PHY392 Solution Set 2 2013

1) for $n = \alpha T_s^\beta$

$$T_s^4 = (\alpha T_s^\beta + 1) T_e^4$$

$$T_s^4 - \alpha T_e^4 T_s^\beta - T_e^4 = 0$$

a) for $\beta = 4$

$$\frac{T_s^4 - T_e^4}{T_e^4 T_s^4} = \alpha$$

$$\frac{1}{T_e^4}\left[1 - \left(\frac{T_e}{T_s}\right)^4\right] = \alpha$$

for $T_s = 288\,K$ and $T_e = 255\,K$

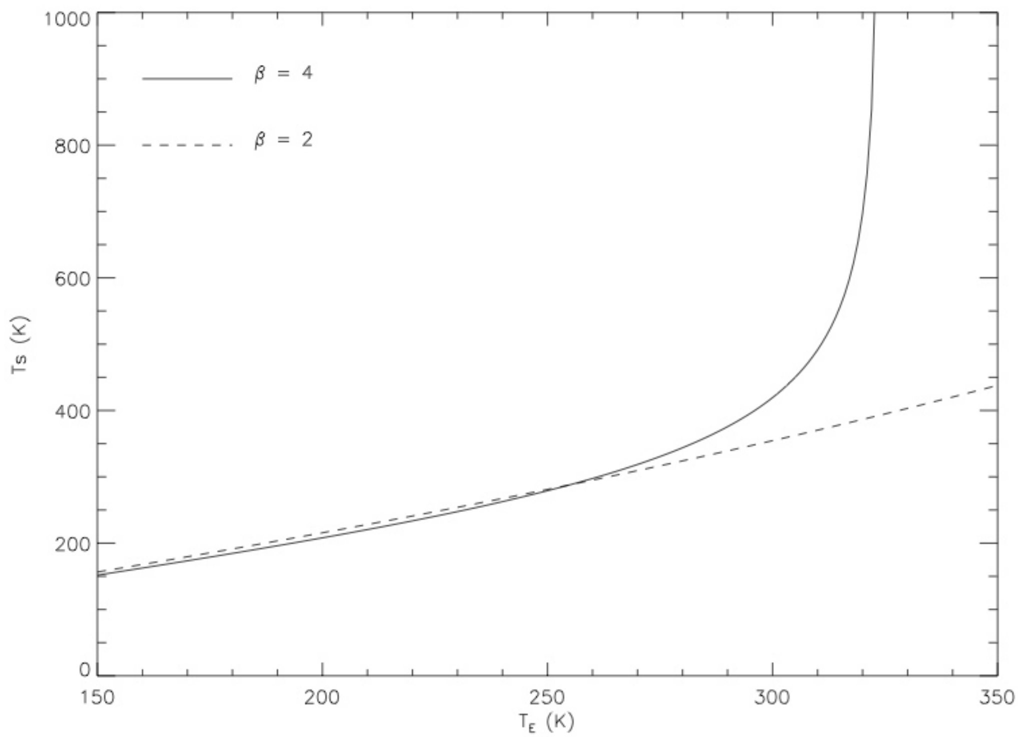
$$\Rightarrow \boxed{\alpha = 9.1 \times 10^{-11}\,K^{-4}}$$

For $\beta = 2$

$$T_s^4 - \alpha T_e T_s^2 - T_e^4 = 0$$

$$\alpha = \frac{1}{T_s^2}\left[\left(\frac{T_s}{T_e}\right)^4 - 1\right] = \boxed{7.56 \times 10^{-6}}$$

b)



I would characterize the case of $\beta = 4$ as an example of a runaway greenhouse effect as the surface temperature diverges for a a finite Te.

Furthermore, from the Clausius-Clapeyron relationship the saturation vapour pressure of water increases exponentially with temperature so we expect an exponential increase in n

c) If the albedo of Venus were $A_p = 0.15$

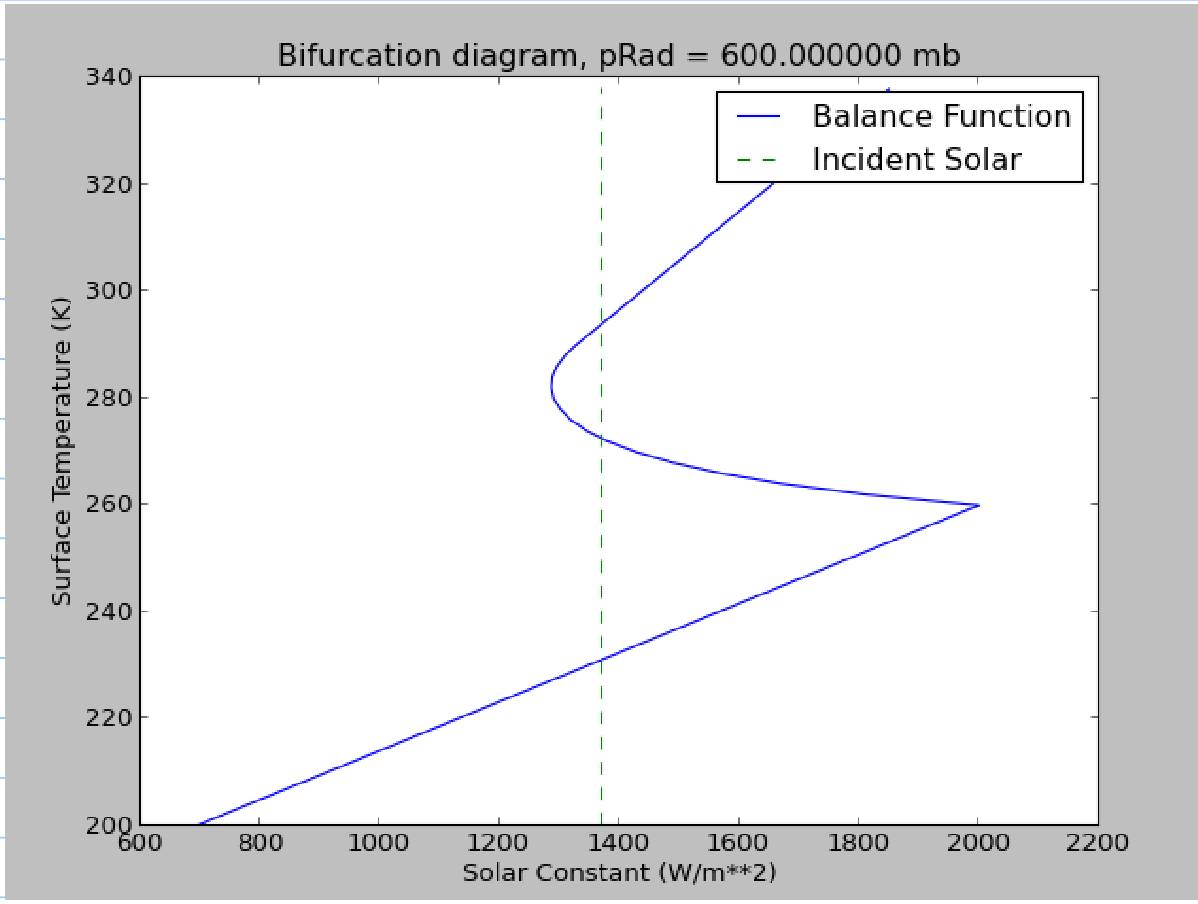
then $$T_e = \left[\frac{S_v}{4}(1 - A_p)\right]^{1/4}$$

$$= \left[\frac{2637}{4}\frac{(1-0.15)}{5.67\times10^{-8}}\right]^{1/4}$$

$$\Rightarrow \boxed{T_e = 315\,K}$$

d) As the atmospheric abundance of water vapour increased on both planets, the surface temperatures increased rapidly because of the greenhouse effect. Because Earth began with a lower emission temperature than Venus (since Earth is farther from the Sun), the partial pressure of water vapour eventually reach the saturation vapour pressure and water condensed out of the atmosphere. On Venus, on the other hand, because of the higher initial temperature ($T_s \gtrsim T_e$ without an absorbing atmosphere) the saturation vapour pressure was never reached, producing a runaway greenhouse effect.

2)



a) Based on the results shown in the plot, we would have to reduce L_0 to about 1280 Wm^{-2} to a transition the Snowball state. This a reduction in the solar constant of about 6.4%.

b) Power from the Sun: $P = 4\pi r_0^2 L_0 = 4\pi r^2 L$

where L is the solar constant at new radius r.

$$\Rightarrow \frac{r^2}{r_0^2} = \frac{L_0}{L} = \frac{1367\ Wm^{-2}}{1280\ Wm^{-2}}$$

$$\Rightarrow \boxed{\frac{r}{r_0} = 1.033}$$

Earth would have to be displaced by 3.3 % to force it into a Snowball state.

c) To prevent a Snowball state, we need L_0 greater than the "point" on the right of the graph, which is $L \approx 2000\ W m^{-2}$

$$\frac{r^2}{r_0^2} = \frac{L_0}{L} = \frac{1367}{2000} \Rightarrow \boxed{\frac{r}{r_0} = 0.827}$$

For Venus, $r = 1.08 \times 10^{11} m$ and

for Earth, $r_0 \simeq 1.5 \times 10^{11} m$

$$\Rightarrow \quad \frac{r}{r_0} = \frac{1.08}{1.5} = 0.72$$

The required distance is greater than the Sun-Venus distance.

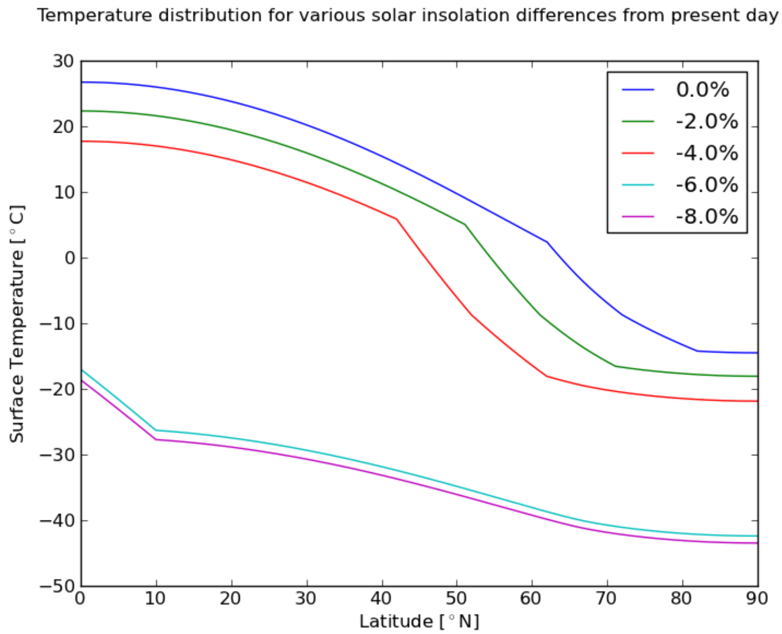


Figure 1: Surface temperature distribution for various values of solar constant

delQp [%]	Global mean temperature [°C]	Sea ice boundary [°N]
0	16.4	72
-2	11.2	61
-4	5.6	52
-6	-30.2	0
-8	-31.6	0

Table 1: Calculated global mean temperature and sea ice boundary for various values of **delQp**.

3 The Budyko model

(a) The model gives a fairly realistic temperature distribution for **delQp**=0%. The temperature is warmest at the equator and decreases towards the pole where there is lower solar insolation and a higher albedo due to the presence of sea ice. The global mean temperature is about 16°C, which is roughly similar to current conditions on Earth.

As **delQp** is decreased, the temperature at all latitudes decreases, as we would expect (see Figure 1 and Table 1). However, the decrease is not smooth: between -4% and -6% there is a much larger change in the temperature distribution than between any other values **delQp**. The reason for this sudden jump is that at **delQp**=-6.0% the Earth has entered a "snowball Earth" state in which the entire surface is covered with ice.

(b) Increasing or decreasing the albedo over either the ocean or sea ice has the effect of, respectively, decreasing or increasing the surface temperature. The surface temperature distribution is more sensitive to changes in the ocean's albedo than changes in the ice's albedo, primarily because the ocean covers much more surface area than the sea ice. See Figure 2.

(c) See attached code for how to properly incorporate the `for` loop and create the appropriate figures.

(d) As seen in Figure 3, at just less than -6% reduction in solar insolation there is a dramatic jump where the Earth becomes entirely ice-covered and the global mean temperature correspondingly decreases. This a substantially larger decrease than in [1] where the value at which the jump occured was around -1.6%. It is similar to the value found in Question 2a (around -6.4%).

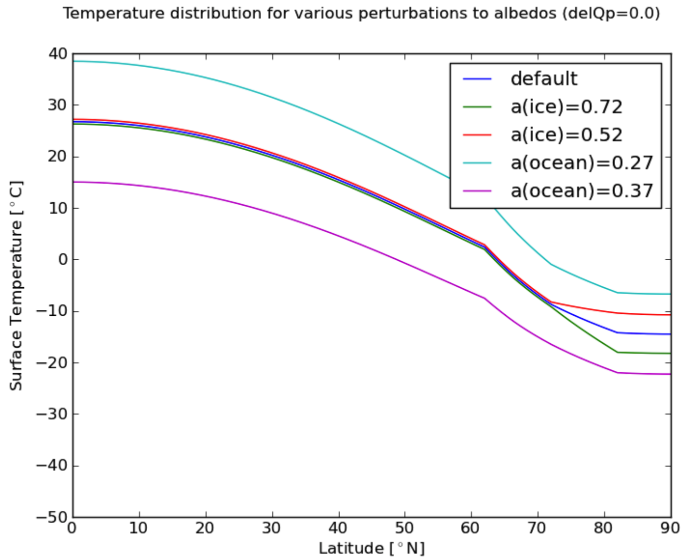


Figure 2: Surface temperature distribution for various perturbations to the surface albedo.

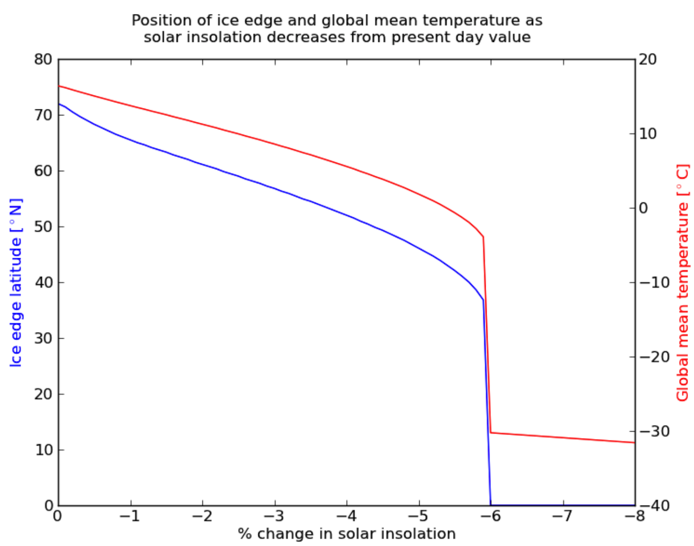


Figure 3: Extent of sea ice (blue) and global mean temperature (red) as a function of **delQp**.

(e) See Figures 4 and 5. As one would expect, by increasing the sea ice albedo, the shift to a snowball Earth state happens sooner (i.e. for a smaller decrease in solar insolation). The opposite is also true: decreasing the sea ice albedo means the shift doesn't happen until there is a larger decrease in insolation.

The effect of changing the albedo of the ocean is reversed. When we increase the albedo of the ocean, even though the global mean temperature is lower for **delQp**=0, the shift to complete sea ice cover doesn't occur until **delQp**=-7.5%. If the albedo of the ocean is decreased, the shift happens sooner.

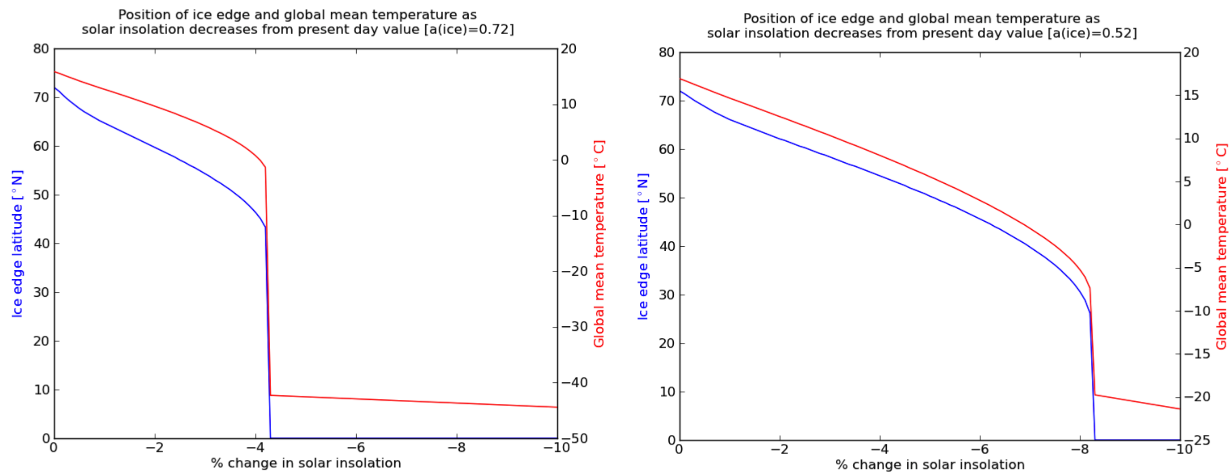


Figure 4: Extend of sea ice and global mean temperature as a function of **delQp** for an ice albedo of 0.72 (left) and 0.52 (right) as opposed to the default value of 0.62.

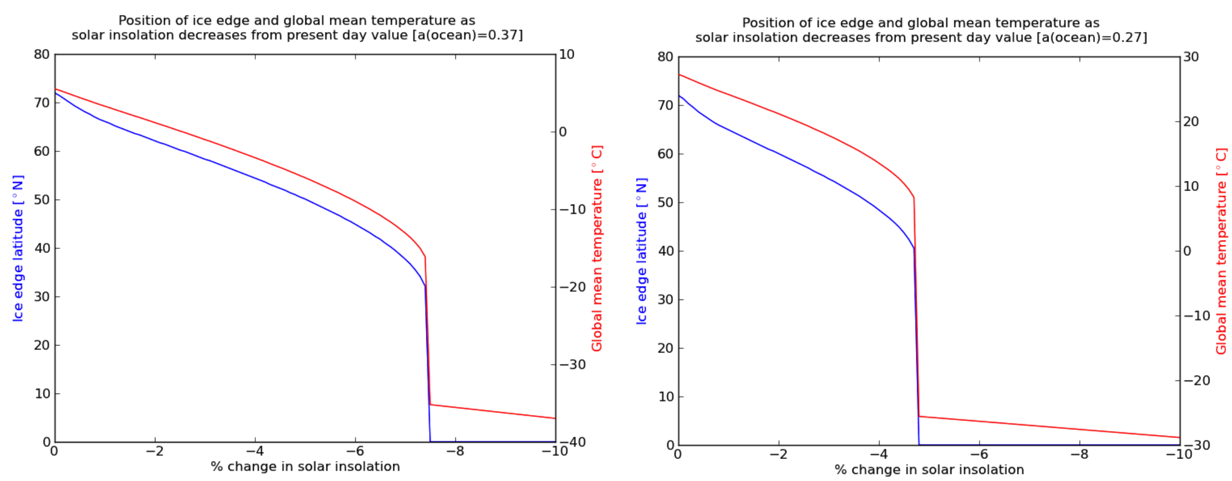


Figure 5: Extend of sea ice and global mean temperature as a function of **delQp** for an ocean albedo of 0.37 (left) and 0.27 (right) as opposed to the default value of 0.32.

A Code for looped script

```
from pylab import *
from numpy import *
import pickle

# INTRODUCTION ###################################################################
# This script implements the energy balance model described in Budyko (1969),
# "The effect of  solar radiation variations on the climate of the Earth",
# Tellus 21, pp. 611-619.

# READ IN INSOLATION DATA ########################################################
# Read array of latitudes and annual mean solar insolation at each latitude.
# Note the solar insolations are calculated by a separate Python script
# 'calc_solar_insolation.py' and then saved in the file 'solar_insolation_data'.
# The calculation is based on Eq. 2.2.25 of "An Introduction to Atmospheric
# Radiation" by K.-N. Liou.
f=open('solar_insolation_data','r')
data=pickle.load(f)  # Unpack data from 'solar_insolation_data'
lat=data[0]    # Array of latitudes at which temperatures will be calculated
Q=data[1]      # Array of annual mean solar insolation at each latitude in 'lat'

# INITIALIZE/DECLARE SOME VARIABLES ##############################################
max_iter=200             # Maximum number of iterations undertaken when calculating temperature based on EBM
tolerance=0.001          # When subsequent iterations of T are within this tolerance at all latitudes, the solution
iceEdgeCurrent=72.0      # Parameter for EBM, declaring current edge of glaciation [degrees latitude]
a=1000.0*14.0            # Parameter for EBM, coefficient [cal cm-2 month-1]
B=1000.0*0.14            # Parameter for EBM, coefficient [cal cm-2 degC-1 month-1]
a1=1000.0*3.0            # Parameter for EBM, coefficient [cal cm-2 month-1]
B1=1000.0*0.10           # Parameter for EBM, coefficient [cal cm-2 degC-1 month-1]
n=0.5                    # Parameter for EBM, cloudiness [dimensionless]
beta=1000.0*0.235        # Parameter for EBM, dependence of horizontal heat transfer on temperature difference from
l=0.0                    # Ratio of ice area change to the whole area of the Earth [dimensionless]
q=0.0                    # Ratio of mean radiation in same zone of ice area change to global mean radiation [dimensio
S=l*q

alpha=zeros(shape(lat))         # Albedo at each latitude [dimensionless]
T=zeros(shape(lat))             # Temperature at each latitude [degrees Celsius]
Tprevious=zeros(shape(lat))     # Temperature at each latitude from previous iteration [degrees Celsius]

Tp=0.0                   # Resulting global average temperatures for delQp change in solar insolation [degrees Celsi
iceEdge=0.0              # Resulting edge of glaciation for delQp change in solar insolation [degrees latitude]

# The following variables are the ones you should adjust
#delQp=-2.0                # Percentage change from present day in solar insolation for which the EBM will be solved
delQp_list=arange(0,-8.1,-.1)
Tp_list=[]
iceEdge_list=[]

albedoOcean=0.32         # Parameter for EBM, albedo of Earth's surface over ocean [dimensionless]
albedoIce=0.62           # Parameter for EBM, albedo of Earth's surface over ice (i.e. glaciated areas) [dimensionle
albedoBoundary=0.5       # Parameter for EBM, albedo of Earth's surface in boundary area between ocean and ice [dime
albedoDiff=albedoIce-albedoOcean    # Difference between albedo over ice and ocean [dimensionless]

# DEFINE TWO FUNCTIONS FOR LATER USE #############################################
def calc_ice_area_change(iceEdgeNew,iceEdgeOld):
    # Given two latitudes (in degrees) representing two ice area boundaries,
    # this function calculates the ratio of ice area change to the entire area
    # of the Earth

    # Calculate ice areas, i.e. surface area north of iceEdge. Note, assuming
    # Earth has unit radius, since ultimately just need ratio of areas.
    iceAreaNew=2*pi*(1-cos(pi/2-deg2rad(iceEdgeNew)))
    iceAreaOld=2*pi*(1-cos(pi/2-deg2rad(iceEdgeOld)))

    # Calculate ratio of difference to surface area of entire Earth
    iceAreaChange=(iceAreaNew-iceAreaOld)/(4*pi)

    return iceAreaChange

def calc_albedo(lat,iceEdge,albedoOcean,albedoIce,albedoBoundary):
    # Calculate albedo at each latitude, given the position of the ice edge, and
    # the albedo over ocean, over ice, and over the boundary between the two.
    #
    # The albedo is assumed to increase linearly from its "ocean" value at
    # iceEdge-boundaryHalfWidth, to its "boundary" value at iceEdge and then to its "ice" value
```

```
    # at iceEdge+boundaryHalfWidth.

    alpha=zeros(size(lat))
    boundaryHalfWidth=10.0

    # set albedo everywhere south of iceEdge-boundaryHalfWidth to albedoOcean
    alpha[lat<(iceEdge-boundaryHalfWidth)]=albedoOcean

    # set albedo to increase linearly from albedoOcean to albedoBoundary between
    # iceEdge-boundaryHalfWidth and iceEdge
    indIceSouthEdge = ((iceEdge-boundaryHalfWidth)<=lat) & (lat<iceEdge)
    alpha[indIceSouthEdge] = (lat[indIceSouthEdge]-(iceEdge-boundaryHalfWidth))*(albedoBoundary-albedoOcean)/boun

    # set albedo to increase linearly from albedoBoundary to albedoIce between
    # iceEdge and iceEdge+boundaryHalfWidth
    indIceNorthEdge = (iceEdge<=lat) & (lat<=(iceEdge+boundaryHalfWidth))
    alpha[indIceNorthEdge] = (lat[indIceNorthEdge]-iceEdge)*(albedoIce-albedoBoundary)/boundaryHalfWidth+albedoBou

    # set albedo to albedoIce everywhere north of iceEdge+boundaryHalfWidth
    alpha[lat>(iceEdge+boundaryHalfWidth)]=albedoIce

    return alpha

# do calculation for each delQp
for delQp in delQp_list:
    # ACTUAL EBM SCRIPT BEGINS BELOW ##################################################

    # calculate global mean of solar insolation received
    Qp=average(Q, weights=cos(deg2rad(lat)))

    # convert percentage change of global solar insolation to dimensional value
    delQp_dim=(delQp/100.0)*Qp

    # Calculate albedo at each latitude based on current ice edge
    alpha=calc_albedo(lat,iceEdgeCurrent,albedoOcean,albedoIce,albedoBoundary)

    # Calculate global mean of albedo
    alphap=average(alpha,weights=cos(deg2rad(lat)))

    # Calculate "present-day" temperature distribution (based on Equations 4 and 5 of Budyko 1969)
    Tp=(Qp*(1-alphap)-a+a1*n)/(B-B1*n)
    T=(Q*(1-alpha)-a+a1*n+beta*Tp)/(beta+B-B1*n)

    # Save this value of "present-day" global mean temperature for later use
    Tpprime=Tp

    # Save the value of the temperature at the current ice edge. This temperature
    # will be used to define the ice edge for other temperature profiles.
    iceEdgeTemp=T[lat==iceEdgeCurrent]

    # MAIN SOLUTION LOOP BEGINS HERE **##############################################
    # Calculate temperature distribution for decreased solar insolation
    # Iteratively solve Eq. 7 of Budyko 1969. Keep looping until either max_iter
    # is reached, or subsequent iterations of T are within the specified tolerance
    j=0
    converged=False
    while j<max_iter and not(converged):
        j+=1

        # Calculate temperature at each latitude based on current values of alpha and S
        T=Q*(1-alpha)*(1+delQp_dim/Qp)-a+a1*n+beta*Tpprime+beta*Qp/(B-B1*n)*(delQp_dim/Qp*(1-alphap-albedoDiff*S)-
        T=T/(beta+B-B1*n)

        if max(abs(T-Tprevious))<tolerance:
            converged=True

        Tprevious=T

        # Find new ice area boundary (i.e. latitude at which temperature=iceEdgeTemp)
        iceEdgeInd=argmin(abs(T-iceEdgeTemp))
        iceEdge=lat[iceEdgeInd]

        # Calculate new albedo based on new ice edge
        alpha=calc_albedo(lat,iceEdge,albedoOcean,albedoIce,albedoBoundary)

        # Calculate global mean of albedo
        alphap=average(alpha,weights=cos(deg2rad(lat)))
```

```
        # Calculate new value for S
        l=calc_ice_area_change(iceEdge,iceEdgeCurrent)
        if iceEdge<iceEdgeCurrent:
            q=average(Q[(iceEdge<lat)&(lat<=iceEdgeCurrent)],weights=cos(deg2rad(lat[(iceEdge<lat)&(lat<=iceEdgeC
        else:
            q=0.0
        S=l*q

    # Calculate global mean temperature
    Tp=average(T,weights=cos(deg2rad(lat)))

    print 'delQp=' + str(delQp) + '%: ice edge=' + str(iceEdge) + ' degN, global mean temperature=' + str(Tp) + '

    # save values of Tp and iceEdge for current value of delQp
    Tp_list.append(Tp)
    iceEdge_list.append(iceEdge)

    # Plot temperature distribution for decreased solar insolation
    #figure(1)
    #plot(lat,T)
    #ylabel('Surface Temperature [$^{\circ}$C]')
    #ylim([-50,30])
    #xlabel('Latitude [$^{\circ}$N]')
    #suptitle('Temperature distribution for '+str(delQp)+'% solar insolation difference from present day')

figure(1)
axes1=subplot(111)
plot(delQp_list,iceEdge_list,'b')
axes2=twinx()
plot(delQp_list,Tp_list,'r')

axes1.set_ylabel('Ice edge latitude [$^{\circ}$N]',color='b')
axes2.set_ylabel('Global mean temperature [$^{\circ}$C]',color='r')
axes1.set_xlabel('% change in solar insolation')
axes1.set_xlim([max(delQp_list),min(delQp_list)])
axes2.set_xlim([max(delQp_list),min(delQp_list)])
suptitle('Position of ice edge and global mean temperature as\nsolar insolation decreases from present day value'

show()
```

References

[1] Budyko, M. I. (1969), The effect of solar radiation variations on the climate of the Earth, *Tellus*, **21**, 611-619.