Package: tgcd (via r-universe)

November 2, 2026

Type Package

Title Thermoluminescence Glow Curve Deconvolution

Version 2.7

Date 2023-09-08

Author Jun Peng [aut, cre], John Burkardt [ctb], Jorge More [ctb], Burton Garbow [ctb], Kenneth Hillstrom [ctb], Linda R. Petzold [ctb], Alan C. Hindmarsh [ctb], R. Woodrow Setzer [ctb], Andrew Horchler [ctb], William Cody [ctb], Cleve Moler [ctb], Jack Dongarra [ctb]

Maintainer Jun Peng <pengjun10@mails.ucas.ac.cn>

Description Deconvolving thermoluminescence glow curves according to various kinetic models (first-order, second-order, general-order, and mixed-order) using a modified Levenberg-Marquardt algorithm (More, 1978) <DOI:10.1007/BFb0067700>. It provides the possibility of setting constraints or fixing any of parameters. It offers an interactive way to initialize parameters by clicking with a mouse on a plot at positions where peak maxima should be located. The optimal estimate is obtained by ``trial-and-error''. It also provides routines for simulating first-order, second-order, and general-order glow peaks.

License GPL-2 | GPL-3

Depends R (>= 3.0.1)

Imports graphics, stats, utils

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

NeedsCompilation yes

Repository https://cran.r-universe.dev

Date/Publication 2023-09-08 15:31:06 UTC

RemoteUrl https://github.com/cran/tgcd

RemoteRef HEAD

RemoteSha 0f1b01ddd8c4049290d2b2d517dce97656f39937

Contents

tgcd-package 2
Kitis 3
Refglow 3
savgol 4
simPeak 5
simqOTOR 7
tgcd 9

Index **20**

`tgcd-package` *tgcd: An R package for analyzing thermoluminescence glow curves*

Description

Package for thermoluminescence glow curve deconvolution (tgcd) and glow peak simulation.

Details

Package:	tgcd
Type:	Package
Version:	2.7
Date:	2023-09-08
License:	GPL-2 | GPL-3

Author(s)

Jun Peng Hunan University of Science and Technology, Xiangtan, China

Package maintainer
Jun Peng <pengjun10@mails.ucas.ac.cn>

Related package projects
R program KMS https://github.com/pengjunUCAS/KMS
R package numOSL https://CRAN.R-project.org/package=numOSL

References

Peng J, Dong ZB, Han FQ, 2016. tgcd: An R package for analyzing thermoluminescence glow curves. SoftwareX, 5: 112-120.

Peng J, Kitis G, Sadek AM, Karsu Asal EC, Li, ZG, 2021. Thermoluminescence glow-curve deconvolution using analytical expressions: A unified presentation. Applied Radiation and Isotopes, 168: 109440.

Kitis *Thermoluminescence glow curves provided by George Kitis*

Description

A total of 22 thermoluminescence glow curves measured from various materials provided by George Kitis.

Usage

```
data(Kitis)
```

Format

A list that contains 22 thermoluminescence glow curves.

Details

This object contains 22 thermoluminescence glow curves (`x001` to `x022`) provided by George Kitis. x001 (Al2O3:C), x002 (CaF2:Dy), x003 (LBO), x004 (Background), x005 (MgO), x006 (BeO), x007 (CaF2:Tm), x008 (Salt), x009 (CaF2:Dy), x010 to x016 (quartz irradiated with dose of 1, 2, 4, 8, 16, 32, and 64 Gy), x017 and x018 (BeO), x019 and x020 (Salt), x021 and x022 (MgO).

Examples

```
# Load package "tgcd".
 require(tgcd)

 data(Kitis)
 names(Kitis)
```

Refglow *Reference glow curves*

Description

Synthetic and measured thermoluminescence glow curves from the GLOCANIN project.

Usage

```
data(Refglow)
```

Format

A list that contains 10 thermoluminescence glow curves.

Details

This object contains 10 thermoluminescence glow curves (`x001` to `x010`) from the GLOCANIN project (Bos et al., 1993, 1994).

References

Bos AJJ, Piters TM, Gomez Ros JM, Delgado A, 1993. An intercomparison of glow curve analysis computer programs: I. Synthetic glow curves. Radiation Protection Dosimetry, 47(1-4), 473-477.

Bos AJJ, Piters TM, Gomez Ros JM, Delgado A, 1994. An intercomparison of glow curve analysis computer programs: II. Measured glow curves. Radiation Protection Dosimetry, 51(4): 257-264.

Examples

```
# Load package "tgcd".
 require(tgcd)

 data(Refglow)
 names(Refglow)
```

---

savgol *Apply a Savitzky-Golay algorithm to smooth thermoluminescence glow curves*

Description

Smooth thermoluminescence glow curves with a Savitzky-Golay smoothing filter and calculate derivatives.

Usage

```
savgol(y, drv, hwd = 3 * (drv + 2), pod = 4)
```

Arguments

`y`	numeric(**required**): the data to be filtered
`drv`	integer(**required**): the order of the derivative to be calculated
`hwd`	integer(with default): half width of the segement used for filter
`pod`	integer(with default): order of the polynomial used for filter

Details

The Savitzky-Golay smoothing algorithm is particularly good at preserving lineshape while removing high frequency squiggles (Press et al., 1986). The procedure can be used to calculate derivatives of thermoluminescence data to identify the location of glow peaks.

Value

The filtered signal, which has the same length as y.

References

Press WH, Teukolsky SA, Vetterling WT, Flannery BP, 1986. Numberic recipes in Fortran 77, the Art of Scientific Computing, second edition.

See Also

tgcd

Examples

```
library(tgcd)
data(Refglow)

x <- Refglow$x009[,1]
y <- Refglow$x009[,2]
y0 <- savgol(y, drv=0)
dy <- savgol(y, drv=1)

plot(x, y, type="p", pch=21, bg="black")
points(x, y0, type="l", col="blue", lwd=2)

plot(x, dy, type="l", col="blue", lwd=2)
abline(h=0, lty="dashed", col="red", lwd=2)
```

---

simPeak *Thermoluminescence glow peak simulation*

Description

Simulating first-order, second-order, or general-order glow peaks.

Usage

```
simPeak(temps, n0, Nn = NULL, bv = NULL, ff,
        ae, hr, typ = c("f", "s", "g"),
        outfile = NULL, plot = TRUE)
```

Arguments

temps	vector(**required**): temperature values (K) where the values of the thermoluminescence intensity will be computed. It needs to be sorted increasingly. A vector of temperature values may be generated using the internal function seq
n0	numeric(**required**): initial concentration of trapped electrons (1/cm^3)

Nn	numeric(**required**): total concentration of the traps in the crystal (1/cm^3)
bv	numeric(**required**): order number for the general order glow peak
ff	numeric(**required**): the frequency factor (1/s)
ae	numeric(**required**): the activation energy (eV)
hr	numeric(with default): the linear heating rate (K/s)
typ	character(with default): the type of a glow peak, typ="f" means first-order, typ="s" means second-order, typ="g" means general-order, default typ="f"
outfile	character(optional): if specified, simulated intensities of glow peaks will be written to a file named "outfile" in CSV format and saved to the current work directory
plot	logical(with default): draw a plot according to the simulated glow peak or not

Details

Function simPeak simulates glow peaks of various orders. The first-, second-, and general-order glow peak can be simulated using the following three ordinary equations, respectively (Pagonis et al., 2006):

$\frac{d_n}{d_T} = \frac{-nSexp(-\frac{E}{kT})}{\beta}$

$\frac{d_n}{d_T} = \frac{-n^2Sexp(-\frac{E}{kT})}{N_n\beta}$

$\frac{d_n}{d_T} = \frac{-n^bSexp(-\frac{E}{kT})}{N_n\beta}$

where n is the concentration of trapped electrons, $\frac{d_n}{d_T}$ the rate of change of the concentration of trapped electrons, S the frequency factor, E the activation energy, T the absolute temperature, k the Boltzmann constant, N_n the total concentration of the traps in the crystal, b the b value (kinetic order), and β the linear heating rate.

The ordinary equation is solved by the Fortran 77 subroutine *lsoda* (original version written by Linda R. Petzold and Alan C. Hindmarsh available at Netlib: https://www.netlib.org/odepack/, modified version by R. Woodrow Setzer from the R package *deSolve* (Soetaert et al., 2010) available at CRAN: https://CRAN.R-project.org/package=deSolve).

Value

Return an invisible list containing the following elements:

temps	a vector of temperature values
tl	values of the thermoluminescence intensity
n	variation of concentration of trapped electrons with temperature
sp	parameters used for describing the shape of a glow peak (Pagonis et al., 2006): the temperature corresponding to half intensity on the left side of the peak (T1); the temperature corresponding to half intensity on the right side of the peak (T2); the temperature corresponding to maximum intensity (Tm); the half-width at the left side of the peak (d1=Tm-T1); the half-width at the right side of the peak (d2=T2-Tm); the total half-width (thw=d1+d2); the symmetry factor (sf=d2/thw)

References

Pagonis V, Kitis G, Furetta C, 2006. Numerical and practical exercises in thermoluminescence. Springer Science & Business Media.

Soetaert K, Petzoldt T, Setzer RW, 2010. Solving Differential Equations in R: Package deSolve. Journal of Statistical Software, 33(9): 1-25.

See Also

tgcd; simqOTOR

Examples

```
# Simulate second-order glow peaks with various
# initial electron trap concentration (n0).
 temps <- seq(400, 600, by=0.5)
 peak1 <- simPeak(temps, n0=0.2e10, Nn=1e10,
   ff=1e19, ae=2.0, hr=1, typ="s")
 peak2 <- simPeak(temps, n0=0.4e10, Nn=1e10,
   ff=1e19, ae=2.0, hr=1, typ="s")
 peak3 <- simPeak(temps, n0=0.6e10, Nn=1e10,
   ff=1e19, ae=2.0, hr=1, typ="s")
 peak4 <- simPeak(temps, n0=0.8e10, Nn=1e10,
   ff=1e19, ae=2.0, hr=1, typ="s")
 peak5 <- simPeak(temps, n0=1.0e10, Nn=1e10,
   ff=1e19, ae=2.0, hr=1, typ="s")
 peaks <- cbind(peak1$tl, peak2$tl, peak3$tl, peak4$tl, peak5$tl)
 matplot(temps, peaks, type="l", lwd=2, lty="solid",
   xlab="Temperature (K)", ylab="TL intensity (counts)")
```

`simqOTOR` *Thermoluminescence glow peak simulation*

Description

Simulating glow peaks according to the one trap-one recombination center (OTOR) model using the quasi-equilibrium approximation.

Usage

```
simqOTOR(temps, n0, Nn, Ah, An, ff, ae,
         hr, outfile = NULL, plot = TRUE)
```

Arguments

`temps`	vector(**required**): temperature values (K) where the values of the thermoluminescence intensity will be computed, it needs to be sorted increasingly
`n0`	numeric(**required**): initial concentration of trapped electrons (1/cm^3)
`Nn`	numeric(**required**): total concentration of the traps in the crystal (1/cm^3)
`Ah`	numeric(optional): probability coefficient of electron recombining with holes in the recombination center (cm^3/s)
`An`	numeric(optional): probability coefficient of electron retrapping in the traps (cm^3/s)
`ff`	numeric(**required**): the frequency factor (1/s)
`ae`	numeric(**required**): the activation energy (eV)
`hr`	numeric(with default): the linear heating rate (K/s)
`outfile`	character(optional): if specified, simulated intensities of glow peaks will be written to a file named `"outfile"` in `CSV` format and saved to the current work directory
`plot`	logical(with default): draw a plot according to the simulated glow peak or not

Details

Function simqOTOR simulates a synthetic glow peak according to the OTOR model using the quasi-equilibrium approximation. This function may be used to simulating glow peaks of first-, second-, and general-order, depending on the given kinetic parameters. The approximate equation of the OTOR model derived using the quasi-equilibrium approximation can be described by (Pagonis et al., 2006):

$$\frac{d_n}{d_T} = \frac{-A_h n^2 Sexp(-\frac{E}{kT})}{[nA_h+(N_n-n)A_n]\beta}$$

where n is the concentration of trapped electrons, $\frac{d_n}{d_T}$ the rate of change of the concentration of trapped electrons, S the frequency factor, E the activation energy, T the absolute temperature, k the Boltzmann constant, N_n the total concentration of the traps in the crystal, A_h the probability coefficient of electron recombining with holes in the recombination center, A_n the probability coefficient of electron retrapping in the traps, and β the linear heating rate.

The ordinary equation is solved by the Fortran 77 subroutine *lsoda* (original version written by Linda R. Petzold and Alan C. Hindmarsh available at Netlib: `https://www.netlib.org/odepack/`, modified version by R. Woodrow Setzer from the R package *deSolve* (Soetaert et al., 2010) available at CRAN: `https://CRAN.R-project.org/package=deSolve`).

Value

Return an invisible list containing the following elements:

`temps`	a vector of temperature values
`tl`	values of the thermoluminescence intensity
`n`	variation of concentration of trapped electrons with temperature
`sp`	parameters used for describing the shape of a glow peak, see function simPeak for details

References

Pagonis V, Kitis G, Furetta C, 2006. Numerical and practical exercises in thermoluminescence. Springer Science & Business Media.

Soetaert K, Petzoldt T, Setzer RW, 2010. Solving Differential Equations in R: Package deSolve. Journal of Statistical Software, 33(9): 1-25.

See Also

tgcd; simPeak

Examples

```
# Synthesizing a glow curve consisting of five glow peaks.
 temps <- seq(330, 730, by=0.5)
 peak1 <- simqOTOR(temps, n0=0.7e10, Nn=1e10, Ah=1e-3, An=1e-7,
   ff=1e14, ae=1.5, hr=1, outfile = NULL, plot = TRUE)
 peak2 <- simqOTOR(temps, n0=0.5e10, Nn=1e10, Ah=1e-7, An=1e-7,
   ff=1e17, ae=1.9, hr=1, outfile = NULL, plot = TRUE)
 peak3 <- simqOTOR(temps, n0=0.2e10, Nn=1e10, Ah=1e-5, An=1e-7,
   ff=1e15, ae=1.45, hr=1, outfile = NULL, plot = TRUE)
 peak4 <- simqOTOR(temps, n0=0.2e10, Nn=1e10, Ah=1e-5, An=1e-7,
   ff=1e9, ae=0.85, hr=1, outfile = NULL, plot = TRUE)
 peak5 <- simqOTOR(temps, n0=0.3e10, Nn=1e10, Ah=1e-7, An=1e-7,
   ff=1e11, ae=1.4, hr=1, outfile = NULL, plot = TRUE)
 peaks <- cbind(peak1$tl, peak2$tl, peak3$tl, peak4$tl, peak5$tl,
   peak1$tl+peak2$tl+peak3$tl+peak4$tl+peak5$tl)
 matplot(temps, y=peaks, type="l", lwd=2, lty="solid",
   xlab="Temperature (K)", ylab="TL intensity (counts)")
```

---

tgcd *Thermoluminescence glow curve deconvolution (tgcd)*

Description

Thermoluminescence glow curve deconvolution according to various first-order, second-order, general-order, and mixed-order empirical expressions.

Usage

```
tgcd(Sigdata, npeak, model = "g1", subBG = FALSE, pickp = "d2",
     pickb = "d0", nstart = 60, kkf = 0.03, mdt = NULL, mwt = NULL,
     mr = NULL, edit.inis = TRUE, inisPAR = NULL, inisBG = NULL,
     hr = NULL, hwd = NULL, pod = NULL, plot = TRUE, outfile = NULL)
```

Arguments

`Sigdata`	matrix(**required**): a 2-column matrix, temperature values (in unit K) and thermoluminescence signal values are stored in the first and second column, respectively
`npeak`	integer(**required**): number of glow peaks, the allowed maximum number of glow peaks is set equal to 13
`model`	character(with default): model used for glow curve deconvolution, `"f1"`, `"f2"`, and `"f3"` for first-order models, `"s1"`, `"s2"` for second-order models, `"g1"`, `"g2"`, `"g3"` for general-order models, `"wo"` and `"lw"` for the Wright Omega and the Lambert W functions, `"m1"`, `"m2"`, and `"m3"` for the mixed-order models (see **Details**)
`subBG`	logical(with default): whether the user want to subtract the background during the deconvolution
`pickp`	character(with default): mode used for initialization of kinetic parameters if `inisPAR=NULL`, `"d0"` and `"d01"` prompt the user to click with a mouse on the original and log-scale glow curves respectively to locate each glow peak, `"d1"`, `"d2"`, `"d3"`, and `"d4"` prompt the user to click with a mouse on the first-, second-, third-, and fourth-derivative of the glow curve respectively to locate each glow peak
`pickb`	character(with default): mode used for initialization of background parameters if `inisBG=NULL`, `"d0"` and `"d01"` prompt the user to click with a mouse on the original and log-scale glow curves respectively to initilize the background parameters
`nstart`	integer(with default): number of trials performed in a "trial-and-error" protocol, the upper limit is set equal to 10000
`kkf`	numeric(with default): factor controlling the range of values from which random starting parameters will be generated during the "trial-and-error" protocol, `0<kkf<1`. For example, if `kkf=0.03` then kinetic parameters will be generated uniformly between `(1.0-kkf)*inisPAR` and `(1.0+kkf)*inisPAR` and background parameters will be generated uniformly between `(1.0-kkf)*inisBG` and `(1.0+kkf)*inisBG`
`mdt`	numeric(optional): allowed minimum distance between `Tm` values of glow peaks. A larger `mdt` prevents the appearance of strongly overlapping peaks
`mwt`	numeric(optional): allowed maximum total half-width of deconvoluted glow peaks. A smaller `mwt` prevents the appearance of glow peaks with large total half-width
`mr`	numeric(optional): allowed minimum resolution of glow peaks. A larger `mr` prevents the appearance of strongly overlapping peaks
`edit.inis`	logical(with default): whether the user want to further modify, constrain, or fix the initialized kinetic (and/or background) parameters through an automatically generated **Dialog Table**
`inisPAR`	matrix(optional): a matrix (`3` or `4` columns) used for storing initial kenetic parameters `Im`, `E`, `Tm`, `b` (or `R`, `a`) (see **Examples**)
`inisBG`	vector(optional): a 3-element vector containing initial background parameters `A`, `B`, and `C` used for background subtraction (see **Examples**)

hr	numeric(optional): linear heating rate used for calculating the frequency factor
hwd	integer(with default): half width (length) of the segement used for Savitzky-Golay smoothing
pod	integer(with default): order of the polynomial used for Savitzky-Golay smoothing
plot	logical(with default): draw a plot according to the fitting result or not
outfile	character(optional): if specified, fitted signal values for each glow peak will be written to a file named "outfile" in CSV format and saved to the current work directory

Details

Function tgcd is used for deconvolving thermoluminescence glow curves according to various kinetic models. In the text below, $I(T)$ is the thermoluminescence intensity as function of temperature T, E the activation energyin ev, k the Boltzmann constant in eV/k, T the temperature in K with constant heating rate K/s, T_m the temperature at maximum thermoluminescence intensity in K, I_m the maximum intensity, b is an extra parameter (the kinetic order) in application of a general-order model, R is an extra parameter in application of the Lambert W (Wright Omega) function, and α is an extra parameter in application of the mixed-order models.

First-order glow peaks appear if the recombination probability (A_m) is greater than that of re-trapping (A_n) during excitation. The three parameters describing a glow peak are: I_m, E, and T_m. Three empirical expressions describing first-order glow peaks are available in function tgcd:
<1>The first type of first-order empirical expression (model="f1") is (Bos et al., 1993a)

$$I(T) = I_m exp(\frac{E}{kT_m} - \frac{E}{kT}) exp[\frac{E}{kT_m}\alpha(\frac{E}{kT_m}) - (\frac{T}{T_m})\alpha(\frac{E}{kT}) exp(\frac{E}{kT_m} - \frac{E}{kT})]$$

where $\alpha(x)$ is a quotient of 4th-order polynomials of the form

$$\alpha(x) = \frac{a_0+a_1x+a_2x^2+a_3x^3+x^4}{b_0+b_1x+b_2x^2+b_3x^3+x^4}$$

$a_0 = 0.267773734$, $a_1 = 8.6347608925$, $a_2 = 8.059016973$, $a_3 = 8.5733287401$,
$b_0 = 3.9584969228$, $b_1 = 21.0996530827$, $b_2 = 25.6329561486$, $b_3 = 9.5733223454$

<2>The second type of first-order empirical expression (model="f2") is (Kitis et al., 1998)

$$I(T) = I_m exp[1 + \frac{E}{kT}\frac{T-T_m}{T_m} - \frac{T^2}{T_m^2} exp(\frac{E}{kT}\frac{T-T_m}{T_m})(1 - \frac{2kT}{E}) - \frac{2kT_m}{E}]$$

<3>The third type of first-order function fits a weibull function (model="f3") (Pagonis et al., 2001)

$$I(T) = 2.713 I_m (\frac{T-T_m}{b} + 0.996)^{15} exp[-(\frac{T-T_m}{b} + 0.996)^{16}]$$

where $b = \sqrt{\frac{242.036T_m^3k^2}{(E+T_mk)^2-7T_m^2k^2}}$

Second-order glow peaks appear if the re-trapping probability is comparable with or greater than that of recombination during excitation. The three parameters describing a glow peak are the same

as those of first-orders. Two empirical expressions describing second-order glow peaks are available in function tgcd:
<4>The first type of second-order empirical expression (`model="s1"`) is (Kitis et al., 1998)

$$I(T) = 4I_m exp(\frac{E}{kT}\frac{T-T_m}{T_m})[\frac{T^2}{T_m^2}(1-\frac{2kT}{E})exp(\frac{E}{kT}\frac{T-T_m}{T_m}) + 1 + \frac{2kT_m}{E}]^{-2}$$

<5>The second type of second-order function fit a logistic asymmetric function (`model="s2"`) (Pagonis and Kitis, 2001)

$$I(T) = 5.2973I_m[1 + exp(-(\frac{T-T_m}{a_2} + 0.38542))]^{-2.4702} exp[-(\frac{T-T_m}{a_2} + 0.38542)]$$

where $a_2 = \sqrt{\frac{1.189T_m^4k^2}{E^2+4ET_mk}}$

General-order glow peaks are produced in intermediate cases (neither of first-order, nor of second-order). The four parameters describing a glow peak are: I_m, E, T_m, and b. Three empirical expressions describing general-order glow peaks are available in function tgcd:
<6>The first type of general-order empirical expression (`model="g1"`) is (Kitis et al., 1998)

$$I(T) = I_m b^{\frac{b}{b-1}} exp(\frac{E}{kT}\frac{T-T_m}{T_m})[(b-1)(1-\frac{2kT}{E})\frac{T^2}{T_m^2}exp(\frac{E}{kT}\frac{T-T_m}{T_m}) + 1 + \frac{2kT_m(b-1)}{E}]^{-\frac{b}{b-1}}$$

<7>The second type of general-order empirical expression (`model="g2"`) is (Gomez-Ros and Kitis, 2002)

$$I(T) = I_m exp(\frac{E}{kT_m} - \frac{E}{kT})[1 + \frac{b-1}{b}\frac{E}{kT_m}(\frac{T}{T_m}exp(\frac{E}{kT_m} - \frac{E}{kT})F(\frac{E}{kT}) - F(\frac{E}{kT_m}))]^{-\frac{b}{b-1}}$$

where $F(x)$ is a rational approximation function of the form

$$F(x) = 1 - \frac{a_0+a_1x+x^2}{b_0+b_1x+x^2}$$

$a_0 = 0.250621$, $a_1 = 2.334733$, $b_0 = 1.681534$, $b_1 = 3.330657$

<8>The third type of general-order empirical expression (`model="g3"`) is (Gomez-Ros and Kitis, 2002)

$$I(T) = I_m exp(\frac{E}{kT_m^2}(T-T_m))[\frac{1}{b} + \frac{b-1}{b}exp(\frac{E}{kT_m^2}(T-T_m))]^{-\frac{b}{b-1}}$$

One trap-one recombination (OTOR) model based semi-analytical expressions have also been applied to fit glow peaks, by using either the Lambert W function (Kitis and Vlachos, 2013; Sadek et al., 2015; Kitis et al., 2016) or the Wright Omega function (Singh and Gartia, 2013; 2014; 2015). The four parameters describing a glow peak are: I_m, E, T_m, and $R = \frac{A_n}{A_m}$ (where A_n and A_m represent the retrapping and recombination probabilities, respectively). Two analytical expressions describing the OTOR model are available in function tgcd:
<9>The semi-analytical expression derived using the Wright Omega function ($R = \frac{A_n}{A_m} < 1$) can be described as (`model="wo"`)

$$I(T) = I_m exp(-\frac{E}{kT}\frac{T_m-T}{T_m})\frac{\omega(Z_m)+[\omega(Z_m)]^2}{\omega(Z)+[\omega(Z)]^2}$$

where $Z_m = \frac{R}{1-R} - log(\frac{1-R}{R}) + \frac{Eexp(\frac{E}{kT_m})}{kT_m^2(1-1.05R^{1.26})}F(T_m, E)$,

and $Z = \frac{R}{1-R} - log(\frac{1-R}{R}) + \frac{Eexp(\frac{E}{kT_m})}{kT_m^2(1-1.05R^{1.26})}F(T, E)$,

<10.1>The semi-analytical expression derived using the Lambert W function for $R = \frac{A_n}{A_m} < 1$ can be described as (`model="lw"`)

$I(T) = I_m exp(-\frac{E}{kT}\frac{T_m-T}{T_m})\frac{W(exp(Z_m))+[W(exp(Z_m))]^2}{W(exp(Z))+[W(exp(Z))]^2}$

where $Z_m = \frac{R}{1-R} - log(\frac{1-R}{R}) + \frac{Eexp(\frac{E}{kT_m})}{kT_m^2(1-1.05R^{1.26})}F(T_m, E)$,

and $Z = \frac{R}{1-R} - log(\frac{1-R}{R}) + \frac{Eexp(\frac{E}{kT_m})}{kT_m^2(1-1.05R^{1.26})}F(T, E)$,

<10.2>The semi-analytical expression derived using the Lambert W function for $R = \frac{A_n}{A_m} > 1$ can be described as (`model="lw"`)

$I(T) = I_m exp(-\frac{E}{kT}\frac{T_m-T}{T_m})\frac{W(-1,-exp(-Z_m))+[W(-1,-exp(-Z_m))]^2}{W(-1,-exp(-Z))+[W(-1,-exp(-Z))]^2}$

where $Z_m = |\frac{R}{1-R}| + log(|\frac{1-R}{R}|) + \frac{Eexp(\frac{E}{kT_m})}{kT_m^2(2.963-3.24R^{-0.74})}F(T_m, E)$,

and $Z = |\frac{R}{1-R}| + log(|\frac{1-R}{R}|) + \frac{Eexp(\frac{E}{kT_m})}{kT_m^2(2.963-3.24R^{-0.74})}F(T, E)$

$F(T_m, E)$ and $F(T, E)$ are described as follows

$F(T_m, E) = T_m exp(-\frac{E}{kT_m}) + \frac{E}{k}Ei(-\frac{E}{kT_m})$,

$F(T, E) = Texp(-\frac{E}{kT}) + \frac{E}{k}Ei(-\frac{E}{kT})$

where $\omega(x)$ and $Ei(x)$ are the wright Omega function and the exponential integral function for variable x, respectively. $W(x)$ and $W(-1, x)$ are the principal and the second branches of the Lambert W function, respectively. The Fortran 90 subroutine used for evaluating the Wright Omega function is transformed from the Matlab code provided by Andrew Horchler (https://github.com/horchler/wrightOmegaq). The Fortran 90 subroutine (original Fortran 77 version by William Cody) used for evaluating the Lambert W function written by John Burkardt is available at https://people.sc.fsu.edu/~jburkardt/f_src/toms743/toms743.f90. The Fortran 90 subroutine used for evaluating the exponential integral function is written by John Burkardt (original Fortran 77 version by William Cody) (https://people.sc.fsu.edu/~jburkardt/f_src/toms715/toms715.f90).

Mixed-order kinetic models introduce an extra parameter $\alpha = \frac{n_0}{n_0+M}$ where n_0 is the initial filled concentration of the active traps and M is the trap concentration of the thermally disconnected deep traps (Sunta et al., 2002). The four parameters describing a glow peak are: I_m, E, T_m, and α. Three empirical expressions describing mixed-order glow peaks are available in function tgcd:
<11>The first type of mixed-order empirical expression (`model="m1"`) is (Kitis and Gomez-Ros, 2000)

$$I(T) = \frac{I_m[exp(\frac{1-\frac{2kT_m}{E}}{R_m})-\alpha]^2 exp(\frac{E}{kT}\frac{T-T_m}{T_m})exp[\frac{T^2}{T_m^2 R_m}exp(\frac{E}{kT}\frac{T-T_m}{T_m})(1-\frac{2kT}{E})]}{exp(\frac{1-\frac{2kT_m}{E}}{R_m})[exp[\frac{T^2}{T_m^2 R_m}exp(\frac{E}{kT}\frac{T-T_m}{T_m})(1-\frac{2kT}{E})]-\alpha]^2}$$

where $R_m = \frac{A_m+\alpha}{A_m-\alpha}$ and $A_m = exp(\frac{A_m-\alpha}{A_m+\alpha}(1-\frac{2kT_m}{E}))$

<12>The second type of mixed-order empirical expression (`model="m2"`) is (Gomez-Ros and Kitis, 2002)

$$I(T) = \frac{4I_m R_m^2 exp(\frac{E}{kT_m}-\frac{E}{kT})exp[R_m\frac{E}{kT_m}(\frac{T}{T_m}exp(\frac{E}{kT_m}-\frac{E}{kT})F(\frac{E}{kT})-F(\frac{E}{kT_m}))]}{(1+R_m)[exp[R_m\frac{E}{kT_m}(\frac{T}{T_m}exp(\frac{E}{kT_m}-\frac{E}{kT})F(\frac{E}{kT})-F(\frac{E}{kT_m}))]-(1-R_m)]^2}$$

where $R_m = (1-\alpha)(1+0.2922\alpha-0.2783\alpha^2)$

$F(x) = 1-\frac{a_0+a_1 x+x^2}{b_0+b_1 x+x^2}$

$a_0 = 0.250621$, $a_1 = 2.334733$, $b_0 = 1.681534$, $b_1 = 3.330657$

<13>The third type of mixed-order empirical expression (`model="m3"`) is (Vejnovic et al., 2008)

$$I(T) = \frac{I_m\alpha(2-l)^2 exp[\frac{T^2}{T_m^2}(\frac{2}{l}-1)exp(\frac{E}{kT_m}\frac{T-T_m}{T_m})(1-\frac{2kT}{E})]exp(\frac{E}{kT_m}\frac{T-T_m}{T_m})}{(l-1)[exp[\frac{T^2}{T_m^2}(\frac{2}{l}-1)exp(\frac{E}{kT_m}\frac{T-T_m}{T_m})(1-\frac{2kT}{E})]-\alpha]^2}$$

where $\alpha = (l-1)exp[\frac{2-l}{l}(1-\frac{2kT_m}{E})]$

The **background** will be subtracted using the following expression if `subBG=TRUE` (Horowitz and Yossian, 1995; Kitis et al., 2012)

$I(T) = A + Bexp(\frac{T}{C})$

where A, B, and C are positive parameters to be optimized.

The procedure minimizes the objective: $fcn = \sum_{i=1}^{n}|y_i^o - y_i^f|, i = 1, ..., n$ where y_i^o and y_i^f denote the *i-th* observed and fitted signal value, respectively, and n indicates the number of data points.

The Levenberg-Marquardt algorithm (More, 1978) (*minpack*: https://netlib.org/minpack/, original Fortran 77 version by Jorge More, Burton Garbow, Kenneth Hillstrom. Fortran 90 version by John Burkardt https://people.sc.fsu.edu/~jburkardt/f_src/minpack/minpack.f90) was modified so as to supports constraints and fixes of parameters. A "trial-and-error" protocol with starting values generated uniformly around the given starting parameters `inisPAR` and `inisBG` is performed repeatedly to search the optimal parameters that give a minimum Figure Of Merit (FOM) value (Balian and Eddy, 1977).

Kinetic parameters can be initialized by the user through argument `inisPAR` or by clicking with a mouse on the plot of the thermoluminescence glow curve showing peak maxima if `inisPAR=NULL`. Background parameters can be initialized by the user through argument `inisBG` or by clicking with a mouse on the plot of the thermoluminescence glow curve to select 4 data points if `inisBG=NULL`.

Parameters can be interactively constrained and fixed by modifying the following elements in a automatically generated **Dialog Table** if `edit.inis=TRUE`:
(1) `INTENS[min,max,ini,fix]`: lower and upper bounds, starting and fixing values of I_m
(2) `ENERGY[min,max,ini,fix]`: lower and upper bounds, starting and fixing values of E
(3) `TEMPER[min,max,ini,fix]`: lower and upper bounds, starting and fixing values of T_m
(4) `bValue[min,max,ini,fix]`: lower and upper bounds, starting and fixing values of b in the general-order model
(5) `rValue[min,max,ini,fix]`: lower and upper bounds, starting and fixing values of R in the OTOR model
(6) `aValue[min,max,ini,fix]`: lower and upper bounds, starting and fixing values of α in the mixed-order model
(7) `BG[min,max,ini,fix]`: lower and upper bounds, starting and fixing values of background parameters A, B, and C

Value

Return an **invisible** list containing the following elements:

`comp.sig`	calculated signal values for each glow peak
`residuals`	calculated residual values
`pars`	optimized parameters stored in a matrix
`BGpars`	optimized background parameters, it returns `NULL` if `subBG=FALSE`
`ff`	calculated frequency factors, it returns `NULL` if `hr=NULL`
`sp`	parameters used for describing the shape of a glow peak, see function simPeak for details
`resolution`	resolutions of optimized glow peaks calculated after Kitis and Pagonis (2019), it returns `NULL` if `npeak=1`
`SSR`	Squared Sum of Residuals
`RCS`	Reduced Chi-Square value
`R2`	squared "pearson" correlation between observed and fitted signals
`FOM`	Figure Of Merit value calculated after Balian and Eddy (1977)

Note

Function tgcd analyzes only thermoluminescence glow curves recorded with linear heating function (LHF) profile. The model to be optimized should not be underdetermined. This means that the number of data points (n_d) should exceed the number of parameters (n_2).

If it is not `NULL` the argument `inisPAR`, the procedure used for initializing kinetic parameters by clicking with a mouse will not be triggered. Similarly, if it is not `NULL` the argment `inisBG`, the procedure used for initializing background parameters by clicking with a mouse will not be triggered.

The user is advocated to use `mwt`, `mdt`, and `mr` to specify the allowed maximum total half-width, the allowed minimum distance, and the allowed minimum resolution respectively to resolve seriously

overlapped glow peaks during the trial-and-error protocol.

Adrie J.J. Bos is appreciated for providing the reference glow curves of the GLOCANIN project to test this routine.

Amr M. Sadek is thanked for providing the Matlab code implementing the Lambert W (Wright Omega) function for reference.

George Kitis is appreciated for giving some useful suggestions to improve the program and providing many experimentally measured thermoluminescence glow curves to check the routine.

References

Balian HG, Eddy NW, 1977. Figure-of-merit (FOM), an improved criterion over the normalized chi-squared test for assessing goodness-of-fit of gamma-ray spectral peaks. Nuclear and Instruments Methods, 145: 389-395.

Bos AJJ, Piters TM, Gomez-Ros JM, Delgado A, 1993a. An intercomparison of glow curve analysis computer programs. IRI-CIEMAT Report, 131-93-005.

Gomez-Ros JM, Kitis G, 2002. Computerised glow curve deconvolution using general and mixed order kinetics. Radiation Protection Dosimetry, 101(1-4): 47-52.

Horowitz YS, Yossian D, 1995. Computerized glow curve deconvolution: application to thermoluminescence dosimetry. Radiation Protection Dosimetry, 60: 1-114.

Kitis G, Carinou E, Askounis P, 2012. Glow-curve de-convolution analysis of TL glow-curve from constant temperature hot gas TLD readers. Radiation Measurements, 47: 258-265.

Kitis G, Gomes-Ros JM, 2000. Thermoluminescence glow-curve deconvolution functions for mixed order of kinetics and continuous trap distribution. Nuclear Instruments and Methods in Physics Research A, 440: 224-231.

Kitis G, Gomes-Ros JM, Tuyn JWN, 1998. Thermoluminescence glow curve deconvolution functions for first, second and general orders of kinetics. Journal of Physics D: Applied Physics, 31(19): 2636-2641.

Kitis G, Pagonis V, 2019. On the resolution of overlapping peaks in complex thermoluminescence glow curves. Nuclear Instruments and Methods in Physics Research, A, 913: 78-84.

Kitis G, Polymeris GS, Pagonis V, 2019. Stimulated luminescence emission: From phenomenological models to master analytical equations. Applied Radiation and Isotopes, 153: 108797.

Kitis G, Polymeris GS, Sfampa IK, Prokic M, Meric N, Pagonis V, 2016. Prompt isothermal decay of thermoluminescence in MgB4O7:Dy, Na and LiB4O7:Cu, In dosimeters. Radiation Measurements, 84: 15-25.

Kitis G, Vlachos ND, 2013. 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.

More JJ, 1978. "The Levenberg-Marquardt algorithm: implementation and theory," in Lecture Notes in Mathematics: Numerical Analysis, Springer-Verlag: Berlin. 105-116.

Pagonis V, Mian SM, Kitis G, 2001. Fit of first order thermoluminescence glow peaks using the weibull distribution function. Radiation Protection Dosimetry, 93(1): 11-17.

Pagonis V, Kitis G, 2001. Fit of second order thermoluminescence glow peaks using the logistic distribution function. Radiation Protection Dosimetry, 95(3): 225-229.

Sadek AM, Eissa HM, Basha AM, Carinou E, Askounis P, Kitis G, 2015. The deconvolution of thermoluminescence glow-curves using general expressions derived from the one trap-one recombination (OTOR) level model. Applied Radiation and Isotopes, 95: 214-221.

Singh LL, Gartia RK, 2013. Theoretical derivation of a simplified form of the OTOR/GOT differential equation. Radiation Measurements, 59: 160-164.

Singh LL, Gartia RK, 2014. Glow-curve deconvolution of thermoluminescence curves in the simplified OTOR equation using the Hybrid Genetic Algorithm. Nuclear Instruments and Methods in Physics Research B, 319: 39-43.

Singh LL, Gartia RK, 2015. Derivation of a simplified OSL OTOR equation using Wright Omega function and its application. Nuclear Instruments and Methods in Physics Research B, 346: 45-52.

Sunta CM, Ayta WEF, Chubaci JFD, Watanabe S, 2002. General order and mixed order of thermoluminescence glow curves-a comparison. Radiation Measurements, 35: 47-57.

Vejnovic Z, Pavlovic MB, Davidovi M, 2008. Thermoluminescence glow curve deconvolution function for the mixed-order kinetics. Radiation Measurements, 43: 1325-1330.

Further reading

Afouxenidis D, Polymeris GS, Tsirliganis NC, Kitis G., 2011. Computerised curve deconvolution of TL/OSL curves using a popular spreadsheet program. Radiation Protection Dosimetry, 1-8.

Bos AJJ, Piters TM, Gomez Ros JM, Delgado A, 1993b. An intercomparison of glow curve analysis computer programs: I. Synthetic glow curves. Radiation Protection Dosimetry, 47(1-4), 473-477.

Chung KS, Choe HS, Lee JI, Kim JL, Chang SY, 2005. A computer program for the deconvolution of thermoluminescence glow curves. Radiation Protection Dosimetry, 115(1-4): 345-349.

Delgado A, Gomez-Ros JM, 2001. Computerised glow curve analysis: a tool for routine thermoluminescence dosimetry. Radiation Protection Dosimetry, 96(1-3): 127-132.

El-Hafez AI, Yasin MN, Sadek AM, 2011. GCAFIT-A new tool for glow curve analysis in thermoluminescence nanodosimetry. Nuclear Instruments and Methods in Physics Research A, 637: 158-163.

Espinosa G, Castano VM, 1991. Automated derivative analysis of thermoluminescence glow curves. Applied Radiation and Isotopes, 42(4): 377-381.

Harvey JA, Rodrigues ML, Kearfott JK, 2011. A computerized glow curve analysis (GCA) method for WinREMS thermoluminescent dosimeter data using MATLAB. Applied Radiation and Isotopes, 69(9):1282-1286.

Horowitz YS, Moscovitch M, 2012. Highlights and pitfalls of 20 years of application of computerised glow curve ananlysis to thermoluminescence research and dosimetry. Radiation Protection Dosimetry, 1-22.

Horowitz Y, Delgado A, Pradhan AS, Yoder RC, 2002. Topics under debate: the use of computerised glow curve analysis will optimise personal thermoluminescence dosimetry. Radiation Protection Dosimetry, 102(3): 269-277.

Karmakar M, Bhattacharyya S, Sarkar A, Mazumdar PS, Singh SD, 2017. Analysis of thermoluminescence glow curves using derivatives of different orders. Radiation Protection Dosimetry, 1-10.

Kiisk V, 2013. Deconvolution and simulation of thermoluminescence glow curves with Mathcad. Radiation Protection Dosimetry, 156(3): 261-267.

Pagonis V, Kitis G, Furetta C, 2006. Numerical and practical exercises in thermoluminescence. Springer Science & Business Media.

Puchalska M, Bilski P, 2006. GlowFit-a new tool for thermoluminescence glow-curve deconvolution. Radiation Measurements, 41(6): 659-664.

Sadek AM, 2013. Test of the accuracy of the computerized glow curve deconvolution algorithm for the analysis of thermoluminescence glow curves. Nuclear Instruments and Methods in Physics Research A, 712: 56-61.

Sadek AM, Kitis G., 2018. Impact of non-fulfillment of the super position principle on the analysis of thermoluminescence glow-curve. Radiation Measurements, 116: 14-23.

Singh TB, Rey L, Gartia RK, 2011. Applications of PeakFit software in thermoluminescence studies. Indian Journal of Pure & Applied Physics, 49: 297-302.

Sunta CM, 2015. Unraveling thermoluminescence. Springer India.

See Also

simPeak; simqOTOR; savgol

Examples

```
# Load the data.
  data(Refglow)
  data(Kitis)

# (1) Deconvolution of a glow curve using 4 peaks (no background subtraction) with
#     the Wright Omega function using specified initial kinetic parameters.

  knPars <-
  cbind(c(400, 550, 850, 1600), # Im
        c(1.4, 1.5, 1.6, 2),    # E
        c(420, 460, 480, 510),  # Tm
        c(0.1, 0.1, 0.1, 0.1))  # R

  dd1 <- tgcd(Refglow$x002, npeak=4, model="wo",
              inisPAR=knPars, nstart=10, edit.inis=FALSE)

  head(dd1$comp.sig)
  dd1$pars
  dd1$sp
  dd1$FOM

# (2) Deconvolution of a glow curve using 5 peaks (with background subtraction) with
#     a mixed-order model using user-supplied intial kinetic and background parameters.
  knPars <-
  cbind(c(46829.06, 187942.43, 121876.22, 110390.55, 67978.33), #Im
        c(1.17, 1.14, 1.57, 0.77, 1.31),                        # E
        c(369.86, 400.69, 428.51,  482.41, 537.28),             # Tm
        c(0.75,  0.81, 0.92, 0.001, 0.29))                      # a
```

```
bgPars <- c(1, 10, 100)  # A, B, C.

dd2 <- tgcd(Kitis$x009, npeak=5, model="m1", subBG=TRUE,
            inisPAR=knPars, inisBG=bgPars, nstart=10, edit.inis=FALSE)

dd2$residual
dd2$SSR
dd2$R2
dd2$BGpars
```

Index

∗ **Deconvolution**
  tgcd-package, 2
∗ **Glow curve deconvolution**
  Kitis, 3
  Refglow, 3
  tgcd, 9
∗ **Glow curve simulation**
  Kitis, 3
  Refglow, 3
∗ **Glow curve**
  tgcd-package, 2
∗ **Glow peak simulation**
  simPeak, 5
  simqOTOR, 7
∗ **Glow peak**
  tgcd-package, 2
∗ **Savitzky-Golay smoothing**
  savgol, 4
∗ **Simulation**
  tgcd-package, 2
∗ **Thermoluminescence**
  Kitis, 3
  Refglow, 3
  savgol, 4
  simPeak, 5
  simqOTOR, 7
  tgcd, 9
  tgcd-package, 2
∗ **package tgcd**
  Kitis, 3
  Refglow, 3
  savgol, 4
  simPeak, 5
  simqOTOR, 7
  tgcd, 9

character, *6*, *8*, *10*, *11*

integer, *4*, *10*, *11*

Kitis, 3

logical, *6*, *8*, *10*, *11*

matrix, *10*

numeric, *4–6*, *8*, *10*, *11*

Refglow, 3

savgol, 4, *18*
seq, *5*
simPeak, 5, *6*, *8*, *9*, *15*, *18*
simqOTOR, 7, *7*, *8*, *18*

tgcd, *5*, *7*, 9, *9*, *11–13*, *15*
tgcd-package, 2

vector, *5*, *8*, *10*