# assignment1_1i

January 13, 2017

GEOG827 Assignment #1<
> M. J. Brodzik<
> due 3/12/17<
>

Question 1.

Using the tables and/or equations in the "Calculating Evaporation" documents posted in blackboard, notes and/or other sources (state the source), express results as mean W/m2 over the day.

Part i) estimate the daily average solar radiation to the top of the Earth's atmosphere at 56oN on July 2nd.

From Table D, "Calculating Evaporation Notes", the Total daily solar radiation, $K_A$, is given in 10-degree latitude increments, for Jun 22 and Jul 15. I begin by plotting these to see how much variability there is in $K_A$:

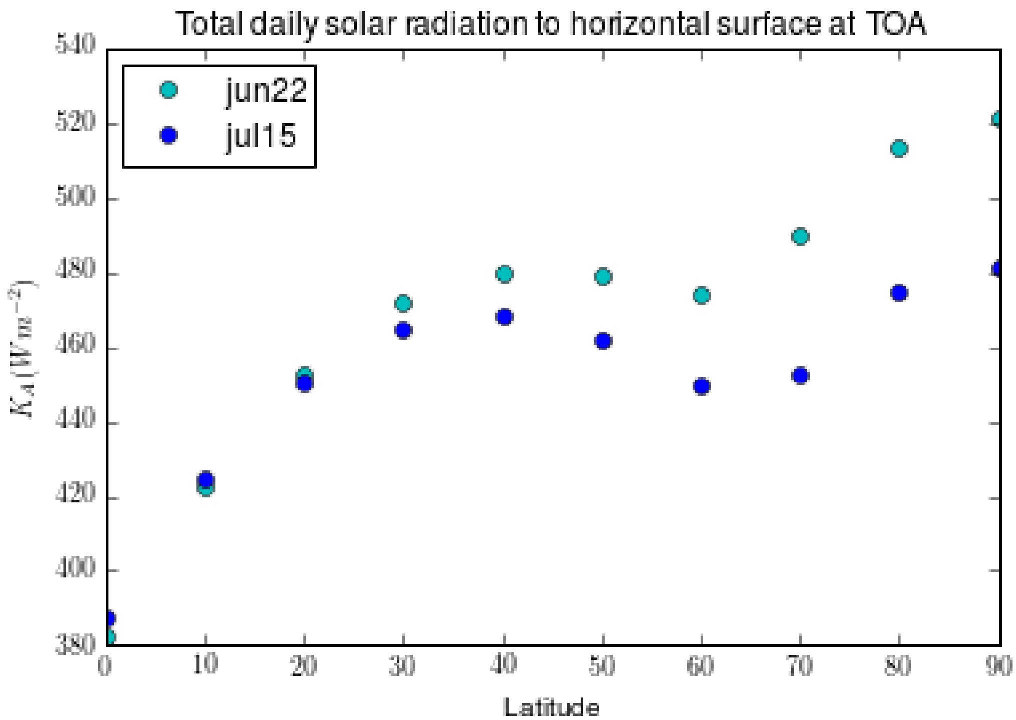
```
In [1]: %pylab inline
        import matplotlib.pyplot as plt
        import numpy as np

        # Define data to plot
        lats = np.arange(10) * 10
        jun22 = np.array([382.68, 422.88, 452.91, 472.29, 480.04,
                          479.07, 474.23, 490.21, 513.46, 521.70])
        jul15 = np.array([387.52, 424.82, 450.49, 465.02, 468.41,
                          462.12, 450.01, 452.43, 474.71, 481.49])

        # Make the plot:
        plt.rc('text', usetex=True)
        fig, ax = plt.subplots(1)
        ax.plot(lats, jun22, 'co', label='jun22')
        ax.plot(lats, jul15, 'bo', label='jul15')
        ax.set_title("Total daily solar radiation to horizontal surface at TOA")
        ax.set_xlabel("Latitude")
        ax.set_ylabel(r'$K_A (W m^{-2})$')
        ax.legend(loc='best', numpoints=1)
        plt.show()
        #fig.savefig('HW1.1i_fig1.png')
```

```
Populating the interactive namespace from numpy and matplotlib

/Users/brodzik/.conda/envs/pmesdr/lib/python2.7/site-packages/matplotlib/font_manager.py:273: UserWarnin
  warnings.warn('Matplotlib is building the font cache using fc-list. This may take a moment.')
```




So although there is an inflection point in the data above 60N, for an estimate I think it's sufficient to just linearly interpolate $K_A$ for the given dates, and then interpolate to 56° N.

In [2]:
```
# define a quick linear interpolation function
# calculate slope and intercept and the value of the line at
# the new value
def linear_model_value_at(x, x1, y1, x2, y2):
    slope = (y2 - y1) / (x2 - x1)
    # y = mx + b ==> b = y - mx
    intercept =  y1 - (slope * x1)
    return (slope * x) + intercept
```

In [3]:
```
KA_56N_jun22 = linear_model_value_at(56., 50., 479.07, 60., 474.23)
KA_56N_jul15 = linear_model_value_at(56., 50., 462.12, 60., 450.01)
```

Linearly interpolate $K_A$ at 56 N to July 2 between Jun22 and Jul15:

In [4]:
```
import datetime
jun22_doy = datetime.datetime(2017, 6, 22).timetuple().tm_yday
jul2_doy = datetime.datetime(2017, 7, 2).timetuple().tm_yday
jul15_doy = datetime.datetime(2017, 7, 15).timetuple().tm_yday
print(jun22_doy, jul2_doy, jul15_doy)

KA_56N_jul2 = linear_model_value_at(jul2_doy,
                                    jun22_doy, KA_56N_jun22,
                                    jul15_doy, KA_56N_jul15)
KA_56N_jul2
```

```
(173, 183, 196)
```

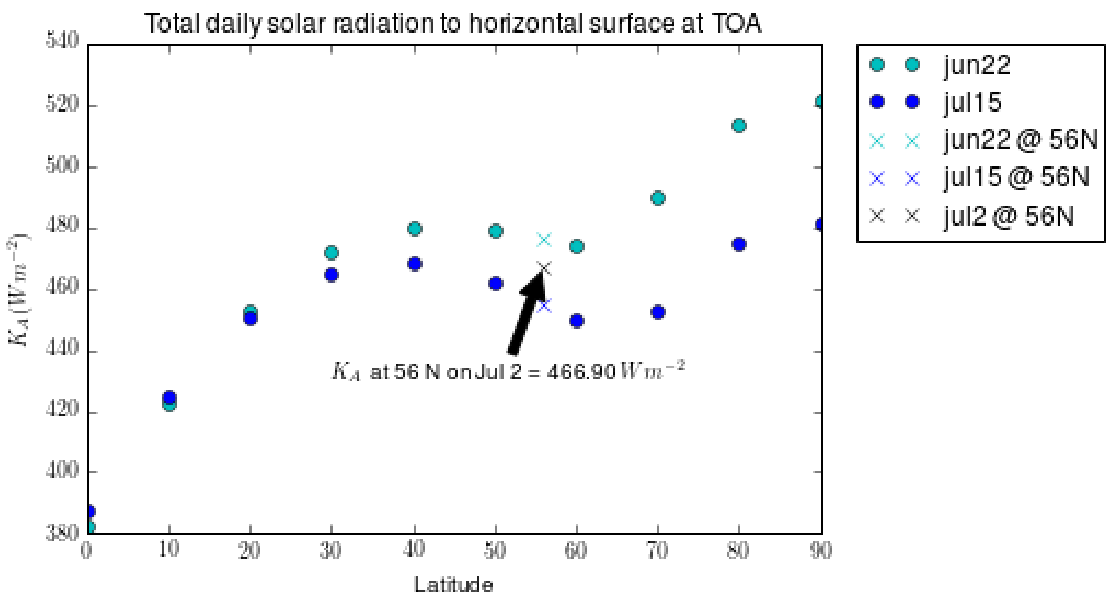
Out[4]: 466.8999130434783

Adding my interpolated values to the plot, I think it's sufficient to use this approximation:

In [8]:
```
fig, ax = plt.subplots(1)

ax.plot(lats, jun22, 'co', label='jun22')
ax.plot(lats, jul15, 'bo', label='jul15')
ax.plot(56, KA_56N_jun22, 'cx', label='jun22 @ 56N')
ax.plot(56, KA_56N_jul15, 'bx', label='jul15 @ 56N')
ax.plot(56, KA_56N_jul2, 'kx', label='jul2 @ 56N')
ax.annotate('$K_A$ at 56 N on Jul 2 = %.2f $W m^{-2}$' % KA_56N_jul2,
            xy=(56, KA_56N_jul2),
            xytext=(30, 430),
            arrowprops=dict(facecolor='k', shrink=0.05))

ax.set_title("Total daily solar radiation to horizontal surface at TOA")
ax.set_xlabel("Latitude")
ax.set_ylabel(r'$K_A (W m^{-2})$')
#ax.legend(loc='best', numpoints=1)
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.show()
#fig.savefig('HW1.1i_fig2.png')
```




So the daily average solar radiation to the top of the Earth's atmosphere at 56° N on July 2nd is 466.90 $Wm^{-2}$.

In [ ]: