Assignment 05

LinQiangMa

December 2023

Problem1

1.1

```
# import libraries
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
import netCDF4
import xarray as xr
import math
#%%
# import data
data = pd.read_csv('global.1751_2014.csv')
observation = pd.read_csv('co2_annmean_mlo.csv',skiprows=70)
# check data
data
#%%
# check data
observation
#%%
# cut the unwanted data  and check data
data = data.iloc[1:,0:2]
data
#%%
# select the data from 1987 to 2004 , transform the year to int
data = data[(data['Year']>='1986')&(data['Year']<='2004')].reset_index(drop=True)
data['Year'] = data['Year'].astype(int)

# check data
data
#%%
# select the data from 1987 to 2004 and check data
obs = observation['346.35'][0:19]
obs
#%%
# create new columns for total carbon emission in unit pg, observation and atmospheric CO2 level in ppm
data['Total carbon emission in pg'] = data['Total carbon emissions from fossil fuel consumption and cem
data['observation'] = obs
data['CO2 level in ppm(no buff)'] = 347
data
#%%
# 1.1
# initialization
# assume that the speed of transfer speed is stationary
k12, k21 = 105/740, 102/900
# the initial value of CO2 in sea and atmosphere
N1, N2 = 740, 900
```

```python
# year
year = 1987
# the total molar of material of atmosphere
n0 = 740*10**21/12/347
#  the emission of every year
gamma =data['Total carbon emission in pg']

#%%
# start iteration
for year in range(1986,2005):
    # calculate the change of CO2 in sea and atmosphere
    dN1 = -k12*N1 + k21*N2 + gamma[year-1986]
    dN2 = k12*N1 - k21*N2
    # update the CO2 in sea and atmosphere
    N1 += dN1
    N2 += dN2
    # update the CO2 level in ppm, because 1986 is known,so the value does not change
    if year!=1986:
        data['CO2 level in ppm(no buff)'][year-1986] = N1*10**21/12/n0

# initialize value again to avoid the error by repeating the iteration
N1, N2 = 740, 900

# check data
data
```

output:

1-10 19 rows × 6 columns pd.DataFrame

	Year	Total carbon emissions...	Total carbon emission in pg	observation	CO2 level in ppm(no buff)	CO2 level in ppm(buff)
0	1986	5583	5.583	347.61	347.000000	347.000000
1	1987	5725	5.725	349.31	349.476592	378.373290
2	1988	5936	5.936	351.69	350.822203	384.121646
3	1989	6066	6.066	353.20	352.200804	385.687094
4	1990	6074	6.074	354.45	353.553674	388.594575
5	1991	6142	6.142	355.70	354.915942	391.102546
6	1992	6078	6.078	356.54	356.226925	393.720567
7	1993	6070	6.070	357.21	357.522571	396.294641
8	1994	6174	6.174	358.96	358.858888	398.934102
9	1995	6305	6.305	360.97	360.243684	401.625967

Figure 1: Output of 1.1

Problem1

1.2

```
# 1.2
# create a new column for CO2 level in ppm with buffer
data['CO2 level in ppm(buff)'] = 347
data
#%%
# initialization
# assume that the speed of transfer speed is stationary
k12, k21 = 105/740, 102/900
# the initial value of CO2 in sea and atmosphere and and the equilibrium value of CO2 in sea
N1, N2, N20= 740, 900, 821
# start year
year = 1986
# the total molar of material of atmosphere
n0 = 740*10**21/12/347
# the emission of every year
gamma =data['Total carbon emission in pg']
#%%
# start iteration
for year in range(1986,2005):
    # calculate the buffer factor
    if year == 1986:
        ksi = 3.69 + 1.86*10**(-2)*data['CO2 level in ppm(buff)'][year-1986] - 1.8*10**(-6)*data['CO2 l
    # calculate the change of CO2 in sea and atmosphere
    dN1 = -k12*N1 + k21*(N20 + ksi*(N2 - N20)) + gamma[year-1986]
    dN2 = k12*N1 - k21*(N20 + ksi*(N2 - N20))
    # update the CO2 in sea and atmosphere
    N1 += dN1
    N2 += dN2
    # update the CO2 level in ppm
    if year!=1986:
        data['CO2 level in ppm(buff)'][year-1986] = N1*10**21/12/n0
    # update the buffer factor
    ksi = 3.69 + 1.86*10**(-2)*data['CO2 level in ppm(buff)'][year-1986] - 1.8*10**(-6)*data['CO2 level

# initialize value again to avoid the error by repeating the iteration
N1, N2 = 740, 900

# check data
data
```

output:

1-10 19 rows × 6 columns pd.DataFrame

	Year	Total carbon emiss...	Total carbon emission in pg	observation	CO2 level in ppm(no buff)	CO2 level in ppm(buff)
0	1986	5583	5.583	347.61	347.000000	347.000000
1	1987	5725	5.725	349.31	349.476592	378.373290
2	1988	5936	5.936	351.69	350.822203	384.121646
3	1989	6066	6.066	353.20	352.200804	385.687094
4	1990	6074	6.074	354.45	353.553674	388.594575
5	1991	6142	6.142	355.70	354.915942	391.102546
6	1992	6078	6.078	356.54	356.226925	393.720567
7	1993	6070	6.070	357.21	357.522571	396.294641
8	1994	6174	6.174	358.96	358.858888	398.934102

Figure 2: Output of 1.2

Problem1

1.3

```
# 1.3
# plot the line and scatter
# creat a figure
plt.figure(figsize=(8,6),dpi=120)
plt.plot(data['Year'],data['observation'],linewidth=0,marker='o',markersize=6,color='blue')
plt.plot(data['Year'][1:],data['CO2 level in ppm(buff)'][1:],linewidth=5,color='black')
plt.plot(data['Year'][1:],data['CO2 level in ppm(no buff)'][1:],color='red')

# set the labels
plt.xlabel('Year',fontsize=20,labelpad=10)
plt.ylabel('CO$_2$ Concentration(ppm)',fontsize=20,labelpad=10)

# set the ticks
plt.xticks(np.arange(1985,2005,5),fontsize=15)
tick = np.arange(340, 431, 5)
tick_labels = [t if t in [360, 380, 400,420] else '' for t in tick]
plt.yticks(tick, tick_labels, fontsize=15)
plt.xlim([1984,2005])
plt.tick_params(axis='both', bottom=True, top=False,direction='in', which='major')


# set the text
plt.text(2000, 385, 'Observation', fontsize=15, color='blue')
plt.text(1987, 415, 'calculation with buffer effect', fontsize=15, color='black')
plt.text(1992, 350, 'calculation without buffer effect', fontsize=15, color='red')

# hide the right and top spines
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
```

output:

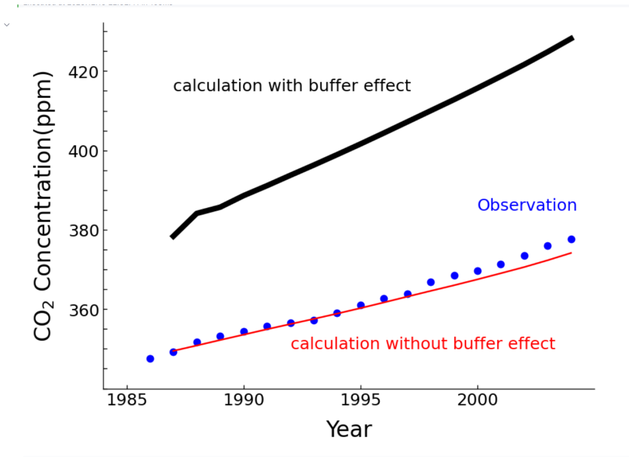

Figure 3: Output of 1.3

Bonus

```
# bonus
# import data
# global fuel emission data
data = pd.read_csv('global.1751_2014.csv')
# observation data
observation = pd.read_csv('trends-in-atmospheric-carbon-dioxide-concentration.csv')
# emission from land use change data
delta =  pd.read_excel('Global_land-use_flux-1850_2005.xls')

# check data
data
#%%
# check data
observation
#%%
# check data
delta
#%%
# cut the unwanted data
observation = observation[(observation['Category']>=1750)&(observation['Category']<=2000)]
# check the result
observation
#%%
# cut the unwanted data  and check data
data = data.iloc[1:,0:2]
data
#%%
# transform the year to int
data['Year'] = data['Year'].astype(int)
#%%
#add a row Of 1750 to correspond to the observation data,
#assuming that the CO2 emission in 1750 is 3 million metric tons of C
data.loc[-1] = [1750,3]
data = data.sort_values('Year').reset_index(drop=True)

# check data
data
#%%
# select the data from 1750 to 2004 and check data
data = data[(data['Year']>=1750)&(data['Year']<=2000)].reset_index(drop=True)

# create new columns for total carbon emission in unit pg
data['Total carbon emission in pg'] =
data['Total carbon emissions from fossil fuel consumption
and cement production (million metric tons of C)'].astype(int)/1000

# create new columns for emission from land use change
data['Emission from land use change'] = delta['Global in pg']

# create a new column for CO2 level
# 288 is the result 615^15/12/n0,n0 = 740*10**21/12/347,
# which is the total molar of material of atmosphere
data['CO2 level in ppm(0.38)'] = 289
data['CO2 level in ppm(0.5)'] = 289
```

```python
# check data
data
#%%
# initialization for beta = 0.38
# assume that the speed of transfer speed is stationary
k12 = 60/615
k21, k23, k24 = 60/842, 9/842, 43/842
k32, k34 = 52/9744, 162/9744
k43, k45 = 205/26280, 0.20/26280
k51 = 0.2/90000000
k67 = 62/731
k71 = 62/1328

# the total molar of material of atmosphere
n0 = 740*10**21/12/347

# the initial value of each reservoir,and the equilibrium value of CO2 in sea
N1, N2, N3, N4, N5, N6, N7 = 615, 842, 9744, 26280, 90000000, 731, 1328
N20 = 842

# beta factor
beta = 0.38

# the initial value of the net primary productivity
f0 = 62

# the initial value of co2 concentration in the atmosphere
P0 = 289

# the fuel emission of every year
gamma = data['Total carbon emission in pg']

# the emission from land use change of every year
delta = data['Emission from land use change']

# start year
year = 1750
#%%
# start iteration
for year in range(1751,2001):
    # calculate the buffer factor
    ksi = 3.69 + 1.86*10**(-2)*data['CO2 level in ppm(0.38)'][year-1751]
    - 1.8*10**(-6)*data['CO2 level in ppm(0.38)'][year-1751]**2
    # calculate co2 flux to the biosphere
    f = f0*(1+beta*math.log(data['CO2 level in ppm(0.38)'][year-1751]/P0))
    # calculate the change of CO2 in sea and atmosphere
    dN1 = -k12*N1 + k21*(N20 + ksi*(N2 - N20)) + gamma[year-1750] - f
    + delta[year-1750]  + k71*N7 + k51*N5
    dN2 = k12*N1 - k21*(N20 + ksi*(N2 - N20)) - k23*N2 + k32*N3 - k24*N2
    dN3 = k23*N2 - k32*N3 - k34*N3 + k43*N4
    dN4 = k24*N2  + k34*N3 - k43*N4 - k45*N4
    dN5 = k45*N4 - k51*N5
    dN6 = f - k67*N6 -2*delta[year-1750]
    dN7 = k67*N6 - k71*N7 + delta[year-1750]
    # update the CO2 in sea and atmosphere
    N1 += dN1
```

```
    N2 += dN2
    N3 += dN3
    N4 += dN4
    N5 += dN5
    N6 += dN6
    N7 += dN7
    # update the CO2 level in ppm
    data['CO2 level in ppm(0.38)'][year-1750] = N1*10**21/12/n0
#%%
# check the result
data
#%%
# initialization for beta = 0.5
# assume that the speed of transfer speed is stationary
k12 = 60/615
k21, k23, k24 = 60/842, 9/842, 43/842
k32, k34 = 52/9744, 162/9744
k43, k45 = 205/26280, 0.20/26280
k51 = 0.2/90000000
k67 = 62/731
k71 = 62/1328

# the total molar of material of atmosphere
n0 = 740*10**21/12/347

# the initial value of each reservoir,and the equilibrium value of CO2 in sea
N1, N2, N3, N4, N5, N6, N7 = 615, 842, 9744, 26280, 90000000, 731, 1328
N20 = 842

# beta factor
beta = 0.50

# the initial value of the net primary productivity
f0 = 62

# the initial value of co2 concentration in the atmosphere
P0 = 289

# the fuel emission of every year
gamma = data['Total carbon emission in pg']

# the emission from land use change of every year
delta = data['Emission from land use change']

# start year
year = 1750
#%%
# start iteration
for year in range(1751,2001):
    # calculate the buffer factor
    ksi = 3.69 + 1.86*10**(-2)*data['CO2 level in ppm(0.5)'][year-1751]
    - 1.8*10**(-6)*data['CO2 level in ppm(0.5)'][year-1751]**2
    # calculate co2 flux to the biosphere
    f = f0*(1+beta*math.log(data['CO2 level in ppm(0.5)'][year-1751]/P0))
    # calculate the change of CO2 in sea and atmosphere
    dN1 = -k12*N1 + k21*(N20 + ksi*(N2 - N20)) + gamma[year-1750] - f + delta[year-1750]
```

```
    + k71*N7 + k51*N5
    dN2 = k12*N1 - k21*(N20 + ksi*(N2 - N20)) - k23*N2 + k32*N3 - k24*N2
    dN3 = k23*N2 - k32*N3 - k34*N3 + k43*N4
    dN4 = k24*N2  + k34*N3 - k43*N4 - k45*N4
    dN5 = k45*N4 - k51*N5
    dN6 = f - k67*N6 -2*delta[year-1750]
    dN7 = k67*N6 - k71*N7 + delta[year-1750]
    # update the CO2 in sea and atmosphere
    N1 += dN1
    N2 += dN2
    N3 += dN3
    N4 += dN4
    N5 += dN5
    N6 += dN6
    N7 += dN7
    # update the CO2 level in ppm
    data['CO2 level in ppm(0.5)'][year-1750] = N1*10**21/12/n0
#%%
# check the result
data
#%%
# plot the line and scatter
# plot the line and scatter
# creat a figure
plt.figure(figsize=(10,6),dpi=120)
plt.plot(observation['Category'],observation['IceCore&MounaLoa'],
linewidth=0,marker='o',markersize=4,color='black')
plt.plot(data['Year'][1:],data['CO2 level in ppm(0.38)'][1:],linewidth=1,color='red')
plt.plot(data['Year'][1:],data['CO2 level in ppm(0.5)'][1:],color='blue')

# set the labels
plt.xlabel('Year',fontsize=15,labelpad=10)
plt.ylabel('CO$_2$ Concentration(ppm)',fontsize=15,labelpad=10)

# set the ticks
tick = np.arange(1750, 2001, 10)
tick_labels = [t if t in np.arange(1800, 2001, 100) else '' for t in tick]
plt.xticks(tick, tick_labels, fontsize=15)
plt.xlim([1750,2009])
tick = np.arange(280, 375, 5)
tick_labels = [t if t in np.arange(280, 361, 20) else '' for t in tick]
plt.yticks(tick, tick_labels, fontsize=15)
plt.ylim([275,375])
plt.tick_params(axis='both', bottom=True, top=False,direction='in', which='major')

# set the text
plt.text(1900, 285, 'observation', fontsize=15, color='black')
plt.text(1825, 300, 'calculation', fontsize=15, color='black')
plt.text(1950, 365, r"$\beta$=0.38", fontsize=15, color='red')
plt.text(2000, 350, r"$\beta$=0.50", fontsize=15, color='blue')
# hide the right and top spines
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
```

output:

1-10 251 rows × 6 columns pd.DataFrame CSV

	Year	Total car...	Total carbon emission in pg	Emission from land use change	CO2 level in ppm(0.38)	CO2 level in ppm(0.5)
0	1750	3	0.003	0.200000	289.000000	289.000000
1	1751	3	0.003	0.203006	288.575519	288.575519
2	1752	3	0.003	0.206012	288.769421	288.774549
3	1753	3	0.003	0.209018	288.953738	288.960893
4	1754	3	0.003	0.212024	289.132713	289.139666
5	1755	3	0.003	0.215030	289.306893	289.311709
6	1756	3	0.003	0.218036	289.476139	289.477062
7	1757	3	0.003	0.221042	289.640240	289.635676
8	1758	3	0.003	0.224048	289.799036	289.787545
9	1759	3	0.003	0.227054	289.952432	289.932727

Figure 4: Output1 of bonous

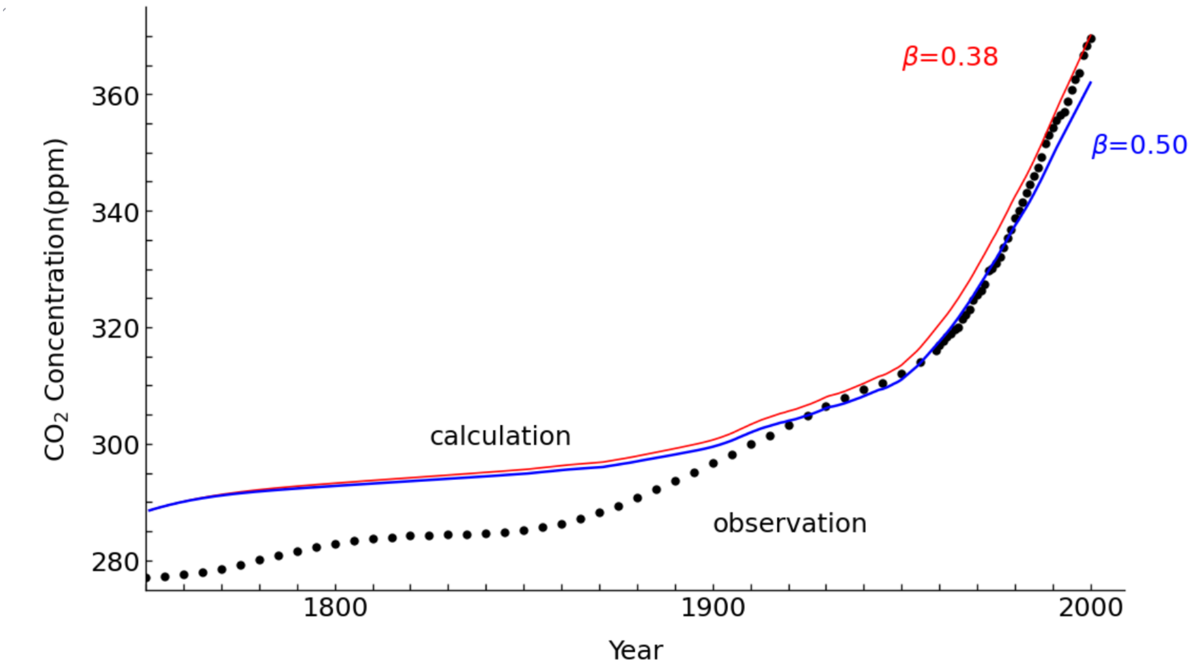

Figure 5: Output2 of bonous