LabVIEW: IV VI

LabVIEW is sophisticated interfacing software used in many research labs. Your task is to learn LabVIEW well enough to write a LabVIEW "program" (normally called a **virtual instrument**, or VI) to generate the IV (current-voltage) curve of a diode. I found this task surprisingly challenging; none of the examples in my LabVIEW textbooks were very similar to what I needed. So, naturally, I googled "LabVIEW diode IV curve," and I did find examples online, but these were so tremendously complicated that I decided it would be easier to start from scratch.

I'd recommend that you work through the following before attempting the IV VI:

- Sections 1.1-1.11, 2.1-2.10, 4.1-4.4, and 4.6-4.10 in Essick, *Hands-On Introduction to LabVIEW for Scientists and Engineers.*
- Activities 7-2, 8-3, and 11-1 in Travis and Kring, *LabVIEW for Everyone*. (In Activity 11-1, the front panel has an array of numeric indicators. First drag an array onto the front panel, and then drag a numeric indicator into the array.)

Although Essick has a lot of indispensable, condensed information, Activities 8-3 and 11-1 from Travis and Kring provided the basic foundation for my VI.

One challenge is the manipulation of the format of the data. In my VI, "dynamic" (time-stamped) data first has to be converted into numeric data. Then, a two-dimensional array (a matrix) of all the data (current vs. voltage) has to be converted into a cluster of two one-dimensional arrays, in order to plot it!

You need to think just a little about hardware. Our data acquisition (DAQ) device, USB-6009, inputs and outputs voltages, not currents. So you'll need to put your diode in series with a resistor and measure the voltage across the resistor to determine current. (You'd want to put the diode in series with a resistor anyway to avoid putting excessive voltage across the diode.)

In your lab report, derive the ideal diode equation. You can study, for example, Lectures 17, 18, 19, and 22 in https://physics.emory.edu/faculty/brody/Advanced%20Lab/phys%20222%20lecture%20notes.pdf, or any other reference you find.

Can you fit the ideal diode equation to your data? Maybe the ideal equation doesn't fit real data very well. You'll at least need to include something called the ideality factor. The ideality factor may be different in different regions of your data. Try fitting different regions of your data separately.

Curve-fitting in Python

To export the data to Python, first copy it into Excel (as an intermediate step). Then open Spyder and run the following Python script by copying it into the "Editor pane" on the left and clicking the green "Run" arrow (if you want Spyder on your computer, download Anaconda from https://www.continuum.io/downloads):

```
import numpy as np, matplotlib.pyplot as plt, pandas as pd
from scipy.optimize import curve_fit
from sklearn.metrics import r2_score
print ('Your x data and y data should appear as columns in Excel\n')
input("Copy your x data to clipboard (CTRL-C) and press Enter.")
x=pd.read_clipboard(header=None)
x=np.array(x).T[0]
input("Copy your y data to clipboard (CTRL-C) and press Enter.")
y=pd.read_clipboard(header=None)
y=np.array(y).T[0]
def func(x,a,b):
   return a*(np.exp(b*x)-1)
popt, pcov = curve_fit(func, x, y,bounds=([0,0],[np.inf,np.inf]))
plt.plot(x,y,'o',label="measured data")
yfit=func(x,*popt)
yplot=func(np.sort(x),*popt)
plt.plot(np.sort(x),yplot,label="theoretical fit")
perr = np.sqrt(np.diag(pcov))
r2=r2_score(y, yfit)
print("The fitting parameters are",popt)
print("The uncertainties in the fitting parameters are",perr)
print("The coefficient of determination is", r2)
plt.legend()
```