Vasilis Pagonis

Luminescence Dosimetry:

Signal Analysis Using Python

EXPECTED DATE OF BOOK PUBLICATION: JUNE 2022

Springer Nature

Preface

ABOUT THIS PDF FILE

This document is a short preview of the book "*Luminescence Dosimetry: Signal Analysis Using Python*" to be published by *Springer International*, which has a tentative date of publication of June 2022.

The document features detailed worked examples of Python code, which were presented in the LED 2021 conference. Users can run immediately the Python codes, and modify them for their experimental data and to explore the various luminescence models. The theory and equations behind each subject are briefly summarized, and appropriate references are given from the literature, so that researchers can look up the details of the theory and the relevant experiments. Each Python code includes comments explaining the structure and the various parts of the code. Researchers can use the Python codes to analyze their own experimental data, and to extract the various parameters describing mathematically the luminescence signals.

USING THE PYTHON CODES

This document assumes some basic knowledge of Python, however I believe that it will be useful for both newcomers to Python, as well as to experienced programmers who wish to learn more about the various luminescence phenomena. The Python codes for all programs in the book are available for downloading from the author's repository at GitHub.

The various Python codes are self contained and ready to run, and represent new codes written by the author.

Experienced programmers will certainly find out that they can improve the Python codes given here, and it is of course possible to make the codes more compact and elegant. However, I chose to provide Python codes which are simple and clear, and which can be easily modified for the purposes of the reader, rather than attempting to create compact codes which may be difficult to follow. I have kept the number of required external Python packages intentionally at a minimum, so that newcomers can follow the Python codes easily. All figures in this document were produced using the Python codes in the book, so that users know immediately what to expect when they run the codes. Additional drawings for the various luminescence models in the book were drawn using *Inkscape*, and the book was produced overall using *Ly*X.

The presented codes and models fall within two broad categories, based on delocalized and localized transitions. In delocalized transition models the conduction and/or valence band participate in the luminescence process. By contrast, in the localized type of models the luminescence process does not involve the energy bands, but rather takes place between the ground an/or excited state of the trapped electron/hole, and an energy level of the recombination center.

This document introduces general examples of experimental luminescence data, and discusses the various experimental techniques for measuring the luminescence signals. These signals include thermoluminescence (TL), Optically stimulated luminescence (OSL), infrared stimulated luminescence (IRSL), as well as the commonly used experimental modes of continuous wave (CW-OSL or CW-IRSL), linearly modulated (LM-OSL or LM-IRSL), constant heating rate TL, isothermal TL (ITL) and time-resolved (TR).

Please keep in mind that these Python codes are a work in progress. While every effort has been and will be made to check their accuracy, these Python codes are nevertheless still under costruction. Users should be aware of this important fact while using these codes in their research.

Westminster, Maryland

Vasilis Pagonis
September 2021

LIST OF ACRONYMS

AF	Anomalous fading
CW-IRSL	Continuous-wave infrared stimulated luminescence
CW-OSL	Continuous-wave optically stimulated luminescence
EST	Excited state tunneling model
FOK	First order kinetics
GOK	General order kinetics
GST	Ground state tunneling model
GOT	General one trap model
IGST	Irradiation ground state model
IMTS	Interactive multi-trap system model
IRSL	Infrared stimulated luminescence
ITL	Isothermal luminescence
KP	Kitis-Pagonis general equation
KP-CW	Kitis-Pagonis CW-IRSL equation
KP-TL	Kitis-Pagonis TL equation
KV	Kitis-Vlachos general equation
KV-CW	Kitis-Vlachos CW-OSL equation
KV-TL	Kitis-Vlachos TL equation
LM-IRSL	Linearly-modulated infrared stimulated luminescence
LM-OSL	Linearly-modulated optically stimulated luminescence
LT	Localized transitions model
MOK	Mixed order kinetics
NMTS	Non interactive multi-trap system model
OSL	Optically stimulated luminescence
OTOR	One trap one recombination center model
PKC	Pagonis-Kitis-Chen dose response equation
PKC-S	Pagonis-Kitis-Chen superlinearity equation
POSL	Pulsed optically stimulated luminescence
PIRSL	Pulsed infrared stimulated luminescence
SLT	Semilocalized transitions model
TA-EST	Thermally-assisted excited state tunneling model
TL	Thermoluminescence
TR-IRSL	Time resolved infrared stimulated luminescence
TR-OSL	Time resolved optically stimulated luminescence

Contents

List of Codes

Chapter 1
TL SIGNALS FROM DELOCALIZED TRANSITIONS: DATA ANALYSIS

1.1 Introduction

In this section we provide an overview of the contents and organization of this chapter on the computerized analysis of experimental TL signals from delocalized transitions.

In Sect.1.2 section we introduce the general principles and mathematical aspects of computerized glow curve deconvolution (CGCD). This is followed in Sect.1.3 by a presentation of the various CGCD equations and their transformed equivalent forms. The CGCD examples in this chapter are based on the `optimize` function in Python, which is discussed in Sect.1.4. Sections 1.5-1.6 contain examples of carrying out CGCD analysis using the analytical solution of the GOT model for TL, i.e. the KV-TL equation and its transformed equivalent form. The CGCD of single peak TL glow curves using mixed order and general order kinetics (MOK, GOK) is discussed in Sects.1.7-1.10.

Sections 1.11-1.14 give example of CGCD using the transformed KV-TL and transformed GOK equation, to analyze the complex TL glow curve of the popular dosimetric material LiF.

1.2 Introduction to the deconvolution of TL glow curves

The subject of computerized curve fitting analysis is an essential part of analysis of TL signals, and several sophisticated CGCD techniques have been developed. The deconvolution of complex glow-curves into their individual glow-peaks is widely applied for dosimetric purposes and for evaluating the trapping parameters E and s using curve fitting methods.

The books by Chen and McKeever[12] and Chen and Pagonis [13] summarized the curve fitting procedures commonly used to analyze multi-peak TL glow curves. They emphasized the primary importance of using a carefully measured TL glow curve, since any errors in measuring the glow curve can lead to the wrong results in the computerized procedures. Such procedures are more likely to yield accurate results in the case of linear superposition of first-order Randall-Wilkins type mathematical expressions. These authors concluded that curve fitting methods using a particular theoretical model should be applied with the utmost care, and extreme caution should be exercised when drawing conclusions from good curve fitting results.

Horowitz and Yossian [22] provided an extensive and detailed review of the subject in a special issue of the journal Radiation Protection Dosimetry. Bos et al.[9] in several important papers presented the results of an evaluation of the capabilities of computer programs to analyze glow curves. This comparative study was carried out in the framework of the GLOw Curve ANalysis INtercomparison (GLOCANIN) project. The papers contain the results of an analysis of thirteen different computer programs involving 11 participants from 10 countries, on both computer generated and on experimentally measured glow curves. The intercomparison project concentrated on the goodness of fit, the determination of the peak area A, the temperature of the peak maximum T_m and the trapping parameters i.e. activation energy E and frequency factor s.

There have been several publications on the development of improved deconvolution procedures for TL can be found in Chung et al. [14, 15, 16], Puchalska et al. [53], Pagonis et al. [45]. There are several Windows open access software interfaces, *GlowFit* and *TLanal*. *GlowFit* can be used for deconvoluting first-order kinetics TL glow curves by using the least squares Levenberg–Marquardt method. *GlowFit* is a Microsoft Windows-operated user-friendly program. The graphic interface enables easy intuitive manipulation of glow-peaks, at the initial stage (parameter initialization) and at the final stage (manual adjustment) of fitting peak parameters to the glow-curves. The program is freely downloadable from the web. The main advantage of *GlowFit* is the ability to resolve complex TL glow-curves consisting of strongly overlapping peaks, as those observed in heavily-doped LiF:Mg,Ti (MTT) detectors. This resolution is achieved mainly by setting constraints or by fixing selected parameters. The initial values of the fitted parameters are placed in the so-called pattern files.

TLanal is also a Windows bases program, which can be used for one kind of TL material (LiF), for one kind of TL reader, for various exposures by radiation, and for various heating profiles. Afouxenidis et al. [3] developed a general CGCD program for TL and OSL data, based on the Excel spreadsheet for Windows.

The analysis of complex TL glow curves is complicated by the fact that the exact number of overlapping peaks is unknown. The technique of CGCD starts by establishing experimentally the number of glow peaks present in a glow curve, by using the well known techniques of $T_{max} - T_{stop}$ and $E - T_{stop}$. Let us denote by $f(T)$ the mathematical function of an individual glow peak.

In the case of a first-order peak, three parameters are involved, namely, the activation energy, E, the frequency factor, s and the initial concentration of trapped carriers, n_0. If we deal with a first-order peak normalized to, say, maximum intensity of unity, the two parameters E and s are sufficient to fully describe the peak. The situation is similar with a second-order peak where 3 parameters are needed to define the glow peak. In the cases of mixed-order and general-order peaks, an additional parameter is required, the mixed order kinetic parameter α in the former case and the general kinetic order b in the latter. These parameters are summarized Table 1.1.

When several glow peaks are involved, the glow curve can be written as

$$I(T) = \sum_{i=1}^{p} \alpha_i f_i(T), \tag{1.1}$$

where α_i is a scaling factor, $f_i(T)$ is the chosen mathematical function for the description of the individual glow peaks (first-, second-, general- or mixed-order), and p is the number of peaks in the glow curve. Thus, when p first-order peaks are involved, $3p$ parameters are required whereas if the individual peaks are of general or mixed order, $4p$ parameters should be considered. Basically, the process of curve fitting, be it for a single peak or a composite glow curve, consists of a first guess of the parameters, evaluating $I(T)$ using Eq. (1.1) and comparing it to the experimental curve. The parameters are then changed so that the difference between the experimental and calculated curves is minimized.

A popular way of doing this is the Levenberg-Marquardt nonlinear least-squares fitting. The least squares fitting procedure minimizes the objective:

$$f = \sum_{i=1}^{n} \left(y_i^{expt} - y_i^{fit} \right)^2, \qquad i = 1 \ldots n \tag{1.2}$$

where y_i^{expt} and y_i^{fit} are the i-th experimental point and the fitted value respectively, and n is the number of data points. When the weights of the experimental data points are known, one can use the "chi-squared" function instead:

$$\chi^2 = \sum_{i=1}^{n} \left(\frac{y_i^{expt} - y_i^{fit}}{w_i} \right)^2$$

where w_i is a weighting parameter.

At the end of the CGCD process of minimization of the objective function, one wishes to evaluate the goodness of fit. Due to random variations in the data points, one cannot expect to end up with that an objective function of zero. The goodness of fit of the equation to the data is often expressed by the Figure of Merit (FOM) which is defined as follows (Balian and Eddy [4]):

$$FOM = \frac{\sum_{i=1}^{n} \mid y_i^{expt} - y_i^{fit} \mid}{\sum_{i=1}^{n} \mid y_i^{expt} \mid}, \qquad i = 1 \ldots n \tag{1.3}$$

where y_i^{expt} and y_i^{fit} were defined above. Since the FOM is normalized by the integral under the curve, the goodness of fit may be compared from one glow curve to another. Fits are considered to be acceptable when the FOM is of a few percent.

Obviously, one wishes to get a *global minimum* of the objective function function, to obtain the best possible set of parameters. Unfortunately, nonlinear functions of this sort usually have many *local minima*, and practically all the methods of minimization lead to a local minimum which is not necessarily global. A wide variety of methods are being used for such minimization and for increasing the probability of approaching the global minimum, even when the initial guess of the set of parameters is rather far from the final optimum. Some of these methods are steepest descent, Newton, quasi-Newton, simulated annealing and tabu search[5] and genetic algorithms(Adamiec et al. [1, 2]).

For an example of analyzing TR-IRSL signals using several of these different algorithms, see for example Pagonis et al.[46].

1.3 CGCD equations and their transformed equivalents

In this chapter we will use analytical equations from three models we examined in the previous chapters, i.e. the analytical Lambert solution of the OTOR, the MOK model, and the GOK model. Table 1.1 summarizes the CGCD equations commonly for analysis of TL peaks. The first six CGCD equations in this Table are based on *delocalized* transition models and presented in this chapter, while the last four equations in this Table are based on *localized* transition models and are studied in two later chapters.

For each of these models, we develop two types of Python codes, based on the original equations in the models and on transformed versions of them. Early in the TL literature, researchers attempted to transform the original analytical equations by replacing the original variables, with equivalent parameters which can be directly evaluated from the experimental data. This idea of transforming the analytical equations was first achieved by Podgorsak et al. [49] for a first order kinetics TL expression derived using a hyperbolic heating function. The next major step in this research area was taken by Kitis et al. [27], who developed transformed equations for first, second and general order kinetics under a linear heating function. In later works transformed equations were developed by Kitis and Gomez-Ros [26] and Gomez-Ros and Kitis [21] for mixed order kinetics and for continuous trap distributions. Additional work by Kitis et al. [25] developed the corresponding transformed equations for an exponential heating function. In the area of OSL, Kitis and

Pagonis [28] developed equations for LM-OSL signals. Recently Sadek et al. [54] transformed the analytical expression derived from the OTOR model, whereas Kitis and Pagonis [30] developed transformed analytical expressions for tunneling recombination from the excited state of a trap.

The general method of developing the transformed equations in these models is described in detail in the review paper by Kitis et al. [32]. The TL transformation is based on replacing two of the variables in the equations with two new variables, which can be estimated directly from the experimental data. Specifically the initial concentration of trapped electrons n_0 and the frequency factor s in the equations, will be replaced with the maximum intensity I_m and the corresponding temperature T_m. In the case of LM-OSL signals, the optical excitation constant λ will be replaced with the time t_m at which the maximum LM-OSL signal occurs.

In the author's opinion, use of the empirical GOK equations listed in rows 5 and 6 of Table 1.1 should be avoided, because the parameters do not have a direct physical meaning. By contrast, the first four equations in this Table have a direct physical meaning, and should be used instead of the GOK to analyze experimental data.

MODEL	**Analytical TL equation**	**Fitting Parameters**
GOT	**KV-TL** Eq.(1.12)-(1.13)	c, s', A, E
GOT	**Transformed KV-TL** Eqs.(1.14)-(1.16)	R, E (optional T_m, I_m)
MOK	**Mixed order kinetics (MOK)** Eqs.(1.17), (1.18)	α, s', A, E
MOK	**Transformed MOK** Eqs.(1.23)-(1.28)	α, E (optional T_m, I_m)
GOK ***(not recommended)***	**GOK** Eq.(1.32)	b, s', A, E
GOK ***(not recommended)***	**Transformed GOK** Eq.(1.33)	b, E (optional T_m, I_m)
TUN	**KP-TL** Eq.(2.11)	B , s, ρ' (E from initial rise data)
TUN	**Transformed KP-TL** Eqs.(2.13)-(2.16)	ρ' (E from initial rise data, T_m, I_m)
LOC	**KP-LOC-TL**	c', s, A, E
LOC	**Transformed KP-LOC-TL**	c', E (optional T_m, I_m)

Table 1.1: Table of the Python CGCD scripts for TL analysis developed in this book. The first six rows in this Table refer to *delocalized* models of TL described in this chapter, while the last four rows are for *localized* TL models which are described in later chapters.

1.4 Using the function `optimize()` in Python

In this chapter we use the `optimize()` package in Python to perform CGCD analysis of TL glow curves. Specifically we will import and use the function `curve_fit` in the form:

```
  scipy.optimize.curve_fit(f, xdata, ydata, p0, bounds)
```

Here f is the function which is used to fit the data (xdata,ydata), `p0` is an array containing the initial guesses for the parameters, and `bounds` specifies the lower and upper bounds on the parameters.

There are two components in the output of the `curve_fit` function. The first component is the array *popt* which contains the optimal values for the parameters, so that the sum of the squared residuals of `f(xdata, *popt) - ydata` is minimized. Here `*popt` is used to pass the array of fitting parameters in the `optimize-curve_fit` function.

The second parameter in the output of the `curve_fit` function is `pcov`; this is a 2-D array containing the estimated covariance of `popt` . The diagonals of `pcov` provide the variance of the best fit estimates for the parameters. In order to compute one standard deviation errors `perr` of the parameters instead of the variance, we can use

```
perr = np.sqrt(np.diag(pcov).
```

As mentioned in the previous section, the quantity to be minimized is the sum of the squared residuals f for unweighted data, and the χ^2 in the case of weighted data.

In the Python codes of this chapter, the residuals are calculated using

```
f(xdata, *popt) - ydata .
```

When applying CGCD methods of analysis, one should keep in mind that the solutions of the best fit process are not unique, and that therein general are infinite combinations of the parameters which could give a very good fit. Indeed, the results of the CGCD procedures are in many cases strongly influenced by the choice of initial values for the parameters.

These are well known standard issues with optimization functions, and they are certainly not unique to luminescence data analysis. It is highly recommended that researchers use several different methods to evaluate the best parameters characterizing a luminescence signal, and not simply use a single fitting method. By using several different methods of analysis to analyze the results of different experiments on the same sample, a better understanding and confidence is obtained for the underlying luminescence process.

1.5 CGCD using the original Lambert-OTOR solution: the KV-TL equation

In the previous chapter we saw the following Lambert-OTOR solution of the OTOR model (Kitis and Vlachos [34], their Eqs.16 and 36):

$$I(T) = \frac{1}{\beta}\,\frac{N\,R}{(1-R)^2}\,\frac{s\,\exp[-E/\,(k_BT)]}{W[e^z]+W[e^z]^2} \tag{1.4}$$

$$z_{TL} = \frac{1}{c} - \ln(c) + \frac{s}{(1-R)\;\beta}\int_{T_0}^{T}\exp[-\frac{E}{k\,T'}]\,dT' \tag{1.5}$$

where the dimensionless positive constant $c > 0$ is defined by:

$$c = \frac{n_0}{N}\,\frac{1-R}{R} \tag{1.6}$$

and n_0, N are the initial and total concentration of filled traps, and $R = A_n/A_m$ is the retrapping ratio in the OTOR model. In these analytical equations W represents the real positive part of the Lambert W function. In fact, we saw that there is a second solution of the OTOR model corresponding to $R > 0$.

For the purposes of CGCD analysis, we need only concern ourselves with the positive real branch of W, since values of the retrapping ratio R in the range $0 < R < 1$ can describe any shape between first and second order kinetics.

We now need to bring the above analytical equations in a form suitable for CGCD analysis. By introducing the variables

$$s' = \frac{s}{(1-R)\;\beta} \tag{1.7}$$

then

$$\frac{1}{\beta}\,\frac{N\,R\,s}{(1-R)^2} = \frac{s}{(1-R)\;\beta}\,\frac{N\,R}{1-R} = s'\frac{n_0}{c} = A \tag{1.8}$$

where we used Eq.(1.6). The TL intensity from Eq.(1.4) becomes:

$$I(T) = \frac{n_0}{c}\,\frac{s'\;\exp[-E/\,(k_BT)]}{W[e^z] + W[e^z]^2} = A\frac{\exp[-E/\,(k_BT)]}{W[e^z] + W[e^z]^2} \tag{1.9}$$

$$z_{TL} = \frac{1}{c} - \ln(c) + s'\int_{T_0}^{T} \exp[-\frac{E}{k\,T'}]\,dT' \tag{1.10}$$

By replacing the exponential TL integral with its analytical form:

$$\int_{T_0}^{T} e^{-\frac{E}{k\,T'}}dT' \simeq \frac{k\,T^2}{E}\;\exp\left[-\frac{E}{k\,T}\right]\;\left(1 - \frac{2kT}{E}\right) \tag{1.11}$$

we obtain the following analytical expression which will be termed the *Kitis-Vlachos TL* equation (*KV-TL*):

The original KV-TL equation for CGCD TL analysis ($R < 1$***)***

$$I(T) = A\,\frac{\exp[-E/\,(k_BT)]}{W[e^z] + W[e^z]^2} \tag{1.12}$$

$$z = \frac{1}{c} - \ln(c) + s'\;\frac{k\,T^2}{E}\;\exp\left[-\frac{E}{k\,T}\right]\;\left(1 - \frac{2kT}{E}\right) \tag{1.13}$$

The three independent fitting parameters in these CGCD equations are c, s' and A. Because these parameters contain mixed algebraic combinations of the initial parameters n_0, s and R, it is not possible to evaluate the parameters s and R individually from the fitting process. However, we can obtain the value of E (in eV), the value of the parameter s' (in units of K^{-1}), the amplitude A (in the same units as the TL intensity of the experimental data) and the dimensionless quantity c.

The following code implements these equations for CGCD analysis, with the four fitting parameters E, s', c and A. The code imports and uses the function `optimize` from `SciPy`, and the function `PrettyTable()` from the package `prettytable`, for formatting and printing the results of the least squares fit in a Table format. Fig.1.5 shows the deconvoluted Reference glow curve #1, from the GLOCANIN project mentioned previously in Sect. 1.2 (Bos et al. [9]). The data was normalized by dividing with the maximum TL intensity, in order to assist the least squares fitting routine in reaching a minimum.

The results show that the activation energy $E = (1.127 \pm 0.004)\, eV$, which is within ~4% of the expected value of $E = 1.182\, eV$. The FOM=0.039 or FOM=3.9%, indicating a good fit to this simulated TL peak. The rest of the best fit parameters are $s' = 2.1 \times 10^{10}$ K^{-1} , $A = 9.6 \times 10^{8}$ and $c = 999.4$; the residuals $y_i^{expt} - y_i^{fit}$ of the best fit are calculated and plotted underneath the data, and they are of the order of 3% of the experimental TL intensity.

Code 1.1: Deconvolution of Glocanin TL with Lambert function

```
# Deconvolution of single TL peak with Lambert-OTOR equation
from scipy import optimize
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable
import warnings
warnings.filterwarnings("ignore")
from scipy.special import lambertw
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
data = np.loadtxt('glocanin1.txt')
x_data,y_data = data[:, 0], data[:, 1]/max( data[:, 1])
kB, beta=  8.617e-5,  1
def W_func(T,A,sprime, E, c):
   expint=kB*(T**2.0)/(beta*E)*np.exp(-E/(kB*T))*(1-2*kB*T/E)
   zTL=(1/c)-np.log(c)+(sprime*expint)
   lam=np.real(lambertw(np.exp(zTL)))
   return A*np.exp(-E/(kB*T))/(lam+lam**2)
params, cov=optimize.curve_fit(W_func,x_data,y_data,([1e10,2e9,\
1.,10]),bounds=((1e8,1e8,.9,1e-4),(1e15,1e14,1.3,1e3)))
```

```
plt.subplot(2,1, 1)
plt.scatter(x_data, y_data, label='Experiment')
plt.plot(x_data, W_func(x_data, *params),
c='r',linewidth=3, label='KV-TL equation')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('TL signal [a.u.]')
plt.xlabel(r'Temperature T [K]')
plt.text(400, .5,'Glocanin')
plt.text(400, .37,'Project')
plt.xlim(375,550);
plt.subplot(2,1, 2)
plt.plot(x_data,W_func(x_data, *params)-\
y_data,c='r',linewidth=2,label='Residuals')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('Residuals')
plt.xlabel(r'Temperature T [K]')
plt.ylim(-.05,.05);
plt.xlim(375,550);
plt.tight_layout()
A=format(params[0],"10.1E")
sprime=format(params[1],"10.1E")
E=round(params[2],3)
c=round(params[3],2)
dsprime = format(np.sqrt(cov[1][1]),"10.1E")
dE = round(np.sqrt(cov[2][2]),3)
res=W_func(x_data, *params)-y_data
FOM=round(100*np.sum(abs(res))/np.sum(y_data),3)
myTable = PrettyTable([ "A", "s' (K^-1)","E(eV)","dE(eV)","c",\
"FOM(%)"])
myTable.add_row([A,sprime,E,dE,c,FOM]);
print(myTable)
```

```
  +------------+------------+-------+--------+--------+--------+
  |     A      | s' (K^-1)  | E(eV) | dE(eV) |   c    | FOM(%) |
  +------------+------------+-------+--------+--------+--------+
  |    9.6E+08 |    2.1E+10 | 1.127 | 0.004  | 999.39 | 3.887  |
  +------------+------------+-------+--------+--------+--------+
```

```
plt.show()
```

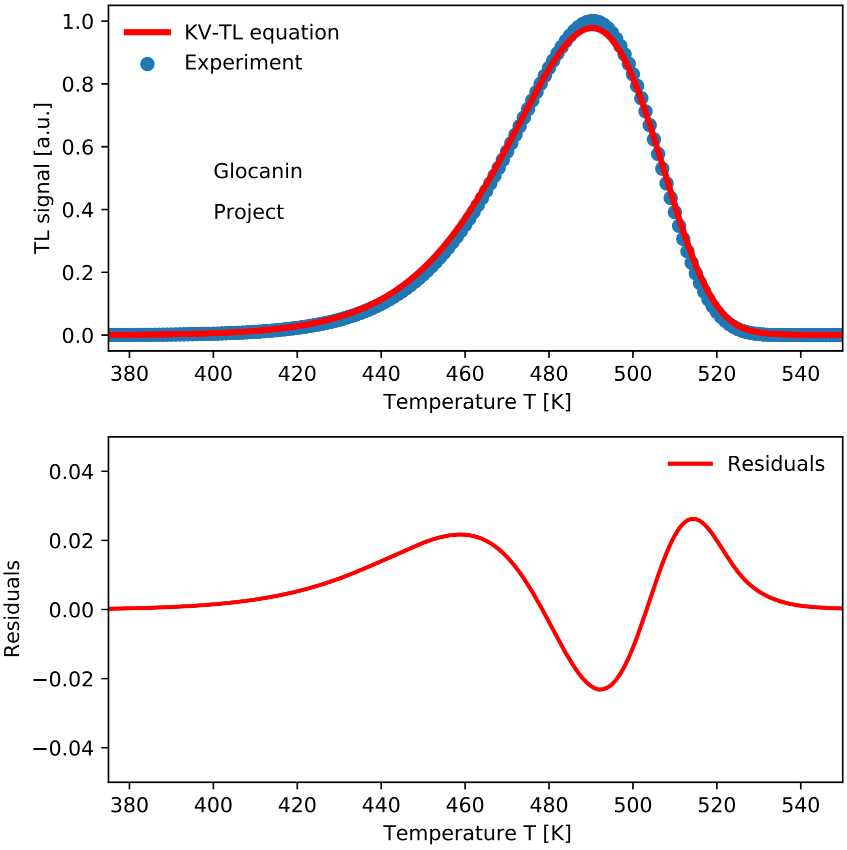


Fig. 1.1: Deconvolution of simulated TL data containing a single peak, using the original *KV-TL* deconvolution equation by Kitis and Vlachos [34]. The data is from Reference glow curve #1 in the intercomparison project GLOCANIN (Bos et al. [9]).

1.6 Deconvolution using the transformed KV-TL equation

By simulating many random values of the parameters in the analytical Lambert solution of the OTOR model, Sadek et al. [54, 32] arrived at the following transformed KV-TL equation, which is very useful for CGCD analysis of TL signals. For a complete discussion of this transformed equation, see the review paper by Kitis et al. [32].

***Transformed KV-TL equations for CGCD** ($c > 0$, $R < 1$)*

$$I(T) = I_m \; \exp\left[-\frac{E}{k}\left(\frac{1}{T}-\frac{1}{T_m}\right)\right]\,\frac{W[e^{z_m}]+W[e^{z_m}]^2}{W[e^{z}]+W[e^{z}]^2} \tag{1.14}$$

$$z = \frac{R}{1-R}-\ln(\frac{1-R}{R})+\frac{E\exp\left[E/\left(kT_m\right)\right]}{kT_m^2(1-1.05\,R^{1.26})}\,\frac{k\,T^2}{E}\;\exp\left[-\frac{E}{k\,T}\right]\;\left(1-\frac{2kT}{E}\right) \tag{1.15}$$

$$z_m = \frac{R}{1-R}-\ln(\frac{1-R}{R})+\frac{(1-2kT_m/E)}{(1-1.05\,R^{1.26})} \tag{1.16}$$

The advantage of using this equation to approximate the TL intensity is that it involves two quantities which are measured experimentally, namely the maximum TL intensity (I_m), and the corresponding temperature (T_m). The activation energy E in this expression and during the computerized fitting procedure is treated as an adjustable parameter.

The following code implements these equations for CGCD analysis, with the two fitting parameters E and R. If we wish to maximize the flexibility of the fitting routine, or if we are dealing with a complex TL glow curve, we can also allow the quantities T_m and I_m to be two additional fitting parameters in the code.

In this example, the residuals are of the order of 1% of the experimental TL intensity. The best fitting parameters are $E = (2.39 \pm 0.04)$ eV and a small value of the retrapping ratio $R = .096$, indicating behavior close to first order kinetics. The FOM=3.1 % indicates a satisfactory fit.

Code 1.2: Deconvolution of LiF peak using the KV-TL equation

```
# Deconvolution of LiF peak using the KV-TL equation
# Deconvolution of single TL peak with Lambert-OTOR equation
from scipy import optimize
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable
import warnings
warnings.filterwarnings("ignore")
from scipy.special import lambertw
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
data = np.loadtxt('LiFcurve5.txt')
x_data,y_data = data[:, 0][0:45], data[:, 1][0:45]/max(\
data[:, 1][0:45])
```

```
x_data=[273+u for u in np.array(x_data)]
x_data=np.array(x_data)
kB=8.617E-5
Imax=max(y_data)
def W_func(T,Tmax,R, E):
    F=kB*(T**2.0)*np.exp(-E/(kB*T))*(1-2*kB*T/E)/E
    Fm=kB*(Tmax**2.0)*np.exp(-E/(kB*Tmax))*(1-2*kB*Tmax/E)/E
    a=kB*Tmax**2.0*(1-1.05*R**1.26)
    Z=R/(1-R)-np.log((1-R)/R)+(F*E*np.exp(E/(kB*Tmax)))/a
    Zm=R/(1-R)-np.log((1-R)/R)+(Fm*E*np.exp(E/(kB*Tmax)))/a
    argW=np.real(lambertw(np.exp(Z)))
    argWm=np.real(lambertw(np.exp(Zm)))
    return Imax*np.exp(-E/(kB*T)*(Tmax-T)/Tmax)*\
         (argWm+argWm**2.0)/(argW+argW**2.0)
params,cov=optimize.curve_fit(W_func,x_data,y_data,\
p0=(490,1e-6,2.0))
plt.subplot(2,1, 1)
plt.plot(x_data, W_func(x_data, *params),'-',linewidth=4)
plt.scatter(x_data, y_data, label='Experiment')
plt.plot(x_data, W_func(x_data, *params),
c='r',linewidth=3, label='KV-TL equation')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('TL signal [a.u.]')
plt.xlabel(r'Temperature T [K]')
plt.text(460, .5,'LiF:Mg,Ti')
plt.text(460, .37,'Remnant TL')
plt.subplot(2,1, 2)
plt.plot(x_data,W_func(x_data, *params)-\
y_data,'o',c='r',linewidth=2,label='Residuals')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('Residuals')
plt.xlabel(r'Temperature T [K]')
plt.ylim(-.2,.2);
plt.tight_layout()
Tmax=round(params[0],1)
R=format(params[1],"10.1E")
E=round(params[2],3)
dE = round(np.sqrt(cov[2][2]),3)
res=W_func(x_data, *params)-y_data
FOM=round(100*np.sum(abs(res))/np.sum(y_data),3)
myTable = PrettyTable([ "Tmax","R", "E(eV)","dE(eV)","FOM(%)"])
myTable.add_row([Tmax,R,E,dE,FOM]);
```

```
print(myTable)

  +-------+------------+-------+--------+--------+
  |  Tmax |     R      | E(eV) | dE(eV) | FOM(%) |
  +-------+------------+-------+--------+--------+
  | 495.9 |    9.6E-02 | 2.387 | 0.037  | 3.088  |
  +-------+------------+-------+--------+--------+

plt.show()
```

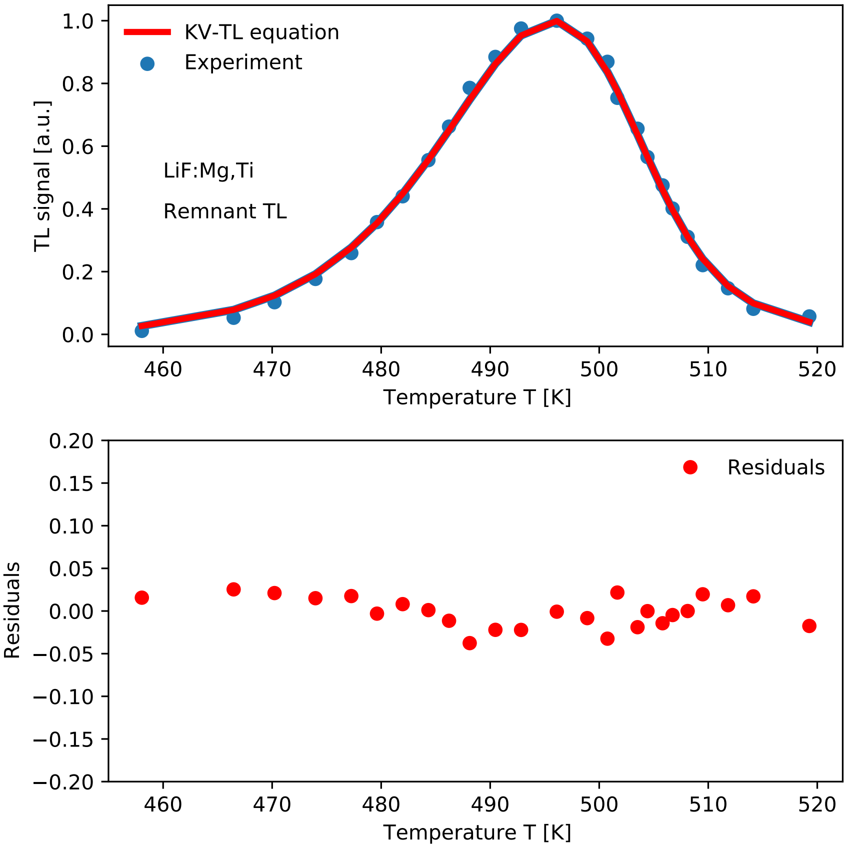


Fig. 1.2: Deconvolution of LiF: Mg,Ti (TLD-100) data containing a single peak, using the transformed KV-TL equation involving the Lambert *W* function.

1.7 CGCD using the original equations for mixed order kinetics

The analytical solution of the MOK model is:

$$I(T) = \frac{g\,N_d^2}{\beta}\,\frac{\alpha\,s\,\exp\left[-E/(k_BT)\right]\,F(T)}{\left[F(T)-\alpha\right]^2} \tag{1.17}$$

$$F(T) = \exp\left\{g\,N_d\,\frac{s}{\beta}\,\frac{k\,T^2}{E}\,\exp\left[-\frac{E}{k\,T}\right]\,\left(1-\frac{2kT}{E}\right)\right\} \tag{1.18}$$

When using these expressions for CGCD analysis, the fitting parameters are α, E, s and n_0.

In order to make these equations more suitable for CGCD analysis, we introduce the following new parameters s' (with units of K^{-1}) and A (with the same units as the TL intensity), instead of the unknown original parameters g, N_d, s:

$$s' = g\,N_d\,\frac{s}{\beta} \tag{1.19}$$

$$A = \frac{g\,N_d^2}{\beta}\alpha \tag{1.20}$$

to obtain the CGCD MOK equations:

Original CGCD MOK equations for TL $(0<\alpha<1)$

$$I(T) = A\,\frac{\exp\left[-E/(k_BT)\right]\,F(T)}{\left[F(T)-\alpha\right]^2} \tag{1.21}$$

$$F(T) = \exp\left\{s'\,\frac{k\,T^2}{E}\,\exp\left[-\frac{E}{k\,T}\right]\,\left(1-\frac{2kT}{E}\right)\right\} \tag{1.22}$$

Figure 1.3 shows the results of fitting a glow curve for Al_2O_3:C using the MOK Eq.(1.21) for TL. The fitting parameters for this model are then E, s', A and α. We find E= (1.074±0.002) eV, s'= 3.7×10^9 K^{-1}, $A = 1.4\times10^{11}$ and $\alpha = 0.329$. The value of α indicates that t the TL glow curve does not have the expected first order kinetics shape. As discussed later in this book, this is caused by the presence of thermal quenching, which distorts the actual shape of the peak, and causes an apparent lower value of E.

Code 1.3: MOK deconvolution of glow curve for Al2O3:C

```
# Deconvolution of Al2O3:C peak with MOK
import numpy as np
import matplotlib.pyplot as plt
from scipy import optimize
from prettytable import PrettyTable
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
data = np.loadtxt('aluminaTL2.txt')
x_data= np.array(data[:, 0])
y_data=np.array(data[:, 1])/max(data[:, 1])
x_data=[273+u for u in np.array(x_data)]
x_data=np.array(x_data)
plt.plot(x_data,y_data,'o')
kB,  beta=  8.617e-5, 1
def TLMOK(T,A,sprime,E,alpha):
    expint=kB*(T**2.0)/E*\
    np.exp(-E/(kB*T))*(1-2*kB*T/E)
    Ft=np.exp(sprime*expint)
    return A*np.exp(-E/(kB*T))*Ft/((Ft-alpha)**2.0)
params, cov=optimize.curve_fit(TLMOK,x_data,y_data,([2e10,2e9,\
1.,.01]),bounds=((1e9,1e8,.9,1e-4),(1e17,1e14,1.3,1)))
plt.subplot(2,1,1)
plt.plot(x_data, TLMOK(x_data, *params),'-',linewidth=4)
plt.scatter(x_data, y_data, label='Experiment')
plt.plot(x_data, TLMOK(x_data, *params),
c='r',linewidth=3, label='MOK equation')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('TL signal [a.u.]')
plt.xlabel(r'Temperature T [K]')
plt.text(520, .7,r'Al$_{2}$O$_{3}$:C')
plt.subplot(2,1,2)
plt.plot(x_data,TLMOK(x_data, *params)-\
y_data,'o',linewidth=2,label='Residuals')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('Residuals')
plt.xlabel(r'Temperature T [K]')
plt.ylim(-.1,.1);
plt.plot(([min(x_data),max(x_data)]),([0,0]),'r-')
plt.tight_layout()
Tmax=x_data[np.argmax(y_data)]
A=format(params[0],"10.1E")
dA = format(np.sqrt(cov[0][0]),"10.1E")
```

```
sprime=format(params[1],"10.1E")
E=round(params[2],3)
dE = round(np.sqrt(cov[2][2]),3)
alpha=round(params[3],5)
res=TLMOK(x_data, *params)-y_data
FOM=round(100*np.sum(abs(res))/np.sum(y_data),2)
myTable = PrettyTable([ "A", "s' (K^-1)","E(eV)","dE(eV)",\
"alpha","FOM(%)"])
myTable.add_row([A,sprime,E,dE,alpha,FOM]);
print(myTable)
```

```
  +------------+------------+-------+--------+--------+--------+
  |     A      | s' (K^-1)  | E(eV) | dE(eV) | alpha  | FOM(%) |
  +------------+------------+-------+--------+--------+--------+
  |    1.4E+11 |    3.7E+09 | 1.074 | 0.002  | 0.3295 |  2.58  |
  +------------+------------+-------+--------+--------+--------+
```

```
plt.show()
```

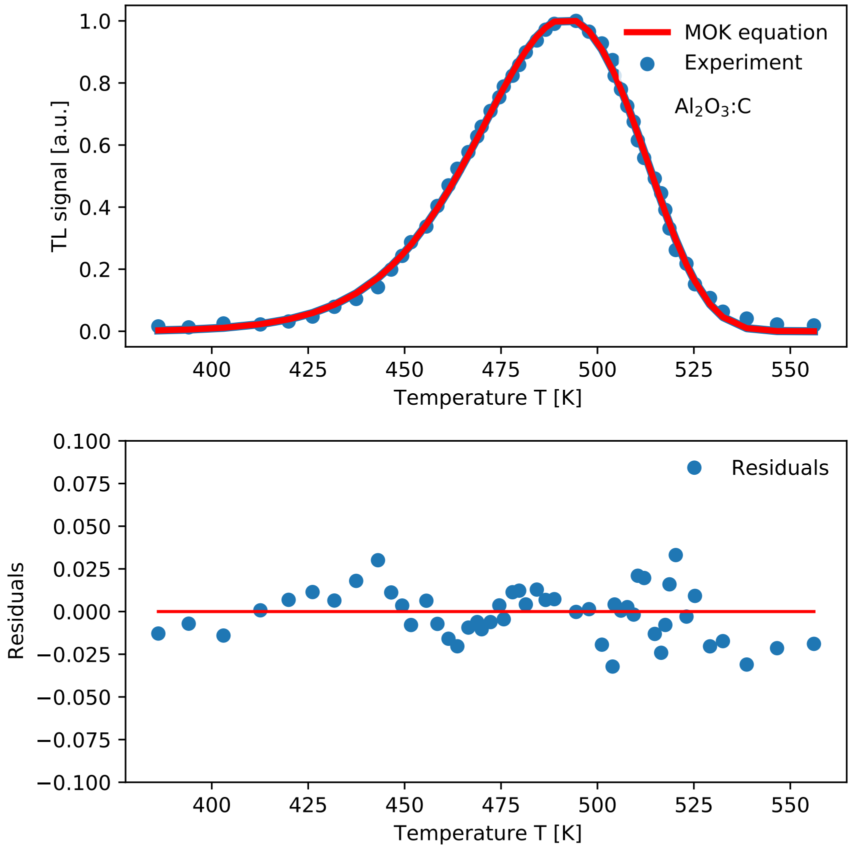


Fig. 1.3: Deconvolution of the main dosimetric peak in Al_2O_3:C, using Mixed Order Kinetics (MOK). Thermal quenching affects the shape of the TL glow curve, as well as the value of E obtained from the CGCD fitting procedure.

1.8 CGCD analysis using the transformed MOK equations

As discussed in the review paper by Kitis et al.[32], Kitis et al. [21] derived a transformed second master equation for TL in the MOK model:

$$I(T) = I(T_m)\,\exp\left[\frac{E\,(T-T_m)}{k\,T\,T_m}\right]\,\frac{(F(T_m)-\alpha)^2}{F(T_m)}\,\frac{F(T)}{(F(T)-\alpha)^2} \tag{1.23}$$

and by using the exponential integral approximation:

$$F(T) = \exp\left\{\frac{1}{f_{MOK}}\,\frac{T^2}{T_m^2}\,\exp\left[\frac{E\,(T-T_m)}{k\,T\,T_m}\right]\,\left(1-\frac{2\,k\,T}{E}\right)\right\}. \tag{1.24}$$

From this equation we obtain the following function $F(T_m)$ by setting $T = T_m$:

$$F(T_m) = \exp\left(\frac{1-(2\,k\,T_m)/E}{f_{MOK}}\right). \tag{1.25}$$

Kitis et al. [21] showed that the quantity f_{MOK} depends only on the MOK parameter α according to Eq.(1.28).

In summary, the following are the equations for TL in the MOK model, in terms of the MOK kinetic parameter α, the temperature of maximum TL intensity T_m, and the activation energy E.

Transformed CGCD equations for TL analysis in MOK model

$$I(T) = I(T_m)\,\exp\left(\frac{E\,(T-T_m)}{k\,T\,T_m}\right)\,\frac{(F(T_m)-\alpha)^2}{F(T_m)}\,\frac{F(T)}{(F(T)-\alpha)^2} \tag{1.26}$$

$$F(T) = \exp\left(\frac{1}{f_{MOK}}\,\frac{T^2}{T_m^2}\,\exp\big(\frac{E\,(T-T_m)}{k\,T\,T_m}\big)\big(1-\frac{2\,k\,T}{E}\big)\right) \tag{1.27}$$

$$f_{MOK} = \frac{2.6-0.9203\,\alpha+0.324\,\alpha^{3.388}}{2.6-2.9203\,\alpha+0.324\,\alpha^{3.338}} \tag{1.28}$$

The following code shows how to use these equations from the MOK model, in order to carry out a CGCD analysis of typical TL data. The sequence of calculations inside the function `MOK(T,alpha,E)` is as follows: first use α, E in order to find f_{MOK} from Eq.(1.28), next calculate $F(T)$ and $F(T_m)$ from Eq.(1.27) using the value of f_{MOK}, and finally calculate the TL intensity using Eq.(1.26).

In the example of Fig.1.4 for a TL glow curve of BeO, the best fit values are $E = (1.21 \pm 0.06)$ eV, and the value of the MOK parameter $\alpha = 10^{-10}$, indicating again the shape of a first order TL peak. The FOM= 3 % and the residuals are of the order of 1%, showing an excellent fit to the data.

Code 1.4: Deconvolution of BeO TL with transformed MOK

```
# Deconvolution of BeO glow curve with transformed  MOK
from scipy import optimize
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable
kB=8.617E-5
def MOK(T,alpha,E,Tm):
     fmok=(2.6-0.9203*alpha+0.324*(alpha**3.338))/(2.6-\
     2.9203*alpha+0.324*(alpha**3.338))
     FT=np.exp((1/fmok)*(T**2.00)/(Tm**2.0)*np.exp(E*(T-Tm)/\
     (kB*T*Tm))*(1-2.0*kB*T/E))
     FTm=np.exp((1-2*kB*Tm/E)/fmok)
     return imax*np.exp(E*(T-Tm)/(kB*T*Tm))*((FTm-alpha)**2.0)\
      *FT/(FTm*((FT-alpha)**2.0))
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
data = np.loadtxt('BeOTL140.txt')
x_data= np.array(data[:, 0])+273
y_data=np.array(data[:, 1])
plt.plot(x_data,y_data,'o')
imax=max(y_data)     #find imax from given data
T=np.arange(390,520,1);
params,cov =optimize.curve_fit(MOK,x_data,y_data,p0=(.5,1,460),
      bounds=((1e-10,.9,450),(0.99999,1.3,480)))
plt.subplot(2,1, 1)
plt.scatter(x_data, y_data, label='Data')
plt.plot(x_data, MOK(x_data,*params[0:3]),
c='r',linewidth=3, label='MOK')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('TL signal [a.u.]')
plt.xlabel(r'Temperature T [K]')
plt.text(400, 1.5e5,'BeO')
plt.subplot(2,1, 2)
plt.plot(x_data,MOK(x_data, *params[0:3])-y_data,'o',c='r',\
linewidth=2,label='Residuals')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('Residuals')
plt.xlabel(r'Temperature T [K]')
plt.tight_layout()
alphafit = format(params[0],"10.1E")
Efit = round(params[1],3)
dalpha, dE ,dTm= np.round(np.sqrt(np.diag(cov)),3)
```

```
res=MOK(x_data, *params)-y_data
FOM=round(100*np.sum(abs(res))/np.sum(y_data),2)
myTable = PrettyTable(["alpha", "E (eV)","dE (eV)",\
"FOM (%)"])
myTable.add_row([alphafit, Efit, dE,FOM]);
print(myTable)

   +------------+--------+---------+---------+
   |   alpha    | E (eV) | dE (eV) | FOM (%) |
   +------------+--------+---------+---------+
   |    1.0E-10 | 1.209  |  0.061  |   3.02  |
   +------------+--------+---------+---------+

plt.show()
```

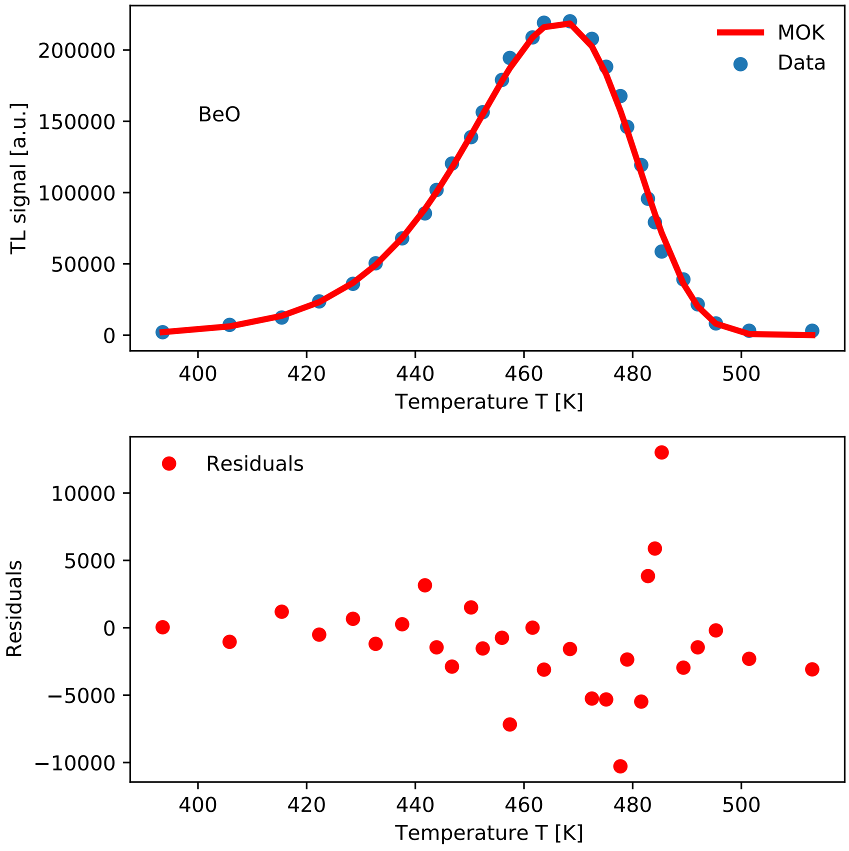


Fig. 1.4: Deconvolution of TL glow curve for BeO, using the transformed Eq.(1.26) for Mixed Order Kinetics (MOK).

1.9 Deconvolution using the original GOK equation

Although the use of the GOK is not recommended by the author, the Python code is included here for completeness. The solution of the GOK equation is:

$$I(T) = A\, n_0\, e^{-\frac{E}{kT}} \left[1 + (b-1)A\, \frac{k_B T^2}{E} \left[e^{-\frac{E}{kT}} \left(1 - \frac{2k_B T}{E}\right)\right]\right]^{-b/(b-1)}$$

$$I(T) = B\, n_0\, e^{-\frac{E}{k_B T}} \left[1 + (b-1)\, B\, \frac{k_B T^2}{E}\, e^{-\frac{E}{k_B T}} \left(1 - \frac{2k_B T}{E}\right)\right]^{-b/(b-1)} \tag{1.29}$$

where $B = (s/\beta)\,(n_0/N)^{b-1}$. We now introduce two new fitting parameters A, s' instead of s, n_0/N:

$$s' = (b-1)B \tag{1.30}$$

$$A = B\, n_0 \tag{1.31}$$

to obtain the following CGCD equations for the GOK model:

CGCD equation using the original GOK model

$$I(T) = A \;\; \exp(-E/kT) \left[1 + s'\, \frac{k\,T^2}{E}\, \exp\left[-\frac{E}{k\,T}\right] \left(1 - \frac{2kT}{E}\right)\right]^{-b/(b-1)} \tag{1.32}$$

The Python code shows application of this CGCD equation to the previously analyzed GLOCANIN TL glow curve.

The results show that the activation energy $E = 1.183\,eV$, which is almost exactly equal to the expected value of $E = 1.182\,eV$. The FOM=0.03% indicating an excellent fit to this simulated TL peak. The rest of the best fit parameters are $s' = 1.0 \times 10^8$ K^{-1} , $A = 3.7 \times 10^{12}$ and $b = 1.001$.

Code 1.5: Deconvolution of GLOCANIN TL using the original GOK

```
# Deconvolution of GLOCANIN TL using the original GOK
import numpy as np
import matplotlib.pyplot as plt
from scipy import optimize
from prettytable import PrettyTable
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
data = np.loadtxt('glocanin1.txt')
x_data,y_data = data[:, 0], data[:, 1]/max(data[:, 1])
kB= 8.617e-5
def TLGOK(T,A,sprime,E,b):
    expint=kB*(T**2.0)/E*\
    np.exp(-E/(kB*T))*(1-2*kB*T)/E
    return A*np.exp(-E/(kB*T))*((1+sprime*expint)**(-b/(b-1)) )
params, cov=optimize.curve_fit(TLGOK,x_data,y_data,([1,2e10,1.,\
1.5]),bounds=((1e-10,1e8,1,1),(1e20,1e14,1.3,1.999)))
plt.subplot(2,1, 1)
plt.scatter(x_data, y_data, label='Experiment')
plt.plot(x_data, TLGOK(x_data, *params),
c='r',linewidth=3, label='GOK equation')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('TL signal [a.u.]')
plt.xlabel(r'Temperature T [K]')
plt.text(400, .5,'Glocanin')
plt.text(400, .37,'Project')
plt.xlim(375,550);
plt.subplot(2,1, 2)
plt.plot(x_data,TLGOK(x_data, *params)-\
y_data,c='r',linewidth=2,label='Residuals')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('Residuals')
plt.xlabel(r'Temperature T [K]')
plt.ylim(-1e-3,1e-3);
plt.xlim(375,550);
plt.tight_layout()
A=format(params[0],"10.1E")
dA = format(np.sqrt(cov[0][0]),"10.1E")
sprime=format(params[1],"10.1E")
E=round(params[2],3)
dE = round(np.sqrt(cov[2][2]),7)
b=round(params[3],3)
db = round(np.sqrt(cov[3][3]),7)
```

```
res=TLGOK(x_data, *params)-y_data
FOM=round(100*np.sum(abs(res))/np.sum(y_data),2)
myTable = PrettyTable([ "A","s' (K^-1)","E(eV)",\
"b","FOM(%)"])
myTable.add_row([A,sprime,E,b,FOM]);
print(myTable)

    +------------+------------+-------+-------+--------+
    |     A      | s' (K^-1)  | E(eV) |   b   | FOM(%) |
    +------------+------------+-------+-------+--------+
    |   3.7E+12  |   1.0E+08  | 1.183 | 1.001 |  0.03  |
    +------------+------------+-------+-------+--------+

plt.show()
```

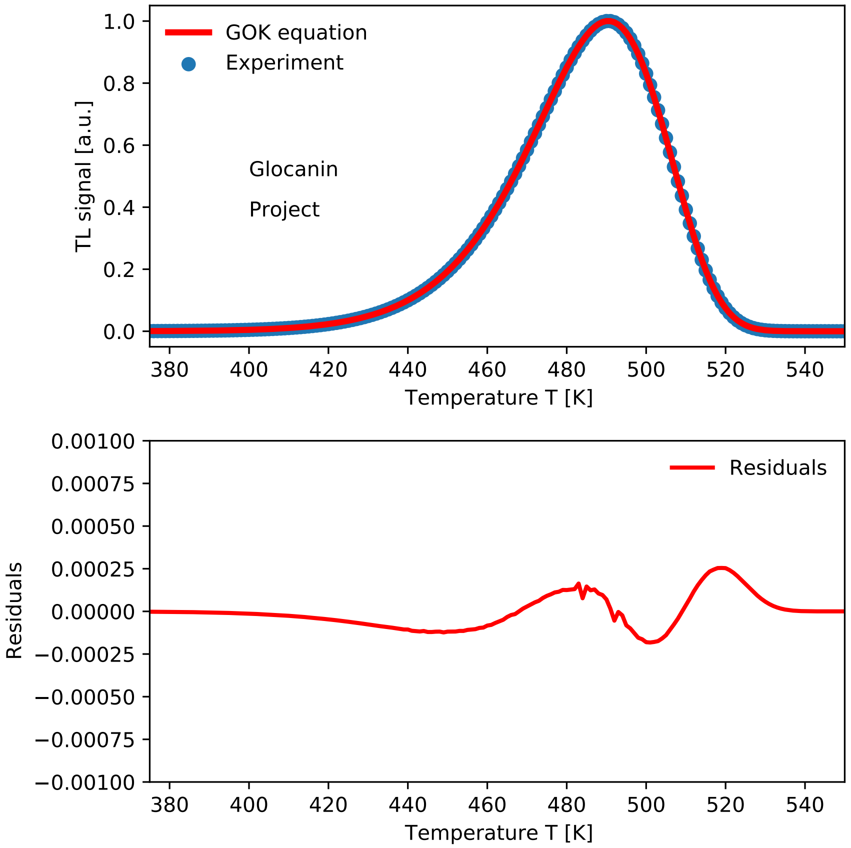


Fig. 1.5: Deconvolution of TL data from GLOCANIN project containing a single peak, using the *GOK* function (Bos et al. [9]).

1.10 Deconvolution using the transformed GOK equation

Kitis et al. ([26],[27]) developed several new analytical expressions for use in computerized glow curve deconvolution (CGCD) analysis, by using the approximation of the TL integral $\int \exp\left(-E/kT'\right) dT'$ in Eq.(1.11). In the case of general order kinetics, these authors derived the following transformed GOK equation:

Transformed equation for CGCD analysis using the GOK model

$$I(T) = I_m b^{\frac{b}{b-1}} e^u \left[1 + Z_m + (b-1)\left(1 - \frac{2kT}{E}\right)\left(\frac{T^2}{T_m^2} e^u\right)\right]^{-\frac{b}{b-1}} \tag{1.33}$$

$$u = \frac{E}{kT}\,\frac{T - T_m}{T_m} \tag{1.34}$$

$$Z_m = 1 + (b-1)\frac{2kT_m}{E} \tag{1.35}$$

In this expression b is the kinetic order of the TL process, I_m=maximum intensity of the glow peak (in counts per K), T_m=temperature at peak maximum (in K). The accuracy of this expression has been tested widely by researchers for both synthetic and for experimental glow curves, and it was found that it can describe accurately glow peaks for first, second and general order kinetics (Bos [8]). However, we recommend against using the GOK for CGCD analysis, since the parameters in the equation do not have a direct physical meaning.

It is noted that mathematically Eq.(1.33) diverges when $b = 1$ because of the presence of the term $b/(b-1)$. However, this equation gives reliable results even in the case of first order kinetics, in which it yields a value of b very close to 1. For the example in Al_2O_3:C data of Fig.1.6, we obtain $b = 1.09 \pm 0.02$ and $E = (1.00 \pm 0.01)$ eV. In this material, the presence of thermal quenching distorts the actual shape of the peak, and causes an apparent lower value of E. For a detailed study of thermal quenching in this material, see Kitis [24] .

Code 1.6: Deconvolution of Al2O3:C glow curve (GOK)

```
# Deconvolution of Al2O3:C glow curve (GOK)
from scipy import optimize
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable
import warnings
warnings.filterwarnings("ignore")
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
data = np.loadtxt('aluminaTL2.txt')
x_data,y_data = data[:, 0], data[:, 1]
x_data=[273+u for u in np.array(x_data)]
x_data=np.array(x_data)
kB=8.617E-5
imax=max(y_data)
def GOK_func(T, Tmax,b, En):
    return imax* np.exp(En/(kB*T)*(T-Tmax)/Tmax)*(b**\
    ((b/(b-1))))*((1+(b-1)*2*kB*Tmax/En+(b-1)*(1-2*kB*T/\
    En)*np.exp(En/(kB*T)*(T-Tmax)/Tmax)*(T**2.0)/(Tmax**\
    2.0))**(b/(1-b)))
params, cov=optimize.curve_fit(GOK_func,x_data,\
y_data,bounds=((460,1.001,1),(520,2.0,1.3)))
plt.subplot(2,1, 1)
plt.scatter(x_data, y_data, label='Experiment')
plt.plot(x_data, GOK_func(x_data, *params),
c='r',linewidth=3, label='GOK equation')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('TL signal [a.u.]')
plt.xlabel(r'Temperature T [K]')
plt.text(400, 1e6,'Al$_{2}$O$_{3}$:C')
plt.xlim(375,550);
plt.subplot(2,1, 2)
plt.plot(x_data,GOK_func(x_data, *params)-\
y_data,'o',c='r',linewidth=2,label='Residuals')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('Residuals')
plt.xlabel(r'Temperature T [K]')
plt.ylim(-4e5,4e5,'o');
plt.xlim(375,550);
plt.tight_layout()
Tmax=params[0]
s=np.exp(params[2]/(kB*Tmax))*(params[2]/(kB*(Tmax**2.0)))
sf=format(s, "10.1E")
Tmax=format(params[0],"10.1E")
b=round(params[1],3)
```

```
E=round(params[2],4)
db = round(np.sqrt(cov[1][1]),3)
dE = round(np.sqrt(cov[2][2]),3)
res=GOK_func(x_data, *params)-y_data
FOM=round(100*np.sum(abs(res))/np.sum(y_data),2)
myTable = PrettyTable([ "b","db", "E(eV)","dE(eV)",\
"s(s^-1)","FOM(%)"])
myTable.add_row([b,db,E,dE,sf,FOM]);
print(myTable)

   +-------+-------+-------+--------+------------+--------+
   |   b   |   db  | E(eV) | dE(eV) |   s(s^-1)  | FOM(%) |
   +-------+-------+-------+--------+------------+--------+
   | 1.094 | 0.025 |  1.0  | 0.014  |    8.5E+08 |  2.91  |
   +-------+-------+-------+--------+------------+--------+

plt.show()
```

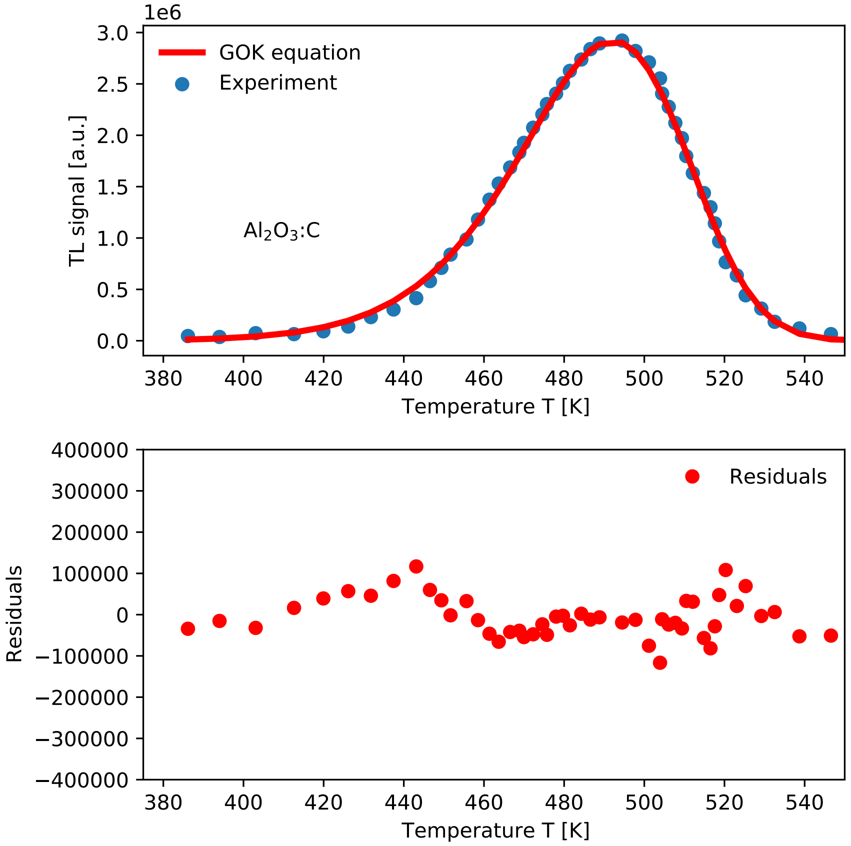


Fig. 1.6: Deconvolution of TL data containing a single peak for Al_2O_3:C , using the transformed GOK function.

1.11 CGCD of multipeak TL data using the transformed KV-TL equation

In this section we analyze TL glow curve with two peaks from dosimetric material LiB_4O_7:Cu,In (LBO) (Kitis et al. [33]), using the transformed KV-TL equation.

The code reads the *.txt* file containing pairs (T,TL) of temperature (in K) and TL intensity (in counts or counts/s etc), and the user provides initial estimates of the maximum TL intensity I_m, of the activation energy E, of the temperature of maximum TL intensity T_m, and of the recombination ratio R. The experimental data in Fig.1.8 are for sample LBO, from the experimental work by Kitis et al. [33].

When applying CGCD analysis, the estimated values of the retrapping ratios R and the activation energies E should be examined carefully and critically, and one should not accept blindly the results of the CGCD analysis. For example, if one finds that the *higher* temperature peak has $E_2 < E_1$, then the experimental data should be further analyzed using different methods. For a more complete analysis of this experimental data for sample LBO using several methods of analysis, see Kitis et al. [33].

Discuss the results of CGCD!

Code 1.7: Deconvolution LBO data using transformed KV-TL equation

```
# Deconvolution using transformed Lambert equation
# ====================================================
from scipy import optimize
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable
import matplotlib.pyplot as plt
from scipy.special import lambertw
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
data = np.loadtxt('lbodata.txt')
```

```
x_data,y_data = data[:, 0], data[:, 1]
kB=8.617E-5
def W_func(T,Imax,Tmax,R, E):
    F=kB*(T**2.0)*np.exp(-E/(kB*T))*(1-2*kB*T/E)/E
    Fm=kB*(Tmax**2.0)*np.exp(-E/(kB*Tmax))*(1-2*kB*Tmax/E)/E
    a=kB*Tmax**2.0*(1-1.05*R**1.26)
    Z=R/(1-R)-np.log((1-R)/R)+(F*E*np.exp(E/(kB*Tmax)))/a
    Zm=R/(1-R)-np.log((1-R)/R)+(Fm*E*np.exp(E/(kB*Tmax)))/a
    argW=np.real(lambertw(np.exp(Z)))
    argWm=np.real(lambertw(np.exp(Zm)))
    return Imax*np.exp(-E/(kB*T)*(Tmax-T)/Tmax)*\
    (argWm+argWm**2.0)/(argW+argW**2.0)
def total_TL(T, Imax1,Tmax1,R1, E1,Imax2,Tmax2,R2, E2):
    return W_func(T,Imax1,Tmax1,R1, E1)+W_func(T,Imax2,Tmax2,\
R2, E2)
params, cov=optimize.curve_fit(total_TL,x_data,y_data,
p0=(119,463,1e-4,1.1,20,524,1e-4,1.2))
plt.subplot(2,1, 1)
plt.plot(x_data,y_data,'o',label='LBO data')
plt.plot(x_data, total_TL(x_data, *params),'+-',\
label='Fitted GOK')
plt.plot(x_data, W_func(x_data, *params[0:4]),'-',\
label='peak 1')
plt.plot(x_data, W_func(x_data, *params[4:8]),'-',\
label='peak 2')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('TL signal [a.u.]')
plt.xlabel(r'Temperature T [K]')
plt.ylim(0,140);
plt.text(400, 100,'Sample')
plt.text(400, 80,'LBO')
plt.subplot(2,1, 2)
plt.scatter(x_data,total_TL(x_data, *params)
   -y_data,c='r',linewidth=2,label='Residuals')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('Residuals')
plt.xlabel(r'Temperature T [K]')
plt.ylim(-10,10);
plt.tight_layout()
imax1,Tmax1, R1,E1=int(params[0]),round(params[1],2),\
round(params[2],2),round(params[3],3)
imax2,Tmax2,R2,E2=int(params[4]),round(params[5],2),\
round(params[6],2),round(params[7],3)
beta= 1
dR1 = round(np.sqrt(cov[2][2]),2)
dE1 = round(np.sqrt(cov[3][3]),2)
dR2 = round(np.sqrt(cov[6][6]),2)
dE2 = round(np.sqrt(cov[7][7]),2)
res=total_TL(x_data, *params)-y_data
FOM=round(100*np.sum(abs(res))/np.sum(y_data),2)
myTable = PrettyTable(["Imax", "Tmax","R", "dR","E(eV)",\
"dE(eV)","FOM(%)"])
myTable.add_row([imax1,Tmax1,R1,dR1,E1,dE1,FOM]);
myTable.add_row([imax2,Tmax2,R2,dR2,E2,dE2,"-"]);
print(myTable)
```

```
  +------+--------+------+------+-------+--------+--------+
  | Imax |  Tmax  |  R   |  dR  | E(eV) | dE(eV) | FOM(%) |
  +------+--------+------+------+-------+--------+--------+
  | 118  | 464.35 | 0.04 | 0.02 | 1.221 |  0.01  |  2.7   |
  |  23  | 518.24 | 0.63 | 0.69 | 1.668 |  0.46  |   -    |
  +------+--------+------+------+-------+--------+--------+
```

```
plt.show()
```

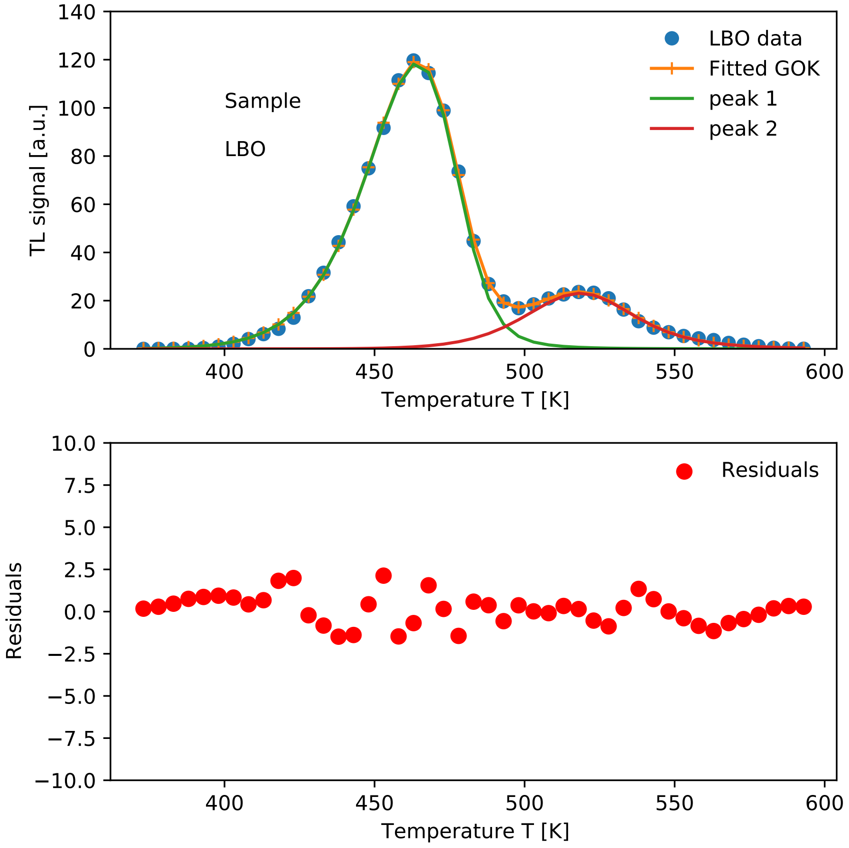


Fig. 1.7: Deconvolution of TL data containing two peaks from the dosimetric material LBO, using the transformed KV-TL equation based on the Lambert W function. For more details of the experiment, see Kitis et al. [33].

1.12 CGCD of complex TL glow curves using the transformed GOK equation

Figure shows the application of the transformed GOK equation, to analyze the same TL glow curve as in the previous section, from sample LBO (Kitis et al. [33]).

The s values of the two peaks are found using the following GOK expression, which contains the kinetic parameters b, E , the heating rate β, and the temperature of maximum intensity T_m:

$$s = \frac{\beta\, E}{k_B T_m^2}\exp\left(\frac{E}{k_B T_m}\right)\left[1 + (b-1)\, 2k_B T/E\right]^{-1} \tag{1.36}$$

Code 1.8: Deconvolution of TL user data (.txt file, GOK)

```
# Deconvolution of TL user data (.txt file, GOK)
from scipy import optimize
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable
import warnings
warnings.filterwarnings("ignore")
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
data = np.loadtxt('lbodata.txt')
x_data,y_data = data[:, 0], data[:, 1]
kB=8.617E-5
def TL(T, imax,b, En,Tmax):
    return imax* np.exp(En/(kB*T)*(T-Tmax)/Tmax)*(b**\
((b/(b-1))))*((1+(b-1)*2*kB*Tmax/En+(b-1)*(1-2*kB*T/En)*\
np.exp(En/(kB*T)*(T-Tmax)/Tmax)*(T**2.0)/(Tmax**2.0))**\
(b/(1-b)))
def total_TL(T, imax1,b1, En1,Tmax1, imax2,b2, En2,Tmax2):
    return TL(T, imax1,b1, En1,Tmax1)+TL(T,imax2,b2, En2,Tmax2)
params, cov = optimize.curve_fit(total_TL, x_data,
y_data,bounds=((80,1.001,.8,430, 20,1.001,.8,480),
(140,2.0,1.3,480, 40,2.0,1.3,540)))
plt.subplot(2,1, 1)
plt.scatter(x_data, y_data, label='Experiment')
plt.plot(x_data, total_TL(x_data, *params),
c='r',linewidth=3, label='GOK equation')
plt.plot(x_data,TL(x_data,*params[0:4]))
plt.plot(x_data,TL(x_data,*params[4:8]))
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('TL signal [a.u.]')
plt.xlabel(r'Temperature T [K]')
plt.ylim(0,140);
plt.text(400, 100,'Sample')
plt.text(400, 80,'LBO')
plt.subplot(2,1, 2)
plt.scatter(x_data,total_TL(x_data, *params)
   -y_data,c='r',linewidth=2,label='Residuals')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('Residuals')
plt.xlabel(r'Temperature T [K]')
plt.ylim(-10,10);
plt.tight_layout()
imax1,b1,E1,Tmax1=int(params[0]),round(params[1],2),\
round(params[2],2),int(params[3]),
imax2,b2,E2,Tmax2=int(params[4]),round(params[5],2),\
round(params[6],2),int(params[7]),
beta= 1
s1=np.exp(E1/(kB*Tmax1))*(beta*E1/(kB*(Tmax1**2.0)))\
/(1+(b1-1)*2*kB*Tmax1/E1)
s1=format(s1,"10.2E")
s2=np.exp(E2/(kB*Tmax2))*(beta*E2/(kB*(Tmax2**2.0)))\
/(1+(b2-1)*2*kB*Tmax2/E2)
s2=format(s2,"10.2E")
db1 = round(np.sqrt(cov[1][1]),2)
dE1 = round(np.sqrt(cov[2][2]),2)
db2 = round(np.sqrt(cov[5][5]),2)
dE2 = round(np.sqrt(cov[6][6]),2)
res=total_TL(x_data, *params)-y_data
FOM=round(100*np.sum(abs(res))/np.sum(y_data),2)
myTable = PrettyTable(["Imax", "b","db", "E(eV)",\
"dE(eV)", "s(s^-1)","FOM(%)"])
myTable.add_row([imax1,b1,db1,E1,dE1,s1,FOM]);
myTable.add_row([imax2,b2,db2,E2,dE2,s2,"-"]);
print(myTable)
```

```
  +------+------+------+-------+--------+------------+--------+
  | Imax |  b   |  db  | E(eV) | dE(eV) |  s(s^-1)   | FOM(%) |
  +------+------+------+-------+--------+------------+--------+
  | 116  | 1.06 | 0.03 |  1.24 |  0.02  |   2.10E+12 |  3.05  |
  |  23  | 1.5  | 0.22 |  1.3  |  0.18  |   2.73E+11 |   -    |
  +------+------+------+-------+--------+------------+--------+
```

```
plt.show()
```

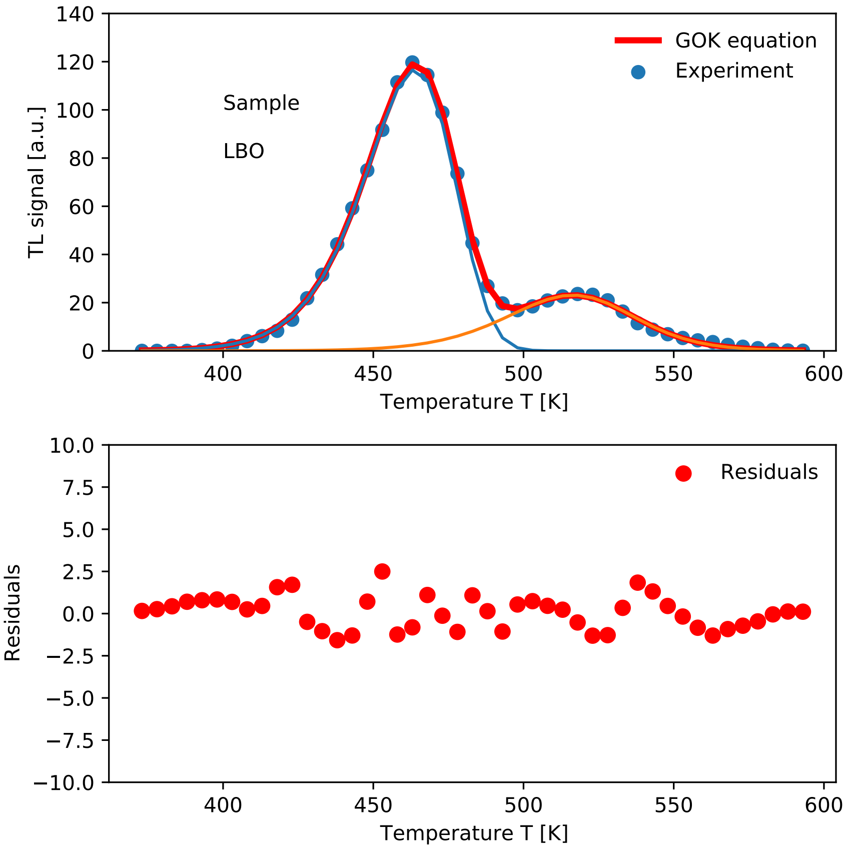


Fig. 1.8: Deconvolution of experimental data for dosimetric material LBO with 2 peaks, by using the transformed GOK equation. For more experimental details, see Kitis et al. [33].

1.13 CGCD of LiF glow curve using the transformed KV-TL equation

In this section we use the transformed KV-TL equations to fit a 9-peak LiF glow curve from the GLOCANIN project (Bos et al. [9, 10])

This Python code can be easily generalized to a TL glow curve with any number of peaks. In some situations, when attempting to analyze a multipeak

TL glow curve as in this case, it may be necessary to increase the maximum number of iterations, by using e.g. the parameter *maxfev=10000* when calling the *optimize.curve_fit* function.

Code 1.9: Deconvolution of 9-peak glow curve using the transformed KV-eqt

```
#Deconvolution of 9-peak glow curve using the transformed
# KV-eqt
from scipy import optimize
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable
import warnings
warnings.filterwarnings("ignore")
from scipy.special import lambertw
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
data = np.loadtxt('Refglow009.txt')
x_data,y_data = data[:, 0], data[:, 1]
plt.scatter(x_data, y_data, label='Experiment')
kB=8.617E-5
def TL(T,imax,R,E,Tmax):
    F=kB*(T**2.0)*np.exp(-E/(kB*T))*(1-2*kB*T/E)/E
    Fm=kB*(Tmax**2.0)*np.exp(-E/(kB*Tmax))*(1-2*kB*Tmax/E)/E
    a=kB*Tmax**2.0*(1-1.05*R**1.26)
    Z=R/(1-R)-np.log((1-R)/R)+(F*E*np.exp(E/(kB*Tmax)))/a
    Zm=R/(1-R)-np.log((1-R)/R)+(Fm*E*np.exp(E/(kB*Tmax)))/a
    argW=np.real(lambertw(np.exp(Z)))
    argWm=np.real(lambertw(np.exp(Zm)))
    return imax*np.exp(-E/(kB*T)*(Tmax-T)/Tmax)*\
    (argWm+argWm**2.0)/(argW+argW**2.0)
def total_TL(T, *inis):
    u=np.array([0 for i in range(len(x_data))])
    imaxs,     Rs,        Es,         Tmaxs=\
    inis[0:9],inis[9:18],inis[18:27], inis[27:36]
    for i in range(8):
        u=u+TL(T,imaxs[i],Rs[i], Es[i],Tmaxs[i])
    return u
inis = (9824,21009,27792,50520,7153,5496,6080,1641,2316,
0.01,0.01,.01,.01,.01,.01,.01,.01,.01,
1.24,1.36,2.10, 2.65,1.43, 1.16,2.48,2.98,2.25,
 387,428,462,488,493,528,559,585, 602)
params, params_covariance = optimize.curve_fit(total_TL,\
x_data,y_data,p0=inis)
plt.subplot(2,1, 1)
plt.scatter(x_data, y_data, label='Experiment')
plt.plot(x_data, total_TL(x_data,
*params),c='r',linewidth=3, label='CGCD')
for i in range(0,9):
    plt.plot(x_data, TL(x_data, *params[i:36:9]))
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('TL signal [a.u.]')
plt.xlabel(r'Temperature T [K]')
plt.text(350, 58000,'TLD-100')
plt.text(350, 50000,'GLOCANIN')
plt.text(350, 42000,'Refglow#9')
plt.subplot(2,1, 2)
plt.scatter(x_data,total_TL(x_data, *params)
   -y_data,c='r',linewidth=2,label='Residuals')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('Residuals')
plt.xlabel(r'Temperature T [K]')
plt.ylim(-2000,2000);
plt.tight_layout()
res=total_TL(x_data, *params)-y_data
FOM=round(100*np.sum(abs(res))/np.sum(y_data),2)
myTable = PrettyTable(["Imax", "R", "E (eV)", "Tmax (K)"])
for i in range(0,9):
    myTable.add_row(np.round(params[i:36:9],2));
print(myTable)
```

```
  +----------+------+--------+----------+
  |   Imax   |  R   | E (eV) | Tmax (K) |
  +----------+------+--------+----------+
  | 9681.17  | 0.0  |  1.22  |  387.33  |
  | 20436.75 | 0.06 |  1.32  |  428.24  |
  | 24498.06 | 0.21 |  1.77  |  461.87  |
  | 51888.71 | 0.05 |  2.3   |  488.05  |
  | 8209.94  | 0.11 |  1.44  |  490.33  |
  | 5253.75  | 0.07 |  1.0   |  525.88  |
  | 5312.01  | 0.2  |  2.32  |  557.91  |
  | 2945.38  | 0.86 |  1.6   |  590.38  |
  |  2316.0   | 0.01 |  2.25  |  602.0   |
  +----------+------+--------+----------+

print('FOM=',FOM,' %')

  FOM= 1.03  %

plt.show()
```

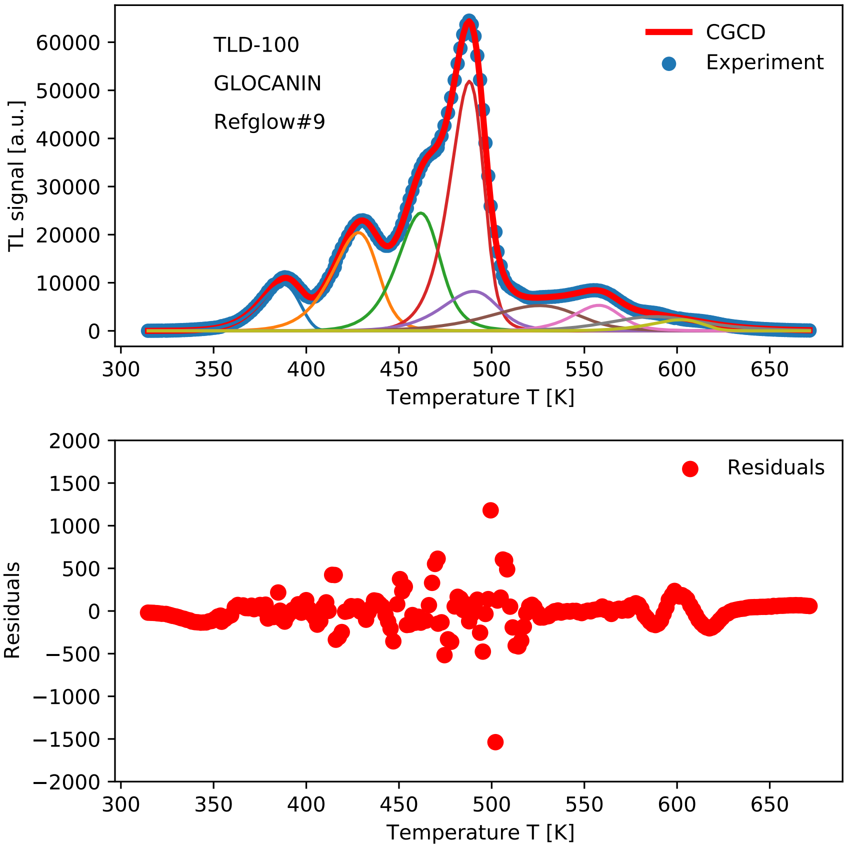


Fig. 1.9: Deconvolution of a glow curve from the GLOCANIN project using 9 peaks with the transformed KV-TL equation, and with user-supplied initial kinetic parameters (Bos et al. [9]). Bottom: Histogram of residuals $y_i^{expt} - y_i^{fit}$ from the best fit shown in (a), for the nine-peak TL glow curve of the GLOCANIN project, with a GOK model.

1.14 CGCD of LiF glow curve using the transformed GOK equation

In this section we use the transformed GOK equations, with the four fitting parameters: I_M, T_M, E and b. As discussed previously, the *curve_fit* function uses the Levenberg-Marquardt algorithm and also supports constraining and/or fixing the parameters.

The code consists of a call to the GOK model represented by Eq.(1.33). The data analyzed in the example of Fig.1.10 represent a typical TL glow curve for the well-known material LiF (TLD-700).

The Python code calls two functions, *TL* and *total_TL* to evaluate the intensity of individual peaks and also the sum of the 9 peaks, respectively. The FOM= 0.84 % which indicates an excellent fit to the experimental data. The residuals are calculated as are the difference between the experimental and fitted values ($y_i^{expt} - y_i^{fit}$), and a histogram plot of the distribution of the residuals of the fitting process is shown underneath the best fit plot to the data. The symmetric shape of the distribution of residuals indicates the absence of non-systematic sources of error in the fitting process.

The FOM= 0.99 % indicating an excellent fit to the data. However, as mentioned previously, the parameters from the GOK model have no physical meaning.

Code 1.10: Deconvolution of 9-peak Glocanin TL data (GOK)

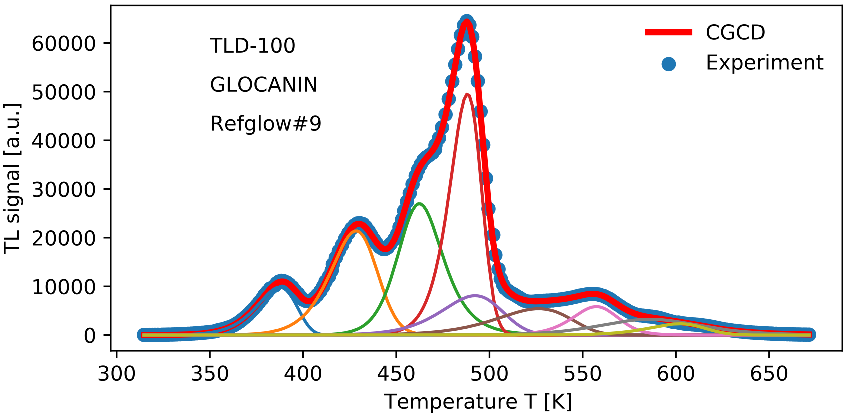
```
from scipy import optimize
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable
import warnings
warnings.filterwarnings("ignore")
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
data = np.loadtxt('Refglow009.txt')
x_data,y_data = data[:, 0], data[:, 1]
plt.scatter(x_data, y_data, label='Experiment')
kB=8.617E-5
def TL(T, imax,b, En,Tmax):
    return imax* np.exp(En/(kB*T)*(T-Tmax)/Tmax)*(b**\
    ((b/(b-1))))*((1+(b-1)*2*kB*Tmax/En+(b-1)*(1-2*kB*T/\
    En)*np.exp(En/(kB*T)*(T-Tmax)/Tmax)*(T**2.0)/(Tmax**\
    2.0))**(b/(1-b)))
def total_TL(T, *inis):
    u=np.array([0 for i in range(len(x_data))])
    imaxs,     bs,        Es,         Tmaxs=\
    inis[0:9],inis[9:18],inis[18:27], inis[27:36]
    for i in range(8):
        u=u+TL(T,imaxs[i],bs[i], Es[i],Tmaxs[i])
    return u
inis = (9824,21009,27792,50520,7153,5496,6080,1641,2316,
1.02, 1.15, 1.99,1.20, 1.28,1.19,1.40,1.01,1.18,
1.24,1.36,2.10, 2.65,1.43, 1.16,2.48,2.98,2.25,
 387,428,462,488,493,528,559,585, 602)
params, params_covariance = optimize.curve_fit(total_TL,\
x_data,y_data,p0=inis)
plt.subplot(2,1, 1)
plt.scatter(x_data, y_data, label='Experiment')
plt.plot(x_data, total_TL(x_data,
*params),c='r',linewidth=3, label='CGCD')
for i in range(0,9):
    plt.plot(x_data, TL(x_data, *params[i:36:9]))
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('TL signal [a.u.]')
plt.xlabel(r'Temperature T [K]')
plt.text(350, 58000,'TLD-100')
plt.text(350, 50000,'GLOCANIN')
plt.text(350, 42000,'Refglow#9')
plt.subplot(2,1, 2)
res=total_TL(x_data, *params)-y_data;
plt.hist(res,22,label='Residuals');
plt.ylabel('Distribution of residuals')
plt.xlabel(r'Residuals')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.tight_layout()
FOM=round(100*np.sum(abs(res))/np.sum(y_data),2)
myTable = PrettyTable(["Imax", "b", "E (eV)", "Tmax (K)"])
for i in range(0,9):
    myTable.add_row(np.round(params[i:36:9],2));
print(myTable)
```

```
  +----------+------+--------+----------+
  |   Imax   |  b   | E (eV) | Tmax (K) |
  +----------+------+--------+----------+
   | 9834.82  | 1.03 |  1.22  |  387.38  |
   | 21411.88 | 1.26 |  1.4   |  428.39  |
   | 26999.01 | 2.2  |  2.3   |  462.36  |
   | 49530.68 | 1.17 |  2.61  |  488.2   |
   | 8084.15  | 1.15 |  1.36  |  492.17  |
   |  5418.4  | 1.0  |  1.16  |  526.22  |
   | 5842.23  | 1.54 |  2.75  |  557.31  |
   | 3242.52  | 1.78 |  1.46  |  585.71  |
   |  2316.0  | 1.18 |  2.25  |  602.0   |
   +----------+------+--------+----------+

print('FOM=',FOM,' %')

   FOM= 0.99  %

plt.show()
```



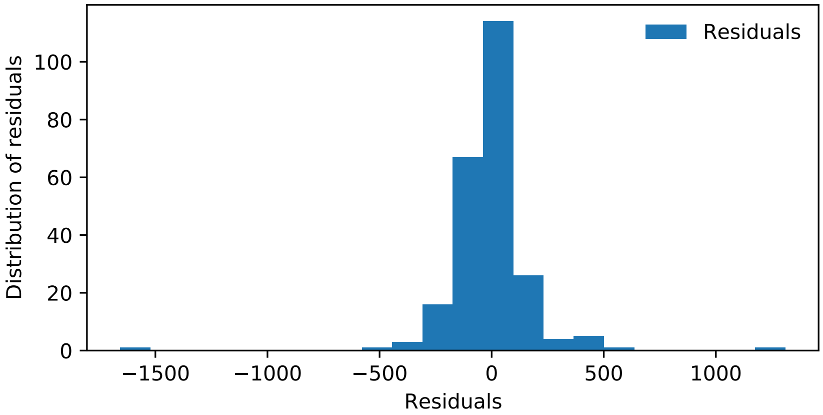


Fig. 1.10: Deconvolution of a glow curve from the GLOCANIN project using 9 peaks with a GOK model, and with user-supplied initial kinetic parameters (Bos et al. [9]). Bottom: Histogram of residuals $y_i^{expt} - y_i^{fit}$ from the best fit shown in (a), for the nine-peak TL glow curve of the GLOCANIN project, with a GOK model.

Table 1.2 summarizes the results of analyzing the same experimental data for TLD-700, using three different methods: the KV equation of this chapter based on the Lambert function, the GOK method is based on the Python code of this chapter, and the GOK method within the R package *tgcd* developed by Peng et al. [48].

From this Table it is clear that the CGCD results from these unrestricted least squares codes are not unique. Even though the three codes started with the *same* initial set of parameter guesses, they all produced very different E values for the 9 peaks.

When using this type of code for routine analysis of TL glow curves (e.g. in medical dosimetry), researchers fix several of the fitting parameters in the CGCD codes, so that the results from different TL glow curves of the same material can be compared to each other.

For more details of the models and correspodning R codes in this chapter, see the recent book by Pagonis [39].

peak #	T$_{max}$ (K)	E ,eV	E ,eV	E ,eV
		(KV)	(GOK)	(*tgcd* R code)
1	387	1.22	1.22	1.24
2	428	1.32	1.40	1.36
3	462	1.77	2.30	2.06
4	488	2.30	2.61	2.50
5	493	1.44	1.36	1.41
6	527	1.00	1.16	1.47
7	559	2.32	2.75	2.18
8	585	1.60	1.46	3.34
9	602	2.25	2.25	2.25

Table 1.2: Summary of E values obtained from fitting the same TL glow curve using the KV equation, the GOK empirical equations. The last column shows the results obtained using the GOK CGCD function in the R package *tgcd* (Peng et al. [48]).

Chapter 2
TL FROM QUANTUM TUNNELING PROCESSES: DATA ANALYSIS

2.1 Example of analyzing experimental data for the anomalous fading effect

In this chapter we present examples of analyzing experimental TL data for materials exhibiting quantum tunneling processes. We use the KP-TL analytical equation to analyze TL from feldspars and oter dosimetric materials. We also discuss how to analyze samples which have undergone optical or thermal pretreatments before measurement of their RTL signals.

Extensive experimental and modeling studies revealed a time dependent localized tunneling/recombination probability in a variety of dosimetric materials. Of major interest is the "anomalous fading" of luminescence signals observed mainly in feldspars and apatites (Lamothe et al. [35], and references therein). These studies have provided very strong evidence that the anomalous fading effect is due to quantum mechanical tunneling from the ground state of the trap to the luminescence center (Visocekas et al. [62], [63]).

Pagonis and Kitis [42] discussed the traditional method of analyzing anomalous fading effects, in which the decay of luminescence signal after the end of irradiation is written as (Li and Li [37], Polymeris et al. [52], Lamothe et al. [35]):

$$\frac{L}{L_c} = 1 - \frac{g}{100} \log_{10} \left(\frac{t}{t_c} \right) \tag{2.1}$$

or, in terms of the natural logarithm function instead of the base-10 logarithm (Polymeris et al. [52]):

$$\frac{L}{L_c} = 1 - K \ln \left(\frac{t}{t_c} \right) \tag{2.2}$$

$$g = 230.2\, K \tag{2.3}$$

where the dimensionless g factor describes the percentage loss of luminescence signal per decade of time, and t_c is the time when the first measurement L_c

takes place. In practical situations one plots the remnant luminescence signal L as a function of $\log_{10}(t)$, and the slope of the linear part of this graph is the g-factor. According to this definition, the g-value depends on the exact point in time (t_c) in which the first measurement is carried out, as well as on the choice of the linear part of the experimental data (see Li and Li [37], for a more detailed discussion).

An example of how the g-factor is evaluated from experimental data is shown in the following code, using the experimental data by Polymeris et al. [52], on Durango apatite. Figure 2.1a shows the remnant TL signal measured at two different times. The results of the calculation are shown in Fig.2.1b. In this method one finds the slope K of the graph of L/L_c vs $\ln(t)$, and then Eq.(2.3) is used to find the value of g.

In this example we obtain g =20.5% per decade, which is a high g-value characteristic of several types of apatites.

Code 2.1: Anomalous fading (AF) and the g-factor

```
# Anomalous fading (AF) and calculating  the g-factor
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from prettytable import PrettyTable
def linearFunc(x2,intercept,slope):
    y2 = intercept + slope * x2
    return y2
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
data = np.loadtxt('durnago0sok.txt')
x0, y0 = np.array(data[:, 0]), np.array(data[:, 1])
data = np.loadtxt('durango10daysok.txt')
x1, y1 = np.array(data[:, 0]), np.array(data[:, 1])
data = np.loadtxt('DurangoAFdataok.txt')
x2, y2 = np.array(data[:, 0]), np.array(data[:, 1])
plt.subplot(1,2, 1)
plt.plot(x0,y0,'b^-',label='t=0s')
plt.plot(x1,y1,'ro-',label='t=10 days')
plt.title('(a)')
plt.ylim(0,1.1e5);
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('Remnant TL signal [a.u.]')
plt.xlabel(r'Temperature [$^{o}$C]')
plt.text(100, 6e4,'Durango')
plt.text(100,5e4,'apatite')
#######
plt.subplot(1,2, 2)
plt.plot(x2,y2,"o",c="r")
plt.ylabel('Remnant TL signal [a.u.]')
plt.xlabel(r'ln(time)')
plt.title('(b)')
plt.text(5,.9,"g-factor")
plt.text(5,.85,r"analysis")
plt.text(5,.76,r"g=21.0%")
params,cov=curve_fit(linearFunc,x2,y2)
plt.plot(x2,linearFunc(x2,*params),label='Fit')
inter,slope = np.round(params,4)
d_inter,d_slope = np.round(np.sqrt(np.diag(cov)),3)
residuals = y2- linearFunc(x2, *params)
ss1=np.sum(residuals**2.0)
ss2 = np.sum((y2-np.mean(y2))**2.0)
rsqr=np.round(1-ss1/ss2,3)
g=np.round(-230.2*slope)
myTable=PrettyTable(['g (%)','slope (a.u.)','dslope (a.u.)',\
'R^2'])
myTable.add_row([g,-slope, d_slope, rsqr]);
print(myTable)
```

```
  +-------+--------------+---------------+-------+
  | g (%) | slope (a.u.) | dslope (a.u.) |  R^2  |
  +-------+--------------+---------------+-------+
  |  21.0 |    0.0892    |     0.003     | 0.989 |
  +-------+--------------+---------------+-------+
```

```
plt.tight_layout()
plt.show()
```

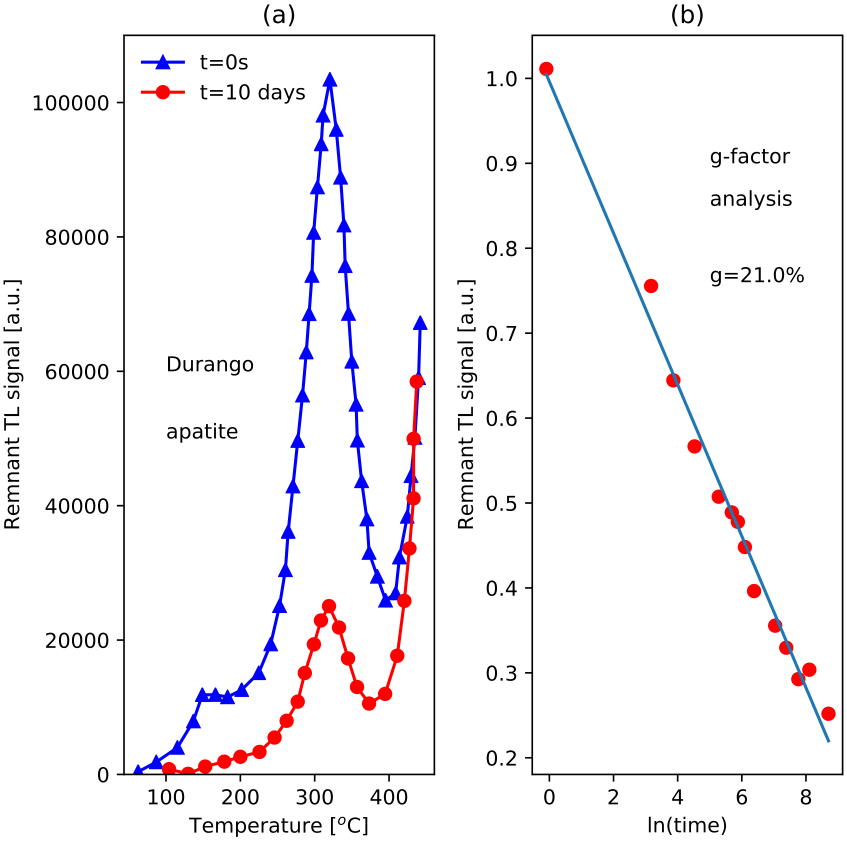


Fig. 2.1: Anomalous fading effect in Durango apatite. (a) The TL signal is measured immediately after irradiation, and after 10 days have elapsed at room temperatures. (b) Analysis of the remnant TL signal from (a), to obtain the *g*-factor for this materials. For more details, see Polymeris et al. [52].

In the next two sections we give specific examples of using the general KP equations (2.5), to analyze experimental CW-IRSL and TL data, by using appropriate Python codes.

2.2 Fitting single-peak TL data from unfaded samples using the KP-TL equation

The Kitis-Pagonis analytical solution of the EST model

Kitis and Pagonis [29] derived an analytical equation solution for the EST model, by considering quasi-equilibrium conditions (QE). One of the assumptions in the model is that the concentration of electrons $n_e(t)$ in the excited state at any time is several orders of magnitude smaller than the concentration of electrons $n_g(t)$ in the ground state of the system. Furthermore, the relaxation of the excited states is much faster than the time scales of TL experiments. Therefore, one can model the excited state in the quasi-steady approximation. Specifically, the time scale for electronic relaxation processes involving the excited states is of order of ns or ms. On the other hand the time scale for TL processes is of the order of s.

Based on these observations, Kitis and Pagonis [29] carried out extensive algebra, and obtained the following analytical solutions for the remaining concentration of electrons $n(t)$ during a TL measurement:

Loss of charge due to excited state tunneling

$$n(t) = n_0 \exp\left(-\rho' \, \ln\left[1 + \frac{z\, s_{tun}}{B} \int_0^t A(t)\, dt\right)\right]^3\right) \tag{2.4}$$

Notice that this equation for the loss of charge $n(t)$ due to *excited state tunneling* in the EST model, is almost identical to corresponding equation for the loss of charge due to *ground state tunneling* in the GST model. As in the original paper by Jain et al. [23], Kitis and Pagonis [29] assumed that $s_{tun} = B$ in order to simplify the notation in the model. This assumption does not alter the results from the model, because the equations contain the combination $A(t)s_{tun}/B$, rather than the variables $A(t)$, s_{tun} and B separately.

The corresponding luminescence intensity $I(t)$ is found from the derivative $-dn/dt$ of Eq.(2.4), and we will refer to this analytical equation as the *general Kitis-Pagonis equation (KP)*:

The general KP-equation for luminescence intensity

$$I(t) = 3\, n_0\, \rho'\, z\, p(t)\, F(t)^2\, e^{-F(t)}\, e^{-\rho'(F(t))^3} \tag{2.5}$$

$$F(t) = \ln\left(1 + \frac{z\, s_{tun}}{B} \int_0^t A(t)\, dt\right) \tag{2.6}$$

Equation (2.5) along with Eq.(2.6), has been termed the *fifth master equation* in the review paper by Kitis et al. [32]. These analytical expressions can characterize TL, IRSL and ITL signals in the TLT model, as long as one is dealing with freshly irradiated samples, i.e. samples which have not undergone any thermal or optical treatments after irradiation.

The function $F(t)$ takes the following form for the various stimulated luminescence phenomena.

In the case of TL, we have $A(t) = s_{th} \exp\left(-E/k_B T\right)$ where E, s_{th} are the thermal parameters for the trap, and $F(t)$ has the form:

$$F_{\mathrm{TL}}(t) = \ln\left(1 + \frac{z\, s_{th}\, s_{tun}}{B}\,\frac{1}{\beta}\int_{T_0}^{T} e^{-\frac{E}{k_B\, T'}}\, dT'\right) \tag{2.7}$$

$$F_{\mathrm{TL}}(t) = \ln\left(1 + \frac{z\, s}{\beta}\int_{T_0}^{T} e^{-\frac{E}{k_B\, T'}}\, dT'\right) \tag{2.8}$$

where $s = s_{th}\; s_{tun}/B$ is the effective rate constant (or effective frequency factor) characterizing the TL process in the EST model.

This paragraph and equations should be moved to ITL chapter. Finally, for ITL decay curves, we have $A(t) = s_{th}\exp\left(-E/k_BT_{ISO}\right)$ where E, s_{th} are the thermal parameters for the trap and T_{ISO} is the temperature of the isothermal process, and $F(t)$ has the form:

$$F_{\mathrm{ITL}}(t) = \ln\left(1 + \frac{z\, s_{th}\, s_{tun}}{B}\,\exp\left(-E/k_BT_{ISO}\right)t\right) = \ln\Big(1 + z\,A\,t\Big) \tag{2.9}$$

where $A = s_{th}\; s_{tun}\exp\left(-E/k_BT_{ISO}\right)/B$ (s^{-1}) is the effective rate constant characterizing the ITL process in the EST model.

The fifth master equation Eq.(2.5) was tested by Kitis and Pagonis [29], by comparing it with the numerical solution of the differential equations in the EST model (Kitis and Pagonis [29]; [30]; Pagonis and Kitis [42]). This equation has been also tested extensively during the past decade, by comparing it with many different types of experimental signals, from different types of natural and artificial dosimetric materials (Sfampa et al. [57], Sahiner et al. [55], Kitis et al. [33], Polymeris et al. [50]).

For freshly irradiated samples, the TL signals are analyzed using the following KP equation for the intensity of a TL signal (see Eqs.29-30 in Kitis and Pagonis [29]):

$$I_{\mathrm{TL}}(t) = 3\, n_0\, \rho'\, z\, p(t)\, F(t)^2\, e^{-F(t)}\, e^{-\rho'(F(t))^3} \tag{2.10}$$

where $F_{\mathrm{TL}}(t)$ is given above in Eq.(2.7). For practical work fitting experimental data, we use this equation in the following form, which will be hereafter named the *KP-TL equation*:

The KP-TL equation

$$I_{\mathrm{TL}}(t) = \frac{B\,F(t)^2\,\exp\left[-\rho'\,F(T)^3\right]\left(E^2 - 6k_B^2T^2\right)}{Ek_BsT^2z - 2k_B^2sT^3z + \exp\left(E/k_BT\right)E\beta} + bgd \tag{2.11}$$

$$F(T) = \ln\left[1 + \frac{z\, sk_BT^2}{\beta E}\, e^{-\frac{E}{k_B\, T}}\left(1 - \frac{2k_BT}{E}\right)\right] \tag{2.12}$$

The fitting parameters are the scaling constant B, thermal activation energy E (eV), effective frequency factor s (s^{-1}) for the TL process, dimensionless acceptor density ρ' and a constant background bgd (if necessary). Here β is the heating rate, in the following code we use β =1 K/s. Even though the

KP-TL equation looks complex, it is rather easy and straightforward to code it.

In order to obtain reliable results with the KP-TL equation, one *must* constraint the code using experimental data, because there are infinite combinations of the parameters E, s, ρ' which will fit the data. It is best to use the known value of E, which can be obtained from separate initial rise and $T_{\text{max}} - T_{\text{stop}}$ experiments. In the following code we use $E = 1.45$ eV, a value obtained from a separate analysis using the initial rise method, and the least squares fitting parameters are I_0, s, ρ'. The experimental data are from a feldspar sample KST4 which was irradiated and heated up to 300°C, to thermally remove the lower temperature peaks in the TL glow curve, before measuring the TL signal. It is assumed that heating the sample to 300°C does not affect the trap responsible for the TL signal at 320°C.

Notice that in this figure the KP-TL equation fails to describe the TL glow curve very well at low temperatures. This is due to the approximations made during derivation of the KP-TL equation (see the detailed discussion in Kitis and Pagonis [30]).

For more details on this experimental data, see Pagonis et al. [41], and Polymeris et al. [50]. It is also important to remember that the physical meaning of the parameter s in the KP-TL equation is an *effective* frequency factor for the TL process (equal to $s = s_{th}\ s_{tun}/B$), and *not* the frequency factor s_{th} for the thermal activation process.

In order to obtain reliable results with the KP-TL equation, one *must* constraint the code using experimental data, because there are infinite combinations of the parameters B, E, s, ρ' which will fit the data. It is best to use the known value of E, which can be obtained from separate initial rise and $T_{\text{max}} - T_{\text{stop}}$ experiments. In the following code we use $E = 1.45$ eV, a value obtained from a separate analysis using the initial rise method, and the least squares fitting parameters are B, s, ρ'. The experimental data are from a feldspar sample KST4 which was irradiated and heated up to 300°C, to thermally remove the lower temperature peaks in the TL glow curve, before measuring the TL signal. It is assumed that heating the sample to 300°C does not affect the trap responsible for the TL signal at 320°C, and the heating are β =1 K/s.

Notice that in Fig.2.3, the KP-TL equation fails to describe the TL glow curve very well at low temperatures. This is due to the approximations made during derivation of the KP-TL equation (see the detailed discussion in Kitis and Pagonis [30]).

Figure 2.2 shows a second example for sample MBO irradiated with 6 Gy.

The best fitting parameters are $\rho' = 0.0111 \pm 0.0003$, $s = 2.0 \times 10^{12}$ s^{-1},$B =$ 2.0×10^{12} and the FOM= 4.1 %. For an extensive analysis of the experimental data for this Mg_4BO_7: Dy, Na dosimeter, see the detailed description in Pagonis et al. [40].

Code 2.2: Fit MBO data with KP-TL equation

```
# Fit TL with KP-TL (Kitis-Pagonis) analytical equation
from scipy import optimize
from sympy import *
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
data = np.loadtxt('MBO6gynew.txt')
x_data,y_data = data[:, 0], data[:, 1]
z,   kB,      En=\
1.8, 8.617E-5,1.0
def test_func(x, B,rho, s):
    return B* np.exp(-rho*( (np.log(1+z*s*kB*(((x+\
273)**2.0)/np.abs(En))*np.exp(-En/(kB*(x+273)))*\
(1-2*kB*(x+273)/En)))**3.0))*(En**2.0-6*(kB**2.0)*\
     ((x+273)**2.0))*((np.log(1+z*s*kB*(((x+273)**2.0)/\
abs(En))*np.exp(-En/(kB*(x+273)))*(1-2*kB*(x+273)/\
En)))**2.0)/(En*kB*s*((x+273)**2)*z-2*(kB**2.0)*\
s*z*((x+273)**3.0)+np.exp(En/(kB*(x+273)))*En)
params, cov = optimize.curve_fit(test_func,\
x_data, y_data,bounds=(0,[1e20,.02,1e14]))
drho= round(np.sqrt(cov[1][1]),5)
plt.subplot(2,1, 1)
plt.plot(x_data, y_data,'o', c='lightgreen',label='Experiment')
plt.plot(x_data, test_func(x_data, *params),
label='KP-TL equation',   c='black',linewidth=2)
plt.title('(a)')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('TL signal [a.u.]')
plt.xlabel(r'Temperature T [$^{o}$C]')
#plt.text(400, 1.8e4,'KST4 feldspar')
plt.subplot(2,1, 2)
plt.plot(x_data,test_func(x_data, *params)-y_data,'o',\
label='Residuals')
plt.title('(b)')
plt.ylim(-1,1);
plt.ylabel('Residuals')
plt.xlabel(r'Temperature T [$^{o}$C]')
plt.tight_layout()
B,rho, s=format(params[0],"10.2E"),round(params[1],5),\
format(round(params[2],2),"10.2E")
res=test_func(x_data, *params)-y_data
```

```
FOM=round(100*np.sum(abs(res))/np.sum(y_data),2)
plt.show()
```

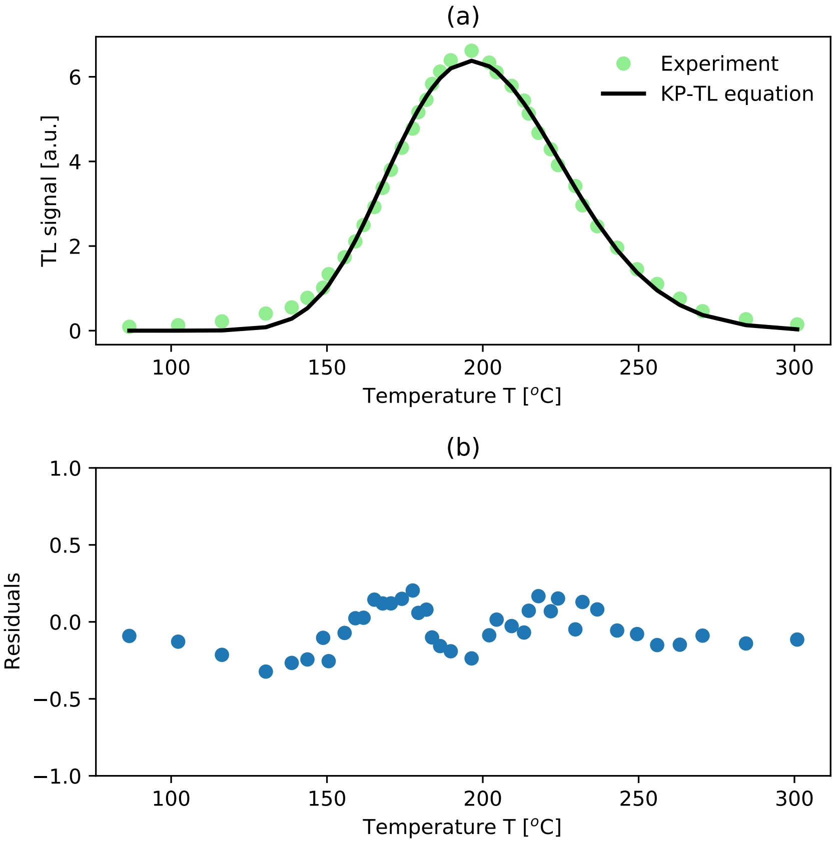


Fig. 2.2: Experimental TL glow curves from freshly irradiated Mg_4BO_7: Dy, Na sample, fitted using the KP-TL analytical equation. For more detailed modeling of experiments on this material, see Pagonis et al. [40].

For more details on the experimental data of Fig.2.3, see Pagonis et al. [41], and Polymeris et al. [50]. It is also important to remember that the physical meaning of the parameter s in the KP-TL equation is an *effective* frequency factor for the TL process (equal to $s = s_{th}\, s_{tun}/B$), and *not* the frequency factor s_{th} for the thermal activation process.

The best fitting parameters are $\rho' = 0.00953 \pm 0.00005$, $s = 3.7 \times 10^{12}$ s^{-1},$B = 3.2 \times 10^{17}$ and the FOM= 5.3 %.

Code 2.3: Fit TL for KST4 feldspar with KP-TL equation

```
# Fit KST4 TL with KP-TL analytical equation
from scipy import optimize
from sympy import *
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
data = np.loadtxt('ph300s0.txt')
x_data,y_data = data[:, 0], data[:, 1]
z,   kB,      En=\
1.8, 8.617E-5,1.45
def test_func(x, B,rho, s):
    return B* np.exp(-rho*( (np.log(1+z*s*kB*(((x+\
273)**2.0)/np.abs(En))*np.exp(-En/(kB*(x+273)))*\
(1-2*kB*(x+273)/En)))**3.0))*(En**2.0-6*(kB**2.0)*\
     ((x+273)**2.0))*((np.log(1+z*s*kB*(((x+273)**2.0)/\
abs(En))*np.exp(-En/(kB*(x+273)))*(1-2*kB*(x+273)/\
En)))**2.0)/(En*kB*s*((x+273)**2)*z-2*(kB**2.0)*\
s*z*((x+273)**3.0)+np.exp(En/(kB*(x+273)))*En)
params, cov = optimize.curve_fit(test_func,\
x_data, y_data,bounds=(0,[1e20,.02,1e14]))
drho= round(np.sqrt(cov[1][1]),5)
plt.subplot(2,1, 1)
plt.plot(x_data, y_data,'o', c='lightgreen',label='Experiment')
plt.plot(x_data, test_func(x_data, *params),
label='KP-TL equation',       c='black',linewidth=2)
plt.title('(a)')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('TL signal [a.u.]')
plt.xlabel(r'Temperature T [$^{o}$C]')
plt.text(400, 1.8e4,'KST4 feldspar')
plt.subplot(2,1, 2)
plt.plot(x_data,test_func(x_data, *params)-y_data,\
label='Residuals')
plt.title('(b)')
plt.ylim(-2000,2000);
plt.ylabel('Residuals')
plt.xlabel(r'Temperature T [$^{o}$C]')
plt.tight_layout()
B,rho, s=format(params[0],"10.2E"),round(params[1],5),\
format(round(params[2],2),"10.2E")
res=test_func(x_data, *params)-y_data
FOM=round(100*np.sum(abs(res))/np.sum(y_data),2)
plt.show()
```

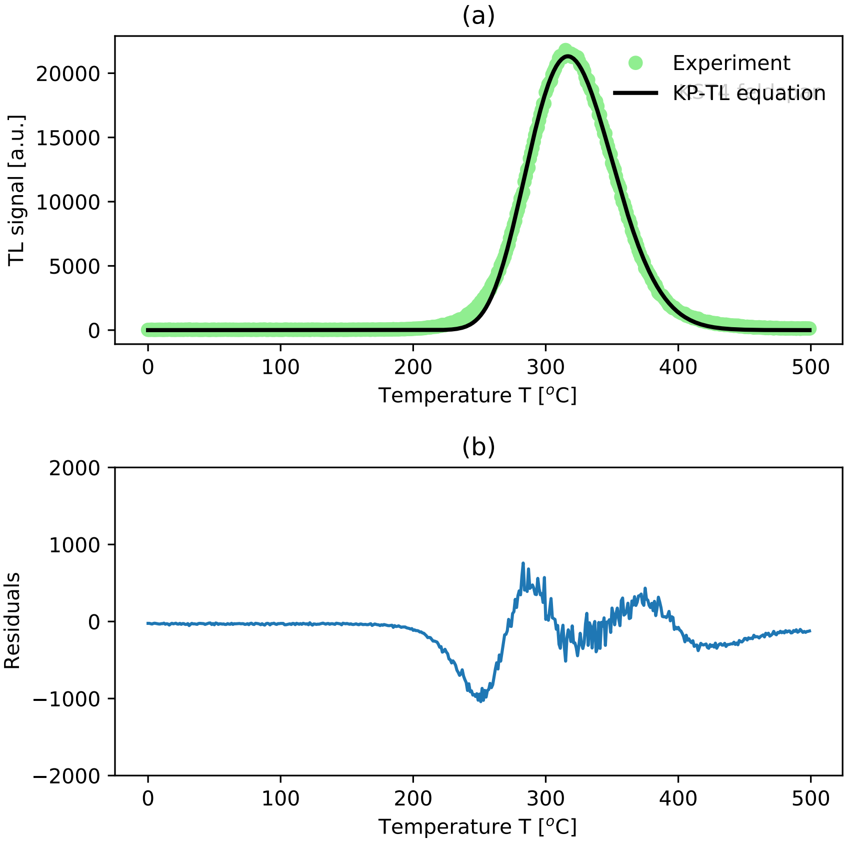


Fig. 2.3: Experimental TL glow curves from freshly irradiated KST4 feldspar sample, fitted using the KP-TL analytical equation. Note that the solid line does not describe the experimental data very accurately at low temperatures, due to the approximations involved in the KP equations. For more details and MC examples, see Pagonis et al. [41].

2.3 Fitting multiple-peak TL data from unfaded samples using the KP-TL equation

In this section we analyze complex TL glow curves in feldspars, using the KP-TL equation. The analysis is based on the following physical assumptions:

(a) Each TL glow curve in a freshly irradiated sample consists of at least 5 constituent glow peaks, each corresponding to a different activation energy E. The E values are determined in a separate fractional glow experiment. The donor-acceptor pairs for each trap are randomly distributed, and *each* trap in the unfaded sample is described by its own nearest neighbor distribution.

(b) There are many more acceptors in the material than donors, so that the material is characterized by a constant acceptor density parameter ρ'. Electrons from all five traps are likely to be accessing the same recombination center, and all traps are characterized by the same constant parameter ρ'.

Fig.2.4 shows the results of DC analysis of the TL glow curves for sample BAL21, based on the above assumptions. The result of the deconvolution of the complex TL glow curve is the sum of 5 individual peaks, located approximately at temperatures of 100, 150, 200, 250, 300°C. The amplitudes A obtained from the least squares fit procedure are $A = 5.9\times10^{16}$, 11.0×10^{16}, 15.5×10^{16}, 3.9×10^{16}, 0.98×10^{16} and the density parameter $\rho'=0.0038$.

For a Monte Carlo method of analyzing TL glow curves from feldspars using a microscopic description of the tunneling process, the reader is referred to Pagonis et al. [47].

Code 2.4: Deconvolution of 5-peak glow curve for BAL21 sample

```
#Deconvolution of 5-peak glow curve for BAL21 sample
# using the  KP-TL equation
from scipy import optimize
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable
import warnings
warnings.filterwarnings("ignore")
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
data = np.loadtxt('initialsensBAL21.txt')
x_data,y_data = data[:, 0], data[:, 1]
plt.scatter(x_data, y_data, label='Experiment')
kB=8.617E-5
nPks=5
z,   kB,     s, =1.8, 8.617E-5,1e12
def TL(T, B,En ,rho):
    return abs(B)* np.exp(-rho*( (np.log(1+z*s*kB*(((T+\
273)**2.0)/np.abs(En))*np.exp(-En/(kB*(T+273)))*\
(1-2*kB*(T+273)/En)))**3.0))*(En**2.0-6*(kB**2.0)*\
     ((T+273)**2.0))*((np.log(1+z*s*kB*(((T+273)**2.0)/\
abs(En))*np.exp(-En/(kB*(T+273)))*(1-2*kB*(T+273)/\
En)))**2.0)/(En*kB*s*((T+273)**2)*z-2*(kB**2.0)*\
s*z*((T+273)**3.0)+np.exp(En/(kB*(T+273)))*En)
def total_TL(T, *inis):
    u=np.array([0 for i in range(len(x_data))])
    Bs, rho=    inis[0:nPks],inis[-1]
    for i in range(nPks):
        u=u+TL(T,Bs[i],Ens[i],rho)
    return u
inis=(5e16,9e16,10e16,1e16,.1e16,.004)
Ens=[.82, .95,1.06,1.19,1.355]

params, params_covariance = optimize.curve_fit(total_TL,\
x_data,y_data,p0=inis)
plt.subplot(2,1, 1)
plt.scatter(x_data, y_data, label='BAL21 feldspar data')
plt.plot(x_data, total_TL(x_data,
*params),c='r',linewidth=3, label='Analytical KV-TL')
plt.plot(x_data, TL(x_data, params[0],Ens[0],params[5]))
for i in range(0,nPks):
     plt.plot(x_data, TL(x_data, params[i],Ens[i],params[5]))
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('TL signal [a.u.]')
plt.xlabel(r'Temperature T [K]')
plt.subplot(2,1, 2)
plt.scatter(x_data,total_TL(x_data, *params)
   -y_data,c='r',linewidth=2,label='Residuals')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('Residuals')
plt.xlabel(r'Temperature T [K]')
plt.ylim(-5e4,5e4);
plt.tight_layout()
res=total_TL(x_data, *params)-y_data
FOM=round(100*np.sum(abs(res))/np.sum(y_data),2)

print('FOM=',FOM,' %')
```

```
  FOM= 5.31  %
```

```
plt.show()
```

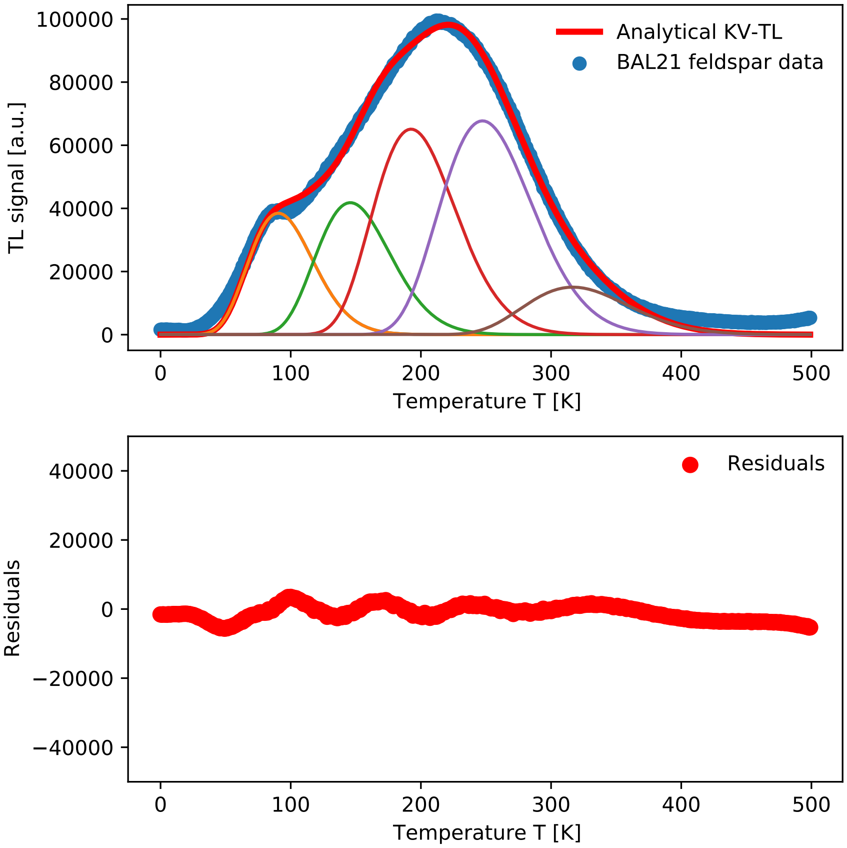


Fig. 2.4: Experimental TL glow curve from freshly irradiated feldspar BAL21 sample, fitted using 5 peaks and the KP-TL analytical equation. For additional modeling of experiments on this material, see Pagonis et al. [47, 50].

2.3.1 The transformed KP-TL equation

The transformed version of the analytical KP-TL equation is (Kitis and Pagonis [30]):

$$I(T) = I_m \, \exp\left[\frac{E\,(T-T_m)}{k\,T\,T_m}\right] \, \frac{\exp\left[-\left(\rho'\,F_m^3 + F_m\right)\right]}{\exp\left[-\left(\rho'\,F^3 + F\right)\right]} \tag{2.13}$$

where the dimensionless functions F, F_m are given by:

$$F = \ln\left[1 + \frac{1}{f_m}\,\frac{T^2}{T_m^2}\,\exp\left[\frac{E\,(T - T_m)}{k\,T\,T_m}\right]\left(1 - \frac{2\,k\,T}{E}\right)\right] \tag{2.14}$$

$$F_m = \ln\left[1 + \frac{1}{f_m}\,\left(1 - \frac{2\,k\,T_m}{E}\right)\right] \tag{2.15}$$

Kitis and Pagonis [30] showed that the dimensionless quantity f_m depends only on the dimensionless acceptor density parameter ρ', according to the empirical equation ([30], their Figure 5b):

$$f_m = 4.90537\ (\rho')^{1.21038} \tag{2.16}$$

The values of the parameters I_m and the corresponding temperature T_m can be determined directly from the experimental data, and the value of the activation energy E can be found by an initial rise analysis of the TL peak. The adjustable fitting parameter in the algorithm is the density parameter ρ', which is bounded in the code within the values of $\rho' = 0.002$ and $\rho' = 0.02$ (Sfampa et al. [56]).

Figure 2.5 shows a second example of the KP-TL algorithm, applied for freshly irradiated sample MBO. The best fitting parameter in this example is $\rho' = 0.0123 \pm 0.0004$, in agreement with the detailed study in Pagonis et al. [40].

Code 2.5: Deconvolution of MBO data with transformed KP-TL equation

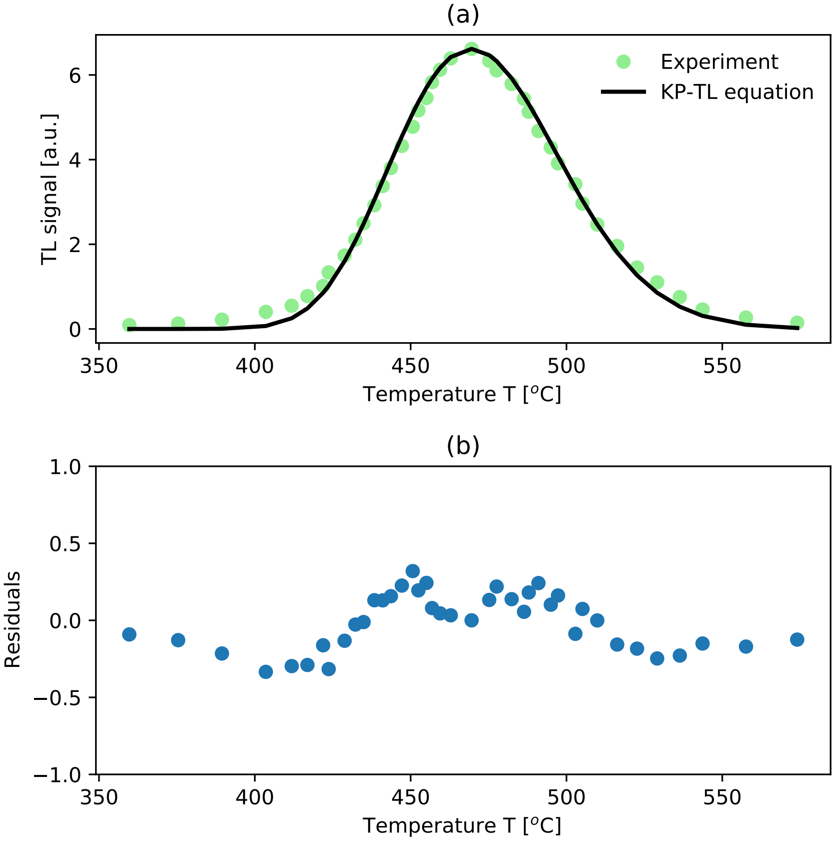


Fig. 2.5: Experimental TL glow curve from freshly irradiated MBO sample, fitted using a single peak with the transformed KP-TL analytical Eqs.(2.13)-(2.16). For additional modeling of experiments on this material, see Pagonis et al. [40].

Chapter 3
ITL SIGNALS: DATA ANALYSIS

3.1 Overview of Isothermal TL (ITL) experiments and models

Isothermal TL signals are usually shapeless decaying functions of time, and it is not always clear how one should analyze them. One of the issues is that we do not know the number of components in the signal, and which model we should use in the analysis. One must keep in mind always that even when an excellent fit is obtained to the experimental data by using a certain model, this does not guarantee in any way that the correct model is being used. Researchers face the exact same issues and unknown circumstances when attempting to analyze the shapeless CW-OSL signals from a dosimetric material.

The most common method of analyzing isothermal and/or CW-OSL signals is by assuming that the signal can be described as the sum of several exponential decay components. As a general rule, one should use as few components as possible to obtain a good fit. As a second step, one can analyze different parts of the signal, by gradually increasing the number of data points used for the fitting process. If the same parameters are obtained in a systematic manner during this type of analysis, then one can have more confidence in the fitting process and in the model.

Another issue that complicates the analysis of ITL/CW-OSL signals is the possible presence of retrapping effects, which can distort the shape of the first order exponential curves. In such cases, one can use the Lambert/OTOR equation, which describes retrapping effects by the retrapping ratio R.

Finally if the material shows anomalous fading effects, one should investigate whether quantum tunneling effects are influencing the shape of the signals. In such cases, one can attempt to use the TLT models discussed in the previous chapter.

For all of the above reasons, isothermal TL data should be analyzed carefully, and almost always in conjunction with other types of data (TL, OSL etc). The importance of combining the results of different methods of analysis can not be overemphasized.

3.2 Isothermal signal analysis within the delocalized TL model

When the thermal stimulation takes place at a constant temperature T_D, the stimulated luminescence signal is called *prompt isothermal decay* (PID), and the function $p(t)$ is given by:

$$p(t) = s\ \exp\left[\frac{-E}{k_B\, T_D}\right] \tag{3.1}$$

where E, s are the thermal kinetic parameters of the trap and k_B is the Boltzmann constant.

In the case of first order kinetics the differential equation is:

$$\frac{dn}{dt} = -n\, p(t) \tag{3.2}$$

whose solution is the simple decaying exponential:

$$n(t) = n_0 \exp\left\{-s\ \exp\left[\frac{-E}{k_B\, T_D}\right] t\right\} \tag{3.3}$$

and the intensity $I(t)$ of the isothermal signal is :

$$I(t) = -\frac{dn}{dt} = n_0 s\ \exp\left[\frac{-E}{k_B\, T_D}\right] \exp\left\{-s\ \exp\left[\frac{-E}{k_B\, T_D}\right] t\right\} \tag{3.4}$$

$$I(t) = -\frac{dn}{dt} = n_0\,(1/\tau)\exp\{-t/\tau\} \tag{3.5}$$

where

$$\tau\,(T_D) = s\ \exp\left[\frac{-E}{k_B\, T_D}\right] \tag{3.6}$$

Taking the natural logarithm of this equation:

Isothermal analysis using Arrhenius plots

$$\ln\tau = \ln s\ -\ \frac{E}{k_B\, T_D} \tag{3.7}$$

The most often used method of analyzing isothermal TL data is applied in two steps. First, one fits the ITL intensity with exponential decay functions, at various temperatures T_D. From these fits one obtains the decay constant $\tau\,(T_D)$.

Secondly, one creates an Arrhenius plot of the values of $\ln\tau\,(T_D)$ as a function of $1/\,(k_B\,T_D)$. According to the last equation, the slope of this best fit line is the activation energy $-E$, and the y-intercept is equal to $\ln s$, so one can also obtain the frequency factor s for the trap.

3.3 Analysis of isothermal TL, initial rise and CGCD of a peak in LiF: Mg,Ti

In this section and the next one, we present detailed analysis of single peaks in two important dosimetric materials, LiF: Mg,Ti (TLD-100) and BeO. Specifically we analyze and compare the results for three different methods: isothermal TL, initial rise and CGCD analysis carried out in the same sample.

In a recent study, Polymeris et al. [51] investigated in a systematic manner whether it is possible to distinguish experimentally between localized and delocalized transitions, for several well known dosimetric materials. Specifically they compared the values of the activation energy E obtained from PID and TL experiments, by using three different methods of analysis of TL data: the initial rise, isothermal decay and peak shape methods.

The material used in their work were commercially available LiF:Mg,Ti chips with dimensions 3mm, thickness 1mm. In addition, they used BeO square discs with dimensions of 4 mm and thickness of 1 mm. These dosimetric materials were selected for the stability and reproducibility of their thermally stimulated signals. Specifically, the TL and PID signals in these materials are found to be very stable and reproducible over many irradiation and readout cycles. This property allows the application of an experimental protocol in which a single aliquot of the sample is used. In addition, sensitivity tests were continuously applied during these experiments, to ensure the robustness of the results. The heating rate was selected at $1^0\,C/s$ and the maximum readout temperature 350^0C.

The experimental procedure used for both materials was as follows. After irradiation with a test dose of 0.5 Gy, the sample is preheated with a constant heating rate up to a temperature T_{pre}, in order remove thermally unstable low temperature TL peaks. After cooling down, the temperature is again raised with a constant heating rate up to a temperature of T_{DEC} where isothermal decay will take place, and the signal is recorded for 150 s. Finally the TL signal is recorder up to the maximum allowed temperature, in order to obtain the remnant-TL signal (RTL). This procedure is then repeated for a new increased isothermal decay temperature T_{DEC}.

During application of the experimental protocol and during analysis of the data, care must be taken to ensure that the thermal cleaning process does not produce any side effects to the main glow peak. The low temperature side of the TL signals is thermally cleaned during the experimental protocol.

Figure 3.1 shows a detailed analysis of isothermal decay signals in LiF: Mg,Ti (TLD-100). The isothermal TL signals were measured at temperatures of 180-210°C. The solid lines in (a) are exponential decay curves $y = A\exp(-\lambda t)$ fitted to the experimental data. Fig.3.1 shows the linear regression fit to the Arrhenius plot of $\ln\lambda$ vs $1/(k_BT)$, yielding the activation energy $E = (2.14 \pm 0.16)$ eV. For more details of this experiment, see Polymeris et al. [51].

Code 3.1: Isothermal analysis for LiF:Mg,Ti

```
# Isothermal analysis for LiF:Mg,Ti
import numpy as np
import matplotlib.pyplot as plt
from scipy import optimize
from prettytable import PrettyTable
from scipy import stats
def expon(x,A,tau):
    return A*np.exp(-x/tau)
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
files=('LiF180.txt','LiF190.txt','LiF210.txt','LiF220.txt')
slopes=[0 for x in range(4)]
plt.subplot(1,2, 1)
plt.title('(a')
plt.ylim(0,1.2);
plt.ylabel('Isothermal signal [a.u.]')
plt.xlabel(r'Time t [s]')
sym=(['o','s','^','x'])
lab=('180$^{o}$C','190$^{o}$C','200$^{o}$C','210$^{o}$C')
for i in range(4):
    dat = np.loadtxt(files[i])
    x_data,y_data = dat[:, 0],  dat[:, 1]
    plt.plot(x_data, y_data,sym[i],label=lab[i])
    params, cov=optimize.curve_fit(expon,x_data,y_data)
    print(params)
    slopes[i]=1/params[1]
    plt.plot(x_data, expon(x_data, *params))
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
temps=([180,190,200,210])
xArrh=[1/(8.617e-5*(x+273)) for x in temps]
```

```
yArrh=np.log(slopes)
slope, intercept,r_value,p_value, std_err=\
stats.linregress(xArrh,yArrh)
plt.subplot(1,2, 2)
plt.plot(xArrh,yArrh,'o')
fit=[x*slope+intercept for x in xArrh]
plt.plot(xArrh,fit,label='Fit')
plt.xlabel('1/(kT) (eV$^{-1}$)')
plt.ylabel(r'ln[$\lambda$(T)]')
plt.text(24.5,-3,'E=-slope')
plt.title('(b)');
plt.tight_layout()
slope, intercept, rsqr, p_value,std_err=\
np.round(slope,2),np.round(intercept,2),np.round(\
r_value**2.0,3),format(p_value,"10.2E"), np.round(std_err,2)
myTable=PrettyTable(['E (eV)','dE (eV)','p-value','R^2'])
myTable.add_row([-slope, std_err, p_value,rsqr]);
print(myTable)
```

```
  +--------+---------+------------+-------+
  | E (eV) | dE (eV) |  p-value   |  R^2  |
  +--------+---------+------------+-------+
  |  2.14  |   0.16  |   5.83E-03 | 0.988 |
  +--------+---------+------------+-------+
```

```
plt.show()
```

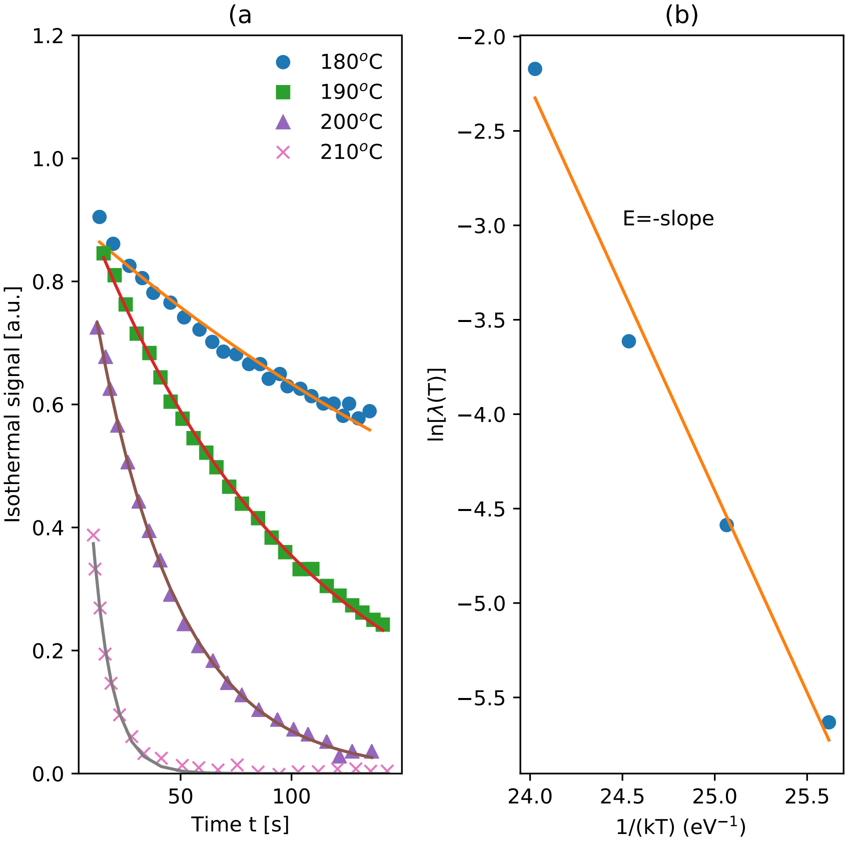


Fig. 3.1: (a) Isothermal TL decay curves for LiF: Mg,Ti (TLD-100) at temperatures of 180-210°C. The solid lines are exponential decay curves fitted to the experimental data. (b) A best line fit to the Arrhenius plot of $\ln\lambda$ vs $1/(k_BT)$ yields the activation energy $E = (2.14 \pm 0.16)$ eV. For more details see Kitis et al. [51].

The following code carries out the initial rise analysis for the same experiment with LiF:Mg,Ti, for TL glow curves obtained after isothermal decay at temperatures in the range 180-210°C. The E values are obtained from the best line fit to the Arrhenius plot of $\ln(TL)$ vs $1/(k_BT)$, and the average activation energy $E = (2.17 \pm 0.13)$ eV (Kitis et al. [51]).

Code 3.2: Initial rise analysis for LiF:Mg,Ti

```
# Initial rise analysis for LiF:Mg,Ti
from scipy import optimize
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable
import warnings
warnings.filterwarnings("ignore")
from scipy import stats
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
files=('LiFTL150degC.txt','LiFcurve3.txt','LiFcurve5.txt')
plt.subplot(1,2, 1)
plt.title('(a)')
plt.ylabel('TL signal [a.u.]')
plt.xlabel(r'Temperature [$^{o}$]')
sym=(['o-','s-','^-'])
sym2=(['o','s','^'])
lab=('150$^{o}$C','164$^{o}$C','178$^{o}$C')
for i in range(3):
    data = np.loadtxt(files[i])
    x_data,y_data = data[:, 0], data[:, 1]
    plt.plot(x_data,y_data,sym[i],label=lab[i],linewidth=3)
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.subplot(1,2, 2)
plt.title('(b)  Initial rise')
plt.ylabel('TL [a.u.]')
plt.xlabel(r'Temperature T [$^{o}$C]')
Evalues=[0 for x in range(len(files))]
Rvalues=[0 for x in range(len(files))]
dEvalues=[0 for x in range(len(files))]
lowpts=([3,3,3])
hipts=([9,9,9])
for i in range(3):
    data = np.loadtxt(files[i])
    x_data,y_data = data[:, 0][lowpts[i]:hipts[i]],\
data[:, 1][lowpts[i]:hipts[i]]
    x_data=[1/(8.617e-5*(273+u)) for u in np.array(x_data)]
    y_data=np.log(y_data)
    plt.plot(x_data,y_data,sym2[i])
    slope, intercept,r_value,p_value, std_err=\
stats.linregress(x_data,y_data)
    fit=[x*slope+intercept for x in x_data]
    plt.plot(x_data,fit,'-',label=lab[i])
    slope,std_err= np.round(slope,2), np.round(std_err,2)
    Evalues[i]=round(slope,3)
    dEvalues[i]=round(std_err,3)
avgE=round(np.mean(Evalues),3)
dE=round(np.std(Evalues),2)
plt.tight_layout()
myTable = PrettyTable([ "Curve","E (eV)","dE (eV)",\
"average E (eV)","stdE (eV)"])
myTable.add_row(["1",Evalues[0],dEvalues[0],avgE,dE]);
myTable.add_row(["2",Evalues[1],dEvalues[1],' ',' ']);
myTable.add_row(["3",Evalues[2],dEvalues[2],' ',' ']);
print(myTable)
```

```
  +-------+--------+---------+----------------+-----------+
  | Curve | E (eV) | dE (eV) | average E (eV) | stdE (eV) |
  +-------+--------+---------+----------------+-----------+
  |   1   | -2.34  |   0.05  |     -2.17      |    0.13   |
  |   2   | -2.03  |   0.1   |                |           |
  |   3   | -2.14  |   0.07  |                |           |
  +-------+--------+---------+----------------+-----------+
```

```
plt.show()
```

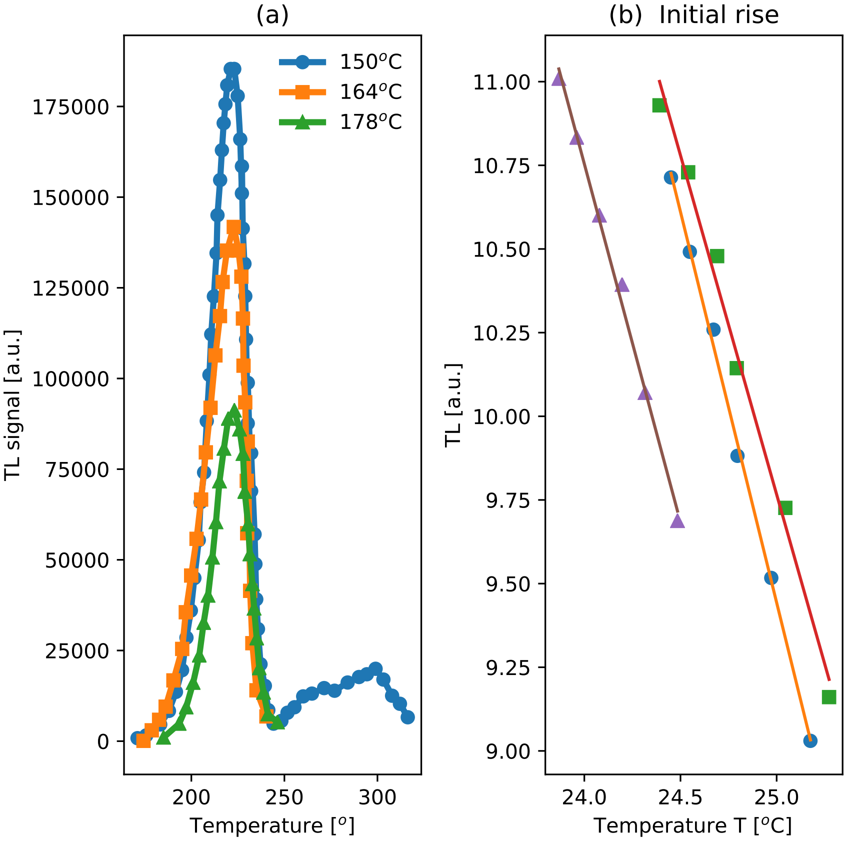


Fig. 3.2: Initial rise analysis of TL glow curves for LiF: Mg,Ti (TLD-100), obtained after isothermal decay at temperatures of 180-210°C. The average activation energy $E = (2.17 \pm 0.13)$ eV. For more details see Kitis et al. [51].

The following code applies CGCD analysis for the RTL glow curve obtained after isothermal decay at 150°C.

The activation energy $E = (2.31 \pm 0.02)$ eV is higher than the E values form IR and isothermal methods, however the estimated errors for the latter two methods are rather large. The three methods agree reasonably within the experimental accuracy of the data.

Code 3.3: CGCD analysis of single TL peak in LiF:Mg,Ti

```
# Deconvolution of LiF  TL peak with KV-TL equation
from scipy import optimize
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable
import warnings
warnings.filterwarnings("ignore")
from scipy.special import lambertw
import os
os.chdir('C:/Users/Bill Pagonis/Desktop/pythonVP')
data = np.loadtxt('LiFTL150degC.txt')
x_data,y_data = data[:, 0][0:45], data[:, 1][0:45]/max( data[:,\
1][0:45])
x_data=[273+u for u in np.array(x_data)]
x_data=np.array(x_data)
kB=8.617E-5
Imax=max(y_data)
def W_func(T,Tmax,R, E):
    F=kB*(T**2.0)*np.exp(-E/(kB*T))*(1-2*kB*T/E)/E
    Fm=kB*(Tmax**2.0)*np.exp(-E/(kB*Tmax))*(1-2*kB*Tmax/E)/E
    a=kB*Tmax**2.0*(1-1.05*R**1.26)
    Z=R/(1-R)-np.log((1-R)/R)+(F*E*np.exp(E/(kB*Tmax)))/a
    Zm=R/(1-R)-np.log((1-R)/R)+(Fm*E*np.exp(E/(kB*Tmax)))/a
    argW=np.real(lambertw(np.exp(Z)))
    argWm=np.real(lambertw(np.exp(Zm)))
    return Imax*np.exp(-E/(kB*T)*(Tmax-T)/Tmax)*\
        (argWm+argWm**2.0)/(argW+argW**2.0)
params,cov=optimize.curve_fit(W_func,x_data,y_data,\
p0=(490,1e-6,2.0))
plt.subplot(2,1, 1)
plt.plot(x_data, W_func(x_data, *params),'-',linewidth=4)
plt.scatter(x_data, y_data, label='Experiment')
plt.plot(x_data, W_func(x_data, *params),
c='r',linewidth=3, label='KV-TL equation')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('TL signal [a.u.]')
plt.xlabel(r'Temperature T [K]')
plt.text(460, .5,'LiF:Mg,Ti')
plt.text(460, .37,'Remnant TL')
plt.subplot(2,1, 2)
plt.plot(x_data,W_func(x_data, *params)-\
y_data,'o',c='r',linewidth=2,label='Residuals')
leg = plt.legend()
```

```
leg.get_frame().set_linewidth(0.0)
plt.ylabel('Residuals')
plt.xlabel(r'Temperature T [K]')
plt.ylim(-.2,.2);
plt.tight_layout()
Tmax=round(params[0],1)
R=format(params[1],"10.1E")
E=round(params[2],3)
dE = round(np.sqrt(cov[2][2]),3)
res=W_func(x_data, *params)-y_data
FOM=round(100*np.sum(abs(res))/np.sum(y_data),3)
myTable = PrettyTable([ "Tmax","R", "E(eV)","dE(eV)","FOM(%)"])
myTable.add_row([Tmax,R,E,dE,FOM]);
print(myTable)
```

```
  +-------+------------+-------+--------+--------+
  |  Tmax |     R      | E(eV) | dE(eV) | FOM(%) |
  +-------+------------+-------+--------+--------+
  | 494.7 |    3.8E-02 | 2.308 | 0.024  | 2.889  |
  +-------+------------+-------+--------+--------+
```

```
plt.show()
```

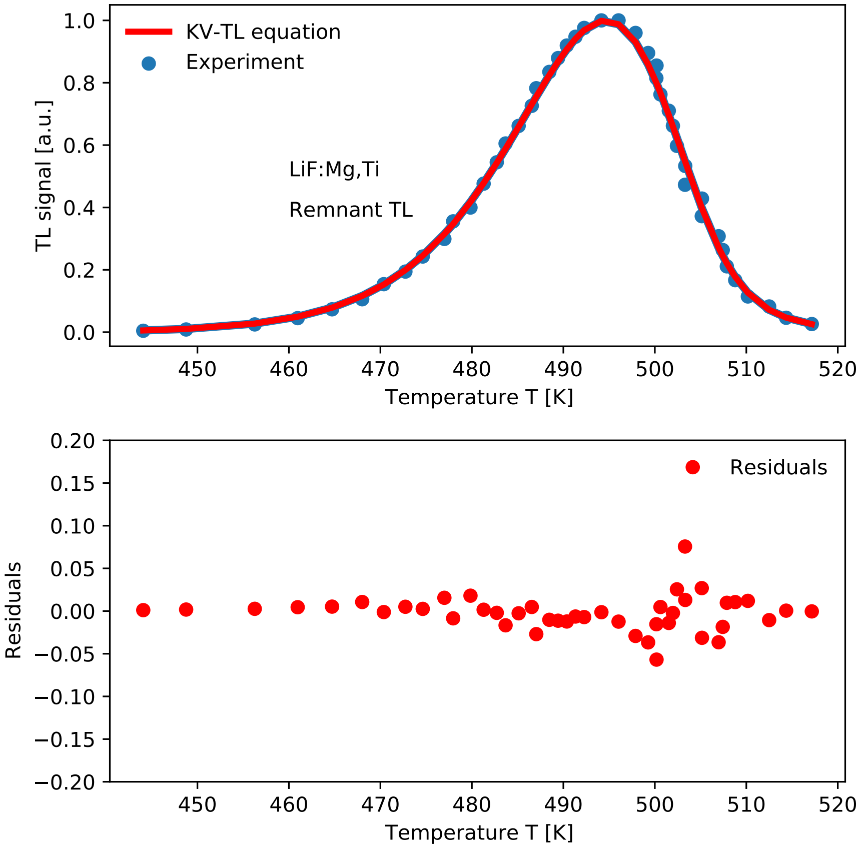


Fig. 3.3: CGCD analysis of TL glow curve for LiF: Mg,Ti (TLD-100). The solid lines is the transformed KV-TL fit to the experimental data. The analysis yields the activation energy $E = (2.31 \pm 0.02)$ eV. For more details see Kitis et al. [51].

In summary, in this section the transformed Lambert-OTOR was applied to the experimental data for TLD-100, and the analysis gave the activation energy $E = (2.31 \pm 0.02)$ eV. The initial rise method gave $E = (2.17 \pm 0.13)$ eV, and the isothermal TL method gave $E = (2.14 \pm 0.16)$ eV. These values are in reasonable agreement despite the CGCD analysis giving a higher value for E, and they within the overlapping ranges of the experimental errors.

Chapter 4
DOSE RESPONSE OF LUMINESCENCE SIGNALS: DATA ANALYSIS

4.1 Overview of ESR and OA experiments

In this section we discuss briefly optical absorption (OA) and electron spin resonance (ESR) experiments, and point out their relationship to TL and OSL data. For a more complete review of these types of experiments in connection with TL/OSL data, the reader is referred to the ESR review paper by Trompier et al. [60], and also to the extensive list of references in the book by Chen and Pagonis [13].

ESR and OA experiments are often measured simultaneously with TL and/or OSL with the same sample. These combined types of experiments produce important information about the underlying luminescence mechanisms, as well as about the nature of the trapping and luminescence centers (see for example the quartz study by Yang and McKeever [65]).

The absorption technique of electron spin (paramagnetic) resonance (ESR, EPR), is used for the study of materials which exhibit paramagnetism because of the magnetic moment of unpaired electrons. ESR spectra are usually presented as plots of the absorption or dispersion of the energy of an oscillating magnetic field of fixed radio frequency, versus the intensity of an applied static magnetic field. Among the wide variety of paramagnetic substances to which ESR spectroscopy has been applied, the one that is of interest in the present context is that of impurity centers in solids, mainly single crystals. The same impurity centers may be associated with TL or OSL, either acting as the charge-carrier traps, or as recombination centers. They may also be responsible for optical absorption.

Since ESR is normally capable of identifying the impurities in the crystal, in cases where the simultaneous TL-ESR measurements show a direct relation between the two phenomena, the identification by ESR may serve as a direct proof for the identity of the impurity involved in the luminescence process. In many cases the OA and/or the ESR signal show a decrease at a certain temperature range, where the trapping state becomes unstable. The instability may be associated with either the thermal release of charge carriers from the paramagnetic impurity, or, alternatively, the filling of paramagnetic impurities which serve as TL recombination centers. In general, it is expected that the TL peak will resemble the negative derivative of the ESR signal, with a close resemblance to OA (Chen and Pagonis [13]).

Figure 4.1 shows an example of the dose dependence of an ESR signal in quartz from Duval [19]. This author measured the dose response curves of the Al center for 15 sedimentary quartz samples from the Iberian Peninsula. The samples were irradiated up to a maximum dose of 23-40 kGy. It was found that the ESR signal grows almost linearly with the absorbed dose for doses above about 4 kGy.

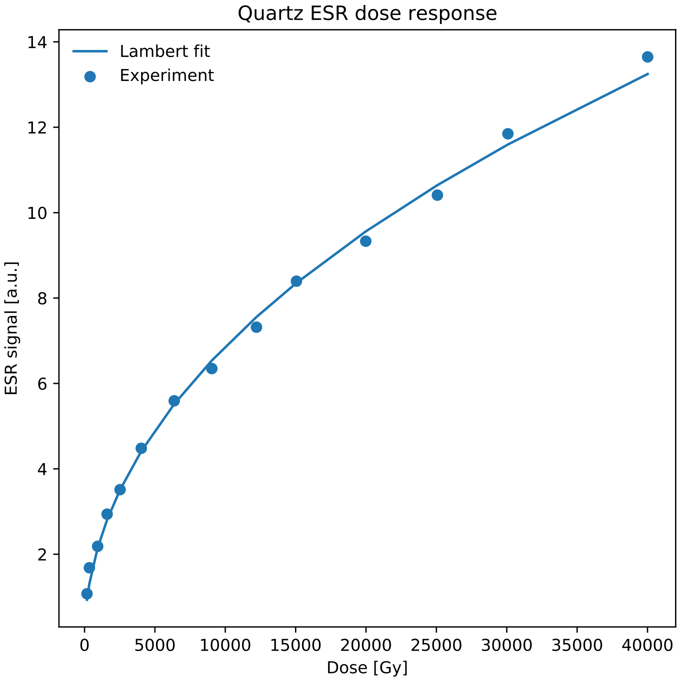


Fig. 4.1: Example of the dose response of ESR data from quartz. Redrawn from Pagonis et al. [43], original data from Duval [19].

4.2 The Pagonis-Kitis-Chen (PKC) equations

The shape of the simulated dose response $n(D)/N$ from Eq.(4.4) depends strongly on the retrapping ratio R, and looks similar to a saturating exponential function (SE). The SE is often used to fit experimental dose responses in a variety of materials, and for a variety of luminescence signals, together with two more general equations, the SEL and DSE functions written as (Berger and Chen [7]):

$$\frac{n(D)}{N} = 1 - \exp\left[-\frac{D}{D_0}\right] \qquad \text{SE} \tag{4.1}$$

$$\frac{n(D)}{N} = B_1\,D + B_2\left(1 - \exp\left[-\frac{D}{D_0}\right]\right) \qquad \text{SEL} \tag{4.2}$$

$$\frac{n(D)}{N} = B_3\left(1 - \exp\left[-\frac{D}{D_{02}}\right]\right) + B_4\left(1 - \exp\left[-\frac{D}{D_{01}}\right]\right) \qquad \text{DSE} \tag{4.3}$$

where D_0 is called the characteristic dose of the trap filling process. B_i ($i = 1 \ldots 4$) are constants, and D_{0i} are two constants characteristic of the sample with the dimensions of dose D. It is important to note that the SE, SEL and DSE are considered more or less empirical analytical equations, and the constants B_i are not usually assigned a direct physical meaning.

Pagonis et al. [43] developed the *exact* analytical solution $n(D)$ of this equation in terms of the Lambert W function:

PKC equation for dose response in the OTOR model when $n(0) = 0$

$$\frac{n(D)}{N} = 1 + \frac{1}{1-R} W\left[(R-1)\exp\Big(R - 1 - D/D_c\Big)\right] \tag{4.4}$$

where the constant D_c is defined as

$$D_c = N/R \tag{4.5}$$

We will refer to this equation as the *Pagonis-Kitis-Chen (PKC) equation.* Equation (4.4) is the new analytical expression for the irradiation stage of the OTOR model, which gives the trap filling ratio $n(D)/N$ as a function of the dose D, by using the Lambert W function. It shows that the function $n(D)/N$ depends only on two parameters, i.e. on the retrapping ratio R and on the constant $D_c = N/R$. The parameter D_c has the same units as the dose D, and depends on the physical properties R, N of the material.

From a physical point of view, the retrapping ratio parameter R can have *any* positive real value, including values $R > 1$. The values $R = 0$ and $R = 1$ correspond to first and second order kinetics. Furthermore, under certain physical assumptions, values of R between 0 and 1 correspond to the empirical general order intermediate kinetic orders (see for example the discussion in Kitis et al. [32]).

As may be expected from a physical point of view, the approach to saturation and the shape of the $n(D)$ function depends on the amount of retrapping, i.e. on the value of the ratio R.

If the traps are not initially empty (i.e. $n(0) = n_0 \neq 0$), the solution of Eq.(4.4) is (Pagonis et al. [43]):

PKC equation for dose response $n(D)$ ***in the OTOR model, when*** $n(0) \neq 0$

$$\frac{n(D)}{N} = 1 + \frac{1}{1-R} W\left[(R-1)\exp\Big(R - 1 - (D + D_{int})/D_c\Big)\right] \tag{4.6}$$

where the constant D_{int} has the same dimensions as the irradiation dose D, and is defined by:

$$D_{int} = D_c\left[n_0/N + \ln(1 - n_0/N)\right] \tag{4.7}$$

Equation (4.6) has the exact same mathematical form as Eq.(4.4), but is shifted along the horizontal dose D-axis by the amount D_{int} given by Eq.(4.7). In the next section, Eq.(4.4) is used to fit experimental data which start at the origin, and Eq.(4.6) is used to fit data which have a non-zero intercept on the y-axis.

4.3 Superlinear dose response equation (PKC-S)

Recently Pagonis et al. [44] inverted these parametric equations, and brought them in the useful form $n_1(D)$ and $n_2(D)$, by carrying out a series expansion in terms of the parameter A_2/A_1. These authors obtained the following Pagonis-Kitis-Chen (PKC-S) equation describing the non-linear dose response of a dosimetric trap:

PKC-S equation for nonlinear dose response

$$\frac{n}{N} = 1 - \left(\frac{1}{B} W\Big[B\exp(B)\exp(-D/D_c)\Big]\right)^{A_2/A_1} . \tag{4.8}$$

where the two constants B, D_c are:

$$B = \frac{N_1(A_1 - A_m)}{A_2 N_2 + A_m N_1}\ , \tag{4.9}$$

$$D_c = \frac{A_2 N_2 + A_m N_1}{A_1}\ . \tag{4.10}$$

The dose response $n(D)/N$ in this rather simple Eq.(4.8) depends on only three parameters, the constants A_2/A_1, B and D_c, which are given by Eqs.(4.9) and (4.10) in terms of the 5 parameters A_m, A_1, A_2, N_1,N_2 in the original model. The parameter D_c has the same dimensions at the irradiation dose D, so that the ratio D/D_c in Eq.(4.8) is dimensionless. As seen from the definition in Eq.(4.9), the parameter B is also dimensionless. The assumptions in deriving these analytical equations are the QE conditions, and the additional condition $A_2/A_1 < 1$, which allowed the series expansion for small values of A_2/A_1.

The overall dose response in this model will depend on the numerical values of the three parameters appearing in these equations: B, D_c, A_2/A_1. As the competitor trap approaches saturation, the dose response of the dosimetric trap n/N becomes superlinear. The initial short linear range in the curve n/N is followed by a range of superlinearity, which eventually becomes sublinear on its way to saturation.The dose response of TL, OSL, OA and ESR signals exhibit nonlinear regions, associated with the phenomena of superlinearity and sublinearity. These are discussed in detail, in connection with recent theoretical research involving the Lambert W function (Pagonis et al. [43]).

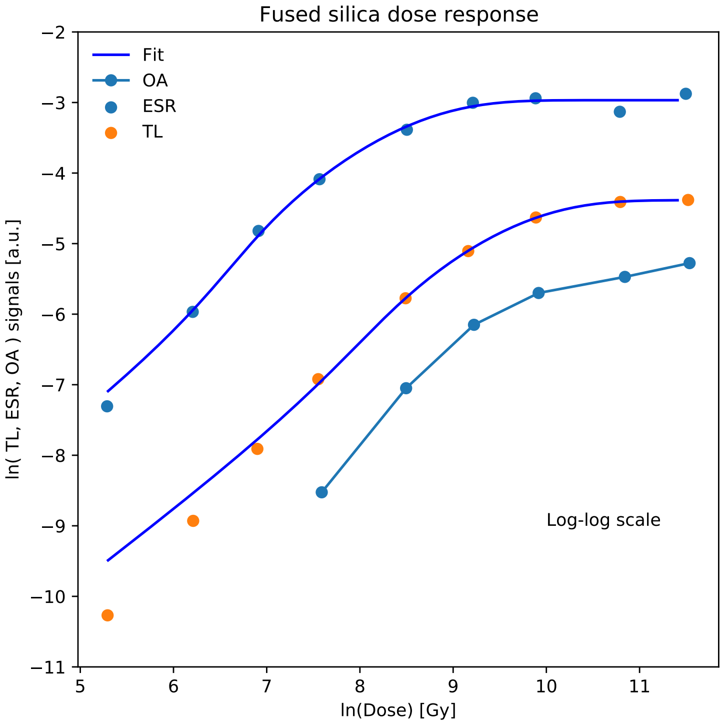


Fig. 4.2: Measurements of TL, ESR and OA signals as a function of the irradiation dose, from a single sample of fused silica. Note the log scale in both axes. For more details see Pagonis et al. [44], original data from Wieser et al. [64].

Figure 4.2 shows simultaneous measurements and possible correlations between the TL, ESR and OA signals from a sample of fused silica (Wieser et al. [64]). This type of multifaceted experiment can be very useful in identifying the source of the various luminescence signals, i.e. the nature of the traps and centers in the dosimetric material.

4.4 Fitting the dose response of luminescence signals using the SE function

The following code fits the dose response of experimental OSL data from Li et al. [36], for a quartz sample from Libya.

Code 4.1: Fit dose response data with saturating exponential

```
#Fit dose response data with Saturating Exponential
import numpy as np
import matplotlib.pyplot as plt
from scipy import optimize
from prettytable import PrettyTable
## fit to SE equation  ----
x_data =np.array([0,50.7117,100.534,152.135,204.626,272.242])
y_data = np.array([0,33.144,42.205,43.1055,44.4157,43.7098])
plt.plot(x_data,y_data,"o")
def lambertfit(x_data,N,Do):
    u=N*(1-np.exp(-x_data/np.abs(Do)))
    return u
init_vals=[20,20]
params, cov = optimize.curve_fit(lambertfit,\
x_data, y_data,p0=init_vals)
dN = round(np.sqrt(cov[0][0]),2)
dDo = round(np.sqrt(cov[1][1]),2)
x_vals=np.arange(0,300,1)
plt.plot(x_vals, lambertfit(x_vals, *params[0:2]),c="b",\
label='SE fit')
plt.scatter(x_data, y_data, label='Experiment')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylim(0,60);
plt.ylabel('OSL (L/T)')
plt.xlabel('Dose [Gy]')
plt.title('OSL dose response')
plt.text(200,20,'Libyan quartz')
plt.text(200,15,'Saturating exponential')
plt.tight_layout()
res=lambertfit(x_data, *params[0:2])-y_data
FOM=round(100*np.sum(abs(res))/np.sum(y_data),2)
myTable = PrettyTable(["N", "dN", "Do (Gy)","d(Do) (Gy)",\
"FOM (%)"])
myTable.add_row([round(params[0],2),dN,round(params[1],2),\
dDo,FOM])
print(myTable)

  +-------+------+---------+------------+---------+
  |   N   |  dN  | Do (Gy) | d(Do) (Gy) | FOM (%) |
  +-------+------+---------+------------+---------+
  | 44.16 | 0.32 |  35.83  |    1.4     |   1.09  |
  +-------+------+---------+------------+---------+

plt.show()
```

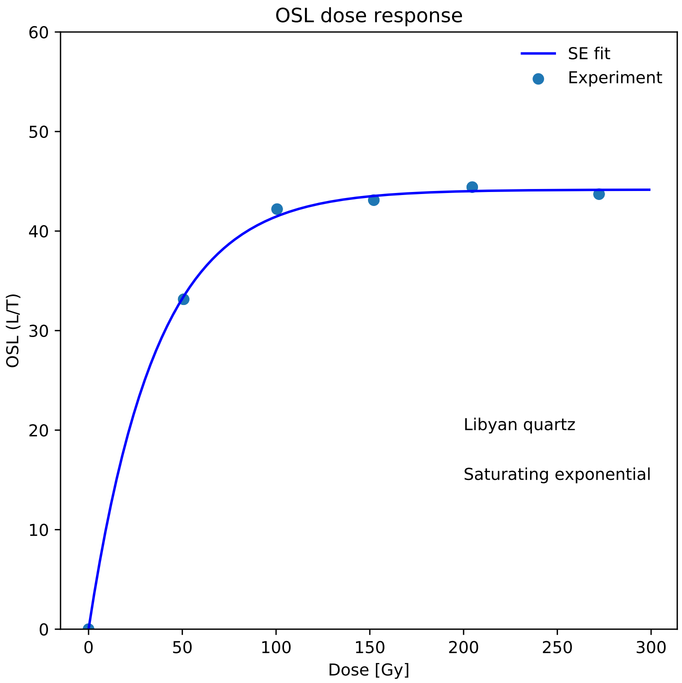


Fig. 4.3: Fitting dose response data with a saturating exponential. For more details see Pagonis et al. [43], original data from Li et al. [36].

4.5 Fitting the dose response using the Lambert equation

Pagonis et al. [43] analyzed the relationship between the Lambert solution Eq.(4.4) and the saturating exponential function. When $R \cong 1$, the Lambert dose response Eq.(4.4) becomes identical with the SE in Eq.(4.1), while for small or large values of R, (e.g. when $R = 0.001$ or $R = 3$), there is disagreement between the two expressions.

These authors chose experimental data from the literature, which could not be fitted with a SE function, and required instead the use of a SEL or a DSE fitting function. They found that Eq.(4.4) provides a satisfactory alternative to the empirical SEL and DSE regression models.

Berger and Chen [7] considered OSL signals, which were measured using the single-aliquot regenerative dose protocol (SAR) on fine grain sedimentary quartz. Figure 4.4a shows the OSL data from Berger and Chen [7], their Fig.1. A preheat of 240°C for 10 s was employed during the SAR protocol, and the beta dose rate was 0.12 Gy/s. The two lines fitted to the data are the Lambert Eq.(4.4), and the DSE function used by Berger and Chen [7]. The two lines are indistinguishable, however it is important to note that the Lambert fitting function contains only two parameters (R, D_c), while the DSE Eq.(4.3) requires 4 fitting parameters.

The following code fits a set of TL data from Berger [6], their Fig.1, using Eq.(4.4). This is a set of additive-dose data for purified volcanic glass, measured at the 321-330 °C temperature range of the glow curves, and preheated for 8 days to remove unstable TL. This type of additive-dose data often contains a non-zero y-intercept, therefore the fitting procedure introduces an extra fitting parameter. The data in Fig.4.4 was fitted with the Lambert Eq.(4.6), with the x-intercept represents the equivalent dose D_E for this sample.

Code 4.2: Fit of experimental TL dose response data using the PKC equation

```
from scipy.special import lambertw
import numpy as np
import matplotlib.pyplot as plt
from scipy import optimize
from prettytable import PrettyTable
## fit to Lambert equation  ----
x_data =np.array([0.00394056,0.00451523,0.39225,0.41203,0.703,\
0.741318,0.742221,1.49553,1.49758,1.5158,2.98473,3.00304,\
3.02282, 5.99852, 6.05755])
```

```
y_data = np.array([2.45103,2.80847,3.97964,4.28586,5.30465,\
5.10007,5.66177,6.21706,7.49364,6.82965,8.50226,7.88934,\
8.19555,11.0809,11.7953])
x_data=np.array(x_data)
y_data=np.array(y_data)
plt.plot(x_data,y_data,"o")
def lambertfit(x_data,N,R,Dc,f):
    u=np.real(N*(1+lambertw((np.abs(R)-1)*np.exp(np.abs(R)-1-\
    (x_data+np.abs(f))/np.abs(Dc)))/(1-np.abs(R))))
    u.astype(float)
    return u
init_vals=[100.0,.1,1e6,2]
params, params_covariance = optimize.curve_fit(lambertfit,\
x_data, y_data,p0=init_vals)
x_vals=np.arange(-1,6,.1)
plt.plot(x_vals, lambertfit(x_vals, *params[0:5]),c="b",\
label='Lambert fit')
plt.scatter(x_data, y_data, label='Experiment')
leg = plt.legend()
plt.ylim(0,13);
plt.xlim(-1,7);
leg.get_frame().set_linewidth(0.0)
plt.ylabel('OSL (L/T)')
plt.xlabel('Dose [Gy]')
plt.title('OSL dose response')
plt.text(3,4,'Volcanic glass')
plt.tight_layout()
res=lambertfit(x_data, *params[0:5])-y_data
FOM=round(100*np.sum(abs(res))/np.sum(y_data),2)
myTable = PrettyTable(["N", "R", "Dc (Gy)","f","FOM (%)"])
myTable.add_row([round(params[0],2),format(np.abs(params[1]),\
"10.2E"),format(np.abs(params[2]), "10.2E"),\
round(params[3],2),FOM])
print(myTable)
```

```
  +--------+------------+------------+------+---------+
  |   N    |     R      |  Dc (Gy)   |  f   | FOM (%) |
  +--------+------------+------------+------+---------+
  | 2001.6 |   2.45E-07 |   3.84E+05 | 0.45 |   5.8   |
  +--------+------------+------------+------+---------+
```

```
plt.show()
```

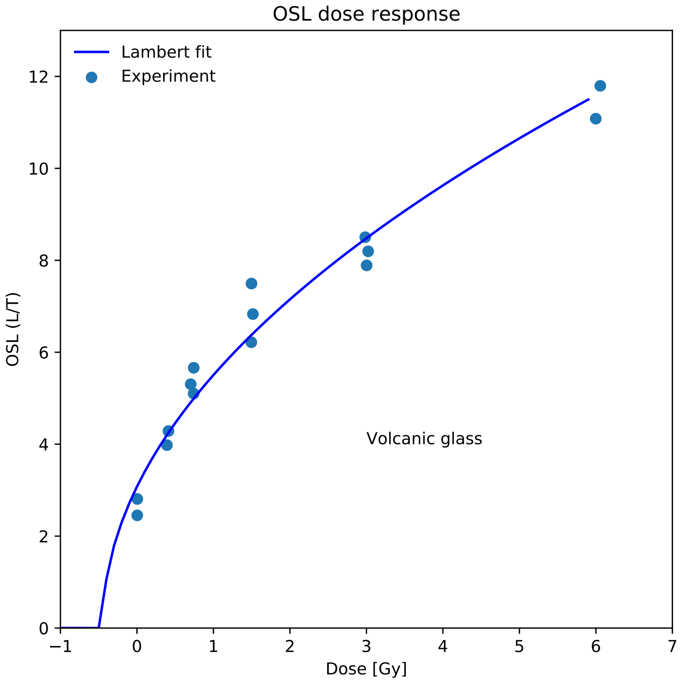


Fig. 4.4: Fit of experimental TL dose response data using the Lambert Eq.(4.6), with a non-zero intercept on the dose axis. For more details see Pagonis et al. [43], original data from Berger [6].

The following code analyzes ESR experimental data from Duval [19], by using the new dose response Eq.(4.4). These authors measured the dose response curves (DRCs) of the Al center from 15 sedimentary quartz samples from the Iberian Peninsula. The samples were irradiated in 11-14 dose steps up to a maximum dose of 23-40 kGy. It was found that the ESR signal grows almost linearly with the absorbed dose for doses above about 4 kGy. In this study it was concluded that *the ESR signal contains at least two components*, with the first component saturating at low doses and the second component showing no saturation even at these very high doses. The solid line in Fig.4.5 is the least squares fits using Eq.(4.4); the observed good fit to the experimental data suggests that *this ESR signal may actually originate in a single trap instead*, which can be described by the new dose response function based on the Lambert function.

```
from scipy.special import lambertw
import numpy as np
```

Code 4.3: Fit of experimental ESR dose response data using Lambert equation

```
import matplotlib.pyplot as plt
from scipy import optimize
from prettytable import PrettyTable
## fit to Lambert equation  ----
TLqzx=([174.13, 345.027, 931.847, 1603.74, 2524.39, 4031.12,
6372.1,9044.1,12217.5,15058.3,19981.5,25072.3,30082.3,40011.4])
TLqzy=([1.07478, 1.68389, 2.18591, 2.93875, 3.51271, 4.48122,
5.59377,6.3484,7.3184,8.39557,9.33133,10.4105,11.8478,13.6478])
x_data=np.array(TLqzx)
y_data=np.array(TLqzy)
def lambertfit(x_data,N,R,Dc):
    u=np.real(N*(1+lambertw((np.abs(R)-1)*np.exp(np.abs(R)-1-\
    x_data/np.abs(Dc)))/(1-np.abs(R))))
    u.astype(float)
    return u
init_vals=[100.0,.1,1e6]
params, cov = optimize.curve_fit(lambertfit,\
x_data, y_data,p0=init_vals)
dN = round(np.sqrt(cov[0][0]),2)
dR = round(np.sqrt(cov[1][1]),2)
dDc = round(np.sqrt(cov[2][2]),2)
plt.plot(x_data, lambertfit(x_data, *params[0:4]),\
label='Lambert fit')
plt.scatter(x_data, y_data, label='Experiment')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('ESR signal [a.u.]')
plt.xlabel('Dose [Gy]')
plt.title('Quartz ESR dose response')
plt.tight_layout()
res=lambertfit(x_data, *params[0:4])-y_data
FOM=round(100*np.sum(abs(res))/np.sum(y_data),1)
myTable=PrettyTable(["N","R","dR","Dc(Gy)",\
"d(Dc)(Gy)","FOM"])
myTable.add_row([round(params[0],1),format(np.abs(params[1]),\
"10.1E"),dR,format(np.abs(params[2]), "10.1E"),dDc,FOM])
print(myTable)
```

```
  +-------+------------+-----+------------+-----------+-----+
  |   N   |     R      |  dR |   Dc(Gy)   | d(Dc)(Gy) | FOM |
  +-------+------------+-----+------------+-----------+-----+
  | 151.1 |    2.5E-09 | inf |    9.5E+06 |    inf    | 2.8 |
  +-------+------------+-----+------------+-----------+-----+

plt.show()
```

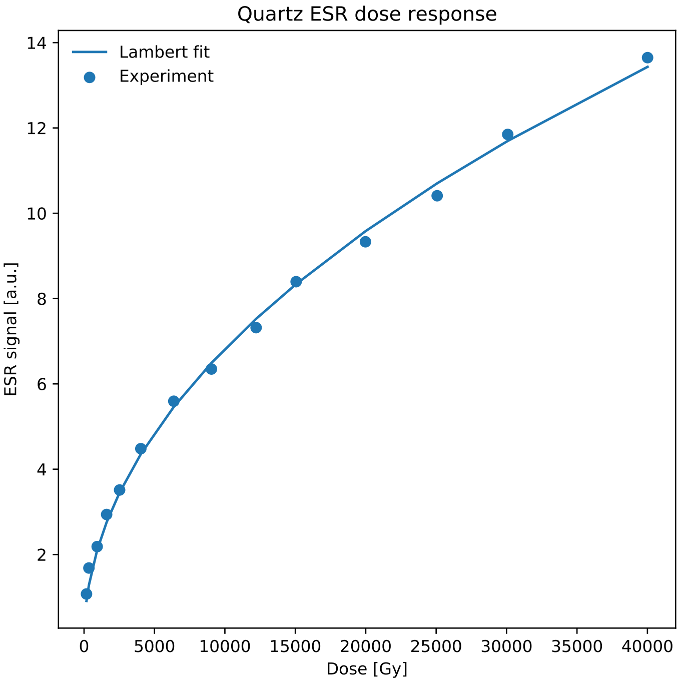


Fig. 4.5: Fit of experimental ESR dose response data using the Lambert Eq.(4.4). For more details see Pagonis et al. [43], original data from Duval [19].

Pagonis et al. [43] also presented sets of experimental ITL data by Vandenberghe et al. [61], which were fitted using the Lambert Eq.(4.4).

The following code analyzes SAR-OSL experimental dose response curves by Timar-Gabor et al. [59] (their Figure 3), for fine grain and coarse grain quartz samples in a loess-palaeosol sequence. The data shown here are for samples MV 10 and MV 8, and were fitted by these authors using a DSE. The solid lines in Fig.4.6 are the least squares fits using the new Eq.(4.4), showing that the Lambert function can be used to describe both types of dose responses in quartz with a smaller number of parameters than the DSE, even at the high doses involved in this experiment.

Code 4.4: Fit of experimental OSL dose response data using W(x)

```
## Fit of quartz OSL dose response data using W(x)
from scipy.special import lambertw
import numpy as np
import matplotlib.pyplot as plt
from scipy import optimize
from prettytable import PrettyTable
t = ([-34.2466, 34.2466, 68.4932, 273.973, 1027.4,1986.3,3013.7,
      5000, 7979.45, 10000])
y = ([1.04664, 0.000978474, 2.76386, 7.24592,12.6008,14.5329,
      15.8956,  17.1905, 17.847, 18.0952])
t2= ([0, 3.5583, 44.1822, 258.718, 1051.62, 2044.98, 3003.94,
      5024.61, 7046.32, 9992.29])
y2 = ([0, 0.93512, 2.61108, 4.99104, 6.36704, 6.42148,
      6.43643, 6.46792, 6.77215, 6.97391])
x_data=np.array(t)
y_data=np.array(y)
#plt.plot(x_data,y_data)
def lambertfit(x_data,N,R,Dc):
    u=np.real(N*(1+lambertw((np.abs(R)-1)*np.exp(np.abs(R)-1-\
    x_data/np.abs(Dc)))/(1-np.abs(R))))
    u.astype(float)
    return u
init_vals=[20.0,.1,1e4]
params, cov = optimize.curve_fit(lambertfit,\
x_data, y_data,p0=init_vals)
dN = round(np.sqrt(cov[0][0]),2)
dR = format(np.sqrt(cov[1][1]),"10.2E")
dDo = round(np.sqrt(cov[2][2]),2)
x_vals=np.arange(0,1e4,1)
plt.subplot(1,2, 1)
plt.plot(x_vals, lambertfit(x_vals, *params[0:4]),\
label='Lambert fit')
plt.scatter(x_data, y_data, label='Experiment')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('OSL (L/T)')
plt.xlabel('Dose [Gy]')
plt.text(3000,10,'Fine grain quartz')
plt.title('(a)')
plt.tight_layout()
myTable=PrettyTable(["N","dN","R","dR","Dc(Gy)","d(Dc)(Gy)"])
```

```
myTable.add_row([round(params[0],1),dN,format(np.abs(\
params[1]),"10.1E"),dR,np.int(np.abs(params[2])),np.int(dDo)])
x_data=np.array(t2)
y_data=np.array(y2)
init_vals=[20.0,.1,1e4]
params, cov = optimize.curve_fit(lambertfit,\
x_data, y_data,p0=init_vals)
dN = round(np.sqrt(cov[0][0]),2)
dR = format(np.sqrt(cov[1][1]),"10.2E")
dDo = round(np.sqrt(cov[2][2]),2)
x_vals=np.arange(0,1e4,1)
plt.subplot(1,2, 2)
plt.plot(x_vals, lambertfit(x_vals, *params[0:4]),\
label='Lambert fit')
plt.scatter(x_data, y_data, label='Experiment')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('OSL (L/T)')
plt.xlabel('Dose [Gy]')
plt.text(2500,4,'Coarse grain quartz')
plt.title('(b)')
plt.tight_layout()
myTable.add_row([round(params[0],1),dN,format(np.abs(\
params[1]),"10.1E"),dR,np.int(np.abs(params[2])),np.int(dDo)])
print(myTable)

  +------+------+------------+------------+--------+-----------+
  |  N   |  dN  |     R      |     dR     | Dc(Gy) | d(Dc)(Gy) |
  +------+------+------------+------------+--------+-----------+
  | 18.2 | 0.87 |   3.5E-02  |   4.78E-02 |  2467  |    689    |
  | 6.6  | 0.08 |   1.8E-08  |   4.56E-03 |  403   |     44    |
  +------+------+------------+------------+--------+-----------+

plt.show()
```

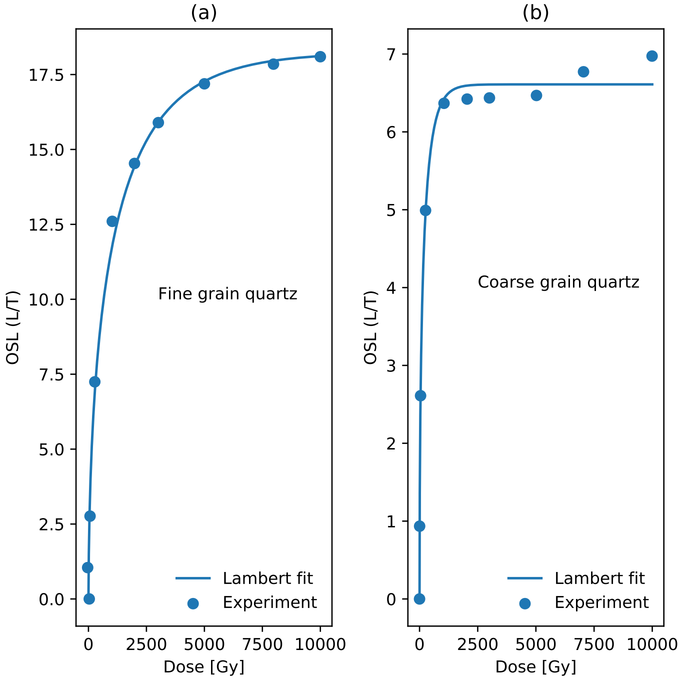


Fig. 4.6: Fit of experimental SAR-OSL experimental dose response data, for (a) fine grain and (b) coarse grain quartz samples, using the Lambert equation. For more details see Pagonis et al. [43], original data by Timar-Gabor et al. [59] (their Figure 3).

As an alternative to the Lambert function, we also recommend fitting a dose response curve using the R package *numOSL* by Peng et al. [48], using the Levenberg-Marquardt algorithm. When using function *fitGrowth*, five models are available: a linear model, a single saturation exponential model, a single saturation exponential plus linear model, a double saturation exponential model, and a general order kinetic model.

4.6 Fitting of superlinear experimental data using the Lambert equation

Figure 4.2 shows ESR, TL and optical absorption data for fused silica, by Wieser et al. [64]. The E_1' center is observed in all forms of quartz and silica, however these authors found an unexpected superlinear dose dependence of the E_1' center in fused silica. Previously a superlinear dose response in quartz had been observed with TL, but not with optical or ESR spectroscopy. They reported a superlinear increase of the E_1' center concentration by irradiation, and compared their results with the growth of the correlated optical E-band absorption, and with TL measurements which showed a correlation of the 415°C glow to the E_1' center. The solid lines in Fig.4.7 represent least squares fits using the analytical PKC Eq.(4.8). The Lambert-based equations in this chapter describe well all three types of dose responses for this material: ESR, TL and OA.

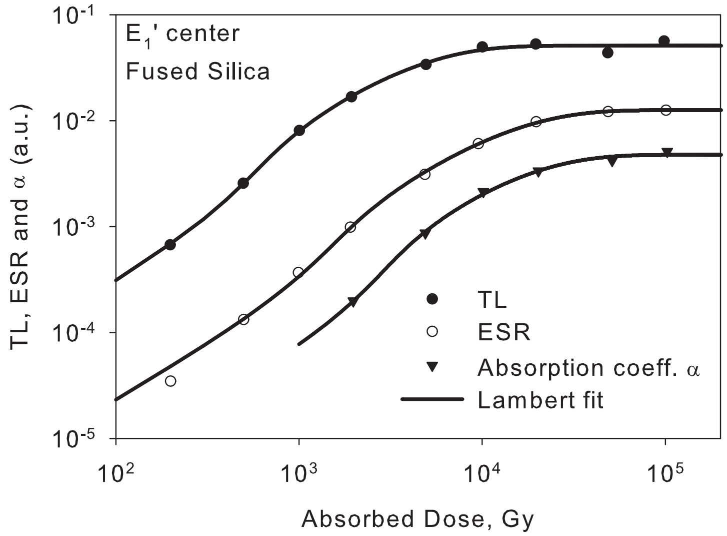


Fig. 4.7: Superlinear dose dependence of the E_1' center concentration (ESR), TL and OA signals, from a single sample of fused silica. Note the log scale in both axes. The solid lines are fitted using the PKC Eq.(4.8). For more details see Pagonis et al. [44], original data from Wieser [64].

As another example of fitted experimental data, Fig.4.8 shows curve 2 from Nikiforov et al. [38], their Fig.7. These authors measured the TL dose response characteristics of anion deficient aluminum oxide single crystals. Crystal 2 is a sample with low initial sensitivity, and the heating rate for these TL measurements was 6 K/s. Once more, the new analytical Eq.(4.8) describes well the variation of the superlinear dose response for the different samples of anion deficient aluminum oxide crystals.

For several additional examples of fitting TL,OSL, ESR data using the Lambert equations, see the papers by Pagonis et al. ([43, 44]).

Code 4.5: TL dose response of anion deficient aluminum oxide

```
# superlinear dose response  of anion deficient aluminum oxide
from scipy.special import lambertw
import numpy as np
import matplotlib.pyplot as plt
from scipy import optimize
from prettytable import PrettyTable
## fit to Lambert equation  ----
## fit to saturation exponential ----
t = ([0.0537568,0.103385,0.156481,0.211929,0.260776,
0.321015,0.36483,0.414625,0.478926,0.535569,0.589476,
0.638899,0.824344,0.951985,1.18991,1.63688,3.19332])
y = ([6694.89,15592.6,24767.6,39360.5,52176.2,78075.6,
101463,131855,189548,227223,272406,376197,469268,
634929,792121,1.16105e6,1.6992e6])
x_data=np.array(t)
y_data=np.array(y)
#plt.plot(x_data,y_data)
def lambertfit(x_data,N,B,Dc,beta):
    u=np.real(lambertw((np.abs(B))*np.exp(np.abs(B)-(x_data/\
    np.abs(Dc))))/(np.abs(B)))
    u=N*(1-(u**beta))
    u.astype(float)
    return u
init_vals=[2e6,10,.01,.01]
params, cov = optimize.curve_fit(lambertfit,\
x_data, y_data,p0=init_vals)
dN = int(np.sqrt(cov[0][0]))
dB = round(np.sqrt(cov[1][1]),2)
dDc = round(np.sqrt(cov[2][2]),3)
dbeta = round(np.sqrt(cov[2][2]),2)
x_vals=np.arange(0.03,4,.01)
plt.subplot(1,2, 1)
plt.plot(x_vals, lambertfit(x_vals, *params[0:5]),c="b",\
label='Lambert fit')
plt.title('(a)')
plt.scatter(x_data, y_data, label='Experiment')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('OSL (L/T)')
plt.xlabel('Dose [Gy]')
plt.text(1.8,.7e6,'Anion-deficient')
plt.text(1.8,.6e6,'Aluminum Oxide')
plt.text(1.8,.5e6,'Linear scale')
plt.subplot(1,2, 2)
plt.title('(b)')
plt.plot(np.log(x_vals), np.log(lambertfit(x_vals, \
*params[0:5])),c="b",label='Lambert fit')
plt.scatter(np.log(x_data), np.log(y_data), label='Experiment')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('OSL (L/T)')
plt.xlabel('Dose [Gy]')
plt.text(-1,10,'Log-log scale')
plt.title('(b)')
plt.tight_layout()
myTable = PrettyTable(["N", "B","dB", "Dc (Gy)",\
"d(Dc","beta","dbeta"])
myTable.add_row([format(params[0],"10.1E"),\
round(np.abs(params[1]),2),dB,round(np.abs(params[2])\
, 2),dDc,round(np.abs(params[3]),5),dbeta])
print(myTable)
```

```
  +------------+------+------+---------+-------+------+-------+
  |     N      |  B   |  dB  | Dc (Gy) |  d(Dc | beta | dbeta |
  +------------+------+------+---------+-------+------+-------+
  |    2.1E+06 | 6.87 | 1.44 |   0.05  | 0.012 | 0.03 |  0.01 |
  +------------+------+------+---------+-------+------+-------+
```

```
plt.show()
```

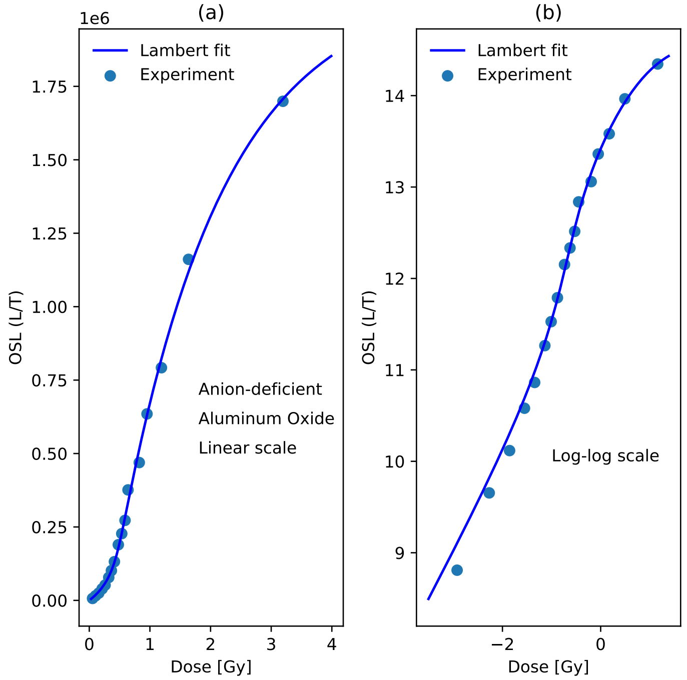


Fig. 4.8: TL dose response of an anion deficient aluminum oxide single crystal, which had low initial sensitivity to irradiation. The superlinear dose response is fitted with the PKC-S equation. For more details see Nikiforov et al. [38], see also the discussion in Pagonis et al. [44].

In the following code and Fig.4.9, we show how to fit experimental data for the supralinearity index $f(D)$, by using the analytical equation. The experimental data are taken from Edmund [20], who studied extensively the superlinear behavior of two Al_2O_3:C probes (probe A and B).

Code 4.6: Fit to Supralinearity index f(D) using the PKC-S equation

```
# Fit to PKC equation for Supralinearity index f(D)
from scipy.special import lambertw
import numpy as np
import matplotlib.pyplot as plt
from scipy import optimize
from prettytable import PrettyTable
```

```
## fit to PKC equation  ----
t = ([0.0811131, 0.171804,0.450923,0.857988,1.88341,
4.44069,8.44947,17.2683,40.7152,83.2104,176.246,
415.552,819.456,1674.74,3948.68,8983.32])
y = ([1.03326,0.983911,1.08074,1.10465,1.12844,1.28023,
1.37731,1.50481,1.76636,1.79021,1.45428, 0.813382,
0.428722,0.208669,0.0799713,0.0305689])
x_data=np.array(t)
y_data=np.array(y)
#plt.plot(x_data,y_data)
def lambertfit(x_data,N,B,Dc,beta):
    u=np.real(lambertw((np.abs(B))*np.exp(np.abs(B)-(x_data/\
np.abs(Dc))))/(np.abs(B)))
    u=N*(1-(u**beta))/x_data
    u.astype(float)
    return u
init_vals=[100,10,1,.01]
params, params_covariance = optimize.curve_fit(lambertfit,\
x_data, y_data,p0=init_vals)
x_vals=np.arange(0.1,1e4,.1)
plt.subplot(1,2, 1)
plt.plot(x_vals,lambertfit(x_vals,*params[0:5]),c="b",\
label='Lambert fit')
plt.scatter(x_data, y_data, label='Experiment')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('Supralinearity index f(D)')
plt.title('(a)')
plt.xlabel('Dose [Gy]')
plt.text(4000,.65,'Probe A')
plt.text(4000,.5,'Linear scale')
# ==========================================================
plt.subplot(1,2, 2)
plt.title('(b)')
plt.plot(np.log(x_vals),lambertfit(x_vals,*params[0:5]),\
c="b",label='Lambert fit')
plt.scatter(np.log(x_data), y_data, label='Experiment')
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.ylabel('Supralinearity index f(D)')
plt.xlabel('ln(Dose) [Gy]')
plt.text(-2,.5,'Log-Linear scale')
plt.title('(a)')
plt.tight_layout()
```

```
res=lambertfit(x_data,*params[0:5])-y_data
FOM=round(100*np.sum(abs(res))/np.sum(y_data),2)
myTable = PrettyTable(["N", "B", "Dc (Gy)","beta","FOM (%)"])
myTable.add_row([round(params[0],2),format(np.abs(params[1]),\
"10.2E"),format(np.abs(params[2]), "10.2E"),round(np.abs(\
params[3]),4),FOM])
print(myTable)

    +--------+------------+------------+--------+---------+
    |   N    |     B      |  Dc (Gy)   |  beta  | FOM (%) |
    +--------+------------+------------+--------+---------+
    | 374.39 |   1.24E+00 |   5.17E+00 | 0.0332 |   3.68  |
    +--------+------------+------------+--------+---------+

plt.show()
```

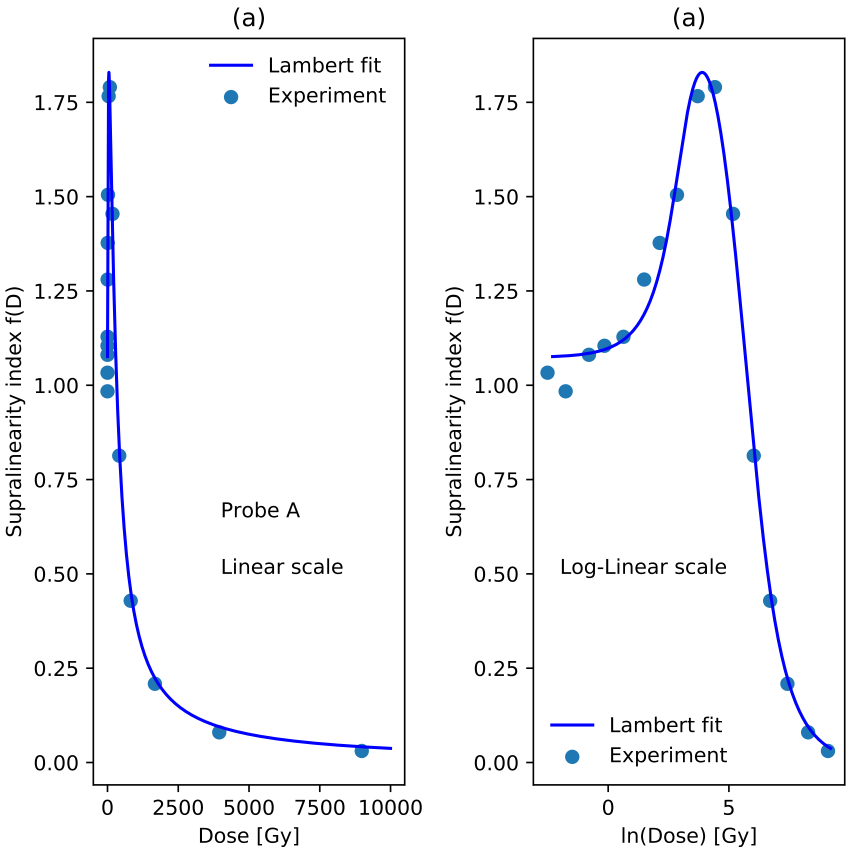


Fig. 4.9: Fit to the superlinear behavior of an Al_2O_3:C probe using the PKC superlinearity equations. For more details of the analysis see Pagonis et al. [44], original experimental data from Edmund [20].

4.7 On the importance of the W function in describing luminescence phenomena

During the past decade, the Lambert function has been used to describe analytical equations for a variety of luminescence phenomena. In this section we summarize these applications in phenomenological luminescence models (TL, OSL etc), as they were reviewed by Kitis et al. [32]. For a general discussion of the properties of the W function and its many uses in science, the reader is referred to the original papers by Corless et al. ([17, 18]).

Description of the readout stage in the OTOR model

The Lambert W function was first used by Kitis and Vlachos [34], to solve the OTOR model during the readout stage, in order to obtain an analytical expression for TL, OSL etc. Even though the OTOR model has been studied for almost 50 years, this was the first time that an analytical solution was obtained. Later Singh and Gartia [58] also solved the OTOR model using the equivalent Wright ω function. In all modern software these functions are built-in functions like the common transcendental functions of sine, cosine, logarithm etc.

Description of the readout stage in the OTOR model

In this chapter we saw the new analytical Lambert equations developed by Pagonis et al. [43], which described the dose response $n(D)$ in the OTOR model. The analytical equations is characterized by the retrapping ratio R and the dose parameter D_c in the model. For values of $R \cong 1$ the new analytical equation can be approximated very well by a SE function, while for $R \leq 0.1$ or $R > 1$ the two functions diverge significantly.

The results of this chapter showed that the new dose response function in Eq.(4.4) is a more general function than the SE, SEL and DSE equations. In addition, the proposed fitting function contains a smaller number of parameters than the SEL and DSE, and is based on physically meaningful parameters. Importantly, the new equation provides a simpler interpretation of the shape of the dose response curve, than the SEL and DSE. This is because by using the Lambert solution, the data is interpreted as the dose response of a *single trap*, instead of the two traps implied in the DSE and SEL functions.

Description of superlinear dose response in the model by Chen/Bowman

In this chapter we saw that Pagonis et al. [44] also obtained the PKC analytical solution of the dose response $n(D)$, within the model by Bowman and Chen [11].

If we consider only the very simple OTOR model, a linear-saturation behavior is always expected. This case has been studied in detail in Pagonis et al. [43], who found the analytical equation for the trap filling ratio n/N:

$$\frac{n(D)}{N} = 1 - \frac{1}{R-1}\, W\left[(R-1)\exp\Big(R - 1 - D/D_c\Big)\right]. \tag{4.11}$$

In this OTOR model, the function $n(D)/N$ depends only on two parameters, i.e. on the retrapping ratio R and on the dose scaling constant D_c. The parameter D_c has the same units as the dose D, and depends on the physical properties of the material. By using the definition of the supralinearity index $f(D)$, we obtain:

$$f_{\mathrm{OTOR}}(D) = \frac{1}{k_{\mathrm{OTOR}} D}\left\{1 - \frac{1}{R-1} W\left[(R-1)\exp\Big(R - 1 - D/D_c\Big)\right]\right\} \tag{4.12}$$

where k_{OTOR} is a constant with dimensions of $1/D$ (usually Gy^{-1}), and represents the slope in the initial linear part of the dose response curve.

Eq.(4.12) is the analytical equation for the supralinearity index $f_{\mathrm{OTOR}}(D)$ in the OTOR model. In this model the index depends on only *two* parameters, the retrapping ratio R and the normalized characteristic dose D_c.

In real materials several trapping states and recombination centers exist, which take part in the irrasdiation and readout stages. The effects of competition can explain a wealth of different dose responses associated with TL, OSL, ITL, OA and ESR signals. As far as superlinear dose dependence is concerned, it has been shown that competition during excitation can yield one kind of behavior, while competition during read-out yields another kind of dependence. Simulations have also shown the combined effect of both kinds of competition (see for example Chen and Pagonis [13]).

In the 2T1C model, the superlinear dose response can be described by the analytical PKC equation (Pagonis et al. [44]):

$$\frac{n\,(D)}{N} = 1 - \left(\frac{1}{B} W\left[B\exp\left(B\right)\exp\left(-D/D_c\right)\right]\right)^{A_2/A_1}. \tag{4.13}$$

In this 2T1C model, the function $n(D)/N$ depends on *three* parameters, i.e. on the dimensionless constant B, on the ratio of trapping coefficients A_2/A_1 of the two competing traps, and on the dose scaling constant D_c. The parameter D_c has the same units as the dose D, and the other two parameters B, A_2/A_1 are dimensionless.

The analytical PKC-S Eq.(4.8) is characterized by the dimensionless constant B, by the ratio of trapping coefficients A_2/A_1 of the two competing traps, and by the dose scaling constant D_c. The parameter D_c has the same units as the dose D, and the other two parameters B, A_2/A_1 are dimensionless.

Description of localized transitions in the LT model

Kitis and Pagonis [31] obtained also the analytical solution of the LT model, in terms of the Lambert W function.

Model Name	**Fitting Parameters**
KV-TL equation Eq.(1.4)	R , E, s, n_0 Retrapping ratio R, Energy E, frequency factor s, initial trap concentration n_0
Transformed KV-TL Eqs.(1.14)-(1.16)	R, E (optional parameters T_m, I_m) (Temperature T_m at maximum TL intensity I_m)
KV-CW equation	R, σ, n_0 Retrapping ratio R, optical cross section σ, finitial trap concentration
KV-LM equation	R, σ, n_0 Retrapping ratio R, optical cross section σ, finitial trap concentration
Transformed KV-LM equation	R, σ (optional parameters t_m, I_m) Retrapping ratio R, optical cross section σ, (time t_m at maximum TL intensity I_m)
Dose response PKC equation	R, D_c
Superlinear dose response PKC-S equation	B, D_c, β (=A_2/A_1)

Table 4.1: Table of deconvolution functions based on the Lambert function W.

References

1. G Adamiec, A Bluszcz, R Bailey, and M Garcia-Talavera. Finding model parameters: Genetic algorithms and the numerical modelling of quartz luminescence. *Radiation Measurements*, 41(7-8):897–902, 2006.
2. G Adamiec, M Garcia-Talavera, R M Bailey, and P I de La Torre. Application of a genetic algorithm to finding parameter values for numerical simulation of quartz luminescence. *Geochronometria*, 23:9–14, 2004.
3. D Afouxenidis, G S Polymeris, N C Tsirliganis, and G Kitis. Computerised curve deconvolution of TL/OSL curves using a popular spreadsheet program. *Radiation Protection Dosimetry*, 149:363–370, May 2012.
4. H G Balian and N W Eddy. Figure-of-merit (FOM), an improved criterion over the normalized chi-squared test for assessing goodness-of-fit of gamma-ray spectral peaks. *Nuclear Instruments and Methods*, 145:389–395, 1977.
5. R. Battiti and G. Tecchiolli. Simulated annealing and Tabu search in the long run: A comparison on QAP tasks. *Computers & Mathematics with Applications*, 28(6):1 – 8, 1994.
6. G W Berger. Regression and error analysis for a saturating-exponential-plus-linear model. *Ancient TL*, 8(3):23–25, 1990.
7. G W Berger and R Chen. Error analysis and modelling of double saturating exponential dose response curves from SAR OSL dating. *Ancient TL*, 29(1):9–14, 2011.
8. A J J Bos. Thermoluminescence as a research tool to investigate luminescence mechanisms. *Materials (Basel, Switzerland)*, 10, November 2017.
9. A J J Bos, T M Piters, J M Gómez-Ros, and A Delgado. An Intercomparison of Glow Curve Analysis Computer Programs: I. Synthetic Glow Curves. *Radiation protection dosimetry*, 47:473–477, 1993.
10. A.J.J. Bos, T.M. Piters, J.M. Gómez Ros, and A. Delgado. An Intercomparison of Glow Curve Analysis Computer Programs: II. Measured Glow Curves. *Radiation Protection Dosimetry*, 51(4):257–264, 02 1994.
11. S G E Bowman and R Chen. Superlinear filling of traps in crystals due to competition during irradiation. *Journal of Luminescence*, 18-19:345 – 348, 1979.
12. R Chen and S W S McKeever. *Theory of thermoluminescence and related phenomena*. World Scientific, Singapore, 1997.
13. R Chen and V Pagonis. *Thermally and Optically Stimulated Luminescence: A Simulation Approach*. John Wiley & Sons, Chichester, 2011.
14. K.S. Chung, H.S. Choe, J.I. Lee, and J.L. Kim. An algorithm for the deconvolution of the optically stimulated luminescence glow curves involving the mutual interactions among the electron traps. *Radiation Measurements*, 46(12):1598 – 1601, 2011.

15. K.S. Chung, J.I. Lee, and J.L. Kim. A computer program for the deconvolution of the thermoluminescence glow curves by employing the interactive trap model. *Radiation Measurements*, 47(9):766 – 769, 2012.
16. K.S. Chung, C.Y. Park, J.I. Lee, and J.L. Kim. Thermoluminescence glow curve deconvolution of LiF:Mg,Cu,Si with more realistic kinetic models. *Radiation Measurements*, 59:151 – 154, 2013.
17. R M Corless, G H Gonnet, D G E Hare, D J Jerey, and D E Knuth. On the Lambert W function. *Advances in Computational Mathematics*, 5:329–359, 1996.
18. R M Corless, D J Jerey, and D E Knuth. A sequence series for the Lambert W function. *In Proceedings of the International Symposium on Symbolic and Algebraic Computation , ISSAC*, pages 133–140, 1997.
19. M Duval. Dose response curve of the ESR signal of the aluminum center in quartz grains extracted from sediment. *Ancient TL*, 30(2):1–9, 2012.
20. J M Edmund. *Effects of temperature and ionization density in medical luminescence dosimetry using Al2O3:C (PhD Thesis, Riso, Denmark)*. PhD thesis, Risø National Laboratory, 2007. Riso-PhD-38(EN).
21. J M Gómez-Ros and G Kitis. Computerized glow-curve deconvolution using mixed and general order kinetics. *Radiation Protection Dosimetry*, 101:47–52, 2002.
22. Y.S. Horowitz and D. Yossian. Computerised Glow Curve Deconvolution: Application to Thermoluminescence Dosimetry. *Radiation Protection Dosimetry*, 60(1):3–3, 06 1995.
23. M Jain, B Guralnik, and M T Andersen. Stimulated luminescence emission from localized recombination in randomly distributed defects. *Journal of Physics: Condensed Matter*, 24(38):385402, 2012.
24. G. Kitis. Confirmation of the influence of thermal quenching on the initial rise method in alpha Al_2O_3:C. *physica status solidi (a)*, 191(2):621–627, 2002.
25. G. Kitis, E. Carinou, and P. Askounis. Glow-curve de-convolution analysis of TL glow-curve from constant temperature hot gas TLD readers. *Radiation Measurements*, 47:258 – 265, 2012.
26. G Kitis and J M Gómez-Ros. Glow curve deconvolution functions for mixed order kinetics and a continuous trap distribution. *Nucl. Instrum. Methods A 440*, 440:224–231, 1999.
27. G Kitis, J M Gómez-Ros, and J W N Tuyn. Thermoluminescence glow curve deconvolution functions for first, second and general order kinetics. *J. Phys. D: Appl. Phys.*, 31:2666–2646, 1998.
28. G Kitis and V Pagonis. Computerized curve deconvolution analysis for LM-OSL. *Radiation Measurements*, 43:737 – 741, 2008.
29. G Kitis and V Pagonis. Analytical solutions for stimulated luminescence emission from tunneling recombination in random distributions of defects. *Journal of Luminescence*, 137:109–115, 2013.
30. G Kitis and V Pagonis. Properties of thermoluminescence glow curves from tunneling recombination processes in random distributions of defects. *Journal of Luminescence*, 153:118–124, 2014.
31. G Kitis and V Pagonis. Localized transition models in luminescence: A reappraisal. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 432:13 – 19, 2018.
32. G Kitis, G S Polymeris, and V Pagonis. Stimulated luminescence emission: From phenomenological models to master analytical equations. *Applied Radiation and Isotopes*, 153:108797, 2019.
33. G Kitis, G S Polymeris, I K Sfampa, M Prokic, N Meriç, and V Pagonis. Prompt isothermal decay of thermoluminescence in Mg_4BO_7: Dy, Na and Li_4BO_7:Cu,In dosimeters. *Radiation Measurements*, 84:15–25, 2016.
34. G Kitis and N D Vlachos. General semi-analytical expressions for TL, OSL and other luminescence stimulation modes derived from the OTOR model using the Lambert W-function. *Radiation Measurements*, 48:47 – 54, 2013.

35. M Lamothe, M Auclair, C Hamzaoui, and S Huot. Towards a prediction of long-term anomalous fading of feldspar IRSL. *Radiation Measurements*, 37(4):493–498, 2003.
36. B Li, Z Jacobs, and R G Roberts. Investigation of the applicability of standardised growth curves for OSL dating of quartz from Haua Fteah cave, Libya. *Quaternary Geochronology*, 35:1 – 15, 2016.
37. B Li and S H Li. Investigations of the dose-dependent anomalous fading rate of feldspar from sediments. *Journal of Physics D: Applied Physics*, 41(22):225502, 2008.
38. S V Nikiforov, V S Kortov, and M G Kazantseva. Simulation of the superlinearity of dose characteristics of thermoluminescence of anion-defective aluminum oxide. *Physics of the Solid State*, 56(3):554–560, March 2014.
39. V. Pagonis. *Luminescence: Data Analysis and Modeling Using R*. Use R! Springer International Publishing, 2021.
40. V Pagonis, N Brown, G S Polymeris, and G Kitis. Comprehensive analysis of thermoluminescence signals in Mg_4BO_7: Dy, Na dosimeter. *Journal of Luminescence*, 213:334 – 342, 2019.
41. 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.
42. V Pagonis and G Kitis. Mathematical aspects of ground state tunneling models in luminescence materials. *Journal of Luminescence*, 168:137–144, 2015.
43. V Pagonis, G Kitis, and R Chen. A new analytical equation for the dose response of dosimetric materials, based on the Lambert W function. *Journal of Luminescence*, 225:117333, 2020.
44. V Pagonis, G Kitis, and R Chen. Superlinearity revisited: A new analytical equation for the dose response of defects in solids, using the Lambert W function. *Journal of Luminescence*, 227:117553, 2020.
45. V Pagonis, S M Mian, and G Kitis. Fit of first order thermoluminescence glow peaks using the weibull distribution function. *Radiation protection dosimetry*, 93:11–17, 2001.
46. V Pagonis, P Morthekai, Singhvi A K, Thomas J, Balaram V, Kitis G, and Chen R. Time-resolved infrared stimulated luminescence signals in feldspars: Analysis based on exponential and stretched exponential functions. *Journal of Luminescence*, 132(9):2330–2340, 2012.
47. Vasilis Pagonis, George Kitis, and George S. Polymeris. Quantum tunneling processes in feldspars: Using thermoluminescence signals in thermochronometry. *Radiation Measurements*, 134:106325, 2020.
48. J Peng, Z Dong, and F Han. tgcd: An R package for analyzing thermoluminescence glow curves. *SoftwareX*, 5:112–120, 2016.
49. E. B. Podgorsak, P. R. Moran, and J. R. Cameron. Interpretation of resolved glow curve shapes in LiF(TLD-100) from 100 to 500K. *pp 1-8 of Proceedings of the Third International Conference on Luminescence Dosimetry. /Mejdahl, V. (ed.). Risoe, Denmark Danish Atomic Energy Commission (1971).*, 1971.
50. G S Polymeris, V Pagonis, and G Kitis. Thermoluminescence glow curves in preheated feldspar samples: An interpretation based on random defect distributions. *Radiation Measurements*, 97:20 – 27, 2017.
51. G S Polymeris, V Pagonis, and G Kitis. Investigation of thermoluminescence processes during linear and isothermal heating of dosimetric materials. *Journal of Luminescence*, 222:117142, 2020.
52. G S Polymeris, N Tsirliganis, Z Loukou, and G Kitis. A comparative study of the anomalous fading effects of TL and OSL signals of Durango apatite. *Physica Status Solidi (a)*, 203(3):578–590, 2006.
53. M. Puchalska and P. Bilski. Glowfit-a new tool for thermoluminescence glow-curve deconvolution. *Radiation Measurements*, 41(6):659 – 664, 2006.

54. A M Sadek, H M Eissa, A M Basha, E Carinou, P Askounis, and G Kitis. The deconvolution of thermoluminescence glow-curves using general expressions derived from the one trap-one recombination (OTOR) level model. *Applied radiation and isotopes : including data, instrumentation and methods for use in agriculture, industry and medicine*, 95:214–221, January 2015.

55. E Şahiner, G Kitis, V Pagonis, N Meriç, and G S Polymeris. Tunnelling recombination in conventional, post-infrared and post-infrared multi-elevated temperature IRSL signals in microcline K-feldspar. *Journal of Luminescence*, 188:514–523, 2017.

56. I K Sfampa, G S Polymeris, V Pagonis, E Theodosoglou, N.C. Tsirliganis, and G. Kitis. Correlation of basic TL, OSL and IRSL properties of ten K-feldspar samples of various origins. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 359:89 – 98, 2015.

57. I K Sfampa, G S Polymeris, N Tsirliganis, V Pagonis, and G Kitis. Prompt isothermal decay of thermoluminescence in an apatite exhibiting strong anomalous fading. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 320:57–63, 2014.

58. L L Singh and R K Gartia. Theoretical derivation of a simplified form of the OTOR/GOT differential equation. *Radiation Measurements*, 59:160 – 164, 2013.

59. A Timar-Gabor, A Vasiliniuc, D A G Vandenberghe, C Cosma, and A G Wintle. Investigations into the reliability of SAR-OSL equivalent doses obtained for quartz samples displaying dose response curves with more than one component. *Radiation Measurements*, 47(9):740 – 745, 2012.

60. F Trompier, C Bassinet, S Della Monaca, A Romanyukha, R Reyes, and I Clairand. Overview of physical and biophysical techniques for accident dosimetry. *Radiation Protection Dosimetry*, 144:571–574, March 2011.

61. D A G Vandenberghe, M Jain, and A S Murray. Equivalent dose determination using a quartz isothermal TL signal. *Radiation Measurements*, 44(5):439 – 444, 2009.

62. R Visocekas. Tunnelling in afterglow: its coexistence and interweaving with thermally stimulated luminescence. *Radiation Protection Dosimetry*, 100:45–53, 2002.

63. R Visocekas, V Tale, A Zink, and I Tale. Trap spectroscopy and tunnelling luminescence in feldspars. *Radiation Measurements*, 29:427–434, 1998.

64. A Wieser, Y Göksu, D F Regulla, and A Waibel. Unexpected superlinear dose dependence of the E1' centre in fused silica. *International Journal of Radiation Applications and Instrumentation. Part D. Nuclear Tracks and Radiation Measurements*, 18(1):175 – 178, 1991.

65. X H Yang and S W S McKeever. The pre-dose effect in crystalline quartz. *Journal of Physics D: Applied Physics*, 23(2):237, 1990.

Index

Author Index