

## Source code

```
# import necessary modules
import pandas as pd
import numpy as np
from scipy import stats
from pandas import DataFrame
import matplotlib.pyplot as plt
from matplotlib import ticker
from IPython.display import Latex
# 2mM K3FeCN6, different scan rate
c_2mM_sr_001 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安法/original data/K3FeCN6/2mM_K3FeCN6/2mM_0.01.csv', header = None)
c_2mM_sr_002 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安法/original data/K3FeCN6/2mM_K3FeCN6/2mM_0.02.csv', header = None)
c_2mM_sr_004 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安法/original data/K3FeCN6/2mM_K3FeCN6/2mM_0.04.csv', header = None)
c_2mM_sr_006 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安法/original data/K3FeCN6/2mM_K3FeCN6/2mM_0.06.csv', header = None)
c_2mM_sr_008 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安法/original data/K3FeCN6/2mM_K3FeCN6/2mM_0.08.csv', header = None)
# data cleaning
for item in [c_2mM_sr_001, c_2mM_sr_002, c_2mM_sr_004, c_2mM_sr_006, c_2mM_sr_008]:
    for i in range(24):
        item.drop([i], inplace = True)
    # reset the index, making the index of first row be 0
    item.index = range(len(item))
    item.columns = ['Potential/V', 'Current/A']
# convert string to float
c_2mM_sr_001 = DataFrame(c_2mM_sr_001, dtype=float)
c_2mM_sr_002 = DataFrame(c_2mM_sr_002, dtype=float)
c_2mM_sr_004 = DataFrame(c_2mM_sr_004, dtype=float)
c_2mM_sr_006 = DataFrame(c_2mM_sr_006, dtype=float)
c_2mM_sr_008 = DataFrame(c_2mM_sr_008, dtype=float)
# make the plot
fig, ax = plt.subplots(figsize = (10,8))
ax.plot(c_2mM_sr_001['Potential/V'].tolist(), c_2mM_sr_001['Current/A'].tolist())
ax.plot(c_2mM_sr_002['Potential/V'].tolist(), c_2mM_sr_002['Current/A'].tolist())
ax.plot(c_2mM_sr_004['Potential/V'].tolist(), c_2mM_sr_004['Current/A'].tolist())
ax.plot(c_2mM_sr_006['Potential/V'].tolist(), c_2mM_sr_006['Current/A'].tolist())
ax.plot(c_2mM_sr_008['Potential/V'].tolist(), c_2mM_sr_008['Current/A'].tolist())
ax.legend(['Scan Rate:0.01V/s', 'Scan Rate:0.02V/s', 'Scan Rate:0.04V/s', 'Scan Rate:0.06V/s', 'Scan Rate:0.08V/s'], fontsize=12)
ax.invert_xaxis()
ax.set_xlabel('Potential/V', fontname='Calibri', fontsize=16)
ax.set_ylabel('Current/A', fontname='Calibri', fontsize=16)
plt.xticks(fontname='Calibri', fontsize=14)
plt.yticks(fontname='Calibri', fontsize=14)
plt.ylim(-3.0e-5, 3.5e-5)
plt.title('I-V curve for 2mM $K_3Fe(CN)_6$ under different scan rate',
```

```

fontsize=14, fontname='Calibri')
formatter = ticker.ScalarFormatter(useMathText=True)
formatter.set_scientific(True)
formatter.set_powerlimits((-1,1))
ax.yaxis.set_major_formatter(formatter)
plt.show()
# Scan rate 0.02V/s, different concentration
c_0mM_sr_002 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安法/original data/K3FeCN6/Scan_Rate_0.02/0mM_0.02.csv', header = None)
c_04mM_sr_002 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安法/original data/K3FeCN6/Scan_Rate_0.02/0.4mM_0.02.csv', header = None)
c_1mM_sr_002 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安法/original data/K3FeCN6/Scan_Rate_0.02/1mM_0.02.csv', header = None)
c_4mM_sr_002 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安法/original data/K3FeCN6/Scan_Rate_0.02/4mM_0.02.csv', header = None)
# data cleaning
for item in [c_0mM_sr_002, c_04mM_sr_002, c_1mM_sr_002, c_4mM_sr_002]:
    for i in range(24):
        item.drop([i], inplace = True)
    # reset the index, making the index of first row be 0
    item.index = range(len(item))
    item.columns = ['Potential/V', 'Current/A']
# convert string to float
c_0mM_sr_002 = DataFrame(c_0mM_sr_002, dtype=float)
c_04mM_sr_002 = DataFrame(c_04mM_sr_002, dtype=float)
c_1mM_sr_002 = DataFrame(c_1mM_sr_002, dtype=float)
c_4mM_sr_002 = DataFrame(c_4mM_sr_002, dtype=float)
# make the plot
fig, ax = plt.subplots(figsize = (10,8))
ax.plot(c_0mM_sr_002['Potential/V'].tolist(), c_0mM_sr_002['Current/A'].tolist())
ax.plot(c_04mM_sr_002['Potential/V'].tolist(),
c_04mM_sr_002['Current/A'].tolist())
ax.plot(c_1mM_sr_002['Potential/V'].tolist(), c_1mM_sr_002['Current/A'].tolist())
ax.plot(c_2mM_sr_002['Potential/V'].tolist(), c_2mM_sr_002['Current/A'].tolist())
ax.plot(c_4mM_sr_002['Potential/V'].tolist(), c_4mM_sr_002['Current/A'].tolist())
ax.legend(['c: 0mM', 'c: 0.4mM', 'c: 1mM', 'c: 2mM', 'c: 4mM'], fontsize=12)
ax.invert_xaxis()
ax.set_xlabel('Potential/V', fontname='Calibri', fontsize=16)
ax.set_ylabel('Current/A', fontname='Calibri', fontsize=16)
plt.xticks(fontname='Calibri', fontsize=14)
plt.yticks(fontname='Calibri', fontsize=14)
plt.title('I-V curve for scan rate=0.02V/s, under different concentration of $K_{3}Fe(CN)_{6}$', fontsize=14, fontname='Calibri')
formatter = ticker.ScalarFormatter(useMathText=True)
formatter.set_scientific(True)
formatter.set_powerlimits((-1,1))
ax.yaxis.set_major_formatter(formatter)
plt.show()
# 4mM ascorbic acid, different scan rate
aa_4mM_sr_001 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安法/original data/ascorbic_acid/4mM_0.01.csv', header = None)
aa_4mM_sr_002 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安法/original data/ascorbic_acid/4mM_0.02.csv', header = None)
aa_4mM_sr_004 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安

```

```

法/original data/ascorbic_acid/4mM_0.04.csv', header = None)
aa_4mM_sr_006 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安
法/original data/ascorbic_acid/4mM_0.06.csv', header = None)
aa_4mM_sr_008 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安
法/original data/ascorbic_acid/4mM_0.08.csv', header = None)
aa_4mM_sr_010 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安
法/original data/ascorbic_acid/4mM_0.1.csv', header = None)
aa_4mM_sr_020 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安
法/original data/ascorbic_acid/4mM_0.2.csv', header = None)
# data cleaning
for item in [aa_4mM_sr_001, aa_4mM_sr_002, aa_4mM_sr_004, aa_4mM_sr_006,
aa_4mM_sr_008, aa_4mM_sr_010, aa_4mM_sr_020]:
    for i in range(24):
        item.drop([i], inplace = True)
    # reset the index, making the index of first row be 0
    item.index = range(len(item))
    item.columns = ['Potential/V', 'Current/A']
# convert string to float
aa_4mM_sr_001 = DataFrame(aa_4mM_sr_001, dtype=float)
aa_4mM_sr_002 = DataFrame(aa_4mM_sr_002, dtype=float)
aa_4mM_sr_004 = DataFrame(aa_4mM_sr_004, dtype=float)
aa_4mM_sr_006 = DataFrame(aa_4mM_sr_006, dtype=float)
aa_4mM_sr_008 = DataFrame(aa_4mM_sr_008, dtype=float)
aa_4mM_sr_010 = DataFrame(aa_4mM_sr_010, dtype=float)
aa_4mM_sr_020 = DataFrame(aa_4mM_sr_020, dtype=float)
# make the plot
fig, ax = plt.subplots(figsize = (10,8))
ax.plot(aa_4mM_sr_001['Potential/V'].tolist(),
aa_4mM_sr_001['Current/A'].tolist())
ax.plot(aa_4mM_sr_002['Potential/V'].tolist(),
aa_4mM_sr_002['Current/A'].tolist())
ax.plot(aa_4mM_sr_004['Potential/V'].tolist(),
aa_4mM_sr_004['Current/A'].tolist())
ax.plot(aa_4mM_sr_006['Potential/V'].tolist(),
aa_4mM_sr_006['Current/A'].tolist())
ax.plot(aa_4mM_sr_008['Potential/V'].tolist(),
aa_4mM_sr_008['Current/A'].tolist())
ax.plot(aa_4mM_sr_010['Potential/V'].tolist(),
aa_4mM_sr_010['Current/A'].tolist())
ax.plot(aa_4mM_sr_020['Potential/V'].tolist(),
aa_4mM_sr_020['Current/A'].tolist())
ax.legend(['Scan Rate:0.01V/s', 'Scan Rate:0.02V/s', 'Scan Rate:0.04V/s', 'Scan
Rate:0.06V/s', 'Scan Rate:0.08V/s', 'Scan Rate:0.10V/s', 'Scan Rate:0.20V/s'],
fontsize=12)
ax.invert_xaxis()
ax.set_xlabel('Potential/V', fontname='Calibri', fontsize=16)
ax.set_ylabel('Current/A', fontname='Calibri', fontsize=16)
plt.xticks(fontname='Calibri', fontsize=14)
plt.yticks(fontname='Calibri', fontsize=14)
plt.title('I-V curve for 4mM ascorbic acid under different scan rate',
fontsize=16, fontname='Calibri')
formatter = ticker.ScalarFormatter(useMathText=True)
formatter.set_scientific(True)
formatter.set_powerlimits((-1,1))

```

```

ax.yaxis.set_major_formatter(formatter)
plt.show()
# Ip,a-c and Ip,c-c plots
df_002Vs_K3FeCN6 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安法/data_tables/0.02Vs_K3FeCN6.csv', usecols=[1,2,3,4,5,6,7])
# linear regression fitting
# Ip,c
stats.linregress(df_002Vs_K3FeCN6['c(K3FeCN6) mM'].tolist(),
df_002Vs_K3FeCN6['Ip,c'].tolist())
# Ip,a
stats.linregress(df_002Vs_K3FeCN6['c(K3FeCN6) mM'].tolist(),
df_002Vs_K3FeCN6['Ip,a'].tolist())
# fitting expressions of Ip,c and Ip,a
def fit_Ipc(x):
    return 7.0498432601880875*10**(-6)*x+3.7423197492163103*10**(-7)
def fit_Ipa(x):
    return -7.183072100313479*10**(-6)*x-3.870532915360517*10**(-7)
# plot for Ip,c
fig, ax = plt.subplots(figsize=(8,6))
# concentration points to plot fitted expression of Ip,c
c_points = np.linspace(0, 4, 100)
ax.plot(df_002Vs_K3FeCN6['c(K3FeCN6) mM'].tolist(),
df_002Vs_K3FeCN6['Ip,c'].tolist(), '-o')
ax.plot(c_points, [fit_Ipc(c) for c in c_points])
ax.legend(['original data', 'fitting curve'], fontsize=12)
ax.set_xlabel('Concentration/mM', fontname='Calibri', fontsize=14)
ax.set_ylabel('$I_{p,c}/A$', fontname='Calibri', fontsize=14)
plt.xticks(fontname='Calibri', fontsize=12)
plt.yticks(fontname='Calibri', fontsize=12)
plt.title('$I_{p,c}$-concentration: original data and fitting curve',
fontname='Calibri', fontsize=16)
formatter = ticker.ScalarFormatter(useMathText=True)
formatter.set_scientific(True)
formatter.set_powerlimits((-1,1))
ax.yaxis.set_major_formatter(formatter)
plt.show()
# plot for Ip,a
fig, ax = plt.subplots(figsize=(8,6))
# concentration points to plot fitted expression of Ip,c
c_points = np.linspace(0, 4, 100)
ax.plot(df_002Vs_K3FeCN6['c(K3FeCN6) mM'].tolist(),
df_002Vs_K3FeCN6['Ip,a'].tolist(), '-o')
ax.plot(c_points, [fit_Ipa(c) for c in c_points])
ax.legend(['original data', 'fitting curve'], fontsize=12)
ax.set_xlabel('Concentration/mM', fontname='Calibri', fontsize=14)
ax.set_ylabel('$I_{p,a}/A$', fontname='Calibri', fontsize=14)
plt.xticks(fontname='Calibri', fontsize=12)
plt.yticks(fontname='Calibri', fontsize=12)
plt.title('$I_{p,a}$-concentration: original data and fitting curve',
fontname='Calibri', fontsize=16)
formatter = ticker.ScalarFormatter(useMathText=True)
formatter.set_scientific(True)
formatter.set_powerlimits((-1,1))
ax.yaxis.set_major_formatter(formatter)

```

```

plt.show()
# Ip,a-v0.5 and Ip,c-v0.5 plots
df_2mM_K3FeCN6 = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安法/data_tables/2mM_K3FeCN6.csv', usecols=[1,2,3,4,5,6,7])
# linear regression fitting
# Ip,c
stats.linregress([np.sqrt(v) for v in df_2mM_K3FeCN6['Scan Rate (V/s)'].tolist()],
df_2mM_K3FeCN6['Ip,c'].tolist())
# Ip,a
stats.linregress([np.sqrt(v) for v in df_2mM_K3FeCN6['Scan Rate (V/s)'].tolist()],
df_2mM_K3FeCN6['Ip,a'].tolist())
# fitting expressions of Ip,c and Ip,a
def fitting_Ipc(x):
    return 0.00010379902108949476*x+7.193269820884064*10**(-7)
def fitting_Ipa(x):
    return -0.00010233937798399661*x-1.362268009280328*10**(-6)
# plot for Ip,c
fig, ax = plt.subplots(figsize=(8,6))
# scan rate points to plot fitted expression of Ip,c
sr_points = [np.sqrt(scan_rate) for scan_rate in np.linspace(0.01, 0.08, 100)]
ax.plot([np.sqrt(v) for v in df_2mM_K3FeCN6['Scan Rate (V/s)'].tolist()],
df_2mM_K3FeCN6['Ip,c'].tolist(), '-o')
ax.plot(sr_points, [fitting_Ipc(sr) for sr in sr_points])
ax.legend(['original data', 'fitting curve'], fontsize=12)
ax.set_xlabel('$v^{1/2}$', fontname='Calibri', fontsize=14)
ax.set_ylabel('$I_{p,c}/A$', fontname='Calibri', fontsize=14)
plt.xticks(fontname='Calibri', fontsize=12)
plt.yticks(fontname='Calibri', fontsize=12)
plt.title('$I_{p,c}$-$v^{1/2}$: original data and fitting curve',
fontname='Calibri', fontsize=16)
formatter = ticker.ScalarFormatter(useMathText=True)
formatter.set_scientific(True)
formatter.set_powerlimits((-1,1))
ax.yaxis.set_major_formatter(formatter)
plt.show()
# plot for Ip,a
fig, ax = plt.subplots(figsize=(8,6))
# scan rate points to plot fitted expression of Ip,a
sr_points = [np.sqrt(scan_rate) for scan_rate in np.linspace(0.01, 0.08, 100)]
ax.plot([np.sqrt(v) for v in df_2mM_K3FeCN6['Scan Rate (V/s)'].tolist()],
df_2mM_K3FeCN6['Ip,a'].tolist(), '-o')
ax.plot(sr_points, [fitting_Ipa(sr) for sr in sr_points])
ax.legend(['original data', 'fitting curve'], fontsize=12)
ax.set_xlabel('$v^{1/2}$', fontname='Calibri', fontsize=14)
ax.set_ylabel('$I_{p,a}/A$', fontname='Calibri', fontsize=14)
plt.xticks(fontname='Calibri', fontsize=12)
plt.yticks(fontname='Calibri', fontsize=12)
plt.title('$I_{p,a}$-$v^{1/2}$: original data and fitting curve',
fontname='Calibri', fontsize=16)
formatter = ticker.ScalarFormatter(useMathText=True)
formatter.set_scientific(True)
formatter.set_powerlimits((-1,1))
ax.yaxis.set_major_formatter(formatter)
plt.show()

```

```
# 4mM ascorbic acid,  $I_p, a-v^{1/2}$  and  $E_p, a-v^{1/2}$  plots
df_4mM_aa = pd.read_csv('C:/Users/rundong-li/Desktop/物理化学实验/7-循环伏安法/data_tables/4mM_ascorbic_acid.csv', usecols=[1,3,6])
#  $I_p, a-v^{1/2}$ 
fig, ax = plt.subplots(figsize=(8,6))
ax.plot([np.sqrt(v) for v in df_4mM_aa['Scan Rate (V/s)'].tolist()],
df_4mM_aa[' $I_p, a$ '].tolist(), '-o')
ax.set_xlabel('$v^{1/2}$', fontname='Calibri', fontsize=14)
ax.set_ylabel('$I_{p,a}/A$', fontname='Calibri', fontsize=14)
plt.xticks(fontname='Calibri', fontsize=12)
plt.yticks(fontname='Calibri', fontsize=12)
plt.title('$I_{p,a}-v^{1/2}$: ascorbic acid', fontname='Calibri', fontsize=16)
formatter = ticker.ScalarFormatter(useMathText=True)
formatter.set_scientific(True)
formatter.set_powerlimits((-1,1))
ax.yaxis.set_major_formatter(formatter)
plt.show()
#  $E_p, a-v$ 
fig, ax = plt.subplots(figsize=(8,6))
ax.plot(df_4mM_aa['Scan Rate (V/s)'].tolist(), df_4mM_aa[' $E_p, a$ '].tolist(), '-o')
ax.set_xlabel('Scan rate (V/s)', fontname='Calibri', fontsize=14)
ax.set_ylabel('$E_{p,a}/V$', fontname='Calibri', fontsize=14)
plt.xticks(fontname='Calibri', fontsize=12)
plt.yticks(fontname='Calibri', fontsize=12)
plt.title('$E_{p,a}-v$: ascorbic acid', fontname='Calibri', fontsize=16)
formatter = ticker.ScalarFormatter(useMathText=True)
formatter.set_scientific(True)
formatter.set_powerlimits((-1,1))
ax.yaxis.set_major_formatter(formatter)
plt.show()
```