Chapter 6
ITL SIGNALS: MODELS

Code 6.1: Plots of the KV-ITL equation for isothermal analysis

```
#plot KV-ITL equation for delocalized processes
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
from scipy.special import wrightomega
def KVITL(R,pITL):
    c=(n0/N)*(1-R)/R
    zITL=(1/c)-np.log(c)+(pITL*n0/(c*N*R))*t
    lam=wrightomega(zITL)
    ITL=(N*R/((1-R)**2.0))*pITL/(lam+lam**2)
    plt.plot(t,ITL/max(ITL),symbs[j-1],\
    linewidth=2,label=labls[j-1]);
N,     s,   E,  n0 ,  kB=\
1e10, 1e12, 1,  1e9,  8.617e-5
t = np.linspace(0, 100, 100)
Tiso=110+273.15
pITL=s*np.exp(-E/(kB*Tiso))
Rs=[0.01,.1,.95]
symbs=['+','^','o']
labls=['R='+str(x) for x in Rs]
plt.subplot(1,2,1);
for j in range(1,4):
    KVITL(Rs[j-1],pITL)
leg = plt.legend()
```

```
leg.get_frame().set_linewidth(0.0)
plt.xlabel('Time [s]');
plt.text(45,.7,'KV-ITL equation');
plt.ylabel('ITL [a.u]');
plt.title('(a)');
plt.subplot(1,2,2);
Tiso=[110,120,130]
pITL=[s*np.exp(-E/(kB*(x+273.15))) for x in Tiso]
labls=['T='+str(x)+r'$^o$C' for x in Tiso]
for j in range(1,4):
    KVITL(0.1,pITL[j-1])
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.xlabel('Time [s]');
plt.ylabel('ITL [a.u]');
plt.title('(b)');
plt.tight_layout()
plt.show()
```

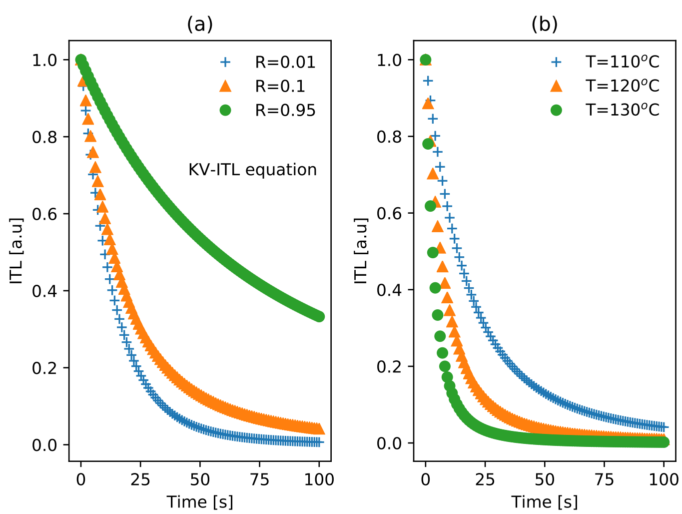

Fig. 6.1: (a) Plot of the KV-ITL Eq.(**??**) for 3 different values of the retrapping ratio R =0.01, 0.1, 0.95. As the value of R decreases, the shape of the ITL curve tends towards the exponential decay curve characteristic of first order kinetics. (b) Plot of the same KV-ITL Eq.(**??**) for different temperatures during the ITL experiment. All signals are normalized to their maximum value.

Code 6.2: Plots of the MOK-ITL and GOK-ITL equation for isothermal analysis

```
#plot MOK-ITL and GOK-ITL equation for delocalized processes
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
def MOKITL( Nd,n):
    alpha, g =n10/(n10+Nd), 1/(N1+Nd)
    Ft=np.exp(g*Nd*lamda*t)
    ITL=g*(Nd**2.0)*alpha*lamda*Ft/((Ft-alpha)**2.0)
    plt.plot(t,np.log(ITL),symbs[n],\
    label=r'$\alpha$='+f'{alpha:.2f}',linewidth=2);
    return ITL
def GOKITL(b,n):
    c=(n10/N1)**(b-1)
    ITL=n10*c*lamda*(1+lamda*c*(b-1)*t)**(-b/(b-1))
    plt.plot(t,np.log(ITL),symbs[n],\
    label=labls[n],linewidth=2);
n10, N1=1e10, 1e10
t = np.linspace(0, 130,20)
N1ds=[1e12,1e10,1e8]
symbs=['+-','^-','o-']
plt.subplot(1,2,1);
lamda=0.05
for j in range(1,4):
    MOKITL(N1ds[j-1],j-1);
plt.text(30,15,"MOK-ITL");
plt.xlabel('Time [s]');
plt.ylabel('ln(ITL)');
plt.title('(a)');
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.subplot(1,2,2);
lamda=0.1
Rs=[1.01,1.5,1.9]
labls=['b='+str(x) for x in Rs]
for j in range(1,4):
    GOKITL(Rs[j-1],j-1);
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.xlabel('Time [s]');
plt.text(30,11,'GOK-ITL');
```

```
plt.ylabel('ln(ITL)');
plt.title('(b)');
plt.tight_layout()
plt.show()
```

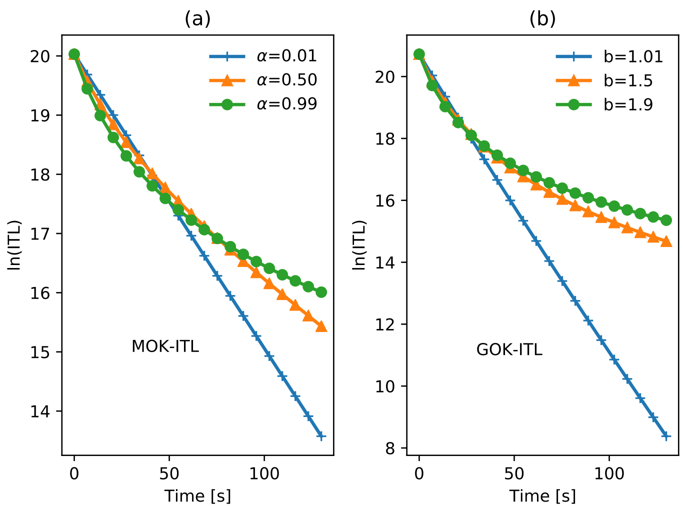

Fig. 6.2: Plots of (a) MOK-ITL Eq.(**??**) for three values of the mixed order kinetics parameter α =0.01, 0.5, 0.99; (b) Plots of GOK-ITL Eq.(**??**) for b =1.01, 1.5, 1.99. All signals are normalized to their maximum value.

Code 6.3: Plots of the KP-ITL equation for isothermal analysis

```
#plot KP-ITL equation for localized processes
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
def KPITL(rho,pITL):
    ITL=np.exp (-rho*(np.log(1 + z*pITL*t))\
    ** 3.0)*(np.log(1+z*pITL*t)**2.0)/(1+t*pITL)
    plt.plot(t,ITL/max(ITL),symbs[j-1], linewidth=2,
    label=labls[j-1]);
s,   E,   kB,       z=\
```

```
1e12,  1,   8.617e-5, 1.8
t = np.linspace(1, 100, 100)
Tiso=170+273.15
pITL=s*np.exp(-E/(kB*Tiso))
rhos=[1e-3,5e-3,1e-2]
labls=[r'$\rho$'+"'="+str(x) for x in rhos]
symbs=['+-','^-','o-']
plt.subplot(1,2,1);
for j in range(1,4):
    KPITL(rhos[j-1],pITL)
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.xlabel('Time [s]');
plt.text(45,.6,'KP-ITL equation');
plt.ylabel('ITL [a.u]');
plt.title('(a)');
plt.subplot(1,2,2);
Tiso=[170,180,190]
pITL=[s*np.exp(-E/(kB*(x+273.15))) for x in Tiso]
labls=['T='+str(x)+r'$^o$C' for x in Tiso]
for j in range(1,4):
    KPITL(0.01,pITL[j-1])
leg = plt.legend()
leg.get_frame().set_linewidth(0.0)
plt.xlabel('Time [s]');
plt.ylabel('ITL [a.u]');
plt.title('(b)');
plt.tight_layout()
plt.show()
```

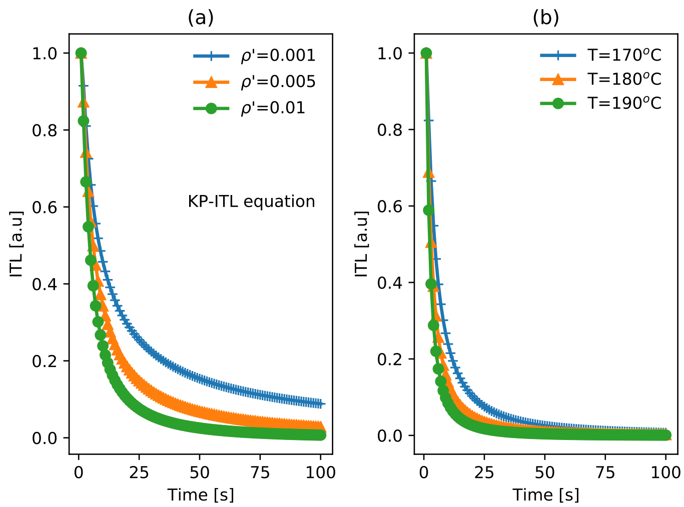

Fig. 6.3: (a) Plot of the KP-ITL Eq.(**??**) for 3 different values of the dimensionless density $\rho' = 0.001, 0.005, 0.01$. As the value of ρ' increases, the ITL curve decreases faster; (b) Plot of the KP-ITL equation for different temperatures during the ITL experiment. All curves are normalized to their maximum value.