ENV 480: Climate Laboratory, Spring 2014

Assignment 1: Intro to Python, global energy balance and albedo feedback
Due: Tuesday February 4 2014

Question 1:
Read through Section 2.1 of the "Fun with Python" document (available on the class web page), and work through all the Python exercises in Canopy on your laptop. Comment on something that you found surprising or interesting in the exercises. Also describe your previous experiences, if any, with computer programming.
If you found nothing surprising or interesting in the exercises, comment on why you are so experienced with Python, and/or why it is so difficult to impress you.

Question 2a:
In class we defined a zero-dimensional energy balance model for the global mean surface temperature T as follows

$$C\,\frac{dT}{dt} = \big(1-\alpha(T)\big)Q - \ OLR(T)$$

$$OLR = \varepsilon\sigma T^4$$

where $C = 4\times10^8$ J m^{-2} K^{-1} is a heat capacity for the atmosphere-ocean column, α is the global mean albedo (which varies with temperature to represent the ice-albedo feedback), $\sigma = 5.67\times10^{-8}$ W m^{-2} K^{-4} is the Stefan-Boltzmann constant, $\varepsilon = 0.612$ is an effective global-mean emissivity for the atmosphere, and $Q = 341.3$ W m^{-2} is the global-mean incoming solar radiation.

We are choosing to represent the ice-albedo feedback as follows:

$$\alpha(T) = \begin{cases} \alpha_i & T \le T_i \\ \alpha_o + \ (\alpha_i - \alpha_o)\dfrac{(T-T_o)^2}{(T_i-T_o)^2} & T_i < T < T_o \\ \alpha_o & T \ge T_o \end{cases}$$

where $\alpha_o = 0.289, \alpha_i = 0.7, T_o = 293\ K, T_i = 260\ K$. Note that this formula reproduces the observed albedo for T = 288 K.

In class we used the following Python code. Enter this code on the command line exactly as written (or experiment with changing whitespace).

```
Q = 341.3

def OLR( T ):
	sigma = 5.67E-8
	epsilon = 0.612
	return epsilon * sigma * T**4

def albedo( T ):
	To = 293.
	Ti = 260.
```

```
    alpha_o = 0.289
    alpha_i = 0.7
    alpha=where(T>Ti,alpha_o+(alpha_i-alpha_o)*(T-To)**2/(Ti-To)**2,alpha_i)
    alpha = where( T >= To, alpha_o, alpha )
    return alpha
```

We also found equilibrium solutions to the global energy balance by plotting the solar and terrestrial radiation as functions of temperature and finding the intersections of the graphs:

```
T = linspace(200, 300)
figure()
plot( T, (1-albedo(T))*Q, T, OLR(T) )
xlabel('Global mean temperature (K)')
ylabel('Energy flux to/from space')
grid()
```

This reveals three possible solutions to the energy budget with very different global mean temperatures.

Imagine the energy output from the sun were different (in fact, it was significantly weaker in the deep past). Re-draw the graph for different values of Q. How do the intersections of the two curves change as you make the solar constant larger and smaller?
What happens to the number of solutions when Q gets very large or very small?
You may need to adjust the temperature range of the graph. Try repeating the above "`linspace`*" command but with different limits.*

You may answer with words and hand-drawn sketches of the graphs, OR you are welcome to figure out how to save your graphs as image files, paste them into a document with the rest of your answers, and submit by email.

Question 2b:

In class we also defined a time-stepping method to investigate the adjustment of the climate system towards equilibrium. This example starts from a temperature of 300 K, uses a discrete time step of one year, steps forward 50 years, and stores the temperature for each year in an array called `Tsteps`.

```
C = 4.0E8
delta_time = 60. * 60. * 24. * 365.
def stepforward( T ):
    return T + delta_time / C * ( (1-albedo(T)) * Q - OLR(T) )

numsteps = 50
Tsteps = zeros(numsteps)
Tsteps[0] = 300.
for n in range(1,numsteps):
    Tsteps[n] = stepforward( Tsteps[n-1] )
```

Make a graph of the output. Is 50 years sufficient for the system to reach equilibrium? Redo the calculation with different initial temperatures. Try to plot the results on the same graph. (Hint: each `plot()` command will draw on the same figure. To create a new blank figure, type `figure()`.)

By choosing different initial temperatures, is it possible to get the system to equilibrate in all of the three different solutions we found above? Why, or why not?