Interview Graphs

January 27, 2017

In [2]:
```
%pylab inline
```

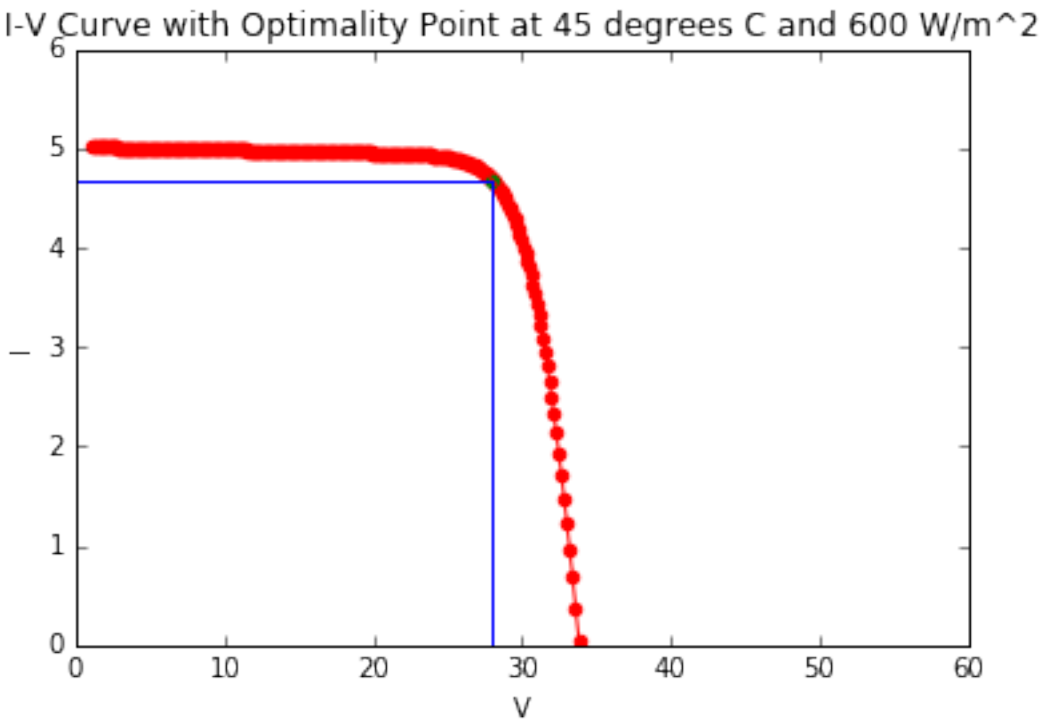
```
Populating the interactive namespace from numpy and matplotlib

/usr/local/lib/python2.7/site-packages/matplotlib/font_manager.py:273: UserWarning: Matplotlib is buildi
  warnings.warn('Matplotlib is building the font cache using fc-list. This may take a moment.')
```

In [3]:
```
import numpy as np
```

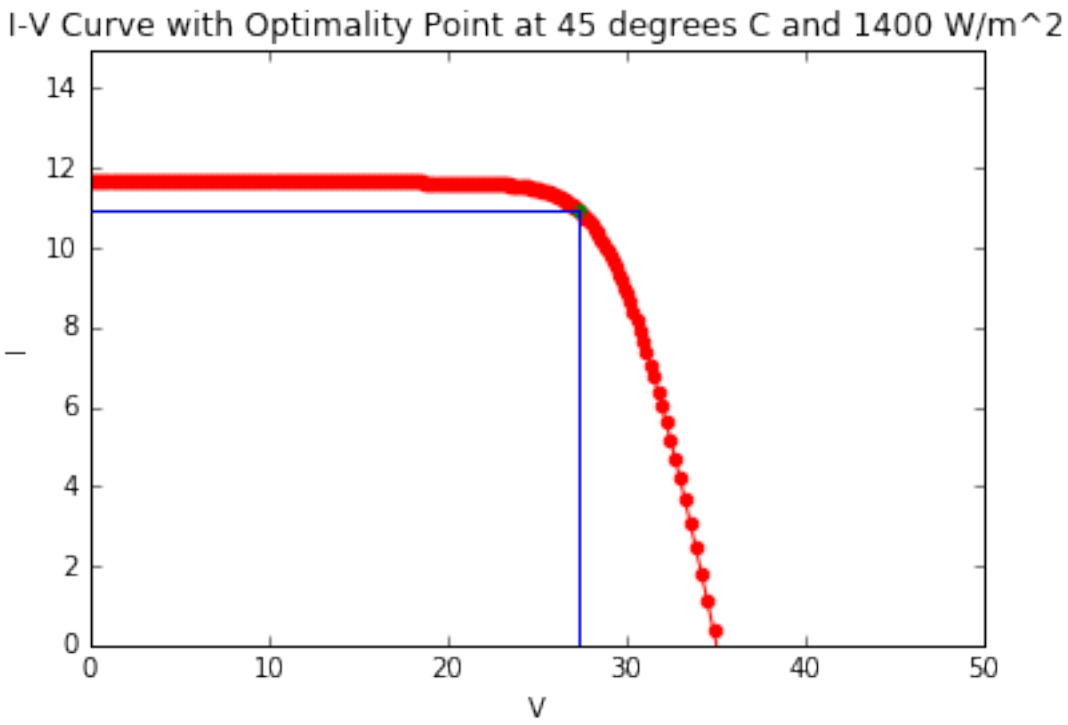
For finding the optimal power points, I simply calculated values of I-V values (and the corresponding power output) along the curve (based on increasing V-diode values). I stored each point as a 3-tuple (power, I, V), in a list, and then sorted the list by power. I then selected the last value in the list (i.e. with the highest power). Then, I applied a modified version of hill climbing to arrive at the optimal power point. Unlike traditional hill climbing, I restrict my hill climb to a step size of +/- 1/n on the nth step, and I restrict the process to a 100 steps. The underlying assumption is that the initial starting point is not far from the point of optimality, so the result of this 100-iteration run should be fairly accurate. The points of optimal power generation that I calculated through this process are displayed below in the graphs.

In [4]:
```
import matplotlib.pyplot as plt
```

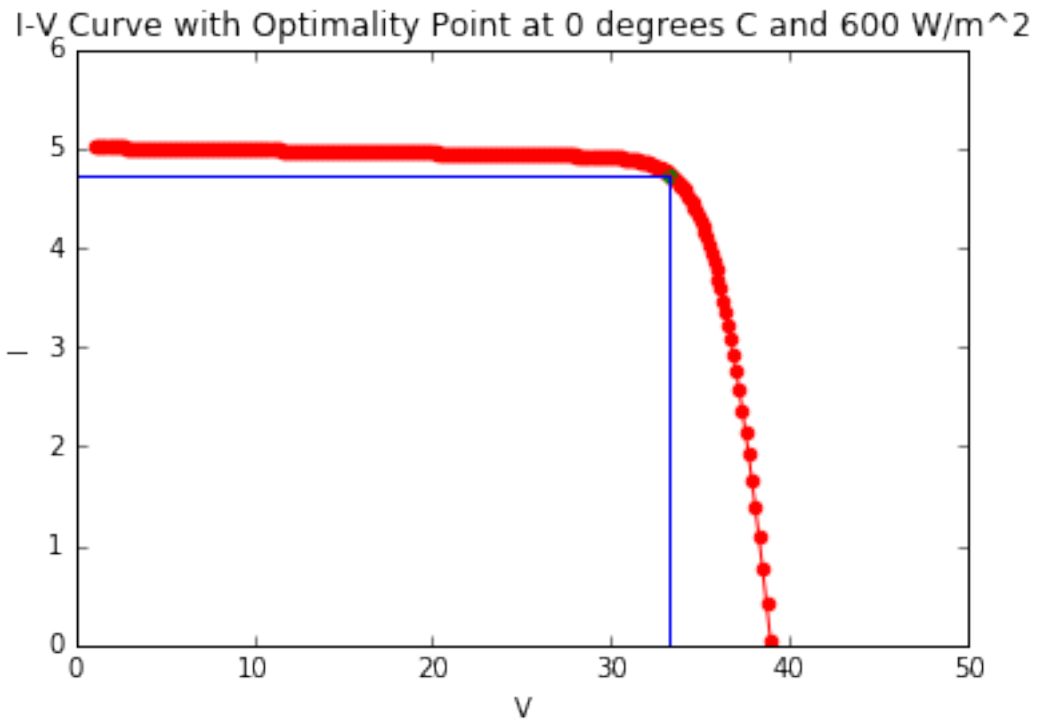
In [48]:
```
v_values = np.array([1.1308480405878272, 1.230952204448065, 1.3310563683117544, 1.431160532179
i_values = np.array([5.0132814872082, 5.012972295570178, 5.012663103921909, 5.012353912262655,
opt_v = np.array([27.9912821026])
opt_i = np.array([4.68012640388])
plt.scatter(v_values, i_values, color = "r")
plt.scatter(opt_v, opt_i, color = 'g')
plt.plot(v_values, i_values, color = "r")
xs = np.linspace(0,27.9912821026,100)
ys = np.array([4.6801264038 for t in xrange(len(xs))])
plt.plot(xs, ys, 'b')
ys = np.linspace(0,4.68012640388,100)
xs = np.array([27.9912821026 for t in xrange(len(ys))])
plt.plot(xs,ys,'b')
plt.ylabel("I")
plt.xlabel("V")
# x = numpy.linspace(0, 150, 1000)
plt.title("I-V Curve with Optimality Point at 45 degrees C and 600 W/m^2")
xlim(0, 60)
ylim(0, 6)
plt.show()
```



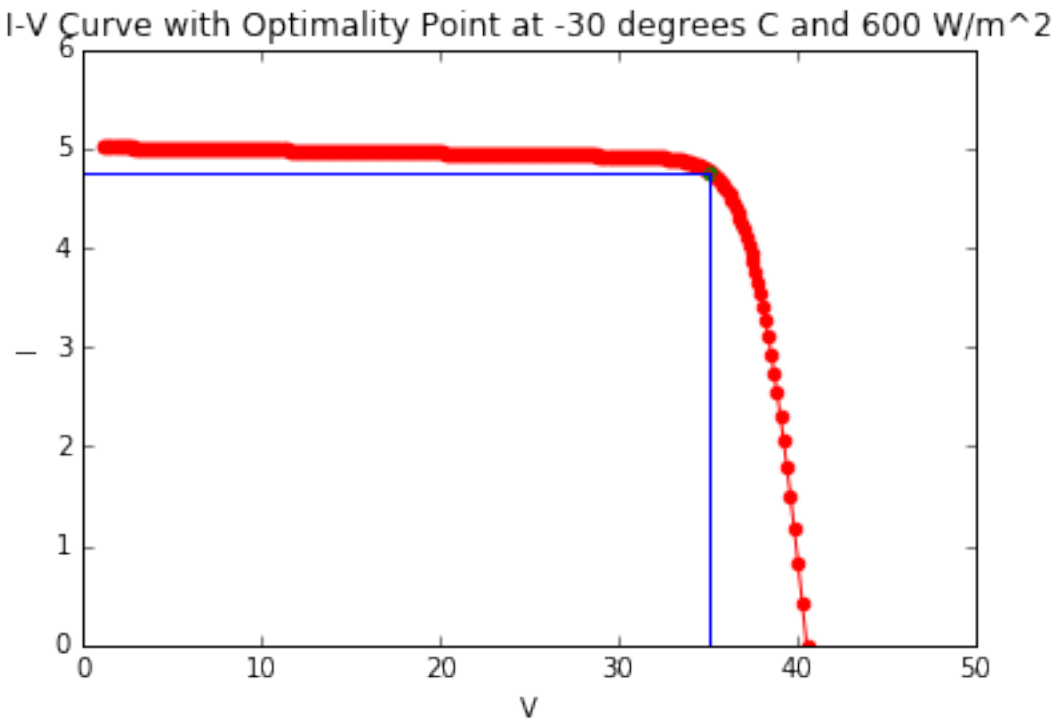

```
In [52]: v_values = np.array([-1.124973664723497, -1.02486950086326, -0.9247653369995703, -0.8246611731
         i_values = np.array([11.709281487208198, 11.708972295570177, 11.708663103921909, 11.7083539122
         opt_v = np.array([27.2737994496])
         opt_i = np.array([10.9581348052])
         plt.scatter(v_values, i_values, color = "r")
         plt.scatter(opt_v, opt_i, color = 'g')
         plt.plot(v_values, i_values, color = "r")
         xs = np.linspace(0,27.3799,100)
         ys = np.array([10.958134 for t in xrange(len(xs))])
         plt.plot(xs, ys, 'b')
         ys = np.linspace(0,10.958135,100)
         xs = np.array([27.3799 for t in xrange(len(ys))])
         plt.plot(xs,ys,'b')
         plt.ylabel("I")
         plt.xlabel("V")
         plt.title("I-V Curve with Optimality Point at 45 degrees C and 1400 W/m^2")
         # x = numpy.linspace(0, 150, 1000)
         xlim(0, 50)
         ylim(0, 15)
         plt.show()
```




```
In [53]: v_values = np.array([1.130848040019651, 1.2309522038321548, 1.331056367644661, 1.4311605314571
         i_values = np.array([5.013281488894728, 5.012972297398396, 5.012663105902056, 5.01235391440571
         opt_v = np.array([33.2773760975])
         opt_i = np.array([4.73337945862])
         plt.scatter(v_values, i_values, color = "r")
         plt.scatter(opt_v, opt_i, color = 'g')
         plt.plot(v_values, i_values, color = "r")
         xs = np.linspace(0,33.2773760975,100)
         ys = np.array([4.73337945862 for t in xrange(len(xs))])
         plt.plot(xs, ys, 'b')
         ys = np.linspace(0,4.73337945862,100)
         xs = np.array([33.2773760975 for t in xrange(len(ys))])
         plt.plot(xs,ys,'b')
         plt.ylabel("I")
         plt.xlabel("V")
         plt.title("I-V Curve with Optimality Point at 0 degrees C and 600 W/m^2")
         # x = numpy.linspace(0, 150, 1000)
         xlim(0, 50)
         ylim(0, 6)
         plt.show()
```



In [66]:
```
v_values = np.array([1.1308480400193874, 1.2309522038318663, 1.3310563676443457, 1.43116053145
i_values = np.array([5.013281488895511, 5.012972297399252, 5.012663105902992, 5.01235391440673
opt_v = np.array([35.1174419673])
opt_i = np.array([4.7562430289])
plt.scatter(v_values, i_values, color = "r")
plt.scatter(opt_v, opt_i, color = 'g')
plt.plot(v_values, i_values, color = "r")
xs = np.linspace(0,35.1174419673,100)
ys = np.array([4.7562430289 for t in xrange(len(xs))])
plt.plot(xs, ys, 'b')
ys = np.linspace(0,4.7562430289,100)
xs = np.array([35.1174419673 for t in xrange(len(ys))])
plt.plot(xs,ys,'b')
plt.ylabel("I")
plt.xlabel("V")
plt.title("I-V Curve with Optimality Point at -30 degrees C and 600 W/m^2")
# x = numpy.linspace(0, 150, 1000)
xlim(0, 50)
ylim(0, 6)
plt.show()
```


I-V Curve with Optimality Point at -30 degrees C and 600 W/m^2
I
V
0
1
2
3
4
5
6
0
10
20
30
40
50

In [ ]: