# Low-Temperature Measurements Code

Jack Symonds

Measuring $\tau_c$:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import scipy.signal as sig

class temptime:
    def __init__(self, filename):
        df = pd.read_csv(filename)
        self.times = df['Time (s)']
        self.temps = df['Temperature (K)']

    def plot(self, ax, distance):
        peaki = sig.find_peaks(self.temps, distance=distance)[0]
        tc_meas = np.array([])
        enumpeaki = np.array(list(enumerate(peaki)))
        for it, i in enumpeaki[1:-1]:
            ax.axvline(self.times[i], color='lightgray', linestyle='--')
            tc = self.times[enumpeaki[it,1]] - self.times[enumpeaki[it-1,1]]
            tc_meas = np.append(tc_meas, tc)

        print(tc_meas)
        tc = np.mean(tc_meas)

        ax.plot(self.times, self.temps, marker='o', label=r'period: $\tau _c={:.2f}$'.format(tc))
        plt.legend()



fig, ax = plt.subplots()
ax.set_xlabel('Time (s)')
ax.set_ylabel('Temperature (K)')
ax.set_title('Temperature-Time')

#tc = temptime('tc.csv')
#tc.plot(ax, 10)
pi = temptime('tc.csv')
pi.plot(ax, 10)
fig.savefig('images/tauc.png')

plt.show()
```

Varying $P$ Values:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import scipy.signal as sig

class temptime:
    def __init__(self, filename):
        self.df = pd.read_csv('./pid/'+filename)

    def plot(self, ax, color,P,I, tstab,tset):
        times = self.df['Time (s)']
        temps = self.df['Temperature (K)']


        #ax.fill_between(times, temps, avgT, color='lightgrey')
        tstabi = np.argmin(times-tstab)
        ax.axvline(tstab,color=color,linestyle='--',linewidth=0.75)
        ntemps = temps[tstabi:]
        rms = np.sqrt((1/(ntemps.size-1))*np.sum((ntemps-np.mean(ntemps))**2))
        label = r'P = %.2f' % P
        ax.plot(times, temps, marker='', color=color,label=label,linewidth=0.8)
        row = [r'$P=%.2f$' % P, r'$I=%.2f$'%I, r'$t_{stab}\approx %.2f$ s'%round(tstab,2), r'$\sigma _T=%.4f$' %
        cellText.append(row)


fig, ax = plt.subplots()
plt.subplots_adjust(bottom=0.16)
ax.grid(True)
ax.set_xlabel('Time (s)')
ax.xaxis.tick_top()
ax.xaxis.set_label_position('top')
ax.set_ylabel('Temperature (K)')
cellText=[]

#pi = temptime('vp2.csv')
#pi.plot(ax, 'black',1.6, 4.4, 10)




vp2 = temptime('p3-2.csv')
vp3 = temptime('p0-8.csv')
vp4 = temptime('p2.csv')
pi = temptime('p1-6.csv')

vp4.plot(ax, 'green',2, 4.4, 40, 10)
vp3.plot(ax, 'orange',0.8, 4.4, 30, 10)
vp2.plot(ax, 'blue',3.2, 4.4, 50, 10)
pi.plot(ax, 'black',1.6, 4.4, 25, 10)

plt.legend()
ax.table(cellText=cellText, cellLoc='left')
fig.savefig('images/ptrace.png')
plt.show()

fig2, ax2 = plt.subplots()
ax2.grid(True)
ax2.set_xlabel('Time (s)')
ax2.xaxis.tick_top()
ax2.xaxis.set_label_position('top')
ax2.set_ylabel('Temperature (K)')
cellText=[]

pi2 = temptime('50Kpi.csv')
pi2.plot(ax2, 'blue', 19.2,4.8,60,50)
ax2.table(cellText=cellText, cellLoc='left')
fig2.savefig('images/50Kpi.png')
plt.show()
```

Varying $I$ Values:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

boltz = 1.380649*10**(-23)
room_temp = 293
elem = 1.60217662*10**(-19)

class IV:
    def __init__(self, filename):
        path = './iv/'
        self.df = pd.read_csv(path+filename, sep='\t').sort_values(by='V')
        self.v = self.df['V'] - 0.001*self.df['Va']
        self.i = self.df['I']

    def plot(self, ax, label):
        label = '{} K'.format(label)
        ax.scatter(self.v, self.i, label=label)
        ax.errorbar(self.v, self.i, xerr=0.005, fmt='none', label='_nolabel_')

        #lx = self.v.min()
        #mx = self.v.max()
        #m, b = np.polyfit(self.v, self.i, 1)
        #x = np.linspace(lx*0.95, mx, 10)
        #label = r'$I=%1.2f \ V%1.2f\qquad V_\mathrm{int}=%1.3f$' % (m, b, vint)
        #ax.plot(x, m*x + b, linestyle='--')
        plt.legend()

fig, ax = plt.subplots()
ax.grid(True)
ax.set_title(r'$I-V$ Curve')
ax.set_xlabel('Voltage (V)')
ax.set_ylabel(r'Current ($\mu$A)')
ax.axhline(0, color='black', linewidth=0.75)

#iv10 = IV('IV10.tsv')
#iv10.plot(ax, 10)
iv50 = IV('IV50.tsv')
iv100 = IV('IV100.tsv')
iv150 = IV('IV150.tsv')
iv200 = IV('IV200.tsv')
iv250 = IV('IV250.tsv')
ivrt = IV('IVrt.tsv')

iv250.plot(ax, 250)
iv50.plot(ax, 50)
iv100.plot(ax, 100)
iv150.plot(ax, 150)
iv200.plot(ax, 200)
ivrt.plot(ax, 294.5)

fig.savefig('images/iv.png')
plt.show()
```

1 V Lock-In Output Superconductor Measurments:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

temps = []
rows = []
class voltemp:
    def __init__(self, filename):
        df = pd.read_csv(filename)
        self.times = np.array(df['Time (s)'])
        self.temps = np.array(df['Temperature (K)'])
        self.amps = np.array(1050*1000*df['Amplitude (V)'])

    def plot(self, ax, label):
        ax.scatter(self.temps, self.amps, label=label, marker='.')
        r1 = np.max(self.amps)
        r0 = np.min(self.amps)
        ticks = ax.get_yticks()
        nticks = [r0,0.1*r1,0.5*r1,0.9*r1,r1]
        ax.axhline(0.5*r1, color='black', linewidth=0.5)
        if np.array_equal(ticks,np.array(nticks)) != True:
            ticksn = np.append(ticks, nticks)
            ax.set_yticks(ticksn)
            for tick in ticksn:
                ax.axhline(tick, color='grey', linewidth=0.75)

        t10i = np.argmin(abs(self.amps - 0.1*r1))
        t10 = self.temps[t10i]
        ax.axvline(t10, linestyle = '--', linewidth=0.5,color = 'green')

        t50i = np.argmin(abs(self.amps - 0.5*r1))
        t50 = self.temps[t50i]
        ax.axvline(t50, linestyle = '--', linewidth=0.5,color='yellow')

        t90i = np.argmin(abs(self.amps - 0.9*r1))
        t90 = self.temps[t90i]
        ax.axvline(t90, linestyle = '--',linewidth=0.5,color='red')

        global temps
        temps.append([t10,t50,t90])
        global rows
        rows.append([label,r'$T_{10}=%.3f$'%t10,r'$T_{50}=%.3f$'%t50,r'$T_{90}=%.3f$'%t90,r'$T_{90}-T_{10}=%.3f$


fig, ax = plt.subplots()
plt.subplots_adjust(bottom=0.2)
ax.set_yticks([])
ax.set_xlim(9.8,10.2)
ax.set_title('Ramp (1 V Lock-In Output)')
ax.set_xlabel('Temperature (K)')
ax.xaxis.tick_top()
ax.xaxis.set_label_position('top')
ax.set_ylabel(r'Resistance ($\mu \Omega$)')

rampu = voltemp('rampu.csv')
rampd = voltemp('rampd.csv')
srampu = voltemp('srampu2.csv')
srampd = voltemp('srampd.csv')

rampu.plot(ax, 'fast and up')
rampd.plot(ax, 'fast and down')
srampu.plot(ax, 'slow and up')
srampd.plot(ax, 'slow and down')


#vrampu = voltemp('5Vsrampu.csv')
#vrampd = voltemp('5Vsrampd.csv')
#
#vrampu.plot(ax, '5 V up')
```

```
#vrampd.plot(ax, '5 V down')
#temps = np.array(temps)
#avgT10 = np.mean(temps[:,0])
#avgT50 = np.mean(temps[:,1])
#avgT90 = np.mean(temps[:,2])
#ax.axvline(avgT10,color='green')
#ax.axvline(avgT50,color='yellow')
#ax.axvline(avgT90,color='red')
#rows.append([['',r'$\langle T_{10}\range=%.3f$'%avgT10,r'$\langle T_{50}\range=%.3f$'%avgT50,r'$\langle T_{90
ax.table(cellText=rows,cellLoc='left')
plt.legend()
fig.savefig('./images/superc.png')
plt.show()
```

5 V Lock-In Output Superconductor Measurments:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

temps = []
rows=[]
class voltemp:
    def __init__(self, filename):
        df = pd.read_csv(filename)
        self.times = np.array(df['Time (s)'])
        self.temps = np.array(df['Temperature (K)'])
        self.amps = np.array(1050*df['Amplitude (V)'])

    def plot(self, ax, label):
        ax.scatter(self.temps, self.amps, label=label, marker='.')
        r1 = np.max(self.amps)
        r0 = np.min(self.amps)

        t10i = np.argmin(abs(self.amps - 0.1*r1))
        t10 = self.temps[t10i]
        ax.axvline(t10, linestyle = '--', linewidth=0.75,color = 'green')

        t50i = np.argmin(abs(self.amps - 0.5*r1))
        t50 = self.temps[t50i]
        ax.axvline(t50, linestyle = '--', linewidth=0.75,color='yellow')

        t90i = np.argmin(abs(self.amps - 0.9*r1))
        t90 = self.temps[t90i]
        ax.axvline(t90, linestyle = '--',linewidth=0.75,color='red')

        global temps
        temps.append([t10,t50,t90])
        global rows
        rows.append([label,r'$T_{10}=%.3f$'%t10,r'$T_{50}=%.3f$'%t50,r'$T_{90}=%.3f$'%t90,r'$T_{90}-T_{10}=%.3f$


fig, ax = plt.subplots()
plt.subplots_adjust(bottom=0.25)
ticks = [0,0.039,0.202]
ax.set_yticks(ticks)
for tick in ticks: ax.axhline(tick,color='grey',linewidth=0.75)
ax.set_xlim(9.8,10.2)
ax.set_title('Ramp (5 V Lock-In Output)')
ax.set_xlabel('Temperature (K)')
ax.xaxis.tick_top()
ax.xaxis.set_label_position('top')
ax.set_ylabel(r'Resistance (m$\Omega$)')

rampu = voltemp('rampu.csv')
rampd = voltemp('rampd.csv')
srampu = voltemp('srampu2.csv')
srampd = voltemp('srampd.csv')

rampu.plot(ax, 'fast and up')
rampd.plot(ax, 'fast and down')
srampu.plot(ax, 'slow and up')
srampd.plot(ax, 'slow and down')


vrampu = voltemp('5Vsrampu.csv')
vrampd = voltemp('5Vsrampd.csv')

vrampu.plot(ax, '5 V up')
vrampd.plot(ax, '5 V down')
plt.table(cellText=rows,cellLoc='left')
plt.legend()
fig.savefig('./images/superchl.png')
plt.show()
```

Diode $IV$ Curves:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

boltz = 1.380649*10**(-23)
room_temp = 293
elem = 1.60217662*10**(-19)

class IV:
    def __init__(self, filename):
        path = './iv/'
        self.df = pd.read_csv(path+filename, sep='\t').sort_values(by='V')
        self.v = self.df['V'] - 0.001*self.df['Va']
        self.i = self.df['I']

    def plot(self, ax, label):
        label = '{} K'.format(label)
        ax.scatter(self.v, self.i, label=label)
        ax.errorbar(self.v, self.i, xerr=0.005, fmt='none', label='_nolabel_')

        #lx = self.v.min()
        #mx = self.v.max()
        #m, b = np.polyfit(self.v, self.i, 1)
        #x = np.linspace(lx*0.95, mx, 10)
        #label = r'$I=%1.2f \ V%1.2f\qquad V_\mathrm{int}=%1.3f$' % (m, b, vint)
        #ax.plot(x, m*x + b, linestyle='--')
        plt.legend()

fig, ax = plt.subplots()
ax.grid(True)
ax.set_title(r'$I-V$ Curve')
ax.set_xlabel('Voltage (V)')
ax.set_ylabel(r'Current ($\mu$A)')
ax.axhline(0, color='black', linewidth=0.75)

#iv10 = IV('IV10.tsv')
#iv10.plot(ax, 10)
iv50 = IV('IV50.tsv')
iv100 = IV('IV100.tsv')
iv150 = IV('IV150.tsv')
iv200 = IV('IV200.tsv')
iv250 = IV('IV250.tsv')
ivrt = IV('IVrt.tsv')

iv250.plot(ax, 250)
iv50.plot(ax, 50)
iv100.plot(ax, 100)
iv150.plot(ax, 150)
iv200.plot(ax, 200)
ivrt.plot(ax, 294.5)

fig.savefig('images/iv.png')
plt.show()
```

Diode $\ln(I) - V$ Curves:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

boltz = 1.380649*10**(-23)
room_temp = 293
elem = 1.60217662*10**(-19)

class IV:
    def __init__(self, filename):
        path = './iv/'
        self.df = pd.read_csv(path+filename, sep='\t').sort_values(by='V')
        self.v = self.df['V'] - 0.001*self.df['Va']
        self.i = np.log(self.df['I'])

    def plot(self, ax, temp):
        ax.errorbar(self.v, self.i, xerr=0.005, fmt='none', label='_nolabel_')

        lx = self.v.min()
        mx = self.v.max()
        m, b = np.polyfit(self.v, self.i, 1)
        x = np.linspace(lx*0.95, mx, 10)
        label='%3d K'%temp
        trend = r'$\qquad \ln(I)=%.3f\ V %+.3f$' % (m, b)
        ideal = elem/(m*boltz*temp)
        i0 = np.exp(b)
        row = [label,trend,r'$n=%.3f$'%ideal,r'$I_0=%e$ $\mu$A'%i0]
        cellText.append(row)
        ax.plot(x, m*x + b, linestyle='--', label=label)
        plt.legend()

fig, ax = plt.subplots()
plt.subplots_adjust(bottom=0.3)
ax.grid(True)
#ax.set_title(r'$\ln(I)$')
ax.set_xlabel('Voltage (V)')
ax.xaxis.tick_top()
ax.xaxis.set_label_position('top')
ax.set_ylabel(r'$\ln(I)$')
ax.set_ylim(-1,12)
ax.axhline(0, color='black', linewidth=0.75)
cellText=[]

#iv10 = IV('IV10.tsv')
#iv10.plot(ax, 10)
iv50 = IV('IV50.tsv')
iv100 = IV('IV100.tsv')
iv150 = IV('IV150.tsv')
iv200 = IV('IV200.tsv')
iv250 = IV('IV250.tsv')
ivrt = IV('IVrt.tsv')

iv250.plot(ax, 250)
iv50.plot(ax, 50)
iv100.plot(ax, 100)
iv150.plot(ax, 150)
iv200.plot(ax, 200)
ivrt.plot(ax,294.5)

plt.table(cellText=cellText, cellLoc='left')
fig.savefig('images/logiv.png')
plt.show()
```

$V(T)$:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

boltz = 1.380649*10**(-23)
room_temp = 293
elem = 1.60217662*10**(-19)

class VT:
    def __init__(self, filename):
        path = './iv/'
        self.df = pd.read_csv(path+filename, sep='\t').sort_values(by='T')
        self.t = self.df['T']
        self.v = self.df['V'] - 0.001*self.df['Va']

    def plot(self, ax, label, color):
        label = r'$I\approx %3d$'%label
        ax.errorbar(self.t, self.v, xerr=0.005, fmt='none', label='_nolabel_')

        lx = self.t.min()
        mx = self.t.max()
        m, b = np.polyfit(self.t, self.v, 1)
        x = np.linspace(lx*0.95, mx, 10)
        trend = r'$\qquad V = %.3f\ T %+.3f$' % (m, b)
        ax.scatter(self.t, self.v, label=label, color=color)
        ax.plot(x, m*x + b, linestyle='--',color=color)
        Eg = b*elem/elem
        row = [label,trend,r'$E_g=%.3f$ eV'%Eg]
        global cellText
        cellText.append(row)
        plt.legend()

fig, ax = plt.subplots()
plt.subplots_adjust(bottom=0.15)
ax.grid(True)
ax.set_xlabel('Temperature (K)')
ax.xaxis.tick_top()
ax.xaxis.set_label_position('top')
ax.set_ylabel('Voltage (V)')
ax.axhline(0, color='black', linewidth=0.75)
cellText=[]

vt50 = VT('vt50.tsv')
vt200 = VT('vt200.tsv')

vt50.plot(ax,50,'grey')
vt200.plot(ax,200,'black')

plt.table(cellText=cellText, cellLoc='left')
fig.savefig('images/vt.png')
plt.show()
```