**

s	<b>numeric (required)</b> : Escape frequency of the trap (s <sup>-1</sup> ).
E	<b>numeric (required)</b> : Thermal activation energy of the trap (eV).
times	<b>numeric (with default)</b> : the number of MC runs.
clusters	<b>numeric (with default)</b> : the number of clusters.
N_e	<b>integer (with default)</b> : The number of electrons.
n_filled	<b>integer (with default)</b> : The number of electron traps that are filled at the beginning of the simulation.
R	<b>numeric (with default)</b> : The retrapping ratio.
method	<b>character (with default)</b> : sequential 'seq' or parallel processing 'par'
output	<b>character (with default)</b> : output is either the 'signal' (the default) or 'remaining_e' (the remaining charges, electrons, in the trap)
...	further arguments

**Details**

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

where in the function N := N\_e := n := n\_filled

**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 Phenomena R package version 0.1.0.9000-73.

**Author(s)**

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

**References**

Pagonis, V., Friedrich, J., Discher, M., Müller-Kirschbaum, A., Schlosser, V., Kreutzer, S., Chen, R. and Schmidt, C., 2019. Excited state luminescence signals from a random distribution of defects: A new Monte Carlo simulation approach for feldspar. Journal of Luminescence 207, 266–272. doi: [10.1016/j.jlumin.2018.11.024](https://doi.org/10.1016/j.jlumin.2018.11.024)

Reuven, C. and S. McKeever, 1997. Theory of thermoluminescence and related phenomena.

## 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)

#' @examples
##=====##
## Example 2: Plot multiple TL stimulation TL curves in R with varying params
##=====##
## Not run:
# define your parameters
times=seq(100,450,1)
s=rep(3.5e12,4)
E=rep(1.45,4)
R<-c(0.7e-6,1e-6,0.01,0.1)
clusters=1000
N_e =c(400, 500, 700, 400)
n_filled =c(400, 500, 300, 70)
method="par"
output ="signal"
col=c(1,2,3,4) # different colours for the individual curves
plot_uncertainty <- c(TRUE,TRUE,TRUE,TRUE) # do you want to see the uncertainty?
add_TF <- c(FALSE,rep(TRUE, (length(R)-1)))
for (u in 1:length(R)){
  results <-run_MC_TL_DELOC(times=times, s=s[u],E=E[u], 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 = FALSE, col=col[u], main=" your plot", ylim=c(0,20))
}
legend("topright",ncol=5,cex=0.55,title = "parameters" ,legend=c(paste0("E = ", E),
                           paste0("s = ", s),
                           paste0("n_filled = ", n_filled),
                           paste0("N_e = ", N_e),
                           paste0("R = ", R)), text.col=col)

## 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-luminescence (LM-OSL) using the generalized one trap (GOT) model. Localized refers to the lack of involvement of the conduction band.

## Usage

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

## Arguments

s	<b>numeric (required)</b> : Escape frequency of the trap ( $s^{-1}$ ).
E	<b>numeric (required)</b> : Thermal activation energy of the trap (eV).
times	<b>numeric (with default)</b> : The number of Mc runs.
clusters	<b>numeric (with default)</b> : The number of clusters.
n_filled	<b>integer (with default)</b> : The number of electron traps that are filled at the beginning of the simulation.
r	<b>numeric (with default)</b> : The retrapping ratio.
method	<b>character (with default)</b> : sequential 'seq' or parallel processing 'par'
output	<b>character (with default)</b> : output is either the 'signal' (the default) or 'remaining_e' (the remaining charges, electrons, in the trap)
...	further arguments

## Details

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

where in the function  $n := n\_filled$

## Value

This function returns an **array** with dimension  $\text{length}(\text{times}) \times \text{length}(r) \times \text{clusters}$

## 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 Phenomena R package version 0.1.0.9000-73.

## Author(s)

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

## References

Pagonis, V., Friedrich, J., Discher, M., Müller-Kirschbaum, A., Schlosser, V., Kreutzer, S., Chen, R. and Schmidt, C., 2019. Excited state luminescence signals from a random distribution of defects: A new Monte Carlo simulation approach for feldspar. Journal of Luminescence 207, 266–272. doi: [10.1016/j.jlumin.2018.11.024](https://doi.org/10.1016/j.jlumin.2018.11.024)

## 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 using Tunnelling Transition*


---

## Description

Runs a Monte-Carlo (MC) simulation of thermo-luminescence (TL) using the tunneling (TUN) model. Tunneling transitions refers to the direct movement of electrons from a trap directly to the recombination centre.

## 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	<b>list (required)</b> : Escape frequency of the trap ( $s^{-1}$ ).
E	<b>numeric (required)</b> : Thermal activation energy of the trap (eV).
rho	<b>numeric (required)</b> : The calculated dimensionless Charge density.
r_c	<b>numeric (with default)</b> : The dimensionless minimal critical radius.
times	<b>vector (with default)</b> : The number of MC runs.
clusters	<b>numeric (with default)</b> : The number of clusters.
N_e	<b>numeric (with default)</b> : The number of electrons
delta.r	<b>numeric (with default)</b> : The appropriate distance interval along the r axis (dimensionless).
method	<b>character (with default)</b> : sequential 'seq' or parallel processing 'par'
output	<b>character (with default)</b> : output is either the 'signal' (the default) or 'remaining_e' (the remaining charges, electrons, in the trap)
...	further arguments

## Details

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

where in the function  $N := N_e := \rho := \rho'$  } := \code{r\_c} } := \code{\rho'\_c}

## 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., Kreutzer, S., 2019. run\_MC\_TL\_TUN(): Run Monte-Carlo Simulation for TL using Tunnelling 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 0.1.0.9000-73.

## Author(s)

Johannes Friedrich, University of Bayreuth (Germany), Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, 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](#)

Pagonis, V., Friedrich, J., Discher, M., Müller-Kirschbaum, A., Schlosser, V., Kreutzer, S., Chen, R. and Schmidt, C., 2019. Excited state luminescence signals from a random distribution of defects: A new Monte Carlo simulation approach for feldspar. Journal of Luminescence 207, 266–272. doi: [10.1016/j.jlumin.2018.11.024](#)

**Further reading** Aitken, M.J., 1985. Thermoluminescence dating. 276-280. doi: [10.1002/gea.3340020110](#)

## Examples

```
## Not run:
##=====##
## Example 1: Simulate TL measurement
##=====##
run_MC_TL_TUN(s = 3.5e12,
              E = 1.45,
              rho = 0.015,
              r_c = 0.85,
              times = 200:500) %>%
  plot_RLumCarlo(legend = T)

## End(Not run)
```

# Index

array, [3](#), [6](#), [8](#), [10](#), [13](#), [14](#), [17](#), [19](#), [21](#)

character, [3](#), [5](#), [6](#), [8](#), [10](#), [11](#), [13–15](#), [17](#), [19](#), [20](#)

data.frame, [2](#)

integer, [3](#), [6](#), [8](#), [10](#), [13](#), [14](#), [17](#), [19](#)

list, [20](#)

logical, [2](#)

numeric, [2–6](#), [8–11](#), [13–15](#), [17](#), [19](#), [20](#)

plot\_RLumCarlo, [2](#)

run\_MC\_CW\_IRSL\_LOC, [3](#)

run\_MC\_CW\_IRSL\_TUN, [4](#)

run\_MC\_CW\_OSL\_DELOC, [6](#)

run\_MC\_ISO\_DELOC, [8](#)

run\_MC\_ISO\_LOC, [9](#)

run\_MC\_ISO\_TUN, [11](#)

run\_MC\_LM\_OSL\_DELOC, [12](#)

run\_MC\_LM\_OSL\_LOC, [14](#)

run\_MC\_LM\_OSL\_TUN, [15](#)

run\_MC\_TL\_DELOC, [16](#)

run\_MC\_TL\_LOC, [18](#)

run\_MC\_TL\_TUN, [20](#)

vector, [15](#), [20](#)