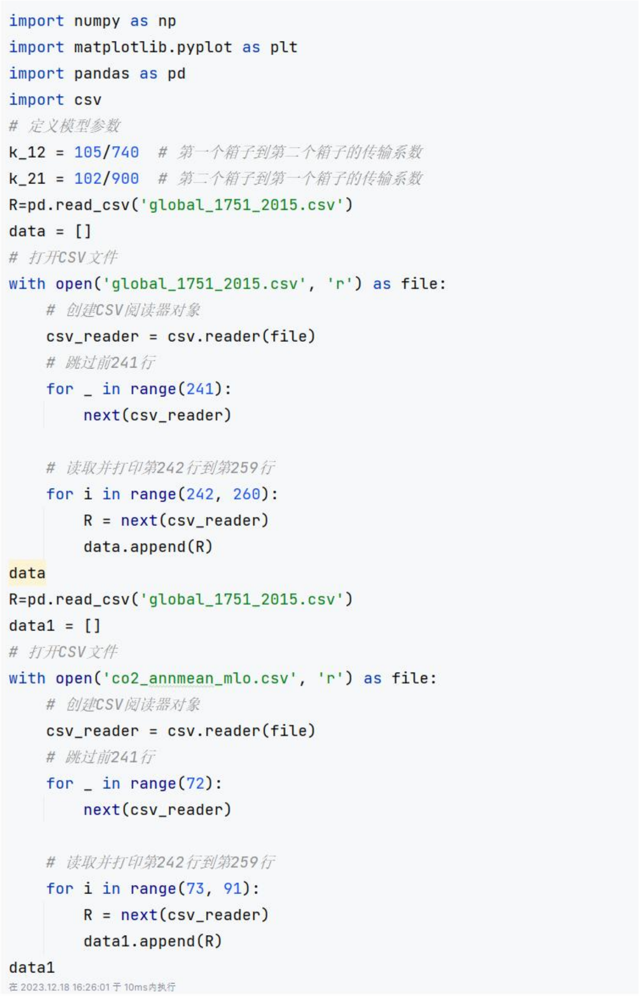
Report

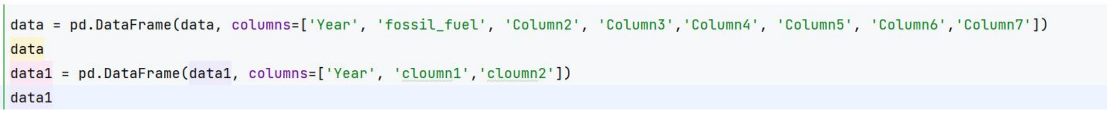
1.1、without the buffer effect



```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import csv
# 定义模型参数
k_12 = 105/740  # 第一个箱子到第二个箱子的传输系数
k_21 = 102/900  # 第二个箱子到第一个箱子的传输系数
R=pd.read_csv('global_1751_2015.csv')
data = []
# 打开CSV文件
with open('global_1751_2015.csv', 'r') as file:
    # 创建CSV阅读器对象
    csv_reader = csv.reader(file)
    # 跳过前241行
    for _ in range(241):
        next(csv_reader)

    # 读取并打印第242行到第259行
    for i in range(242, 260):
        R = next(csv_reader)
        data.append(R)
data
R=pd.read_csv('global_1751_2015.csv')
data1 = []
# 打开CSV文件
with open('co2_annmean_mlo.csv', 'r') as file:
    # 创建CSV阅读器对象
    csv_reader = csv.reader(file)
    # 跳过前241行
    for _ in range(72):
        next(csv_reader)

    # 读取并打印第242行到第259行
    for i in range(73, 91):
        R = next(csv_reader)
        data1.append(R)
data1
```

在 2023.12.18 16:26:01 于 10ms内执行

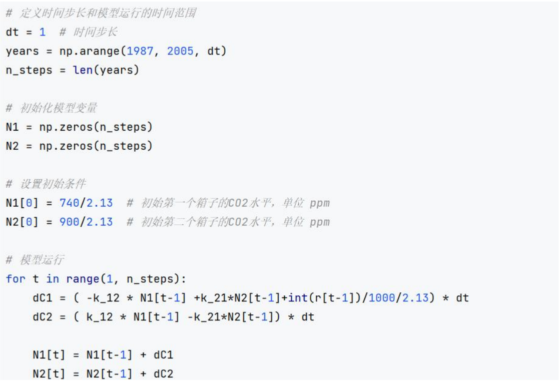
读取 CSV 文件并且存取 1987 年到 2004 年的数据



```python
data = pd.DataFrame(data, columns=['Year', 'fossil_fuel', 'Column2', 'Column3','Column4', 'Column5', 'Column6','Column7'])
data
data1 = pd.DataFrame(data1, columns=['Year', 'cloumn1','cloumn2'])
data1
```

将 fossil fuel 数据以及观测到的数据提取出来

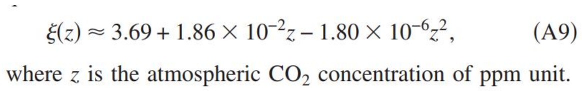
制订箱体模型，设置初始条件，通过 for 循环，实现每年的传输，得出每年二氧化碳浓度的变化。



```python
# 定义时间步长和模型运行的时间范围
dt = 1  # 时间步长
years = np.arange(1987, 2005, dt)
n_steps = len(years)

# 初始化模型变量
N1 = np.zeros(n_steps)
N2 = np.zeros(n_steps)

# 设置初始条件
N1[0] = 740/2.13  # 初始第一个箱子的CO2水平，单位 ppm
N2[0] = 900/2.13  # 初始第二个箱子的CO2水平，单位 ppm

# 模型运行
for t in range(1, n_steps):
    dC1 = ( -k_12 * N1[t-1] +k_21*N2[t-1]+int(r[t-1])/1000/2.13) * dt
    dC2 = ( k_12 * N1[t-1] -k_21*N2[t-1]) * dt

    N1[t] = N1[t-1] + dC1
    N2[t] = N2[t-1] + dC2
```

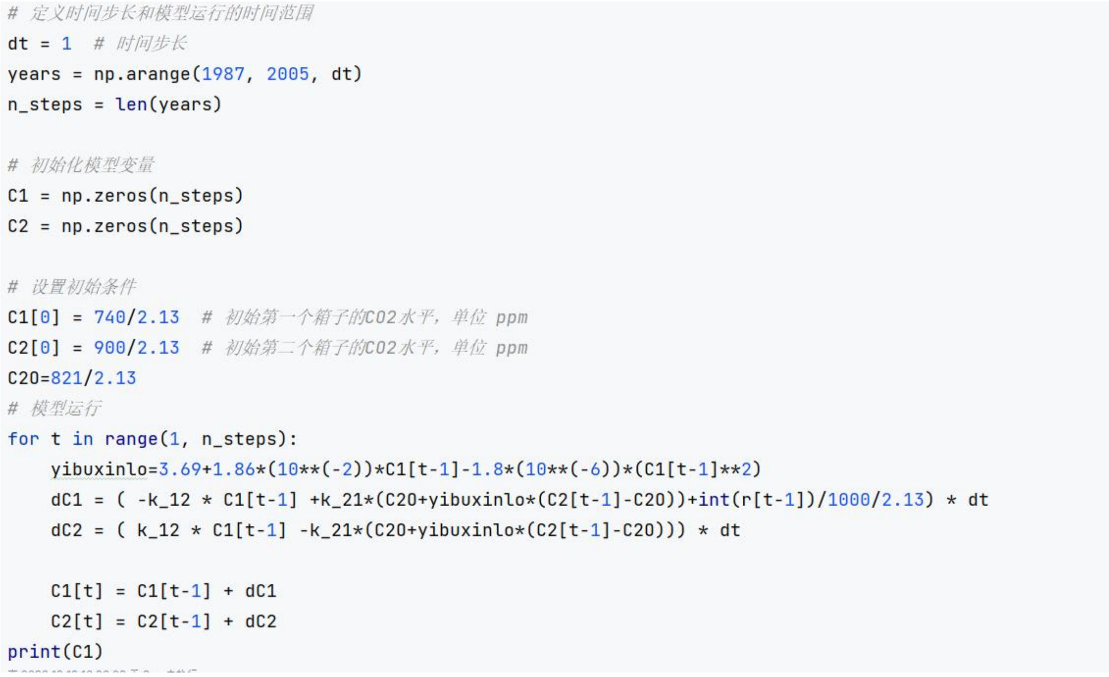
1.2、With the buffer effect

设置初始条件，通过以下公式建立方程



$$\xi(z) \approx 3.69 + 1.86 \times 10^{-2} z - 1.80 \times 10^{-6} z^2, \qquad \text{(A9)}$$

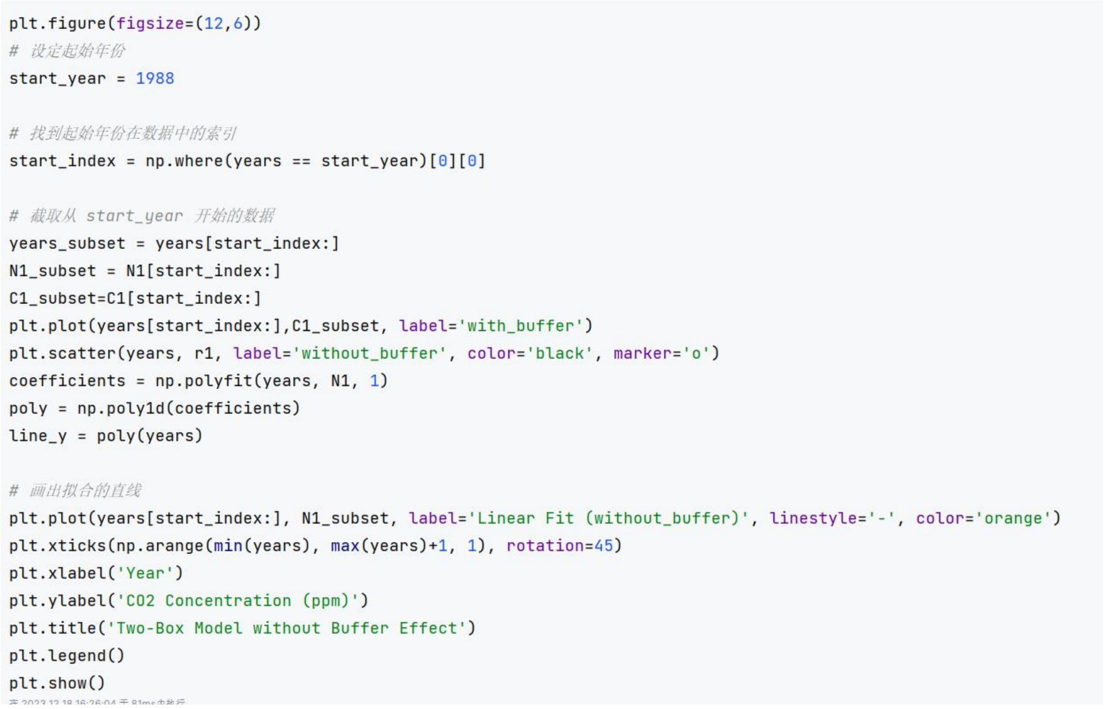
where z is the atmospheric CO_2 concentration of ppm unit.



```
# 定义时间步长和模型运行的时间范围
dt = 1  # 时间步长
years = np.arange(1987, 2005, dt)
n_steps = len(years)

# 初始化模型变量
C1 = np.zeros(n_steps)
C2 = np.zeros(n_steps)

# 设置初始条件
C1[0] = 740/2.13  # 初始第一个箱子的CO2水平，单位 ppm
C2[0] = 900/2.13  # 初始第二个箱子的CO2水平，单位 ppm
C20=821/2.13
# 模型运行
for t in range(1, n_steps):
    yibuxinlo=3.69+1.86*(10**(-2))*C1[t-1]-1.8*(10**(-6))*(C1[t-1]**2)
    dC1 = ( -k_12 * C1[t-1] +k_21*(C20+yibuxinlo*(C2[t-1]-C20))+int(r[t-1])/1000/2.13) * dt
    dC2 = ( k_12 * C1[t-1] -k_21*(C20+yibuxinlo*(C2[t-1]-C20))) * dt

    C1[t] = C1[t-1] + dC1
    C2[t] = C2[t-1] + dC2
print(C1)
```

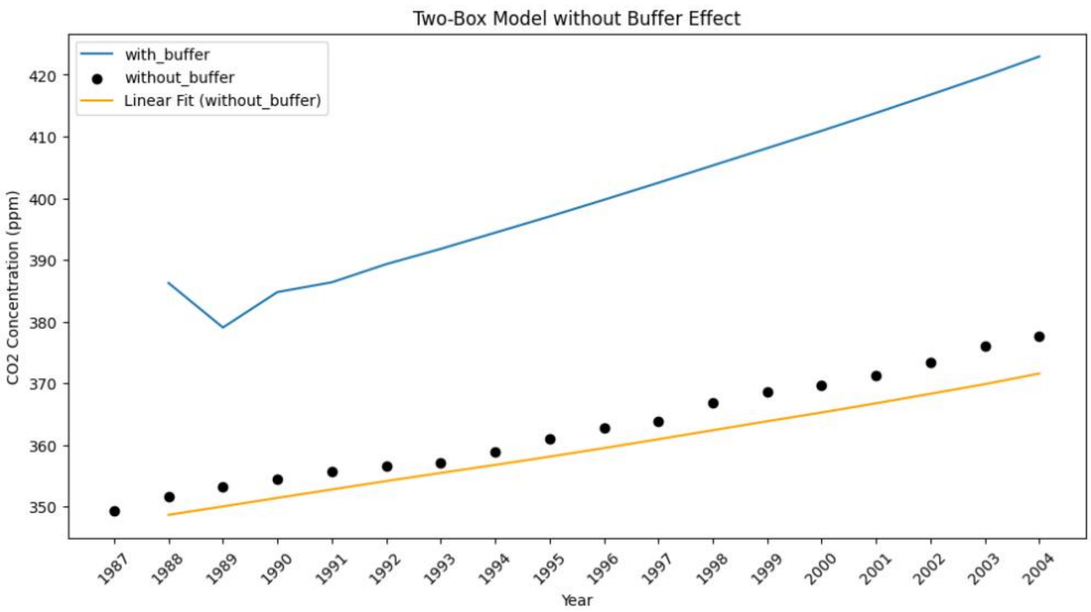
1.3

绘图，将真实观测的二氧化碳浓度绘制为散点图，将上述两个处理以后的 N1,C1 变量绘制出折线图



```
plt.figure(figsize=(12,6))
# 设定起始年份
start_year = 1988

# 找到起始年份在数据中的索引
start_index = np.where(years == start_year)[0][0]

# 截取从 start_year 开始的数据
years_subset = years[start_index:]
N1_subset = N1[start_index:]
C1_subset=C1[start_index:]
plt.plot(years[start_index:],C1_subset, label='with_buffer')
plt.scatter(years, r1, label='without_buffer', color='black', marker='o')
coefficients = np.polyfit(years, N1, 1)
poly = np.poly1d(coefficients)
line_y = poly(years)

# 画出拟合的直线
plt.plot(years[start_index:], N1_subset, label='Linear Fit (without_buffer)', linestyle='-', color='orange')
plt.xticks(np.arange(min(years), max(years)+1, 1), rotation=45)
plt.xlabel('Year')
plt.ylabel('CO2 Concentration (ppm)')
plt.title('Two-Box Model without Buffer Effect')
plt.legend()
plt.show()
```

结果如下


Two-Box Model without Buffer Effect
with_buffer
without_buffer
Linear Fit (without_buffer)
CO2 Concentration (ppm)
420
410
400
390
380
370
360
350
Year
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004