RLumCarlo: Simulating Cold Light using Monte Carlo Methods

by Sebastian Kreutzer, Johannes Friedrich, Vasilis Pagonis, Christian Laag, Ena Rajovic, and Christoph Schmidt

Abstract The luminescence phenomena of insulators and semiconductors (e.g., natural minerals such as quartz) have various application domains. For instance, Earth Sciences and archaeology exploit luminescence as a dating method. Herein, we present the R package **RLumCarlo** implementing sets of luminescence models to be simulated with Monte Carlo (MC) methods. MC methods make a powerful ally to all kinds of simulation attempts involving stochastic processes. Luminescence production is such a stochastic process in the form of charge (electron-hole pairs) interaction within insulators and semiconductors. To simulate luminescence-signal curves, we distribute single and independent MC processes to virtual MC clusters. **RLumCarlo** comes with a modularized design and consistent user interface: (1) C++ functions represent the modeling core and implement models for specific stimulations modes. (2) R functions give access to combinations of models and stimulation modes, start the simulation and render terminal and graphical feedback. The combination of MC clusters supports the simulation of complex luminescence phenomena.

1 Introduction

Light is perhaps the most basic everyday experience. Light emission that is *not* caused by the heating of a substance is called luminescence or 'cold light'. Various fields exploit this phenomenon. For instance, Earth Sciences and archaeology determine the timing of past events (e.g., last sunlight exposure or heating) with a technique called luminescence dating. Since 2012, the luminescence-dating (or more general trapped-charge dating) community has gradually adapted R as a universal tool to analyze, model, and visualize their data. Relevant related CRAN packages are: **BayLum** (Bayesian modeling: Philippe et al., 2019; Christophe et al., 2018), **Luminescence** (luminescence- data analysis, Kreutzer et al., 2012, 2020), **numOSL** (luminescence-data analysis, Peng et al., 2013; Peng and Li, 2018), **RLumModel** (luminescence-data modeling, Friedrich et al., 2016, 2020), **RLumShiny** (graphical interface to functions for plotting and calculation in the framework of luminescence-data analysis, Burow et al., 2016, 2019), and **tgcd** (curve deconvolution, Peng et al., 2016; Peng, 2020).
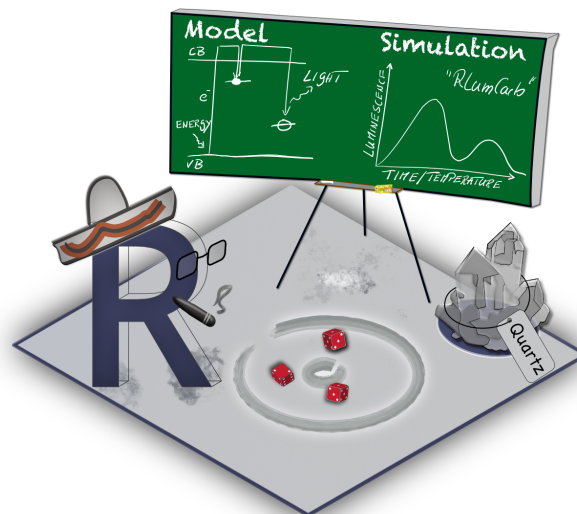

Figure 1: **RLumCarlo** simulates luminescence production in natural minerals such as quartz using Monte Carlo methods. Input is fairly simple (energy-band) models simulating movements of, e.g., electrons in the crystal lattice of quartz. The probability of observing such transitions is a function of energy input in the form of light, heat, or ionizing radiation (e.g., β- or γ-radiation). Th output of **RLumCarlo** is a luminescence curve. This is the light output observed when electrons descend from a higher energy state to a lower energy state emitting part of the energy difference as light.

The luminescence production process is a stochastic process involving discrete random state transitions of subatomic particles. In the case of luminescence, this translates to electrons (and holes) moving to different energy levels, e.g., in the crystal lattice of the natural mineral quartz. Such processes are ideal for Monte Carlo (MC) simulations, and their application has a long and propelling history in physics (cf. Landau and Binder, 2015). Figure 1 summarizes the purpose of **RLumCarlo** developed to simulate luminescence signals in semiconductors and insulators (e.g., quartz) using MC methods. To that end, **RLumCarlo** employs simple (energy-band) models that describe the physical processes in, e.g., the quartz crystal, to simulate xy-curves (luminescence curves). The modeling output expresses the evolution of the light production (luminescence process) over time.

Our contribution, and so **RLumCarlo**, sits on precedent work by Pagonis and Chen (2015), Pagonis et al. (2014), Pagonis et al. (2019), and Pagonis et al. (2020). The included collection of energy-band models for different stimulation modes adapted to MC methods are valuable for, e.g., studying the impact of model parameters on the signal-related stochastic uncertainties or statistic effects in tiny, dosimetric systems. Technically, our approach is closely related to the simulation of birth-and-death processes (for a review on birth-and-death process cf. Novozhilov et al., 2006). Each simulation run describes a Markov process. However, in our case, we allow only a reduction of an initial number of particles (i.e., only death processes).

Herein, we will not derive the full theoretical background of the models, but we will focus on the technical aspects of the package design and the integration of the MC methods. Such a presentation was beyond the scope of previous articles (e.g., Pagonis et al., 2019, 2020), but it is likely of interest to a broader community.

We structured our contribution as follows. The introduction continues with a brief paragraph on luminescence and the term 'cold light'. After that, we detail the rationales for our contribution by recalling conventional modeling approaches in the field. Readers familiar with these topics may safely skip this part. The subsequent section outlines the concept and the implementation of **RLumCarlo**, including code examples. The remainder addresses (A) the implementation of a virtual dosimetric system to simulate weak spatial correlation of dosimetric cluster groups. (B) We outline how **RLumCarlo** can simulate more complex models compared to other solutions, with respect to its strengths and limitations. An outlook outlining the potential to implement more interactions between models will close our manuscript.

'Cold light' in a nutshell

Light emissions of semiconductors or insulators *after* exposure to ionizing radiation is a luminescence phenomenon now and then paraphrased as 'cold light'. The term luminescence relates to light production *not* purely caused by the heating of a substance, a condition called 'incandescence' or black body radiation, but a phenomenon expressing the inherent capacity of a material (dosimeter) to emit light (energy) in the ultraviolet to infrared wavelength range (e.g., Newton Harvey, 1957; Mahesh et al., 1989). Heat-related luminescence phenomena of solids have been explored systematically in physics since the 1930s (Urbach, 1930) to characterize materials and understand charge transfers in dosimeters (e.g., McKeever, 1983; Mahesh et al., 1989). The amount of luminescence, in the context of this manuscript, correlates to the energy absorbed by a dosimeter during ionizing irradiation. The closest analogy to a dosimeter is a battery that can be charged by, e.g., γ-radiation and discharges while emitting light. Natural minerals such as quartz or feldspar are dosimeters. Defects and impurities in their crystal lattice can trap charges (electrons or holes) in metastable states between valence and conduction band. The time an electron spends in such a state can vary from a fraction of a second to millions of years, depending on the crystal lattice configuration and the environmental conditions. The amount of (potential) energy held by an electron in such a center is approximately the energy difference between the valence band and the energy level of the center. A transition of the electron to a lower energy state may lead to a photon emission of the form $E_{photon} = \hbar\omega_{nm} = E_n - E_m$[1]. Energy input ('stimulation') can move the electron out of the defect and eventually it recombines with a hole trapped at another defect ('recombination centre'). Types of stimulation methods relevant for our contribution are heat (thermally stimulated luminescence, TL), visible light (optically stimulated luminescence, OSL), and infrared light (infrared stimulated luminescence, IRSL).

Luminescence phenomena have versatile use in the fields of personal, medical, and accidental dosimetry (e.g., Yukihara and McKeever, 2011). As aforementioned, in Earth Sciences and archaeology, the luminescence of natural minerals gained considerable attractiveness as a dating method (luminescence dating). First attempts exploiting luminescence signals as a chronological tool reach back to the 1950s (Daniels et al., 1953; Houtermans and Stauffer, 1957; Grögler et al., 1958). Nevertheless, it needed a few decades more before the method took off and became today one of the most frequently

---

[1]$\hbar$ (eV s): Planck constant divided by 2π; ω (radians per s): frequency; E_n (eV): higher energy state; E_m (eV): lower energy state

used dating methods on sediments for the last 250,000 years and beyond (e.g., Aitken, 1985, 1998; Bateman, 2019).

2 Towards Monte Carlo simulations

To explain luminescence production, Johnson (1939) and Randall and Wilkins (1945) introduced the first basic energy-band models. Today, most of the commonly accepted luminescence models use series of more or less complicated systems of differential equations (for an overview, see Chen and McKeever, 1997; Bøtter-Jensen et al., 2003; Chen et al., 2011) employing energy-band models. Those models provide a proper phenomenological match with measured data for various experimental designs by simulating electronic transitions. 'Conventional' energy-band models available to simulate luminescence production are developed as a set of nonlinear differential equations. This brings some limitations:

1. The models become complex easily and cannot be solved analytically.
2. If numerical methods are used, some equations are numerically unstable, which may lead to wrong simulation results.
3. A convenient assumption in many of such models is a great abundance of spatially uniformly distributed traps and recombination centers. However, this is not always the most prudent assumption. A spatial correlation and cluster formation of centers may exist for various reasons (cf. Mandowski and Świaltek, 1992; Chen et al., 2011; Horowitz et al., 2017).
4. Deterministic models do not consider stochastic uncertainties and simulated curves are 'noise free'. These limits subsequent analyses for materials where such uncertainties would matter due to the low, finite, number of charge carriers, and in these scenarios, simulation results are used as reference data to test statistical models used for luminescence data analysis in general.

Modeling code for simulating luminescence production was often written with the tools at hand, e.g., *Mathematica* (e.g., Pagonis et al., 2006), which has led to a fragmentation of incompatible solutions. In 2016, Friedrich et al. (2016) introduced **RLumModel** (Friedrich et al., 2020), pooling available kinetic (non-MC) models available for the luminescence production in quartz. A tantamount suite of R code was presented simultaneously by Peng and Pagonis (2016). We will compare results from **RLumModel** and **RLumCarlo** at the end of this manuscript.

MC simulations offer an alternative and are indispensable if the simulation of defect clusters in combination with the analysis of stochastic uncertainties is desired. Usually, the underlying models are very simple, but can be combined to describe complex systems. Important early work simulating TL using MC methods goes back to Mandowski and Świaltek (1992) and Kulkarni (1994). Mandowski and Świaltek (1992) tried to overcome the prerequisite of a large number of sample carriers, and Kulkarni (1994) investigated MC methods to overcome very long calculation times encountered for numerical calculations in particular scenarios. Kulkarni (1994) (p. 103) also reported a "statistical fluctuation" (noise like scatter) caused by the MC simulations but considered this more as a disadvantage. Later, Pagonis et al. (2020) explicitly exploited this as a feature, similar to birth-and-death processes and their related random uncertainties, to investigate specifically the stochastic uncertainties.

Before we start to detail **RLumCarlo**, a preceding note of caution: Any attempt to answer the question of whether a particular model may better explain the one or the other effect measured in luminescence studies would open Pandora's box (e.g., Horowitz et al., 2017). Consequently, we will not engage in such a discussion. What we have implemented so far in **RLumCarlo** can be modified and exchanged. However, the underlying design concept remains applicable.

3 The concept of RLumCarlo

RLumCarlo implements energy-band models in a modular approach. Each model can simulate only an isolated effect (e.g., a single curve, see below), but the package design allows various combinations, e.g., in the form of clusters. Hence, **RLumCarlo** can evolve beyond a specific mathematical model through a combination of simple models.

To that end, **RLumCarlo** differs fundamentally from **RLumModel**, where the collected models allow the simulation of complex phenomena and even entire measurement sequences (Friedrich et al., 2016) but are self-contained by design. In other words, simulations cannot evolve beyond a specific mathematical model selected by the user. In **RLumCarlo**, the implemented energy-band models can simulate only isolated effects (e.g., a single curve, see below), but the package design allows a combination in the form of clusters. Throughout the text, we will use the word 'clusters' to (1) ascribe virtual units used in the MC simulation to run independent random processes (henceforth MC clusters)
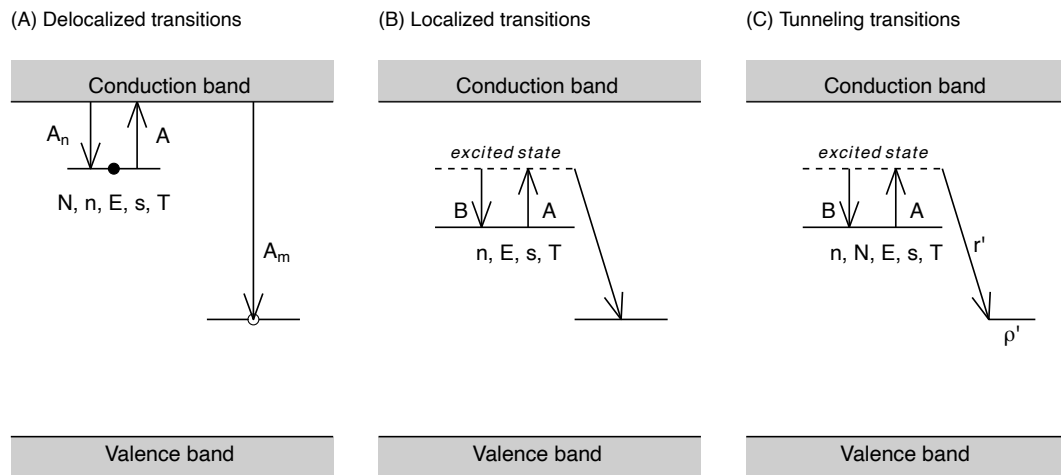
Figure 2: Energy-band model representation of the models implemented in **RLumCarlo**. Letters represent physical model parameters, and arrows indicate allowed transitions. (A) Delocalized transition: the model consists of one single trap and one recombination center. Transition processes involve the conduction band. (B) Localized transition: the model consists of two sub-conduction band energy levels. Charge transitions do not involve the conduction band but take place locally with a constant recombination rate. (C) Tunneling transition: The model consists of one trap and one recombination center. Transitions take place from the excited state into the recombination center without involving the conduction band, and the recombination rate depends on the distance between electrons and holes. All symbols are detailed in the package manual, the package vignette, and the main text. Side note: For the creation of the plots we used **scales** and **ggplot2** (https://github.com/JohannesFriedrich/EnergyBandModels).

and (2) to define groups of defects (defect clusters) e.g., defined by their spatial distance. Only the latter carry a physical meaning.

Implemented energy-band models

To date, **RLumCarlo** ships three simple energy-band models (Figure 2) to simulate luminescence production using (A) delocalized transitions, (B) localized transitions, and (C) excited state tunneling transitions. The models are distinguished by the allowed routes of electrons involved in the luminescence process from one energy state to another. Only the first model (Figure 2A) involves the conduction band, while the models in Figures 2B and C limit the allowed electron pathways to energy levels below the conduction band.

While the parameters differ from model to model and depend on the stimulation mode (heat or light, continuous or ramped), key entities remain alike across the models, such as the trap depth (the energy difference of the electron state from the conduction band) E (eV), the attempt to escape frequency of an electron from the trap (short: frequency factor) s (s^{-1}), the temperature T (K), and the trapped concentration of electrons n (cm^{-3}) in the trap. N (models A and B) is the total number of available electrons in cm^{-3} and ρ' is the dimensionless density of recombination centers (model C, Huntley, 2006). The symbols A_n, A_m, and A (model A), B and A (model B), and B, A, and r' (model C) plotted next to the arrows in Figure 2 parametrize, simply put, the rates of the electronic transitions.

The conditions of the simulations are defined through these parameters, with n being the crucial number. Once the electrons have all recombined, the simulation may still continue, but the luminescence signal is zero. As we will detail below, the essential point of the MC simulation, from the physical point of view, is that these concentrations become dimensionless, absolute numbers in a finite system.

Each model supports up to four different stimulation modes (Figure 3), i.e., the type of energy input (light or heat) and its modulation (continuous or ramped).

As an example, we will detail the mathematical background and its implementation for delocalized transitions below. For all other models, we may refer to the cited literature as well as the package manual.
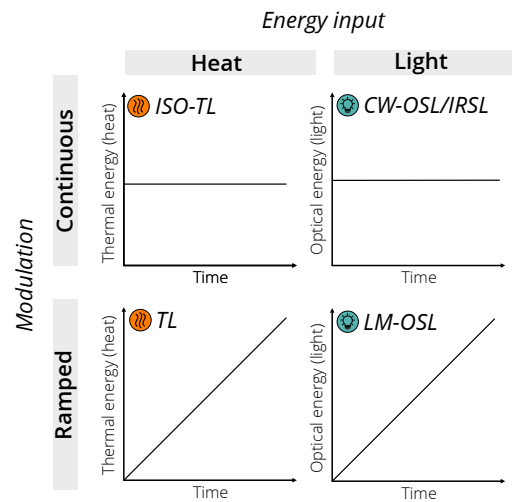
Figure 3: Stimulation modes in **RLumCarlo** applicable to the models. ISO-TL: isothermal TL, i.e., constant stimulation temperature over time. TL: thermal luminescence, i.e., temperature ramps (approximately linearly) over time. CW-OSL/IRSL: continuous wave optically stimulated luminescence respectively infrared stimulated luminescence, constant optical stimulation over time. LM-OSL: linearly modulated optically stimulated luminescence, i.e., linearly ramped optical stimulation over time.

Conceptional overview of the implementation

The basic implementation of the MC processes as a software algorithm consists of two nested loops. The outer loop iterates over a time $0 < t \leqslant t_{max}$ with $t \in \mathbb{R} > 0$. The inner part loops over particles $0 < j \leqslant n$ with $n \in \mathbb{Z}$. The model tests a random number, newly sampled with replacement in each run, against a threshold P. If the sampled random number is smaller than P, the absolute number of particles is reduced by one. The code below shows the basic algorithm outlined above for the radioactive decay, which we have chosen because it can be found in standard textbooks (e.g., Landau and Binder, 2015). Below we used R code for illustrative reasons, while the package implementation is written in C++.

```
n <- 1000
t <- 1:100
P <- 0.2
remaining <- numeric(length(t))

for (t in t) {
  for (j in 1:n) {
    if (runif(1) < P && n > 0)
      n <- n - 1

  }

  if (n > 0)
    remaining[t] <- n

}
```

For example, the algorithm starts with 1,000 particles. In time instant t_1, the random number was smaller for two particles j_6 and j_{576}. Hence, in time instant t_2, the inner loop iterates only over $j = \{1, 2, ..., 998\}$ particles. The absolute number of remaining particles for each t is stored in a vector of length $t_{max}/\Delta t$. This vector is the observed signal curve (in the case of luminescence, the righthand side graph on the green board in Figure 1).

Implementation example for the OTOR model

The implementation for luminescence production in **RLumCarlo** is very similar. To exemplify the adaptation of the models to be run as an MC simulation, we have selected the one-trap one-recombination

center (OTOR) model (based on Halperin and Braner, 1960) for TL. Our description below follows Pagonis and Chen (2015).

The OTOR model for TL can be expressed with the following set of differential equations:

$$\frac{dn}{dt} = -ns\,exp\left(-\frac{E}{k_BT}\right) + n_c(N-n)A_n \tag{1}$$

$$\frac{dn_c}{dt} = -\frac{dn}{dt} - n_cmA_m \tag{2}$$

$$I_{TL}(t) = -\frac{dm}{dt} = n_cmA_m. \tag{3}$$

Beyond already mentioned symbols, we used in the equations I_{TL}, the time-dependent intensity, and n_c (cm^{-3}), the current concentration of electrons in the conduction band. The concentration of recombination centers is represented by m (cm^{-3}), where for reasons of charge neutrality $m = n + n_c$. A_n and A_m (both in cm^3 s^{-1}) are the capture coefficients for traps and recombinations centers, respectively. k_B (eV K^{-1}) is the Boltzmann constant and T (K), the absolute temperature.

By assuming quasi-static equilibrium conditions (Chen et al., 2011)

$$\left|\frac{dn_c}{dt}\right| \ll \left|\frac{dn}{dt}\right|, \left|\frac{dm}{dt}\right| \; ; \; n_c \ll n, \; n \simeq m, \tag{4}$$

the resulting TL intensity becomes the general one trap equation, GOT:

$$I_{TL}(t) = -\frac{dn}{dt} = s\,exp\left(-\frac{E}{k_BT}\right)\frac{A_mn^2}{(N-n)A_n + nA_m}. \tag{5}$$

$$T = T_0 + \beta \times t, \tag{6}$$

with T (K) and T_0 (K) being temperatures, β (K s^{-1}) the (heating) rate, and t (s) the simulation time. $p(t) = s\,exp\left(-\frac{E}{k_BT}\right)$ is the rate of thermal excitation, and $R = \frac{A_n}{A_m}$ is the dimensionless retrapping ratio. The translation into a finite system with a discrete distribution of charge carriers (cf. Mandowski and Swiatek, 1991, 1994), can be expressed through

$$\chi n, \chi N \to \bar{n}, \bar{N}, \tag{7}$$

and the differential equation becomes a difference equation:

$$I_{TL}(t) = -\frac{1}{\beta}\frac{\Delta\bar{n}}{\Delta t} = p(t)\frac{\bar{n}^2}{\bar{N}R + \bar{n}(1-R)}. \tag{8}$$

χ (cm^3) is a constant, $\bar{n}, \bar{N} \in \mathbb{Z}$, and $\Delta t = 1$ s is an appropriate time interval. R is the dimensionless re-trapping ratio in the finite system. To simulate the luminescence process, the related Markov process renders similar to the theory of birth-and-death processes (e.g., Novozhilov et al., 2006), where the population (here of electrons) decreases over continuous time with the probability to observe a transition within Δt being $P = \mu_{\bar{n}}\Delta t$ (here $\mu_{\bar{n}}$ is the "death-rate" in s^{-1}) until the population is depleted. The so-called 'brute force' approach (e.g., Landau and Binder, 2015) tests the population of electrons ($\bar{n}$) per integer time step sequentially by comparing it against a random number sampled with replacement from a continuous distribution $r \sim \mathcal{U}(0,1)$ against the conditional probability P for an electron to get evicted from the trap. In our case, P is calculated as follows:

$$p(t) \times \delta t \times \frac{n}{\bar{N}R + \bar{n}(1-R)}. \tag{9}$$

The factor δt allows values of $\Delta t \neq 1$ while ensuring that $P \ll 1$. $p(t)$ depends on the stimulation mode and the chosen model. For TL (functions named run_MC_TL_<model>()) and isothermal TL (functions named run_MC_ISO_<model>()) applying the localized or delocalized model $p(t)$ becomes:

$$p(t) = s\,exp\left(-\frac{E}{k_BT}\right), \tag{10}$$

and for TL, from tunneling transitions it reads:

$$p(t) = s\,exp\left(-\frac{E}{k_BT}\right)exp\left(-\rho'^{-1/3}r'\right), \tag{11}$$

with ρ' being the dimensionless concentration of recombination centers, and r' being the dimensionless tunneling radius (Huntley, 2006). The basic structure in **RLumCarlo** is, however, identical, except for the models based on excited-state tunneling. Here, an additional outer loop iterates over the dimensionless tunneling radius $0 \leqslant r' \leqslant 2$ (Huntley, 2006).
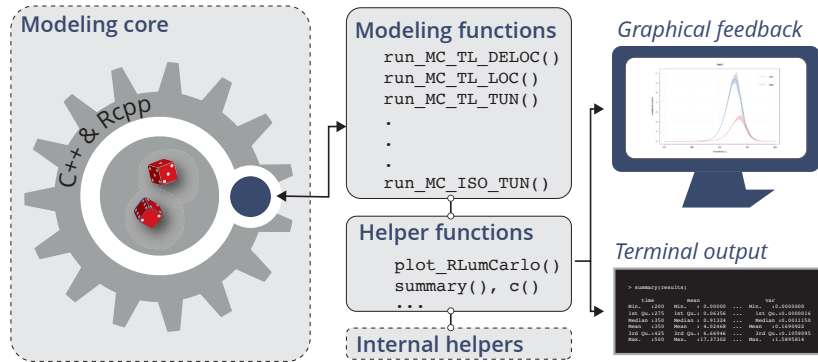
The package design

Figure 4: The conceptional design of **RLumCarlo**. User interaction is realized via exported R functions, one function for each model and stimulation type. For the MC runs, we use C++ functions interfaced via **Rcpp**. Helper functions support graphical and terminal feedback.

In Figure 4, we outline the basic layout of **RLumCarlo**. Two design decisions stand out: (1) Each stimulation mode/energy-band model combination has its own exported R function commencing with the prefix run_MC. (2) These functions interface one C++ function, each via **Rcpp** (Eddelbuettel et al., 2020) (for an overview, see the vignette of **RLumCarlo**). The R functions provide a convenient user interface, and the C++ functions constitute the workhorse, as shown in the modeling core (Figure 4). While the apparent reason for using C++ was speed, the implementation could have been programmed more concisely, i.e., completely in C++ instead of interfacing C++ with R. However, we wanted to allow code inspection by non-specialists from our field, who may wish to implement other models alike. We found that the separation of the user interface (in R) from the modeling core (in C++) aligns best with our premise of simplicity and flexibility.

As indicated above in the example implementation algorithm, each simulation run (Kulkarni, 1994, used the term 'particle tracking') starts with $n > 1$ and ends at t_{max}, while $I(t) = 0$ for $n = 0$. In reality, one has to execute several simulation runs separately (henceforth 'MC clusters', to be distinguished from defect clusters), either to reduce the statistical fluctuation or to estimate the stochastic error (Kulkarni, 1994). **RLumCarlo** runs the simulations in virtual MC clusters on single or multicore systems using **parallel** (R Core Team, 2020), **doParallel** (Microsoft Corporation and Weston, 2020), and **foreach** (Microsoft and Weston, 2020) supported by helper functions (Figure 4), to summarize results and to provide S3-class based graphical output.

4 Simple illustrative examples

Simulations start with a call of the respective function, e.g., for TL using the DELOC model run_MC_TL_DELOC() or run_MC_TL_LOC() for the LOC model, respectively (see Figures 2A-B).

```
results <- run_MC_TL_DELOC(
 s = 3.5e12,
 E = 1.45,
 clusters = 10,
 n_filled = 200,
 R = 1,
 times = 100:450)
```

The function parameter names follow the terminology used for the mathematical expression of the models as closely as possible. For example, E and s are E (trap depth in eV) and s (frequency factor in s^{-1}). R (R in Figure 3) is the so-called re-trapping ratio, expressing the chance an electron of *not* passing into the conduction band, and n_filled is $\bar{n}$, the number of electrons at the beginning of
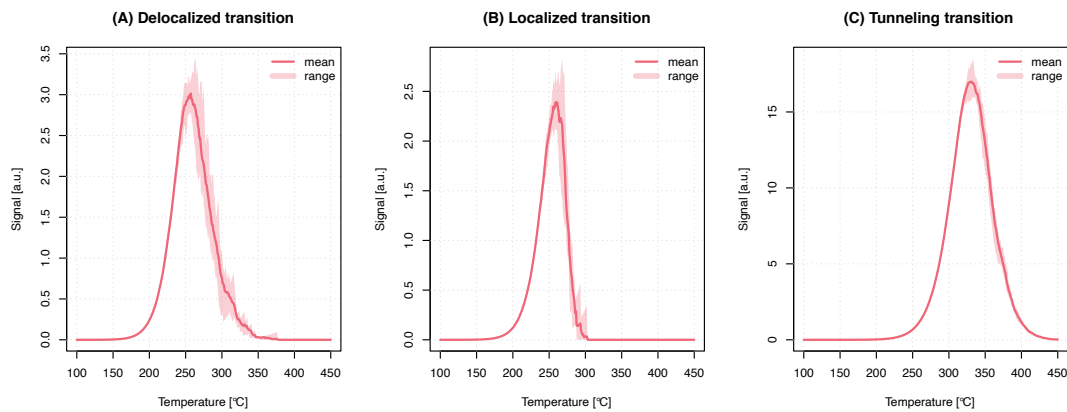
Figure 5: Exemplary comparison of TL signals simulated for all three **RLumCarlo** models: (A) delocalized (DELOC), (B) localized (LOC), and (C) tunneling (TUN). General physical parameters, such as E (1.45 eV), s (3.5×10^{12} s^{-1}), and the stimulation temperature (100–450 °C) were kept constant for all models.
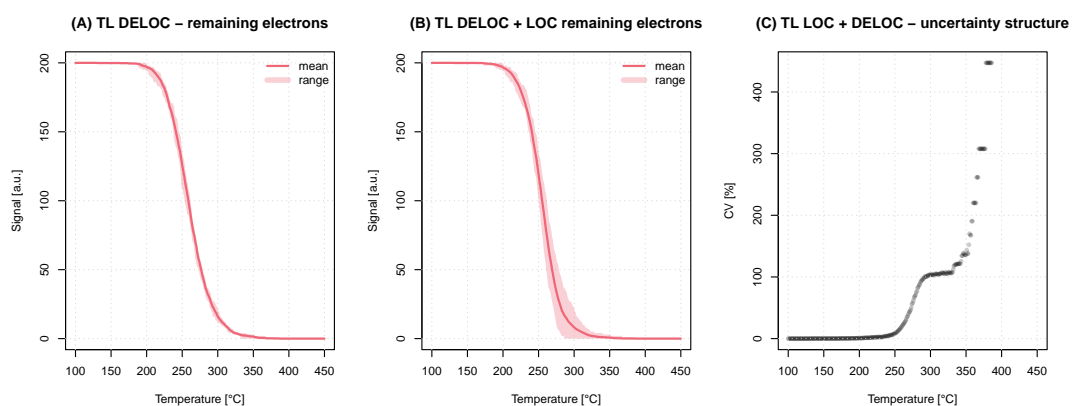

Figure 6: (A) Plot of the remaining electrons for the TL process using the delocalized transition model for which we show the luminescence signal in Figure 5A. (B) Plot of remaining electrons for two models combined in one system. (C) Stochastic uncertainty structure from (B).

the simulation. The duration of the simulation on the time domain (which is not the duration of the computation) is set by `times`. The parameter `clusters` sets the number of MC clusters, i.e., the number of Markov chains. High numbers in `clusters` increase the confidence in the simulation output at the cost of more computation time.

The output can be passed to a dedicated plot function (`plot_RLumCarlo()`). The function supports a couple of standard plot arguments, such as `main` for the title of the plot, which is passed down to `graphics::plot.default()` via `...` (type `?dots` in the R terminal).

```
plot_RLumCarlo(
  object = results,
  legend = TRUE,
  main = "(A) Delocalized transition")
```

The parameter `n_filled` can be a vector enabling different starting conditions for each MC cluster. Figure 5A shows the graphical output for delocalized transitions along with the simulation results for TL stimulation using localized (Figure 5B) and tunneling transitions (Figure 5C). The output is an object of class `RLumCarlo_Model_Output`, which is a `list` comprising a multi-dimensional `array` (one slice per MC cluster) with the resulting luminescence signal and a `numeric` vector for the stimulation time.

Currently we provide S3-generics for `summary()` and `c()`. The first one is also used internally by `plot_RLumCarlo()` to melt the array into a `data.frame` before plotting. The plot output adapts to the used stimulation mode provided via an attribute with each output object.

A straightforward application for this kind of simulation is the study of the impact of physical parameters on the luminescence signal output and the estimation of the stochastic uncertainties, which cannot be achieved with the deterministic approach of differential equations.

We provide more, always up-to-date examples with the package vignette, where we also compiled a table with meaningful physical parameter ranges for each model.

Advanced examples and further considerations

The examples so far presented may not appear very sophisticated, and still, they allow insight that goes beyond a simple educational purpose of simulating luminescence based on phenomenological models. Pagonis et al. (2020), who used a preliminary version of **RLumCarlo**, addressed in detail the stochastic uncertainties of TL and OSL models. These uncertainties come into play in nano-dosimetric materials with a small number of defect clusters where the "finite-size" (Mandowski and Swiatek, 1991) of the system starts to matter in terms of a presumed spatial correlation of defect cluster groups. To some extent, this should also be true for systems exposed to high-energy radiation causing defect clusters (e.g., Mandowski and Swiatek, 1991; Mandowski and Świaltek, 1992). Previously in this paper, we have used the term 'MC clusters'. For a start, in **RLumCarlo**, 'MC clusters' entail independent and continuous Monte Carlo Markov chains employed to simulate luminescence production, starting with a particular number of electrons in the system. Whether the processes are run in parallel or sequentially has no impact on the outcome, except for computation speed. In other words, 'MC clusters' carry no meaning regarding the underlying physics. However, as mentioned above, 'MC clusters' from different models (with the same stimulation mode) can be concatenated (see Figure 6B-C) to simulate defect clusters (also, dosimetric clusters), to which we can attribute physical meaning.

Spatial correlation

To simulate a three-dimensional (dosimetric) system, we can add meaning to MC clusters by reinterpreting them as dosimetric clusters. From the modeling perspective, nothing changes, but MC clusters gain a connotation of having a physical meaning.

Figure 7A illustrates the situation of model combinations transferred into a virtual, three-dimensional dosimetric system. Since all defect clusters are distributed evenly over the system, the distance to each neighboring point is identical, and it is a constant rather than a variable. In other words, the spatial distance between neighboring points does not matter and is of no relevance for the simulation but here chosen for illustrative reasons only. Figure 7B represents a situation that takes one step further. Here, the points are randomly distributed over the system, and points form groups (defect cluster groups). Additionally, **RLumCarlo** supports the mixing of models for the same stimulation mode as in Figure 7A (not shown here). The driving idea of the implementation is the assumption of an individual spatial ordering of defects in a, e.g., quartz crystal to which the luminescence production process might be assigned based on models mentioned above.

Such a system can be created in **RLumCarlo** via `create_ClusterSystem()`. The function distributes points randomly with their coordinates:

$$x_1, y_1, z_1, ..., x_k, y_k, z_k \sim \mathcal{U}(0,1) \mid k \in \mathbb{Z}. \tag{12}$$
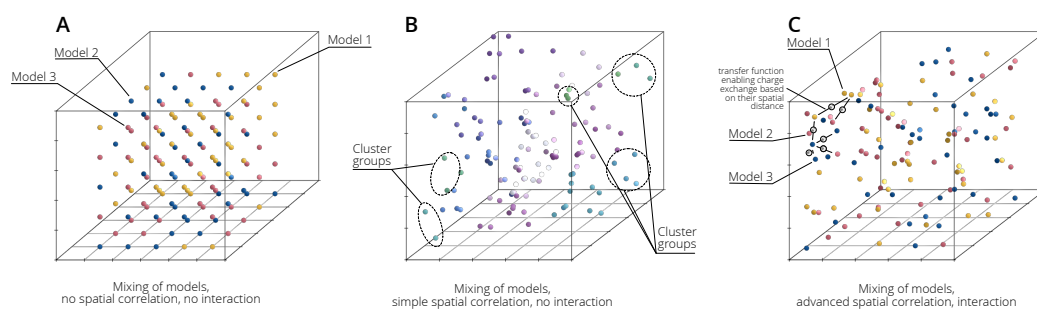

Figure 7: A dosimetric system with three possible approaches of cluster correlation and interaction. (A) Models can be mixed, but no spatial correlation is realized, and no interaction possible. This is the basic mode in **RLumCarlo**. (B) Clusters are grouped by their Euclidean distance, and models can be mixed. Electrons are distributed according to the spatial distance of clusters and also models can be mixed (not shown in the figure). The advanced mode in **RLumCarlo**. (C) Clusters can interact with each other and even exchange electrons. This last stage is subject to future developments of **RLumCarlo**.

Then, the Euclidean distance between the points is determined with `stats:dist()`, which is used by `stats:hclust()` to group the defect clusters (Ξ). To avoid too many small groups, we then cut the cluster tree using `stats:cutree()`, with the outcome shown in Figure 7B. The selection of `stats:hclust()` and `stats:cutree()` for defining the clusters is somewhat arbitrary and might be refined in the future. Therefore, more research, supported by measurements, is needed.

Now, any function from **RLumCarlo** can be used, and the output of `create_ClusterSystem()` is taken as input for the argument `clusters`. For example:

```
run_MC_TL_LOC(s = 3.5e12, E = 1.45, n_filled = 1000,
  clusters = create_ClusterSystem(100))
```

creates a system with 100 randomly distributed defect clusters. If the simulation is run in such a mode, the meaning of `n_filled` changes. Previously, it defined the number of electrons in each cluster ($\bar{n}_{cl_i}$). However, now the same parameter defines the total number of electrons in the entire system. The number of electrons in the ith cluster ($\bar{n}_{cl_i}$) is then an integer fraction of electrons available in each cluster group ($\bar{n}_{\Xi_i} = $ `n_filled`$/N_\Xi$, with N_Ξ the total number of cluster groups). The more clusters are in one group, the less electrons are available per cluster in the group and vice versa. While this is a very simple approach, it allows us to simulate basic spatial correlation. Figure 7C drafts a better way of mimicking spatial interaction of clusters, which is, however, not yet part of **RLumCarlo**. While it would be, based on the designed system, easy from the programming perspective, the needed equations to describe to exchange electrons are yet to be developed.

Comparison to RLumModel

In the remainder, we want to compare simulation results from **RLumCarlo**, with other types of solutions, such as **RLumModel** which uses coupled differential equations to simulate luminescence production. **RLumModel** was selected since it was developed by some of the authors of this contribution. However, in theory, simple scripts using any existing models to simulate luminescence should work as well (as long as the models are comparable).

In contrast to **RLumCarlo**, **RLumModel** input values for physical parameters are preset. **RLumModel** encourages users to write a virtual luminescence signal measurement sequence, which is processed based on a pre-defined model with preset physical parameters.

For the comparison, we have selected a TL curve simulated with the luminescence model for quartz by Bailey (2001).

```
output <- RLumModel::model_LuminescenceSignals(
  sequence = list(IRR = c(20, 10, 1), TL = c(20, 400, 1)),
  model = "Bailey2001"
)
```

The results are shown in Figure 8 (here already with the simulation results from **RLumCarlo** plotted on top of it). The output of **RLumModel** is a three-peak-shaped curve. To simulate the same curve in **RLumCarlo**, we used the parameters from the model by Bailey (2001) (his Table 1), e.g., for the first peak:

```
TL110 <- RLumCarlo::run_MC_TL_DELOC(
  s = 5e+12, E = 0.97, R = 5e-10, times = seq(20, 400, 2),
  N_e = output$`conc. level 1 (TL)`[1,2] / 1e+5)
```

`N_e` was divided by a constant to reduce the computation time. The dimensionless parameter R corresponds to B (s^{-1}) in Bailey (2001). The other two peaks were simulated alike (objects `TL230` and `TL325`) before all three objects were combined via:

```
object <- c(TL110, TL230, TL325)
```

and plotted on the top of the curve derived from **RLumModel**:

```
RLumCarlo::plot_RLumCarlo(
  object = object,
  plot_value = "sum",
  add = TRUE,
  FUN = function(x) {
    x * 1/(1 + (1e+7 * exp(-0.61/(8.617e-5 * (object$time + 273)))))}
)
```
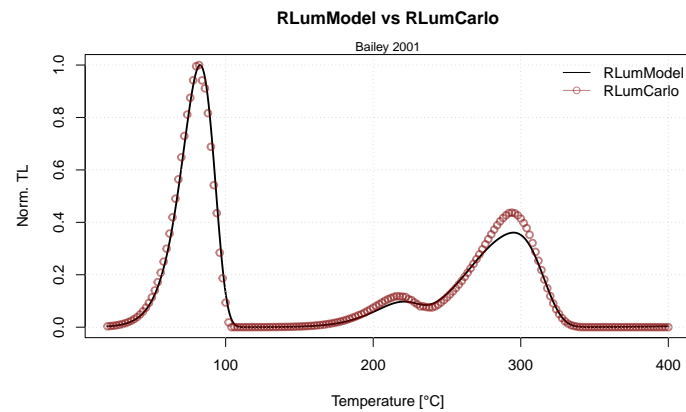
Figure 8: Simulation results of **RLumModel** and **RLumCarlo**. Qualitatively both approaches show a good match.

The argument `plot_value = "sum"` was used to plot the total count sum instead of its average. The additional function injected via the argument `FUN` corrects the TL curves for a phenomenon known as thermal quenching (Wintle, 1975). This is a reduction of luminescence production efficiency at higher temperatures. The chosen quenching parameters follow roughly data measured for quartz by Friedrich et al. (2018). In summary, the results in Figure 8 show that even complex luminescence models can be simulated through the combination of clusters, which brings us back to the initial 'simplicity' premise of **RLumCarlo**. Still, a big 'but' remains. Luminescence models such as those proposed by Bailey (2001) or Pagonis et al. (2008) go beyond single curve simulations. Their purpose is to deliver a general kinetic model for luminescence production of, e.g., quartz, including the simulation of trap filling by irradiation and the simulation of the thermal activation history of the mineral. By contrast, so far all simulations in **RLumCarlo** start with a predefined number of electrons in a trap and are not by default limited to a specific dosimeter. **RLumCarlo** can model more complex luminescence phenomena, but not in a pre-described way out of the box. Instead, **RLumCarlo** is more like a patch box with each model representing a socket ready to be flexibly rewired in many ways to simulate cascades of luminescence production. Due to the nature of the chosen MC approach, in theory (adhering to the patch box picture), the number of sockets is not limited.

5 Summary

The modeling of luminescence phenomena (cold light) of semiconductors and insulators after having received ionizing radiation is a challenging task. MC methods allow setting up flexible and simple systems to simulate luminescence with a finite number of charge carriers. This enables users to address effects usually observed for nano-dosimetric systems, and it provides insight into the stochastic uncertainty structure. We presented **RLumCarlo**, which renders, to our best knowledge, the first open-source and ready-to-use compilation of basic MC luminescence models for different stimulation modes (so far CW-OSL, LM-OSL, ISO-TL, and TL). We showed that the output from different models, which are simulated in separate MC chains in virtual clusters, can be combined to either simulate more complex systems or to mimic simple spatial correlations between cluster groups. The way of the implementation does not limit **RLumCarlo** to a specific dosimeter (e.g., quartz). In this light, **RLumCarlo** can be used in education, and research to test the impact of model parameters, such as cluster sizes and related stochastic uncertainties. Furthermore, **RLumCarlo** can help in in formulating research hypotheses and test them with commonly accepted or new models, still to be developed.

Future work will implement more models to run as MC simulation, e.g., for irradiation processes in crystals (including its luminescence output: radiofluorescence) and for an advanced interaction of clusters.

6 Acknowledgements

Bibliography

M. J. Aitken. *Thermoluminescence dating*. Studies in archaeological science. Academic Press, 1985. [p353]

M. J. Aitken. *An Introduction to Optical Dating*. Oxford University Press, 1998. [p353]

R. M. Bailey. Towards a general kinetic model for optically and thermally stimulated luminescence of quartz. *Radiation Measurements*, 33(1):17–45, 2001. URL https://doi.org/10.1016/S1350-4487(00)00100-1. [p360, 361]

M. D. Bateman. *Handbook of Luminescence Dating*. Whittles Publishing, 2019. [p353]

L. Bøtter-Jensen, S. W. S. McKeever, and A. G. Wintle. *Optically Stimulated Luminescence Dosimetry*. Elsevier Science B.V., 2003. [p353]

C. Burow, S. Kreutzer, M. Dietze, M. C. Fuchs, M. Fischer, C. Schmidt, and H. Brückner. RLumShiny - A graphical user interface for the R Package 'Luminescence'. *Ancient TL*, 34:22–32, 2016. URL http://ancienttl.org/ATL_34-2_2016/ATL_34-2_Burow_p22-32.pdf. [p351]

C. Burow, U. T. Wolpert, and S. Kreutzer. *RLumShiny: 'Shiny' Applications for the R Package 'Luminescence'*, 2019. URL https://CRAN.R-project.org/package=RLumShiny. R package version 0.2.2. [p351]

R. Chen and S. W. S. McKeever. *Theory of Thermoluminescence and Related Phenomena*. WORLD SCIENTIFIC, 1997. [p353]

R. Chen, J. L. Lawless, and V. Pagonis. A model for explaining the concentration quenching of thermoluminescence. *Radiation Measurements*, 46:1380–1384, 2011. URL https://doi.org/10.1016/j.radmeas.2011.01.022. [p353, 356]

C. Christophe, A. Philippe, S. Kreutzer, and G. Guerin. *BayLum: Chronological Bayesian Models Integrating Optically Stimulated Luminescence and Radiocarbon Age Dating*, 2018. URL https://CRAN.R-project.org/package=BayLum. R package version 0.1.3. [p351]

F. Daniels, C. A. Boyd, and D. F. Saunders. Thermoluminescence as a Research Tool. *Science*, 117(3040):343–349, 1953. URL https://doi.org/10.1126/science.117.3040.343. [p352]

D. Eddelbuettel, R. Francois, J. Allaire, K. Ushey, Q. Kou, N. Russell, D. Bates, and J. Chambers. *Rcpp: Seamless R and C++ Integration*, 2020. URL https://CRAN.R-project.org/package=Rcpp. R package version 1.0.5. [p357]

J. Friedrich, S. Kreutzer, and C. Schmidt. Solving ordinary differential equations to understand luminescence: 'RLumModel' an advanced research tool for simulating luminescence in quartz using R. *Quaternary Geochronology*, 35(C):88–100, 2016. URL https://doi.org/10.1016/j.quageo.2016.05.004. [p351, 353]

J. Friedrich, S. Kreutzer, and C. Schmidt. Radiofluorescence as a detection tool for quartz luminescence quenching processes. *Radiation Measurements*, 120:33–40, 2018. URL https://doi.org/10.1016/j.radmeas.2018.03.008. [p361]

J. Friedrich, S. Kreutzer, and C. Schmidt. *RLumModel: Solving Ordinary Differential Equations to Understand Luminescence*, 2020. URL https://CRAN.R-project.org/package=RLumModel. R package version 0.2.7. [p351, 353]

N. Grögler, F. G. Houtermans, and H. Stauffer. Radiation damage as a research tool for geology and prehistory. In *5° Rassegna Internazionale Elettronica E Nucleare, Supplemento Agli Atti Del Congresso Scientifico*, pages 5–15. Sezione Nucleare, 1958. [p352]

A. Halperin and A. A. Braner. Evaluation of Thermal Activation Energies from Glow Curves. *Physical Review*, 117(2):408–415, 1960. URL https://doi.org/10.1103/PhysRev.117.408. [p356]

Y. S. Horowitz, I. Eliyahu, and L. Oster. Kinetic Simulations of Thermoluminescence Dose Response: Long Overdue Confrontation with the Effects of Ionisation Density. *Radiation Protection Dosimetry*, 172(4):524–540, 2017. URL https://doi.org/10.1093/rpd/ncv495. [p353]

F. G. Houtermans and H. Stauffer. Thermolumineszenz als Mittel zur Untersuchung der Temperatur - und Strahlungsgeschichte von Mineralien und Gesteinen. *Helvetica Physica Acta*, 30:274–277, 1957. [p352]

D. J. Huntley. An explanation of the power-law decay of luminescence. *Journal of Physics: Condensed Matter*, 18(4):1359–1365, 2006. URL https://doi.org/10.1088/0953-8984/18/4/020. [p354, 357]

R. P. Johnson. Luminescence of Sulphide and Silicate Phosphors. *Journal of the Optical Society of America*, 29(9):387–391, 1939. URL https://doi.org/10.1364/JOSA.29.000387. [p353]

S. Kreutzer, C. Schmidt, M. C. Fuchs, M. Dietze, M. Fischer, and M. Fuchs. Introducing an R package for luminescence dating analysis. *Ancient TL*, 30(1):1–8, 2012. URL http://ancienttl.org/ATL_30-1_2012/ATL_30-1_Kreutzer_p1-8.pdf. [p351]

S. Kreutzer, C. Burow, M. Dietze, M. C. Fuchs, C. Schmidt, M. Fischer, J. Friedrich, S. Riedesel, M. Autzen, and D. Mittelstrass. *Luminescence: Comprehensive Luminescence Dating Data Analysis*, 2020. URL https://CRAN.R-project.org/package=Luminescence. R package version 0.9.8. [p351]

R. N. Kulkarni. The Development of the Monte Carlo Method for the Calculation of the Thermoluminescence Intensity and the Thermally Stimulated Conductivity. *Radiation Protection Dosimetry*, 51(2): 95–105, 1994. URL https://doi.org/10.1093/oxfordjournals.rpd.a082126. [p353, 357]

D. P. Landau and K. Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge University Press, 2015. [p352, 355, 356]

K. Mahesh, P. S. Weng, and C. Furetta. *Thermoluminescence in solids and its application*. Thermoluminescence in solids and its application. Nuclear Technology Publishing, England, 1989. [p352]

A. Mandowski and J. Świaltek. Monte Carlo simulation of thermally stimulated relaxation kinetics of carrier trapping in microcrystalline and two-dimensional solids. *Philosophical Magazine B*, 65(4): 729–732, 1992. URL https://doi.org/10.1080/13642819208204910. [p353, 359]

A. Mandowski and J. Swiatek. On the determination of trap parameters from TSC spectra in finite-size systems. In *7th International Symposium on Electrets (ISE 7)*, pages 588–593. IEEE, 1991. URL https://doi.org/10.1109/ISE.1991.167278. [p356, 359]

A. Mandowski and J. Swiatek. Monte Carlo simulation of TSC and TL in spatially correlated systems. In *8th International Symposium on Electrets (ISE 8)*, pages 461–466. IEEE, 1994. URL https://doi.org/10.1109/ISE.1994.514813. [p356]

S. W. S. McKeever. *Thermoluminescence of solids*. Thermoluminescence of solids. Cambrigde University Press, 1983. [p352]

Microsoft and S. Weston. *foreach: Provides Foreach Looping Construct*, 2020. URL https://CRAN.R-project.org/package=foreach. R package version 1.5.1. [p357]

Microsoft Corporation and S. Weston. *doParallel: Foreach Parallel Adaptor for the 'parallel' Package*, 2020. URL https://CRAN.R-project.org/package=doParallel. R package version 1.0.16. [p357]

E. Newton Harvey. *A history of luminescence from the earliest times until 1900*. Philadelphia, American Philosophical Society, 1957. [p352]

A. S. Novozhilov, G. P. Karev, and E. V. Koonin. Biological applications of the theory of birth-and-death processes. *Briefings in Bioinformatics*, 7(1):70–85, 2006. URL https://doi.org/10.1093/bib/bbk006. [p352, 356]

V. Pagonis and R. Chen. Monte Carlo simulations of TL and OSL in nanodosimetric materials and feldspars. *Radiation Measurements*, 81:262–269, 2015. URL https://doi.org/10.1016/j.radmeas.2014.12.009. [p352, 356]

V. Pagonis, G. Kitis, and C. Furetta. *Numerical and Practical Exercises in Thermoluminescence*. Springer, 2006. [p353]

V. Pagonis, E. Balsamo, C. Barnold, K. Duling, and S. McCole. Simulations of the predose technique for retrospective dosimetry and authenticity testing. *Radiation Measurements*, 43(8):1343–1353, 2008. URL https://doi.org/10.1016/j.radmeas.2008.04.095. [p361]

V. Pagonis, E. Gochnour, M. Hennessey, and C. Knower. Monte Carlo simulations of luminescence processes under quasi-equilibrium (QE) conditions. *Radiation Measurements*, 67(C):67–76, 2014. URL https://doi.org/10.1016/j.radmeas.2014.06.005. [p352]

V. Pagonis, J. Friedrich, M. Discher, A. Müller-Kirschbaum, V. Schlosser, S. Kreutzer, R. Chen, and C. Schmidt. Excited state luminescence signals from a random distribution of defects - A new Monte Carlo simulation approach for feldspar. *Journal of Luminescence*, 207:266–272, 2019. URL https://doi.org/10.1016/j.jlumin.2018.11.024. [p352]

V. Pagonis, S. Kreutzer, A. R. Duncan, E. Rajovic, C. Laag, and C. Schmidt. On the stochastic uncertainties of thermally and optically stimulated luminescence signals: A Monte Carlo approach. *Journal of Luminescence*, 219:116945, 2020. URL https://doi.org/10.1016/j.jlumin.2019.116945. [p352, 353, 359]

J. Peng. *tgcd: Thermoluminescence Glow Curve Deconvolution*, 2020. URL https://CRAN.R-project.org/package=tgcd. R package version 2.5. [p351]

J. Peng and B. Li. *numOSL: Numeric Routines for Optically Stimulated Luminescence Dating*, 2018. URL https://CRAN.R-project.org/package=numOSL. R package version 2.6. [p351]

J. Peng and V. Pagonis. Simulating comprehensive kinetic models for quartz luminescence using the R program KMS. *Radiation Measurements*, 86:63–70, 2016. URL https://doi.org/10.1016/j.radmeas.2016.01.022. [p353]

J. Peng, Z. Dong, F. Han, H. Long, and X. Liu. R package numOSL: numeric routines for optically stimulated luminescence dating. *Ancient TL*, 31(2):41–48, 2013. URL http://ancienttl.org/ATL_31-2_2013/ATL_31-2_Peng_p41-48.pdf. [p351]

J. Peng, Z. Dong, and F. Han. tgcd: An R package for analyzing thermoluminescence glow curves. *SoftwareX*, pages 1–9, 2016. URL https://doi.org/10.1016/j.softx.2016.06.001. [p351]

A. Philippe, G. Guérin, and S. Kreutzer. BayLum - An R package for Bayesian analysis of OSL ages: An introduction. *Quaternary Geochronology*, 49:16–24, 2019. URL https://doi.org/10.1016/j.quageo.2018.05.009. [p351]

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020. URL https://www.R-project.org/. [p357]

J. T. Randall and M. H. F. Wilkins. Phosphorescence and Electron Traps. I. The Study of Trap Distributions. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 184 (999):365–389, 1945. URL https://doi.org/10.1098/rspa.1945.0024. [p353]

F. Urbach. Zur Lumineszenz der Alkalihalogenide II. Messungsmethoden; erste Ergebnisse; zur Theorie der Thermolumineszenz. *Sitzungsberichte der Akademie der Wissenschaften in Wien, Mathematisch-Naturwissenschaftliche Klasse, Abt. IIa, Mathematik, Astronomie, Physik, Meteorologie und Technik*, 139: 363–372, 1930. [p352]

A. G. Wintle. Thermal Quenching of Thermoluminescence in Quartz. *Geophysical Journal of the Royal Astronomical Society*, 41:107–113, 1975. URL https://doi.org/10.1111/j.1365-246X.1975.tb05487.x. [p361]

E. G. Yukihara and S. W. S. McKeever. *Optically Stimulated Luminescence*. Wiley, 2011. [p352]

Sebastian Kreutzer
Department of Geography & Earth Sciences, Aberystwyth University
Aberystwyth
SY23 3DB, Wales, United Kingdom
IRAMAT-CRP2A, UMR 5060, CNRS-Université Bordeaux Montaigne
Maison de l'Archéologie
Esplanade des Antilles
36607 Pessac Cedex, France

ORCiD: 0000-0001-9166-4563
sebastian.kreutzer@aber.ac.uk

Johannes Friedrich
Chair of Geomorphology, University of Bayreuth

Universitätsstr. 30
95447 Bayreuth, Germany

ORCiD: 0000-0002-0805-9547
johannes.friedrich@posteo.de

Vasilis Pagonis
Physics Department, McDaniel College
Westminster, MD 21157, USA

ORCiD: 0000-0002-4852-9312
vpagonis@mcdaniel.edu

Christian Laag
Université de Paris, Institut de Physique du Globe de Paris, CNRS
75005 Paris, France
Chair of Geomorphology, University of Bayreuth
95447 Bayreuth, Germany

ORCiD: 0000-0002-6012-1029
laag@ipgp.fr

Ena Rajovic
Chair of Geomorphology, University of Bayreuth
95447 Bayreuth, Germany

ena.rajovic@uni-bayreuth.de

Christoph Schmidt
Institute of Earth Surface Dynamics, University of Lausanne
1015 Lausanne, Switzerland

ORCiD: 0000-0002-2309-3209
christoph.schmidt@unil.ch