

## 5.1、

首先，从 CSV 文件中读取数据，该文件包含了 1751 年至 2014 年的全球碳排放数据。

接下来，选择数据的前两列，Year（年份）和 Total carbon emissions from fossil fuel consumption and cement production (million metric tons of C)（来自化石燃料消耗和水泥生产的总碳排放量，以百万公吨为单位）。

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

创建一个新的 DataFrame **ppm\_df**，其中只包括年份和单位已转换的碳排放数据，数据范围为从 1987 年到 2004 年。

初始化一些变量，用于存储模拟结果。然后设置初始条件，包括大气中的碳浓度（N1）、海洋表面的碳浓度(N2)以及大气到海洋和海洋到大气的碳转移系数(k12 和 k21)。

使用欧拉方法模拟从 1987 年到 2004 年期间碳浓度的变化。

这段代码的主要目标是模拟大气中的 CO2 浓度随时间的变化，并考察碳排放对大气中 CO2 浓度的影响。模拟结果以每年的时间步长存储在 calculation\_without\_buffer 列表中，并打印出来。

---

```
Year: 1987, Atmospheric CO2 Level: 348.70 ppm
Year: 1988, Atmospheric CO2 Level: 350.05 ppm
Year: 1989, Atmospheric CO2 Level: 351.44 ppm
Year: 1990, Atmospheric CO2 Level: 352.80 ppm
Year: 1991, Atmospheric CO2 Level: 354.17 ppm
Year: 1992, Atmospheric CO2 Level: 355.48 ppm
Year: 1993, Atmospheric CO2 Level: 356.78 ppm
Year: 1994, Atmospheric CO2 Level: 358.12 ppm
Year: 1995, Atmospheric CO2 Level: 359.51 ppm
Year: 1996, Atmospheric CO2 Level: 360.95 ppm
Year: 1997, Atmospheric CO2 Level: 362.41 ppm
Year: 1998, Atmospheric CO2 Level: 363.86 ppm
Year: 1999, Atmospheric CO2 Level: 365.28 ppm
Year: 2000, Atmospheric CO2 Level: 366.77 ppm
Year: 2001, Atmospheric CO2 Level: 368.31 ppm
Year: 2002, Atmospheric CO2 Level: 369.87 ppm
Year: 2003, Atmospheric CO2 Level: 371.59 ppm
Year: 2004, Atmospheric CO2 Level: 373.43 ppm
```

```

# Year data
years = np.arange(1987, 2005, 1)

# CO2 concentration data, unit: ppm
co2_concentration = np.array(calculation_without_buffer).reshape(-1, 1)

# Convert year and CO2 concentration data to NumPy arrays for linear regression
X = years.reshape(-1, 1) # Reshape to a 2D array to fit the sklearn API
y = co2_concentration

# Create and fit a linear regression model
regressor = LinearRegression()
regressor.fit(X, y)

# Predict using the model
predicted_co2 = regressor.predict(X)

# Plot the original data and the fitted line
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()

```

创建一个包含年份数据的 NumPy 数组，从 1987 年到 2004 年，每年递增 1。

创建一个包含 CO2 浓度数据的 NumPy 数组，单位为 ppm。这些数据在之前的代码段中被计算并存储在 `calculation_without_buffer` 列表中。

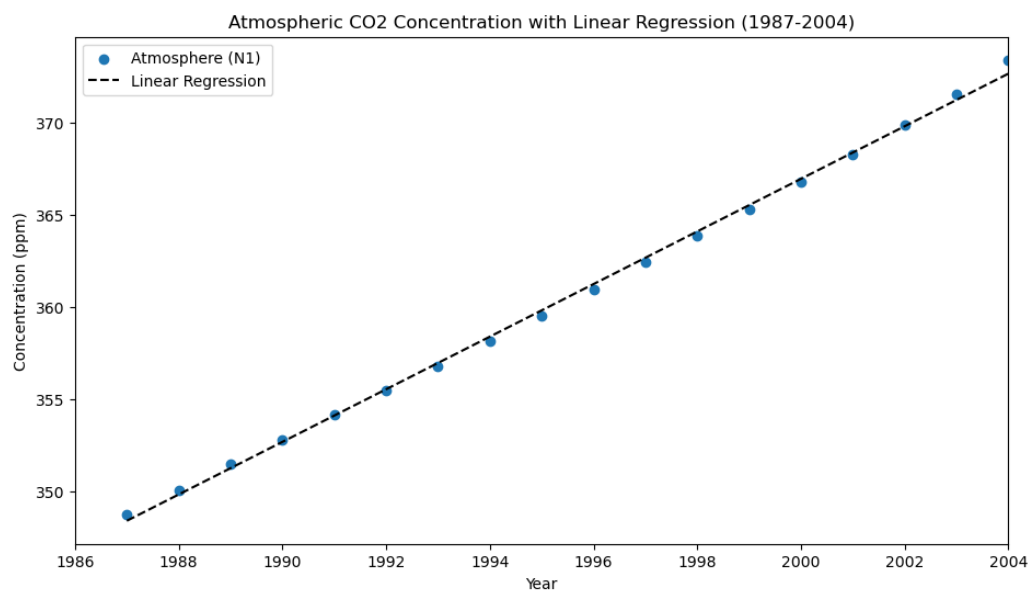
将年份数据和 CO2 浓度数据转换为 NumPy 数组，以便用于线性回归模型。X 是年份数据，y 是 CO2 浓度数据。

创建并拟合一个线性回归模型。线性回归模型会尝试拟合一个线性关系，以预测 CO2 浓度随时间的变化。

使用拟合好的模型进行预测，得到预测的 CO2 浓度值。

绘制散点图以及线性回归拟合线，以可视化模型的性能。原始数据用散点图表示，线性回归拟合线用虚线表示。

这段代码的结果是一个散点图，其中展示了大气 CO2 浓度随时间的变化趋势，以及用线性回归模型拟合的趋势线。



## 5.2、

```
# Initialize variables to store results
calculation_with_buffer = []

# Initial conditions
N1 = 740 # Carbon concentration in the atmosphere
N2 = 900 # Carbon concentration at the ocean surface
k12 = 105 / 740 # Transfer coefficient from atmosphere to ocean
k21 = 102 / 900 # Transfer coefficient from ocean to atmosphere
N20 = 821 # Equilibrium value of carbon at the ocean surface

# Simulate carbon concentration changes from 1987 to 2004
for i in range(num_years):
    year = ppm_df['Year'].iloc[i]
    gama = ppm_df['PGC'].iloc[i]
    z = N1 / 2.13 # Atmospheric CO2 concentration in ppm

    # Calculate the buffer coefficient xi
    xi = 3.69 + 1.86 * 10**(-2) * z - 1.80 * 10**(-6) * z**2

    # Calculate the rate of change of N1 and N2
    dN1_dt = -k12 * N1 + k21 * (N20 + xi * (N2 - N20)) + gama
    dN2_dt = k12 * N1 - k21 * (N20 + xi * (N2 - N20))

    # Numerical integration using Euler's method
    dt = 1 # Time step of 1 year
    N1 += dN1_dt * dt
    N2 += dN2_dt * dt

    # Store the result for each year, converting units to ppm, and save in a list
    calculation_with_buffer.append(N1 / 2.13)

    # Print the result for each year, converting units to ppm
    print(f"Year: {year}, Atmospheric CO2 Level: {N1 / 2.13:.2f} ppm")
```

初始化一个新的空列表 `calculation_with_buffer`, 用于存储具有缓冲效应的模拟结果。

保持之前的初始条件不变, 包括大气中的碳浓度(N1)、海洋表面的碳浓度(N2)、从大气到海洋的转移系数(k12)、从海洋到大气的转移系数(k21)以及海洋表面的碳平衡值(N20)。

在模拟中引入了一个新的变量 `xi`, 用于计算缓冲效应的系数。这个系数会受到大气中 CO2 浓度的影响。

计算 N1 和 N2 的变化率, 考虑了缓冲效应。与之前的模拟不同, 这次计算考虑了缓冲效应对 CO2 转移的影响

使用欧拉方法进行数值积分, 模拟从 1987 年到 2004 年期间的 CO2 浓度变化, 考虑了缓冲效应。

将模拟结果存储在 `calculation_with_buffer` 列表中, 单位转换为 ppm, 并打印每年的结果。

```

Year: 1987, Atmospheric CO2 Level: 386.25 ppm
Year: 1988, Atmospheric CO2 Level: 379.05 ppm
Year: 1989, Atmospheric CO2 Level: 384.78 ppm
Year: 1990, Atmospheric CO2 Level: 386.37 ppm
Year: 1991, Atmospheric CO2 Level: 389.31 ppm
Year: 1992, Atmospheric CO2 Level: 391.79 ppm
Year: 1993, Atmospheric CO2 Level: 394.41 ppm
Year: 1994, Atmospheric CO2 Level: 397.03 ppm
Year: 1995, Atmospheric CO2 Level: 399.73 ppm
Year: 1996, Atmospheric CO2 Level: 402.49 ppm
Year: 1997, Atmospheric CO2 Level: 405.30 ppm
Year: 1998, Atmospheric CO2 Level: 408.11 ppm
Year: 1999, Atmospheric CO2 Level: 410.92 ppm
Year: 2000, Atmospheric CO2 Level: 413.81 ppm
Year: 2001, Atmospheric CO2 Level: 416.77 ppm
Year: 2002, Atmospheric CO2 Level: 419.77 ppm
Year: 2003, Atmospheric CO2 Level: 422.95 ppm
Year: 2004, Atmospheric CO2 Level: 426.29 ppm

```

```

# Year data
years = np.arange(1987, 2005, 1)

# CO2 concentration data, unit: ppm
co2_concentration_2 = np.array(calculation_with_buffer).reshape(-1, 1)

# Convert year and CO2 concentration data to NumPy arrays for linear regression
X_2 = years.reshape(-1, 1) # Reshape to a 2D array to fit the sklearn API
y_2 = co2_concentration_2

# Create and fit a linear regression model
regressor = LinearRegression()
regressor.fit(X_2, y_2)

# Predict using the model
predicted_co2_2 = regressor.predict(X_2)

# Plot the original data and the fitted line
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()

```

创建一个包含年份数据的 NumPy 数组，从 1987 年到 2004 年，每年递增 1，与之前的代码相同。

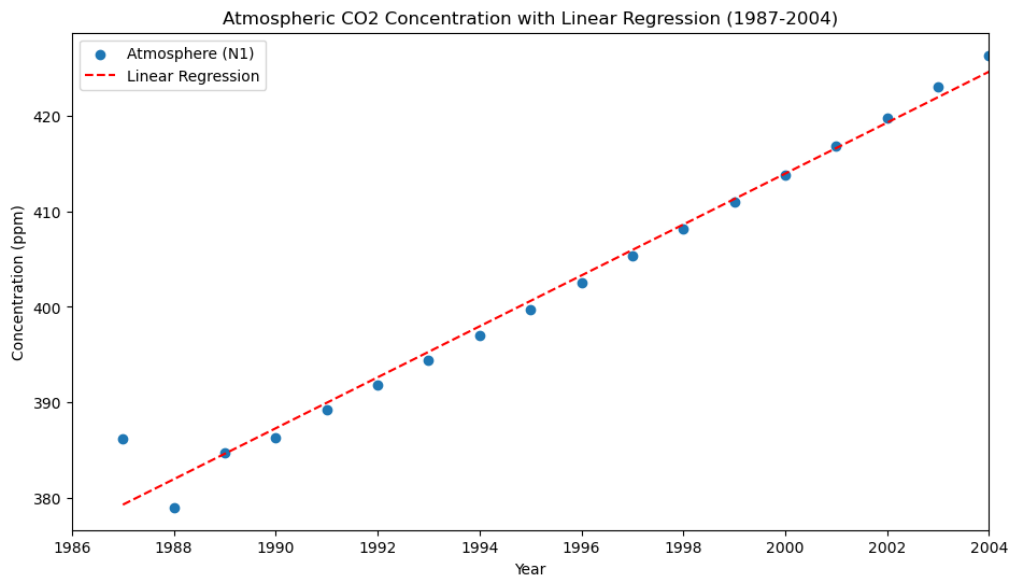
创建一个包含考虑了缓冲效应的 CO2 浓度数据的 NumPy 数组，单位为 ppm。这些数据是在之前的模拟中计算并存储在 calculation\_with\_buffer 列表中的。

将年份数据和 CO2 浓度数据转换为 NumPy 数组，以便用于线性回归模型。X\_2 是年份数据，y\_2 是考虑了缓冲效应的 CO2 浓度数据。

创建并拟合一个线性回归模型，与之前的代码相同

使用拟合好的模型进行预测，得到考虑了缓冲效应的 CO2 浓度的预测值

绘制散点图以及线性回归拟合线，与之前的代码相同。原始数据用散点图表示，线性回归拟合线用红色虚线表示。



5.3、

过滤数据，仅选择年份在 1986 到 2004 之间的数据。

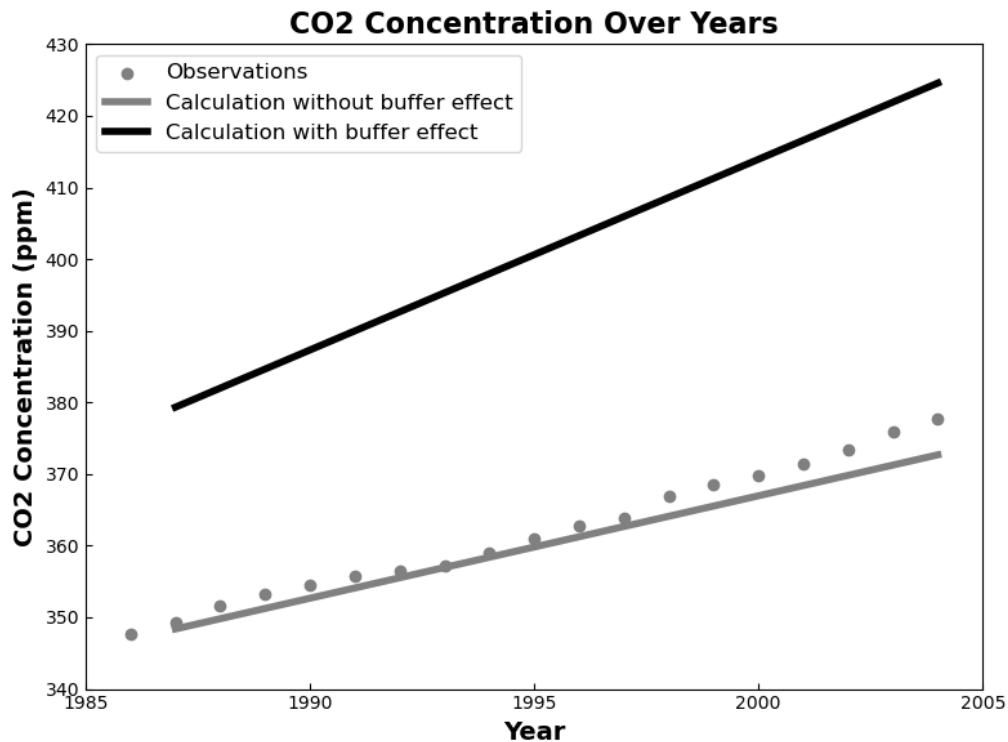
获取筛选后的数据中的年份和 CO2 浓度观测数据。

设置图形的大小和其他绘图参数。

绘制 CO2 浓度的观测值作为散点图。

绘制未考虑缓冲效应的 CO2 浓度模拟结果，以灰色实线表示。绘制考虑了缓冲效应的 CO2 浓度模拟结果，以黑色实线表示。设置坐标轴刻度方向为向内。设置坐标轴标签和图表标题。

设置 x 轴刻度以 5 年间隔显示，并调整 x 轴和 y 轴的范围



#### 5.4、数据读取：

co2\_observations: 读取 1750 年至 2000 年的二氧化碳观测数据。

land\_use\_data: 读取相同时间段内的全球土地使用数据。

fossil\_emissions: 读取 1751 年至 2016 年的全球化石燃料消耗和水泥生产的碳排放数据。

数据预处理：

土地使用数据和化石燃料排放数据被筛选和转换，以便用于后续的计算。

参数定义：

定义了一系列用于模型计算的参数，如各种碳循环过程的速率常数。

初始化：

设置初始条件，包括各个碳储库的初始浓度。

探索 Beta 值：

对两个不同的 Beta 值（0.38 和 0.50）进行迭代计算。Beta 值是影响二氧化碳浓度响应的一个参数。

模型计算：

对 1751 年至 2000 年的每一年进行迭代，计算每年的二氧化碳浓度变化。

使用化石燃料排放和土地使用变化的数据来更新二氧化碳浓度。

考虑不同过程（如植被同化、海洋吸收）对二氧化碳浓度的影响。

绘制图表：

使用 matplotlib 库绘制观测数据和模型预测的二氧化碳浓度随时间的变化图。

在图中标注不同的数据点和文本，如实际观测值、计算值和 Beta 值。

显示图表：

最后，显示所绘制的图表，展示不同 Beta 值下二氧化碳浓度的变化趋势。

此代码的重点在于通过结合实际观测数据和模型计算，分析二氧化碳浓度随时间的变化趋势，以及不同参数设置对预测结果的影响。

