

5.1

导入库

```
import pandas as pd
from sklearn.linear_model import LinearRegression
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
```

读取文件:

```
goble_data=pd.read_csv("global.1751_2014.csv")
```

提取前两列数据:

```
goble_data_convert=goble_data.iloc[:, :2]
```

去掉第一行的内容

```
goble_data_convert_drop=goble_data_convert.drop(0)
```

看一下文件

```
goble_data_convert_drop
```

:

	Year	Total carbon emissions from fossil fuel consumption and cement production (million metric tons of C)
1	1751	3
2	1752	3
3	1753	3
4	1754	3
5	1755	3
...
260	2010	9128
261	2011	9503
262	2012	9673
263	2013	9773
264	2014	9855

264 rows × 2 columns

确保排放量列是浮点类型并直接转换为 pgc

```
goble_data_convert_drop['PGC'] = goale_data_convert_drop['Total carbon emissions from fossil fuel consumption and cement production (million metric tons of C)'].astype(float).apply(lambda x: x * 1e12 / 1e15)
```

astype(float): 确保这一列的数据类型是浮点数 (float)

这一部分将列中的每个值（以百万吨为单位）转换为 petagrams of carbon (PgC)。首先将

百万吨转换为克（乘以 1e12），然后从克转换为 PgC（除以 1e15）。

创建包含特定行的新 DataFrame

```
ppm_df = goble_data_convert_drop[['Year', 'PGC']].loc[237:254]
```

ppm_df

	Year	PGC
237	1987	5.725
238	1988	5.936
239	1989	6.066
240	1990	6.074
241	1991	6.142
242	1992	6.078
243	1993	6.070
244	1994	6.174
245	1995	6.305
246	1996	6.448
247	1997	6.556
248	1998	6.576
249	1999	6.561
250	2000	6.733
251	2001	6.893
252	2002	6.994
253	2003	7.376
254	2004	7.743

初始化变量用于存储结果

```
calculation_without_buffer = []
```

初始条件

```
N1 = 740 # 大气中的碳浓度
```

```
N2 = 900 # 海洋表面的碳浓度
```

```
k12 = 105 / 740 # 大气到海洋的转移系数
```

```
k21 = 102 / 900 # 海洋到大气的转移系数
```

确保 years 的长度与 ppm_df_need 的行数一致

```
num_years = len(ppm_df)
```

模拟计算 1987 年到 2004 年的碳浓度变化

```
for i in range(num_years):
```

```
    year = ppm_df['Year'].iloc[i]
```

```
    gama = ppm_df['PGC'].iloc[i]
```

```
dN1_dt = -k12 * N1 + k21 * N2 + gama
dN2_dt = k12 * N1 - k21 * N2
```

```
# 使用欧拉方法进行数值积分
```

```
dt = 1 # 时间步长为 1 年
```

```
N1 += dN1_dt * dt
```

```
N2 += dN2_dt * dt
```

```
# 存储每年的结果，将单位转换为 ppm 并保存在列表中
```

```
calculation_without_buffer.append(N1 / 2.13)
```

```
# 输出每年的结果，并将单位转换为 ppm
```

```
print(f"年份: {year}, 大气中二氧化碳水平: {N1 / 2.13:.2f} ppm")
```

```
年份: 1987, 大气中二氧化碳水平: 348.70 ppm
年份: 1988, 大气中二氧化碳水平: 350.05 ppm
年份: 1989, 大气中二氧化碳水平: 351.44 ppm
年份: 1990, 大气中二氧化碳水平: 352.80 ppm
年份: 1991, 大气中二氧化碳水平: 354.17 ppm
年份: 1992, 大气中二氧化碳水平: 355.48 ppm
年份: 1993, 大气中二氧化碳水平: 356.78 ppm
年份: 1994, 大气中二氧化碳水平: 358.12 ppm
年份: 1995, 大气中二氧化碳水平: 359.51 ppm
年份: 1996, 大气中二氧化碳水平: 360.95 ppm
年份: 1997, 大气中二氧化碳水平: 362.41 ppm
年份: 1998, 大气中二氧化碳水平: 363.86 ppm
年份: 1999, 大气中二氧化碳水平: 365.28 ppm
年份: 2000, 大气中二氧化碳水平: 366.77 ppm
年份: 2001, 大气中二氧化碳水平: 368.31 ppm
年份: 2002, 大气中二氧化碳水平: 369.87 ppm
年份: 2003, 大气中二氧化碳水平: 371.59 ppm
年份: 2004, 大气中二氧化碳水平: 373.43 ppm
```

```
#选取年份
```

```
years = np.arange(1987, 2005, 1)
```

```
# CO2 浓度数据，单位为 ppm
```

```
co2_concentration_2 = np.array(calculation_without_buffer).reshape(-1, 1)
```

```
# 将年份和 CO2 浓度数据转换为 NumPy 数组，以便进行线性回归
```

```
X = years.reshape(-1, 1) # 转换为二维数组以适应 sklearn 的 API
```

```
y = co2_concentration
```

```
# 创建并拟合线性回归模型
```

```
regressor = LinearRegression()
```

```
regressor.fit(X, y)
```

```
# 使用模型预测
```

```
predicted_co2 = regressor.predict(X)
```

```
# 绘制原始数据和拟合线
```

```
plt.figure(figsize=(11, 6))
```

```
plt.scatter(X, y, label='Atmosphere (N1)')
```

```
plt.plot(X, predicted_co2, color='black', linestyle='--', label='Linear Regression')
```

```
plt.xlabel('Year')
```

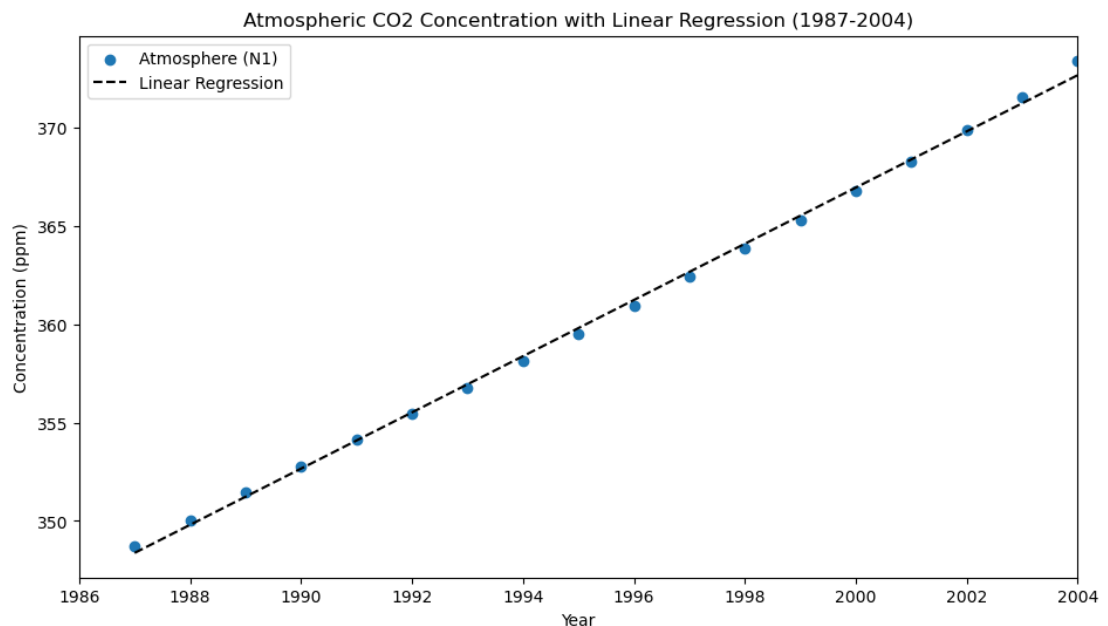
```
plt.ylabel('Concentration (ppm)')
```

```
plt.xlim(1986, 2004)
```

```
plt.title('Atmospheric CO2 Concentration with Linear Regression (1987-2004)')
```

```
plt.legend()
```

```
plt.show()
```



5.2

```
# 初始化变量用于存储结果
```

```
calculation_with_buffer = []
```

```
# 初始条件
```

```
N1 = 740 # 大气中的碳浓度
```

```
N2 = 900 # 海洋表面的碳浓度
```

```
k12 = 105 / 740 # 大气到海洋的转移系数
```

```
k21 = 102 / 900 # 海洋到大气的转移系数
```

```
N20 = 821 # 海洋表层碳的平衡值
```

```
# 模拟计算 1987 年到 2004 年的碳浓度变化
```

```
for i in range(num_years):
```

```
    year = ppm_df['Year'].iloc[i]
```

```
    gama = ppm_df['PGC'].iloc[i]
```

```
    z = N1 / 2.13 # 以 ppm 为单位的大气二氧化碳浓度
```

```
# 计算缓冲系数 xi
```

```
xi = 3.69 + 1.86 * 10**(-2) * z - 1.80 * 10**(-6) * z**2
```

```
dN1_dt = -k12 * N1 + k21 * (N20 + xi * (N2 - N20)) + gama
```

```
dN2_dt = k12 * N1 - k21 * (N20 + xi * (N2 - N20))
```

```
# 使用欧拉方法进行数值积分
```

```
dt = 1 # 时间步长为 1 年
```

```
N1 += dN1_dt * dt
```

```
N2 += dN2_dt * dt
```

```
# 存储每年的结果，将单位转换为 ppm 并保存在列表中
```

```
calculation_with_buffer.append(N1 / 2.13)
```

```
# 输出每年的结果，并将单位转换为 ppm
```

```
print('年份: {year}, 大气中二氧化碳水平: {N1 / 2.13:.2f} ppm')
```

```
年份: 1987, 大气中二氧化碳水平: 386.25 ppm
```

```
年份: 1988, 大气中二氧化碳水平: 379.05 ppm
```

```
年份: 1989, 大气中二氧化碳水平: 384.78 ppm
```

```
年份: 1990, 大气中二氧化碳水平: 386.37 ppm
```

```
年份: 1991, 大气中二氧化碳水平: 389.31 ppm
```

```
年份: 1992, 大气中二氧化碳水平: 391.79 ppm
```

```
年份: 1993, 大气中二氧化碳水平: 394.41 ppm
```

```
年份: 1994, 大气中二氧化碳水平: 397.03 ppm
```

```
年份: 1995, 大气中二氧化碳水平: 399.73 ppm
```

```
年份: 1996, 大气中二氧化碳水平: 402.49 ppm
```

```
年份: 1997, 大气中二氧化碳水平: 405.30 ppm
```

```
年份: 1998, 大气中二氧化碳水平: 408.11 ppm
```

```
年份: 1999, 大气中二氧化碳水平: 410.92 ppm
```

```
年份: 2000, 大气中二氧化碳水平: 413.81 ppm
```

```
年份: 2001, 大气中二氧化碳水平: 416.77 ppm
```

```
年份: 2002, 大气中二氧化碳水平: 419.77 ppm
```

```
年份: 2003, 大气中二氧化碳水平: 422.95 ppm
```

```
年份: 2004, 大气中二氧化碳水平: 426.29 ppm
```

```
# 年份数据
```

```
years = np.arange(1987, 2005, 1)
```

```
# CO2 浓度数据，单位为 ppm
```

```
co2_concentration_2 = np.array(calculation_with_buffer).reshape(-1, 1)
```

```
# 将年份和 CO2 浓度数据转换为 NumPy 数组，以便进行线性回归
```

```
X_2 = years.reshape(-1, 1) # 转换为二维数组以适应 sklearn 的 API
```

```
y_2 = co2_concentration_2
```

```
# 创建并拟合线性回归模型
```

```
regressor = LinearRegression()
```

```
regressor.fit(X_2, y_2)
```

```
# 使用模型预测
```

```
predicted_co2_2 = regressor.predict(X_2)
```

```
# 绘制原始数据和拟合线
```

```
plt.figure(figsize=(11, 6))
```

```
plt.scatter(X_2, y_2, label='Atmosphere (N1)')
```

```
plt.plot(X_2, predicted_co2_2, color='red', linestyle='--', label='Linear Regression')
```

```
plt.xlabel('Year')
```

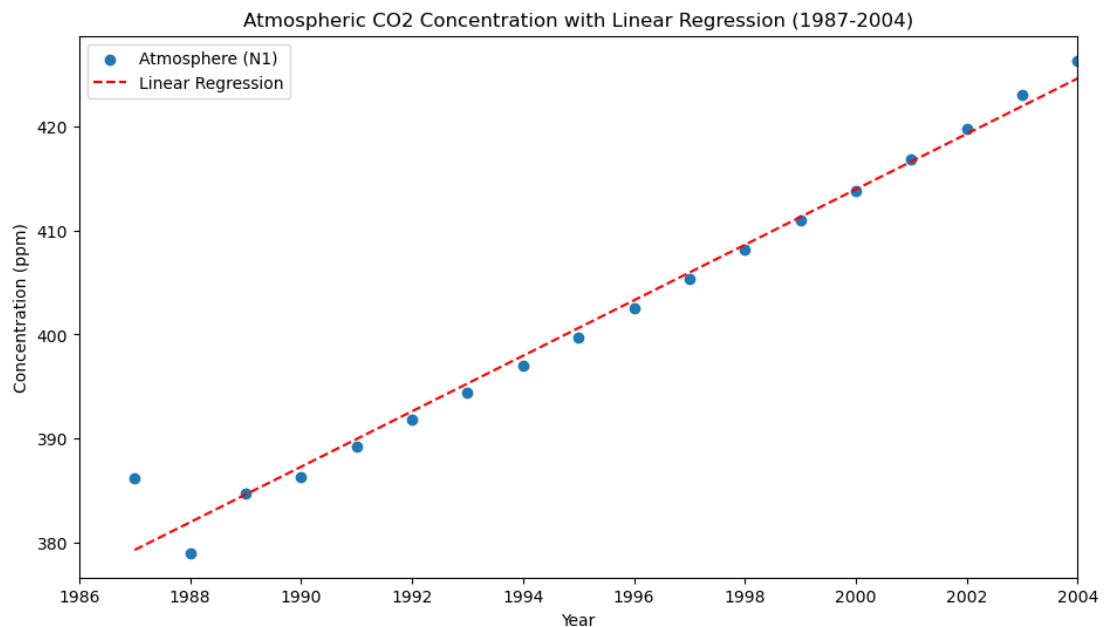
```
plt.ylabel('Concentration (ppm)')
```

```
plt.xlim(1986, 2004)
```

```
plt.title('Atmospheric CO2 Concentration with Linear Regression (1987-2004)')
```

```
plt.legend()
```

```
plt.show()
```



5.3

```
data = pd.read_csv('co2_annmean_mlo.csv', comment='#')
```

这行代码读取名为'co2_annmean_mlo.csv'的 CSV 文件到一个名为 data 的 DataFrame 中。

comment='#'参数告诉 Pandas 忽略以'#'开头的行，因为它们是注释。

```
# 筛选 1986 年到 2004 年的数据
```

```
filtered_data = data[(data['year'] >= 1986) & (data['year'] <= 2004)]
```

```
# 获取筛选后的年份和二氧化碳浓度数据
```

```
years_with_1986 = filtered_data['year']
```

```
observations_with_1986 = filtered_data['mean']
```

这些代码行提取了筛选数据中的年份和对应的平均 CO2 浓度

该脚本使用 plt.scatter 为实际观测值创建一个散点图。

它还使用 plt.plot 绘制了两条折线，代表预测的 CO2 浓度。这些预测可能基于不同的模型或假设（一个考虑缓冲效应，一个不考虑）

```
plt.figure(figsize=(8, 6))
```

```
plt.scatter(years_with_1986, observations_with_1986, color='grey', label='Observations')
```

```
plt.plot(X, predicted_co2, color='grey', linestyle='-', label='calculation without buffer effect',  
linewidth=4)
```

```
plt.plot(X_2, predicted_co2_2, color='black', linestyle='-', label='calculation with buffer effect',  
linewidth=4)
```

```
# 设置刻度朝内
```

```
plt.tick_params(axis='x', direction='in')
```

```
plt.tick_params(axis='y', direction='in', which='both')
```

```
# 设置图表标签和标题
```

```
plt.xlabel('Year', fontsize=14, fontweight='bold')
```

```
plt.ylabel('CO2 Concentration (ppm)', fontsize=14, fontweight='bold')
```

```
plt.title('CO2 Concentration Over Years', fontsize=16, fontweight='bold')
```

```
# 添加网格
```

```
plt.grid(False)
```

```
# Set x-axis ticks with 5-year intervals
```

```
plt.xticks(np.arange(1985, 2006, 5))
```

```
plt.xlim(1985, 2005)
```

```
plt.ylim(340, 430)
```

```
# 添加图例
```

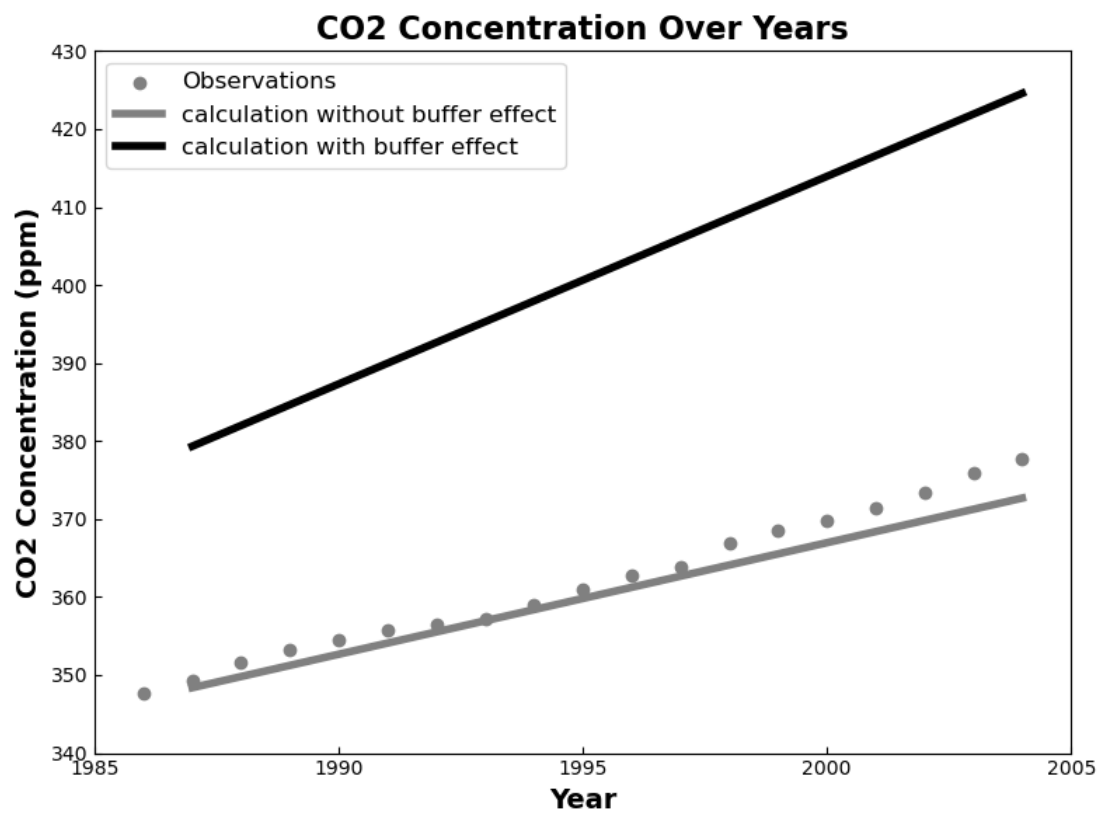
```
plt.legend(fontsize=12)
```

```
# 调整布局
```

```
plt.tight_layout()
```

```
# 显示图表
```

```
plt.show()
```



5.4

该问题我和同学进行讨论，表格数据是经过讨论得到

读取数据

```
co2_observations = pd.read_csv('1750-2000CO2.csv')
land_use_data = pd.read_excel('Global_land-use_flux-1750_2005.xls')
fossil_emissions = pd.read_csv('global_1751_2016.csv')
```

数据预处理

```
land_use_data = land_use_data[['Year', 'Global']]
land_use_data['LandUseChange'] = land_use_data['Global'] / (1000 * 2.13)
```

```
fossil_emissions = fossil_emissions[['Year', 'Total carbon emissions from fossil fuel
consumption and cement production (million metric tons of C)', 'Carbon emissions from
cement production']]
```

```
fossil_emissions['FossilFuelEmissions'] = fossil_emissions.iloc[:, 1] - fossil_emissions.iloc[:, 2]
fossil_emissions['EmissionFactor'] = fossil_emissions['FossilFuelEmissions'] / (1000 * 2.13)
```

参数定义

```
k12, k21, k23, k24, k32, k34, k43, k45, k51, k67, k71 = [60 / 615, 60 / 842, 9 / 842, 43 / 842, 52 / 9744, 162 / 9744, 205 / 26280, 0.2 / 26280, 0.2 / 90000000, 62 / 731, 62 / 1238]
N2_0 = 842 / 2.13
```

初始化

```
initial_conditions = [615 / 2.13, 842 / 2.13, 9744 / 2.13, 26280 / 2.13, 90000000 / 2.13, 731 / 2.13, 1238 / 2.13]
f0 = 62 / 2.13
P0 = 615 / 2.13
```

探索的 Beta 值

```
beta_values = [0.38, 0.5]
results = []
```

```
for beta in beta_values:
```

使用初始条件初始化各个部分

```
N1, N2, N3, N4, N5, N6, N7 = initial_conditions.copy()
atmosphere = [N1]
```

```
for year in range(1751, 2001):
```

获取 gamma 和 delta 值

```
gamma = fossil_emissions[fossil_emissions['Year'] == year]['EmissionFactor'].values[0]
delta = land_use_data[land_use_data['Year'] == year]['LandUseChange'].values[0]
```

计算 xi 值

```
xi = 3.69 + 0.0186 * N1 - 0.0000018 * N1**2
```

```
# 更新 f 值
```

```
f = f0 * (1 + beta * np.log(N1 / P0))
```

```
# 计算各个部分的变化率
```

```
dN1_dt = -k12 * N1 + k21 * (N2_0 + xi * (N2 - N2_0)) + gamma - f + delta + k51 * N5 + k71 * N7
```

```
dN2_dt = k12 * N1 - k21 * (N2_0 + xi * (N2 - N2_0)) - k23 * N2 + k32 * N3 - k24 * N2
```

```
dN3_dt = k23 * N2 - k32 * N3 - k34 * N3 + k43 * N4
```

```
dN4_dt = k34 * N3 - k43 * N4 + k24 * N2 - k45 * N4
```

```
dN5_dt = k45 * N4 - k51 * N5
```

```
dN6_dt = f - k67 * N6 - 2 * delta
```

```
dN7_dt = k67 * N6 - k71 * N7 + delta
```

```
# 更新各个部分的值
```

```
N1 += dN1_dt
```

```
N2 += dN2_dt
```

```
N3 += dN3_dt
```

```
N4 += dN4_dt
```

```
N5 += dN5_dt
```

```
N6 += dN6_dt
```

```
N7 += dN7_dt
```

```
atmosphere.append(N1)
```

```
results.append(atmosphere)
```

```
# 创建图表并设定大小
```

```
plt.figure(figsize=(12, 8))
```

```
# 绘制观测数据的散点图，颜色设置为黑色
```

```
plt.scatter(co2_observations['year'], co2_observations['mean'], label='Observations', color='black')
```

```
# 添加文本标注
```

```
plt.text(1850, 300, 'Calculations', fontsize=12, fontweight='bold')
```

```
plt.text(1900, 290, 'Observations', fontsize=12, fontweight='bold')
```

```
plt.text(1950, 345, 'β=0.38', fontsize=12, fontweight='bold', color='red')
```

```
plt.text(1980, 320, 'β=0.50', fontsize=12, fontweight='bold', color='blue')
```

```
# 直接绘制每个 β 值的结果，颜色分别为红色和蓝色
```

```
plt.plot(range(1750, 2001), results[0], color='red', label='β=0.38')
plt.plot(range(1750, 2001), results[1], color='blue', label='β=0.50')
```

设置图表标签和标题

```
plt.xlabel('Year', fontsize=14, fontweight='bold')
plt.ylabel('CO2 Concentration (ppm)', fontsize=14, fontweight='bold')
plt.title('CO2 Concentration Over Years', fontsize=16, fontweight='bold')
```

显示图例

```
plt.legend()
```

显示图表

```
plt.show()
```

