Chapter 6
LOCALIZED TRANSITIONS AND QUANTUM TUNNELING

Abstract In this chapter we consider the different types of quantum tunneling localized transition models (TLT) models. We describe four different types of TLT models: ground state tunneling models (GST), irradiation ground state tunneling models (IGST), excited state tunneling models (EST) and thermally-assisted excited state tunneling models (TA-EST). We provide R codes for exploring the properties of each model, discuss their physical principles, and code approximate analytical solutions to the differential equations describing each model. R codes are provided for simulating the nearest neighbor distribution in a random distribution of defects in a solid, and also provide an example of analyzing experimental data to obtain the g-factor for the anomalous fading phenomenon (AF). Additional R codes simulate simultaneous irradiation and tunneling, and excited state tunneling phenomena. We show how to analyze experimental TL and OSL data for freshly irradiated samples, using the analytical Kitis-Pagonis equations KP-TL and KP-CW. Finally we present the thermally-assisted excited state model used in low temperature thermochronometry studies, and show how quantum tunneling phenomena can be simulated using the TUN functions in the package *RLumCarlo*.

Code 6.1: The nearest neighbors distribution

```
# Fig 1 in Pagonis and Kulp paper
# Original Mathematica Program written by V Pagonis
# R  version written by Johannes Friedrich
rm(list = ls(all = TRUE)) # empties the environment
library("plot3D")
library("FNN")
## Define Parameters ----
sideX <- 100e-9 # lenght of quader in m
```

```
sideX_nm <- sideX*1e9 # length of quader in nm
N_pts <- 50
alpha <- 9e9
N_centers <- 300
rho <- N_centers/sideX^3
r_prime <- function(r) (4*pi*rho/3)^(1/3) * r * 1e-9
xyz_traps <- data.frame(
  x = sample(1:sideX_nm, N_pts, replace = TRUE),
  y = sample(1:sideX_nm, N_pts, replace = TRUE),
  z = sample(1:sideX_nm, N_pts, replace = TRUE))
xyz_centers <- data.frame(
  x = sample(1:sideX_nm, N_centers, replace = TRUE),
  y = sample(1:sideX_nm, N_centers, replace = TRUE),
  z = sample(1:sideX_nm, N_centers, replace = TRUE))
par(mfrow=c(1,2))
  plot3D::scatter3D(xyz_centers$x, #plot centers (blue)
                xyz_centers$y,cex=1,
                xyz_centers$z, bty = "g", pch = 1, theta = 30,
                phi = 30, col = "blue")
  plot3D::scatter3D(xyz_traps$x, # add traps (red)
                xyz_traps$y, cex=1,
                xyz_traps$z, bty = "g", pch = 17,theta = 30,
                 phi = 30, col = "red",add = TRUE)
legend("topright",bty="n","(a)")
    ## find nearest neighbour
    dist <- FNN::get.knnx(data = as.matrix(xyz_centers),
                          query = as.matrix(xyz_traps),k = 1)
   ## plot histogram
    distance_nm<-as.vector(dist$nn.dist)
    hist(distance_nm,xlim=c(0,22),main=" ")
    ## calc analytical solution
    r <- seq(0, 20, 0.1)
    distr_ana <- 3 * 8 * r_prime(r)^2 * exp(-(r_prime(r)^3))
    ## plot analytical solution
    lines(x = r,  y = distr_ana)
legend("topright",bty="n","(b)")
```

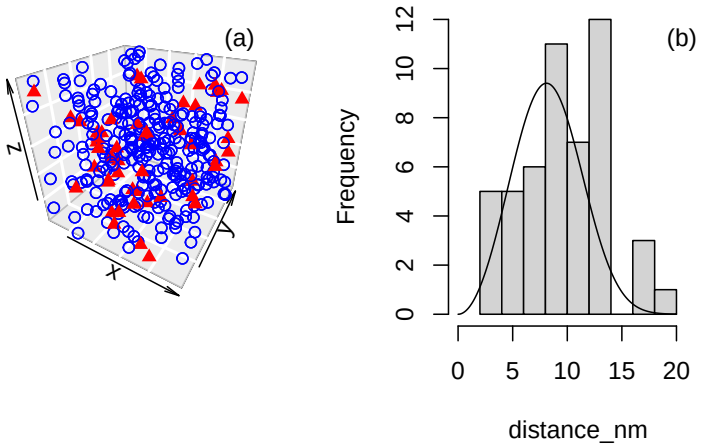

Fig. 6.1: (a) A cube with side $d = 100$ nm contains 50 electrons (triangles) and 300 recombination centers (circles). (b) Histogram of the nearest neighbor distances of electron-acceptor pairs from the cube in (a). The solid line in (b) represents the analytical equation for the distribution of nearest neighbors Eq.(**??**). For more information see Pagonis and Kulp [42].

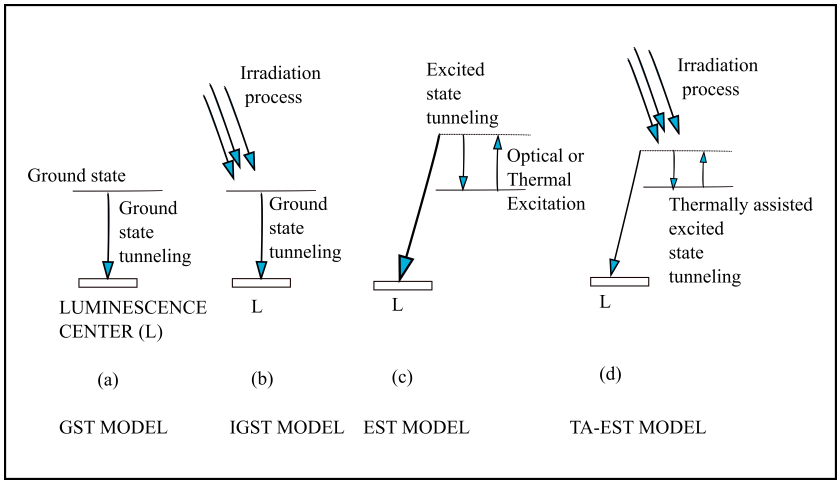

Fig. 6.2: Schematic depiction of several TLT models: (a) The ground state tunneling (GST) model (Tachiya and Mozumder [50], Huntley [11]). (b) The more general irradiation and ground state tunneling (IGST) model studied by Li and Li [23], in which anomalous fading and natural irradiation are taking place simultaneously. (c) The excited state tunneling (EST) model (Jain et al. [12]). (d) Simultaneous irradiation and thermally assisted excited state tunneling (TA-EST) model by Brown et al. [5].

Code 6.2: Time evolution of the nearest neighbors distribution

```
rm(list=ls())
s<-3e15                    # frequency factor
rho<-1e-6                  # rho-prime values 0.005-0.02
rc<-0.0                    # for freshly irradiated samples, rc=0
times<-3.154e7*c(0,1e2,1e4,1e6)          # times in seconds
rprimes<-seq(from=rc,to=2.2,by=0.002)    # rprime=0-2.2

##### function to find distribution of distances ###
fingDistr<-function(tim){3*(rprimes**2.0)*exp(-(rprimes**3.0))*
    exp(-exp(-(rho**(-1/3))*rprimes)*s*tim)}
######

distribs<-sapply(times,fingDistr)
# Plots
cols=c(NA,NA,NA,1:4)
matplot(rprimes,distribs,xlab="Dimensionless distance r'",
ylab="Nearest neighbor Distribution g(r')",type="l",lwd=4)
```

```
legend("topright",bty="n", lty=c(NA,NA,NA,1:4), lwd=4,
col=cols,legend = c("Elapsed", "time"  ," ","t=0 s",
 expression("10"^"2"*" years"),
expression("10"^"4"*" years"),expression("10"^"6"*" years")))
```

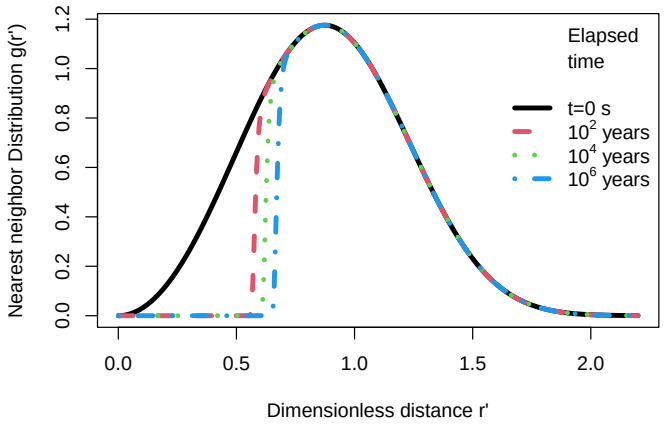

Fig. 6.3: Examples of the nearest neighbor distribution at different times $t = 0, 10^2, 10^4, 10^6$ years, based on Eq.(**??**). The solid black line represents the nearly symmetric distribution $g(r')$ at time $t = 0$. As time increases, the "tunneling front" is the almost vertical line which moves to the right, as more and more electrons are recombining at larger distances r'.

Code 6.3: Ground state tunneling: Remaining electrons n(t)

```
# Simulate Loss of charge due to ground state tunneling (GST)
rm(list=ls())
z<-1.8
s<-3e15                    # frequency factor
rho1<-1e-6                 # rho-prime values
rho2<-5e-6
rho3<-1e-5
years<-1000
elapsedt<-3.154e7*years
```

```
t<-seq(from=1,to=elapsedt,by=1e6)
gr1<-100*exp(-rho1*(log(z*s*t)**3.0))
gr2<-100*exp(-rho2*(log(z*s*t)**3.0))
gr3<-100*exp(-rho3*(log(z*s*t)**3.0))
plot(unlist(t)/(3.154e7),unlist(gr1),type="l",lty=1,lwd=2,
ylim=c(0,130),pch=1,ylab="Remaining electrons [%]",
xlab="Elapsed time t [years]")
lines(unlist(t)/(3.154e7),unlist(gr2),lwd=2,col="red",lty=2)
lines(unlist(t)/(3.154e7),unlist(gr3),lwd=3,col="green",lty=2)
legend("topright",bty="n",lwd=2, lty=c(NA,1,2,3),
legend = c("Acceptor Density"  ,expression("10"^"-6"*""),
expression("5x10"^"-6"*""),expression("10"^"-5"*"")),
 col=c(NA,"black","red","green"))
```

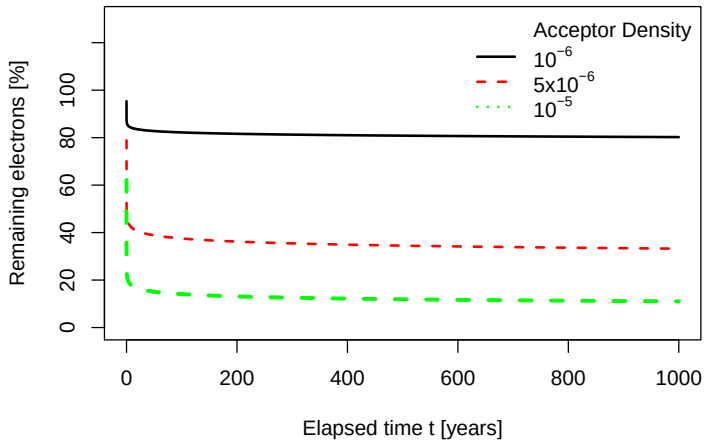

Fig. 6.4: Examples of the loss of charge due to ground state tunneling, for different dimensionless acceptor densities $\rho' = 10^{-6}, 5\times10^{-6}, 10^{-5}$, based on Eq.(**??**). As ρ' increases, the rate of charge loss increases. As the elapsed time t increases, the initial fast loss of charge is followed by a "long tailed" decay, characteristic of quantum tunneling phenomena.

Code 6.4: Anomalous fading (AF) and the g-factor

```
rm(list=ls())
# Load the data of Anomalous fading (AF) and calculate the g-factor
par(mfrow=c(1,2))
mydata <- read.table("durnago0sok.txt")
T<-mydata[,1]
TL<-mydata[,2]/1e5
plot(T,TL,type="o",pch=1,col="red",xlim=c(50,450),ylim=c(0,1.6),
  xlab=expression("Temperature ["^"o"*"C]"),ylab ="TL (a.u.)")
mydata2 <- read.table("durango10daysok.txt")
T2<-mydata2[,1]
TL2<-mydata2[,2]/1e5
lines(T2,TL2,type="o",pch=2,col="blue")
legend("left",bty="n","(a)")
legend("topleft",bty="n",lwd=2, lty=c(NA,NA,1,2,3),
pch=c(NA,NA,1,2),
legend=c(expression('Anomalous', 'fading ', 't=0 s',
't=10 days')),col=c(NA,NA,"red","blue"))
mydata3 <- read.table("DurangoAFdataok.txt")
t<-mydata3[,1]
RTL<-mydata3[,2]
#plot(t,RTL,xlab="ln(t/to)",ylab="Remnat TL [%]")
y<-RTL
x<-t
bestfit<-lm(y~x)
coefficients(bestfit)
plot(x, y, xlab=expression("ln(t/to)"),ylab = "RTL [%]")
abline(lm(y~x))
slope<-abs(coefficients(bestfit)[[2]])
g<-230.2*slope
paste0("g-factor=",round(g,digits=2)," % per decade")
legend("topright",bty="n",
legend=c(expression('Durango apatite',
'g-factor=20.5% ',)))
legend("left",bty="n","(b)")
```

```
  ## (Intercept)           x
  ##  0.99615757 -0.08918921
  ## [1] "g-factor=20.53 % per decade"
```

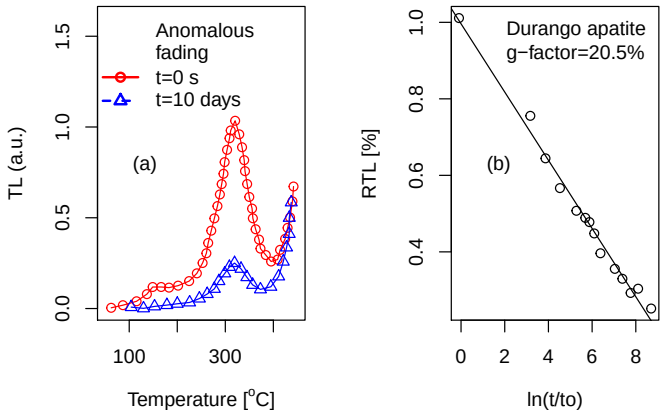

Fig. 6.5: Anomalous fading effect in Durango apatite. (a) The TL signal is measured immediately after irradiation, and after 10 days have elapsed at room temperatures. (b) Analysis of the remnant TL signal from (a), to obtain the g-factor for this materials. For more details, see Polymeris et al. [49].

Code 6.5: Simultaneous irradiation and anomalous fading in nature

```
# Simultaneous irradiation and anomalous fading in nature
rm(list=ls())
s<-3e15                    # frequency factor
rho1<-1e-6                 # rho-prime values 0.005-0.02
rho2<-2e-6
rho3<-3e-6
Do<-538
X<-3/(1000*365*3600*24)       #natural dose rate X=3 Gy/Ka
curve((1-exp(-x/Do))*exp(-rho1*(log(Do*s/X)**3.0)),1,3500, 200,
lty=1,lwd=3,
ylab=expression('L'[FADED]*'  [a.u.]'),xlab="Natural Dose [Gy]",
col=1,ylim=c(0,1.2))
curve((1-exp(-x/Do))*exp(-rho2*(log(Do*s/X)**3.0)),1,3500,200,
col=2,lty=2, lwd=3,   add=TRUE)
curve((1-exp(-x/Do))*exp(-rho3*(log(Do*s/X)**3.0)),1,3500,200,
col=3,   lwd=3,   lty=3,add=TRUE)
```

```
legend("topright",bty="n",lwd=3, lty=c(NA,1,2,3),legend =
c("Acceptor Density"  ,expression("10"^"-6"*""),
expression("2x10"^"-6"*""),expression("3x10"^"-6"*"")),
col=c(NA,1:3))
legend('topleft',bty="n",c("Simultaneous",
"Irradiation and fading") )
```

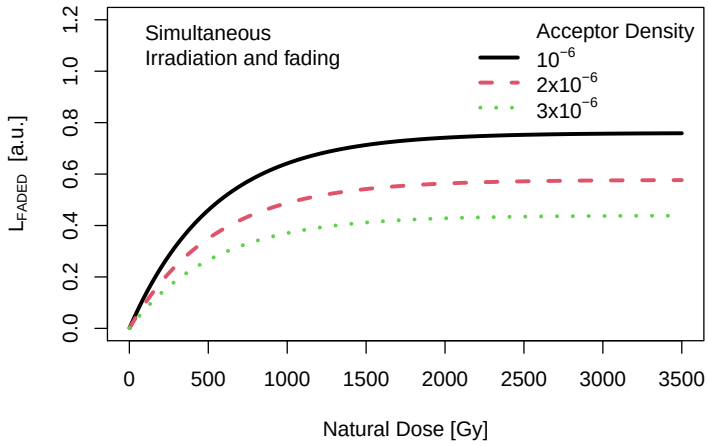

Fig. 6.6: Examples of the accumulated charge $n(t)$ during simultaneous irradiation and fading in nature, due to ground state tunneling, for three different dimensionless acceptor densities $\rho' = 10^{-6}, 2\times10^{-6}, 3\times10^{-6}$. As ρ' increases, the rate of charge accumulation and the corresponding saturation signal *decrease*. For more details, see Pagonis and Kitis [37].

Code 6.6: CW-IRSL data fitted with KP-CW equation

```
#Fit CW-IRSL data with KP-CW equation
rm(list = ls(all=T))
options(warn=-1)
library("minpack.lm")
## fit to analytical KP-CW equation for CW-IRSL (TLT model)-
mydata <- read.table("ph300s0IR.asc")
```

```
t<-as.numeric(gsub(",", ".", gsub("\\.", "", mydata[,1])))
y<-as.numeric(gsub(",", ".", gsub("\\.", "", mydata[,3])))
mydata<-data.frame(t,y)
plot(t,y,log="xy",
xlab="Time [s]",ylab="CW-IRSL [counts/s]",col="black",pch=1)
fit_data <-mydata
fit <- minpack.lm::nlsLM(
 formula=y~ imax*exp (-rho*(log(1 + A*t)) ** 3.0)*
    (log(1 + A*t) ** 2.0)/(1 +  t*A)+bgd,
  data = fit_data,
  start = list(imax=3,A=1.1,rho=0.03,bgd=min(y)))
imax_fit <- coef(fit)[1]
A_fit <- coef(fit)[2]
rho_fit <- coef(fit)[3]
bgd_fit <- coef(fit)[4]
## plot analytical solution
lines(
  x = t,
  y =imax_fit*exp (-rho_fit*(log(1 + A_fit*t)) ** 3.0)*
    (log(1 + A_fit*t) ** 2.0)/(1 +  t*A_fit)+bgd_fit,
  col = "red",lwd=2,log="xy")
legend("topright",bty="n", pch=c(NA,NA,1,NA),lwd=2,
lty=c(NA,NA,NA,"solid"),
c(expression('KST4 feldspar',' ',
'Experiment','KP-CW equation')),col=c(NA,NA,"black","red"))
## print results
cat("Parameters from Least squares fit"," ")
cat("\nImax=",formatC(imax_fit,format="e",digits=2)," cts/s",
sep="    ","A=",round(A_fit,digits=2)," (s^-1)")
cat("\nrho=",round(rho_fit,digits=4),sep=" ",
"  bgd=",round(bgd_fit,digits=2)," cts/s")
```

```
  ## Parameters from Least squares fit
  ## Imax=    2.72e+04     cts/s    A=    7.07     (s^-1)
  ## rho= 0.0073   bgd= 47.05  cts/s
```

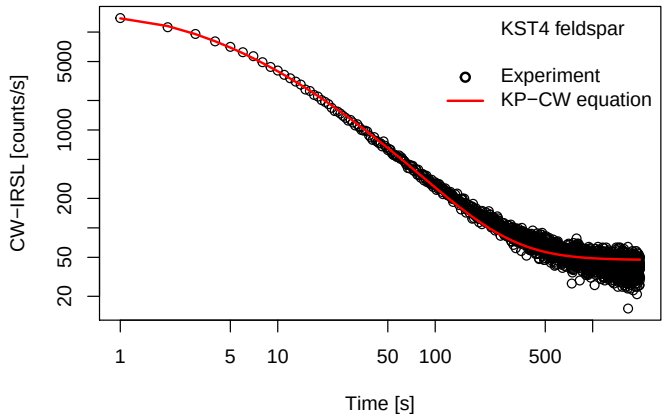

Fig. 6.7: Experimental CW-IRSL glow curves from freshly irradiated KST4 feldspar sample, fitted using the KP-CW analytical Eq.(**??**). For more details and examples, see Pagonis et al. [36].

Code 6.7: Kitis-Pagonis analytical equation for TL (KP-TL)

```
rm(list = ls(all=T))
options(warn=-1)
library("minpack.lm")
library(expint)
## Least squares fit to TL using the KP-TL eqt (TLT model)-
mydata <- read.table("ph300s0.asc")
t<-as.numeric(gsub(",", ".", gsub("\\.", "", mydata[,1])))
y<-as.numeric(gsub(",", ".", gsub("\\.", "", mydata[,3])))
y<-y/max(y)
mydata<-data.frame(t,y)
plot(t,y,xlab="Temperature [\u00B0C]",ylab="Normalized TL",
col="black",pch=1,xlim=c(200,450))
kb<-8.617e-5
z<-1.8
fit_data <-mydata
kB<-8.617E-5
En<-1.45
T<-t+273
```

```
fit <- minpack.lm::nlsLM(
 formula=y~imax* exp(-rho*( (log(1+z*s*kB*((T**2.0)/
abs(En))*exp(-En/(kB*T))*(1-2*kB*T/En)))**3.0))*
(En**2.0-6*(kB**2.0)*(T**2.0))*( (log(1+z*s*kB*((T**2.0)/
abs(En))*exp(-En/(kB*T))*(1-2*kB*T/En)))**2.0)/
(En*kB*s*(T**2)*z-2*(kB**2.0)*s*z*(T**3.0)+exp(En/(kB*T))*En),
  data = fit_data,
start = list(imax=1e12,s=1e11,rho=.009),upper=c(1e20,1e13,.02),
 lower=c(1e11,1e11,.008))
# Obtain parameters from best fit
imax_fit <- coef(fit)[1]
s_fit <- coef(fit)[2]
rho_fit <- coef(fit)[3]
En_fit <- En
## plot analytical solution
lines(
x = t,imax_fit* exp(-rho_fit*( (log(1+z*s_fit*kB*((T**2.0)/
abs(En_fit))*exp(-En_fit/(kB*T))*(1-2*kB*T/En_fit)))**3.0))*
(En_fit**2.0-6*(kB**2.0)*(T**2.0))*((log(1+z*s_fit*kB*((T**2.0)/
abs(En_fit))*exp(-En_fit/(kB*T))*(1-2*kB*T/En_fit)))**2.0)/
(En_fit*kB*s_fit*(T**2)*z-2*(kB**2.0)*s_fit*z*(T**3.0)+
exp(En_fit/(kB*T))*En_fit),col="red",lwd=2)
legend("topleft",bty="n", pch=c(NA,NA,1,NA),lwd=2,
lty=c(NA,NA,NA,"solid"),
c(expression('KST4 feldspar',' ',
'Experiment','KP-TL Eq.')),col=c(NA,NA,"black","red"))
## print results
cat("\nBest fit parameters"," ")
cat("\nImax=", formatC(imax_fit, format = "e", digits = 2),
" s=",formatC(s_fit, format = "e", digits = 2)," (s^-1) ")
cat("\nrho=",round(rho_fit,digits=4),sep=" ","E=",En_fit," eV")
```

```
  ##
  ## Best fit parameters
  ## Imax= 1.40e+13  s= 3.50e+12  (s^-1)
  ## rho= 0.0096 E= 1.45  eV
```

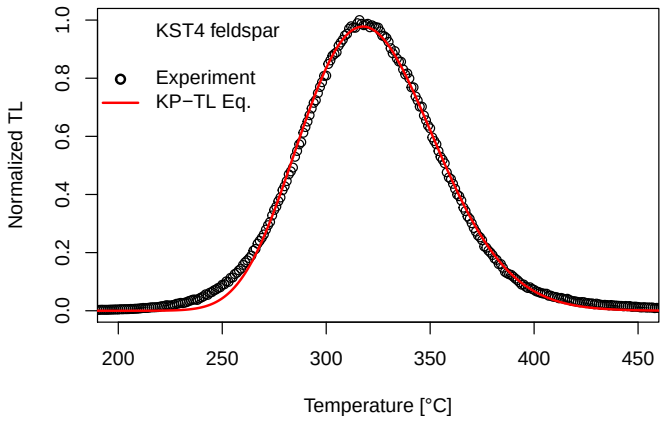

Fig. 6.8: Experimental TL glow curves from freshly irradiated KST4 feldspar sample, fitted using the KP-TL analytical Eq.(**??**). Note that the solid line does not describe the experimental data very accurately at low temperatures, due to the approximations involved in the KP equations. For more details and MC examples, see Pagonis et al. [36].

Function Name	**Description**
plot_RLumCarlo	Plots 'RLumCarlo' modeling results (the averaged signal or the number of remaining electrons), with modeling uncertainties.
run_MC_CW_IRSL_TUN	Simulation of CW-IRSL signals due to tunneling transitions from the excited state of the trap, into a recombination center (RC) for the EST model.
run_MC_ISO_TUN	Simulation of ITL signals due to tunneling from the excited state of the trapped charge, into the RC (EST).
run_MC_LM_OSL_TUN	Simulation of LM-IRSL signals due to tunneling from the excited state of the trapped charge, into the RC (EST).
run_MC_TL_TUN	Simulation of TL signals due to tunneling from the excited state of the trapped charge, into the RC (EST).

Table 6.1: Table of TUNneling functions available in the package *RLumCarlo*.

Process	Parameter	Description	Units	Typical values
TL with tunneling	E	Thermal activation energy	eV	0.5-3
	s	Effective frequency factor	1/s	1E8-1E16
	rho	Dimensionless density of recombination centers ρ'	1	1E-6-1E-2
	r_c	Critical distance (>0) for thermally/optically pretreated samples	1	0.1-0.8
	times	Sequence of time steps (heating rate 1 K/s)	s	0-700
	clusters	Number of MC runs	1	1E1-1E4
	N_e	Total number of electron traps available	1	2-1E5
	delta.r	Increments of the distance r'	1	0.01-0.1
CW-IRSL with tunneling	A	Effective optical excitation rate	1/s	1E-3-1
	rho	Dimensionless density of recombination centers ρ'	1	1E-6-1E-2
	times	Sequence of time steps	s	0-500
	clusters	Number of MC runs	1	1E1-1E4
	N_e	Total number of electron traps available	1	2-1E5
	r_c	Critical distance (>0) for thermally/optically pretreated sample	1	0.1-0.8
	delta.r	Increments of the distance r'	1	0.01-0.1
ISO with tunneling	E	Thermal activation energy	eV	0.5-3
	s	Effective frequency factor	1/s	1E8-1E16
	T	Temperature of the isothermal process	°C	20-300
	rho	Dimensionless density of recombination centers ρ'	1	1E-6-1E-2
	times	Sequence of time steps for simulation	s	0-1000
	clusters	Number of MC runs	1	1E1-1E4
	N_e	Total number of electron traps available	1	2-1E5
	r_c	Critical distance (>0) for thermally/optically pretreated sample	1	0.1-0.8
	delta.r	increments of the distance r'	1	0.01-0.1
LM-OSL with tunneling	A	Effective optical excitation rate	1/s	1E-3-1
	rho	Dimensionless density of recombination centers ρ'	1	1E-6-1E-2
	times	Sequence of time steps	s	0-3000
	clusters	Number of MC runs	1	1E1-1E4
	N_e	Total number of electron traps available	1	2-1E5
	r_c	Critical distance (>0) for thermally/optically pretreated sample	1	0.1-0.8
	delta.r	Increments of the distance r'	1	0.01-0.1

Table 6.2: Table of input parameters for TUN functions in *RLumCarlo*.

Code 6.8: Single plot MC simulations for tunneling CW-IRSL

```
##================================================##
## Example:Single Plot for  tunneling CW-IRSL
##================================================##
rm(list = ls(all=T))
suppressMessages(library("RLumCarlo"))
run_MC_CW_IRSL_TUN(
A = 5,
rho = 5e-3,
times = 0:500,
r_c = 0.5,
delta.r = 1e-2,
method = "par",
output = "signal"
) %>%
#Plot results of the MC simulation
plot_RLumCarlo(norm = F, legend = F)
legend("top",bty="n",legend=c("CW-IRSL","TUN function"))
```

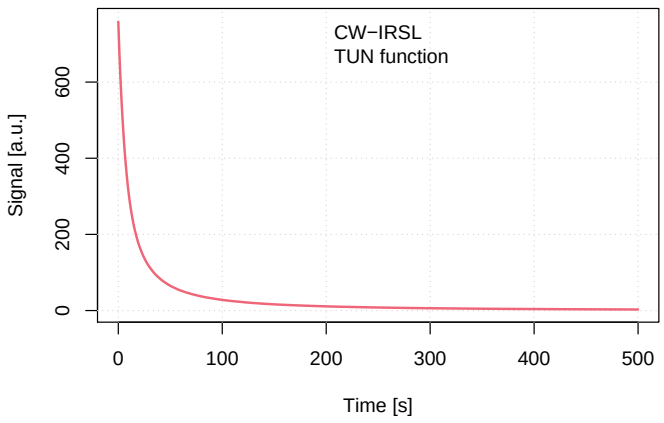

Fig. 6.9: Simulation of CW-IRSL signal using the function *run_MC_CW_IRSL_TUN* in the package *RLumCarlo*.

Code 6.9: Combining two plots in CW-IRSL experiment

```
rm(list = ls(all=T))
library(RLumCarlo)
times <- seq(0, 200)
run_MC_CW_IRSL_TUN(A = 3, rho = 0.003, times = times) %>%
  plot_RLumCarlo(norm = TRUE, lty=1,legend = TRUE)
run_MC_CW_IRSL_TUN(A = 6, rho = 0.003, times = times) %>%
  plot_RLumCarlo(norm = TRUE, lty=2,col="blue",add = TRUE)
legend("top",bty="n",legend=c(expression("CW-IRSL","TUN function",
 "A=3.0 s"^-1*" ","A=6.0 s"^-1*" ")),lty=c(NA,NA,1:2))
```

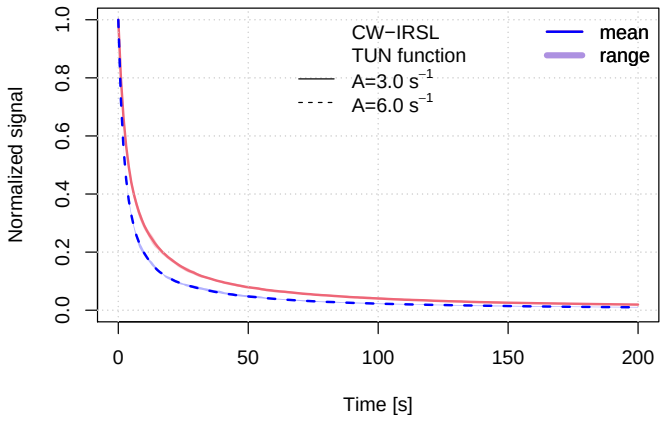

Fig. 6.10: Simulation of two CW-IRSL signals using the function *run_MC_CW_IRSL_TUN* in the package *RLumCarlo*, for two different excitation rates A.

Code 6.10: MC for tunneling ITL: remaining electrons

```
# MC Model for remaining charges and ITL signal
rm(list = ls(all=T))
library(RLumCarlo)
par(mfrow=c(1,2))
results <- run_MC_ISO_TUN(
  E = 1.0,
  s = 1e12,
  T = 250,
  rho = 0.01,
  clusters=100,
  times = seq(0, 200),
  output = "remaining_e"
) %T>%
  plot_RLumCarlo(
    legend = FALSE,
    ylab = "Remaining electrons"  )
legend("topright",bty="n",legend=c("(a)"," ","n(t)",
"TUN function"))

results <- run_MC_ISO_TUN(
  E = 1.2,
  s = 1e12,
  T = 250,
  rho = 0.01,
  times = seq(0, 200)
) %T>%
  plot_RLumCarlo(norm = FALSE, legend = FALSE)
legend("topright",bty="n",legend=c("(b)"," ","ITL signal"))
```

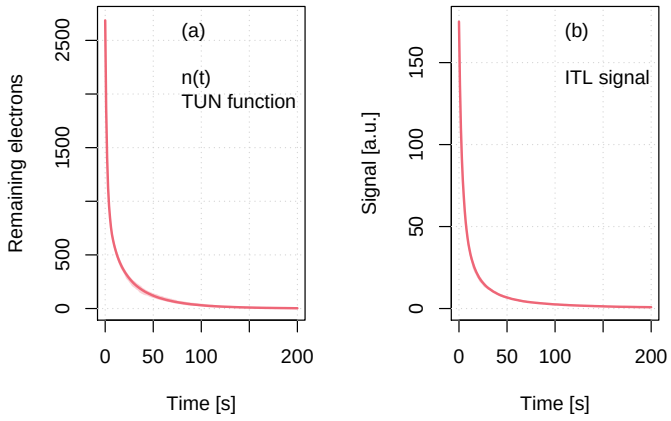

Fig. 6.11: Simulation of (a) remaining electrons $n(t)$ during an ITL experiment, and (b) of the resulting ITL signal, using the function *run_MC_ISO_TUN* in the package *RLumCarlo*.

Code 6.11: Single plot MC simulations for tunneling LM-OSL

```
##=================================================##
## Example 1: MC simulations of tunneling LM_IRSL
##=================================================##
rm(list = ls(all=T))
library(RLumCarlo)
run_MC_LM_OSL_TUN(
A =3.0,
rho = 1e-2,
times = 0:200,
clusters = 100,
N_e = 20,
r_c = 0.001,
delta.r = 1e-1,
method = "par",
output = "signal"
) %>%
```

```
# Plot results of the MC simulation
plot_RLumCarlo(norm = F,legend=F)
legend("topright",bty="n",legend=c("LM-IRSL signal",
"TUN function",""))
```

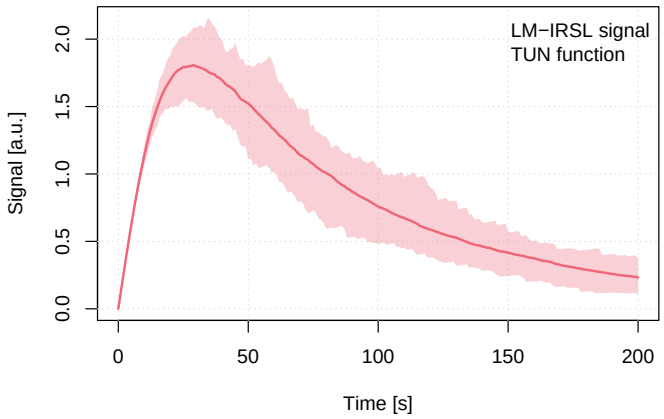

Fig. 6.12: Simulation of LM-IRSL signal using the function *run_MC_LM_OSL_TUN* in the package *RLumCarlo*.

Code 6.12: Simulation of TL from pretreated sample

```
##============================================
## Simulate TL from pretreated sample
##============================================
rm(list = ls(all=T))
library(RLumCarlo)
s <- 3.5e12
rho <- 0.015
E <- 1.45
r_c <- c(0,0.8,1.0)
times <- seq(100, 450) # time = temperature
results <- lapply(r_c, function(x) {
  run_MC_TL_TUN(
```

```
    s = s,
    E = E,
    rho = rho,
    r_c = x,
    times = times
  )})%>%
plot_RLumCarlo(norm = FALSE, legend = TRUE)
legend("topleft",bty="n",legend=c(expression('Critical Radius',
' ','r'[c]*'=0, 0.8, 1.0')) )
```

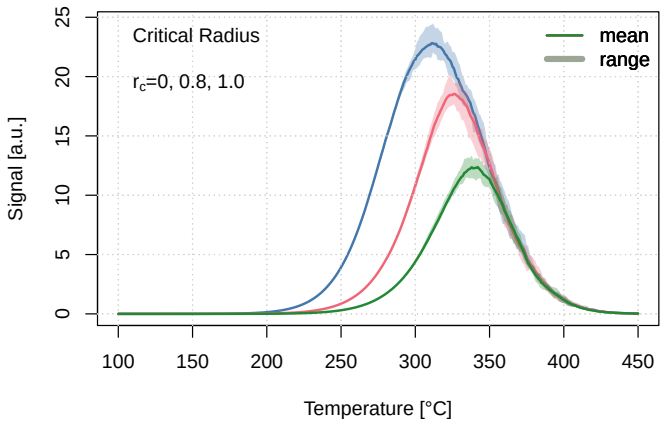

Fig. 6.13: Simulation of TL signal using the function *run_MC_TL_TUN* in the package *RLumCarlo*. The sample has been thermally treated after irradiation, and the effect of the thermal treatment is described by the variable critical radius parameter $r_c' = 0,\ 0.8,\ 1.0$.