# Chapter 9
# MONTE CARLO SIMULATIONS OF LOCALIZED TRANSITIONS

**Abstract** In this chapter we discuss fixed time interval MC simulations of luminescence signals produced by localized transitions in the TLT and LT models. Examples of R codes are provided for TL, CW-IRSL and LM-IRSL signals in the excited state tunneling (EST) model, and the MC results are compared with the analytical Kitis-Pagonis equations (KP-CW, KP-TL). The R codes also provide estimates for the stochastic coefficients of variation CV% for a variety of processes.This chapter concludes with a MC simulation for the LT model.

---

**Code 9.1: Vectorized MC code for tunneling TL transitions (TLT model)**

```
# Vectorized MC code for tunneling transitions (TLT model)
# Original Mathematica program by Vasilis Pagonis
# R version written by Johannes Friedrich, 2018
rm(list = ls(all=T))
rho <- 5e-3
En<-1.43
s<-3.5e12
kB<-8.617e-5
deltat <- 1
times <- seq(0, 500, deltat)
# In this example time =temperature, i.e. a heating rate=1 K/s
r <- seq(0, 2, 0.1)
clusters <- 20
n0<-100
signal<-array(0,dim=c(length(times),
  ncol = length(r), clusters))
# Run MC simulation
```

```
system.time(invisible(for(c in 1:clusters)
{
  for(k in 1:length(r)){
    n <- n0
    for (t in 1:length(times)){
      P <- s*exp(-En/(kB*(t+273)))*exp(-rho^(-1/3) * r[k])
      vec<-rep(runif(n))
      n<-length(vec[vec>P*deltat])
      signal[t,k,c] <- n * P * 3 * r[k]^2 * exp(-r[k]^3) }}})
)
par(mfrow=c(1,2))
# plot an example : the result from the first cluster
matplot(signal[,,1],type = "l",lty="solid",
ylab = "Partial TL glow curves for constant(r')",
ylim=c(0,3.2),xlab=expression("Temperature ["^"o"*"C]"),lwd = 2)
legend("topleft",bty="n",legend=c("(a)"," ","Partial TL",
"glow curves"))
#add the signals from all clusters
sum_signal <- sapply(1:clusters, function(y){
  vapply(1:length(times), function(x){
    sum(signal[x,,y])
  }, FUN.VALUE = 1)  })
# add the signals from all r values
TL <- rowMeans(sum_signal)
# plot and normalize the TL signal
plot(  x = times, y = TL/max(TL),type = "l", lwd = 3,
ylim=c(0,1.6), xlab=expression("Temperature ["^"o"*"C]"),
ylab="Average TL signal")
legend("topleft",bty="n",legend=c("(b)"," ",
"Sum of partial TL","glow curves"))
## plot analytical solution Kitis-Pagonis
z<-1.8
T<-times+273
TLanalyt<-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)
lines(times,TLanalyt/max(TLanalyt),lty="dashed",col="red",
lwd=3)
```

```
##    user  system elapsed
##    1.51    0.00    1.53
```

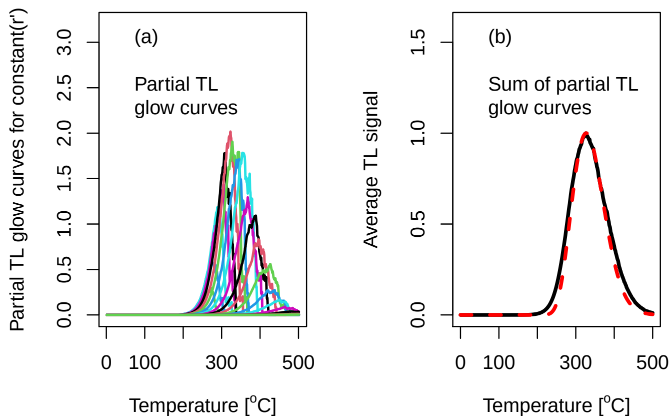


**Fig. 9.1:** MC simulation of TL signals in the TLT model, for the parameters $\rho' = 5 \times 10^{-3}$, $M$ =20 MC runs, $n_0 = 100$ initially trapped electrons, $E = 1.43$ eV, $s = 3.5 \times 10^{13}$ s$^{-1}$. (a) Example of partial TL glow curves evaluated for each distance $r'$ . (b) The sum of the partial TL glow curves from (a), normalized to its maximum. The dashed line in (b) represents the approximate analytical KP-TL equation from Chapter 6, also normalized to its maximum value.

---

**Code 9.2: Vectorized MC code for tunneling CW-IRSL transitions (TLT model)**

```
# Vectorized MC code for tunneling transitions (TLT model)
# Original Mathematica program by Vasilis Pagonis
# R version written by Johannes Friedrich, 2018
rm(list = ls(all=T))
rho <- 5e-3
A<-2
deltat <- 1
times <- seq(1, 400, deltat)
# In this example time =temperature, i.e. a heating rate=1 K/s
r <- seq(0, 2.2, 0.1)
clusters <- 10
n0<-500
signal<-array(0,dim=c(length(times),
```

```
                      ncol = length(r), clusters))
# Run MC simulation
system.time(invisible(for(c in 1:clusters)
{
  for(k in 1:length(r)){
    n <- n0
    for (t in 1:length(times)){
      P <- A*exp(-rho^(-1/3) * r[k])
      vec<-rep(runif(n))
      n<-length(vec[vec>P*deltat])
      signal[t,k,c] <- n * P * 3 * r[k]^2 * exp(-r[k]^3) }}})
)
par(mfrow=c(1,2))
# plot an example : the result from the first cluster
matplot(signal[,,1],type = "l",lty="solid",
ylab = "Partial CW-IRSL curves",
ylim=c(0,14),xlab=expression("Time [s]"),lwd = 2)
legend("topleft",bty="n",legend=c("(a)"," ","Partial CW-IRSL",
" curves"))
#add the signals from all clusters
sum_signal <- sapply(1:clusters, function(y){
  vapply(1:length(times), function(x){
    sum(signal[x,,y])
  }, FUN.VALUE = 1)  })
# add the signals from all r values
TL <- rowMeans(sum_signal)
# plot and normalize the TL signal
plot(  x = times, y = TL/max(TL),type = "l", lwd = 3,
ylim=c(0,1.4), xlab=expression("Time [s]"),
       ylab="Average signal")
legend("topleft",bty="n",legend=c("(b)"," ",
"Sum of partial","CW-IRSL curves"))
## plot analytical solution Kitis-Pagonis
z<-1.8
CWanalyt<-exp(-rho*( (log(1+z*A*times))**3.0))*
  ( (log(1+z*A*times))**2.0)/(1+z*A*times)
lines(times,CWanalyt/max(CWanalyt),lty="dashed",col="red",lwd=3)
```

```
##    user  system elapsed
##    1.35    0.00    1.36
```

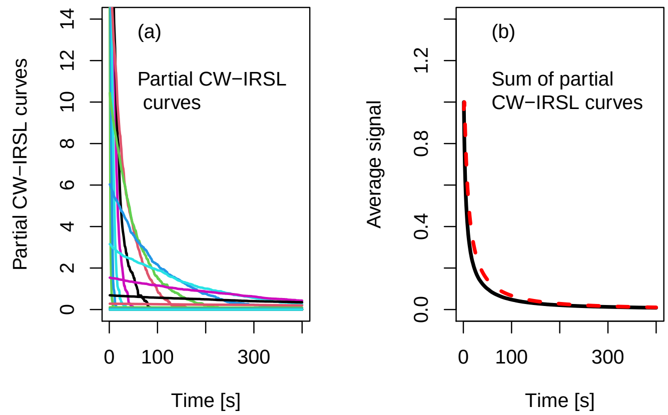


**Fig. 9.2:** MC simulation of CW-IRSL signals in the TLT model, for the parameters $\rho' = 5 \times 10^{-3}$, $M$ =10 MC runs, $n_0 = 500$ initially trapped electrons, and IR excitation rate $A = 2$ s$^{-1}$. (a) Example of partial CW-IRSL curves evaluated for each distance $r'$ . (b) The sum of the partial CW-IRSL curves from (a), normalized to its maximum. The dashed line in (b) represents the approximate analytical KP-CW equation from Chapter 6, also normalized to its maximum value (kitis and Pagonis [13]).

---

**Code 9.3: Vectorized MC code for tunneling LM-IRSL transitions (TLT model)**

```
# Vectorized MC code for tunneling transitions (TLT model)
# Original Mathematica program by Vasilis Pagonis
# R version written by Johannes Friedrich, 2018
rm(list = ls(all=T))
rho <- 5e-3
A<-2                # IR exciation rate in s^-1
deltat <- 1
times <- seq(0, 400, deltat)
r <- seq(0, 2.2, 0.1)
clusters <- 10
n0<-500
signal<-array(0,dim=c(length(times),
ncol = length(r), clusters))
```

```
# Run MC simulation
system.time(invisible(for(c in 1:clusters)
{  for(k in 1:length(r)){
    n <- n0
    for (t in 1:length(times)){
      P <- A*t/max(times)*exp(-rho^(-1/3) * r[k])
      vec<-rep(runif(n))
      n<-length(vec[vec>P*deltat])
 signal[t,k,c] <- n * P * 3 * r[k]^2 * exp(-r[k]^3) }}}))
par(mfrow=c(1,2))
# plot an example : the result from the first cluster
matplot(signal[,,1],type = "l",lty="solid",
ylab = "Partial LM-IRSL curves",
ylim=c(0,7),xlab=expression("Time [s]"),lwd = 2)
legend("topleft",bty="n",legend=c("(a)"," ","Partial LM-IRSL",
" curves"))
#add the signals from all clusters
sum_signal <- sapply(1:clusters, function(y){
  vapply(1:length(times), function(x){
    sum(signal[x,,y])
  }, FUN.VALUE = 1)  })
# add the signals from all r values
TL <- rowMeans(sum_signal)
# plot and normalize the TL signal
plot(  x = times, y = TL/max(TL),type = "l", lwd = 3,
ylim=c(0,1.6), xlab=expression("Time [s]"),
ylab="Average LM-IRSL signal")
legend("topleft",bty="n",legend=c("(b)"," ",
"Sum of partial","LM-IRSL curves"))
## plot analytical solution Kitis-Pagonis
z<-1.8
LManalyt<-exp(-rho*( (log(1+z*A*times^2/(2*max(times))))**3.0))*
times*((log(1+z*A*times^2/(2*max(times))))**2.0)/(1+z*A*times^2/
(2*max(times)))
lines(times,LManalyt/max(LManalyt),lty="dashed",col="red",lwd=3)
```

```
##    user  system elapsed
##    1.75    0.00    1.74
```

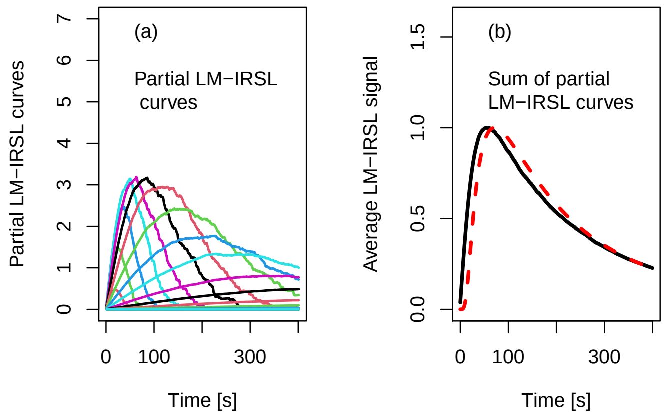


**Fig. 9.3:** MC simulation of LM-IRSL signals in the TLT model, for the parameters $\rho' = 5 \times 10^{-3}$, $M$ =10 MC runs, $n_0 = 100$ initially trapped electrons, $A = 2$ s$^{-1}$. (a) Example of partial LM-IRSL curves evaluated for each distance $r'$ ; (b) The sum of the partial LM-IRSL curves from (a), normalized to its maximum. The dashed line in (b) represents the approximate analytical equation by Kitis and Pagonis [13], also normalized to its maximum value.

---

**Code 9.4: Vectorized MC code for TL in localized TL transitions (LT model)**

```
# Vectorized MC code for localized TL transitions (LT model)
rm(list = ls(all=T))
options(warn=-1)
library(matrixStats)
library(lamW)
mcruns<-100
n0<-500
s<-1e13
E<-1
kb<-8.617e-5
r<-1e2
tmax<-120
deltat<-1
times<-seq(1,tmax,deltat)  #heating rate=1 K/s
nMatrix<-TLMatrix<-matrix(NA,nrow=length(times),ncol=mcruns)
nMC<-TL<-rep(NA,length(times))
```

```
system.time(
  for (j in 1:mcruns){
    n<-n0
    for (t in 1:length(times)){
      vec<-rep(runif(n))
      P<-s*exp(-E/(kb*(t+273)))*n/(r+n)*deltat
      n<-length(vec[vec>P])
      nMC[t]<-n
      TL[t]<-nMC[t]*P}
    nMatrix[,j]<-nMC
    TLMatrix[,j]<-TL })
#Find average avgn, average TL signal,CV[%]
avgn<-rowMeans(nMatrix)
avgTL<-rowMeans(TLMatrix)
sd<-rowSds(TLMatrix)
cv<-100*sd/avgTL
# plots
par(mfrow=c(1,2))
pch<-c(NA,NA,1,NA)
lty<-c(NA,NA,NA,"solid")
col<-c(NA,NA,"black","red")
plot(times,avgTL,ylab="TL",
     xlab=expression("Temperature ["^"o"*"C]"))
# plot analytical solution
k<-function(u) {integrate(function(p){exp(-E/(kb*p))},
300,u)[[1]]}
y1<-lapply(times+273,k)
x<-unlist(273+times)
y<-unlist(y1)
zTL<-(r/n0)-log(n0/r)+(s*y)
lines(x-273,r*s*exp(-E/(kb*x))/(lambertW0(exp(zTL))
+lambertW0(exp(zTL))^2),type="l",col="red")
legend("topleft",bty="n",c("(a)  TL"," ","MC",
                           "Analyt."),pch=pch,lty=lty,col=col)
plot(times,cv,ylab="(b)  CV[%]",ylim=c(0,120),
     xlab=expression("Temperature ["^"o"*"C]"))
legend("topleft",bty="n",c("CV[%]"," ","MC"),
pch=pch,lty=lty,col=col)
```

```
##    user  system elapsed
##    0.24    0.00    0.23
```

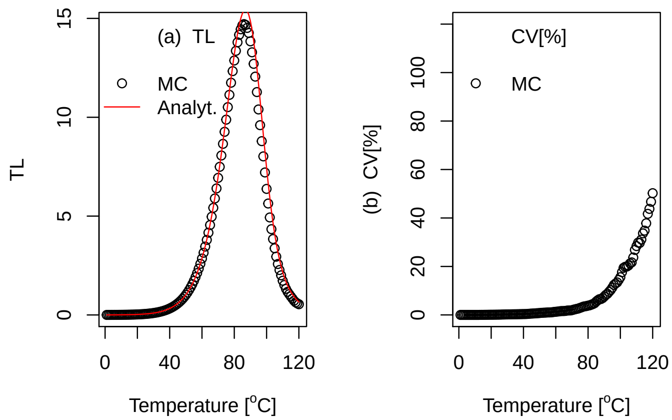


**Fig. 9.4:** (a) MC simulation of TL signals in the LT model, for the parameters $r = 10^2$, $M$ =100 MC runs, $n_0 = 500$, $E = 1$ eV, $s = 10^{13}$ s$^{-1}$. The solid line is the analytical solution by Kitis and Pagonis [13]. (b) The corresponding coefficient of variation *CV[%]*.

retrapping ratio $r = 10^2,\ 10^3,\ 10^4$.

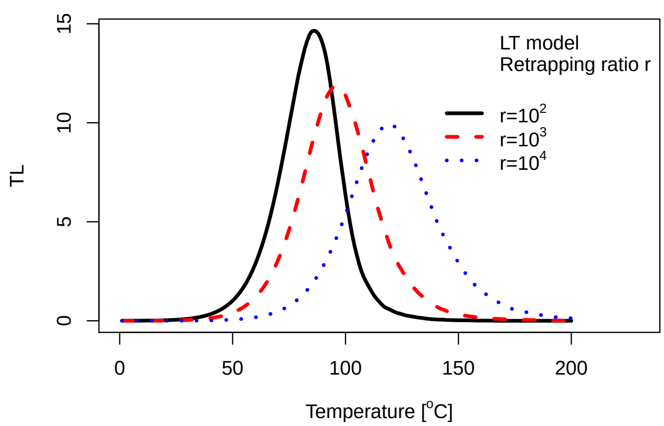


**Fig. 9.5:** MC simulation of TL signals in the LT model, for three different values of the retrapping ratio $r = 10^2,\ 10^3,\ 10^4$ cm$^{-3}$.