In [ ]:

```
#1
import pandas as pd
import os

# 定义数据文件夹路径
data_folder = 'E:/ese5023/homework/assignment5/Global Fossil-Fuel CO2 Emissions/data/CSV-FILES/CSV-FILES'

# 检查全球化石燃料CO₂排放数据文件夹是否存在
if not os.path.exists(data_folder):
    raise FileNotFoundError(f"全球化石燃料CO₂排放数据文件夹 {data_folder} 不存在，请检查路径。")

# 用于存储全球化石燃料CO₂排放数据的列表
fossil_fuels_data_list = []

# 读取全球化石燃料CO₂排放数据
for root, dirs, files in os.walk(data_folder):
    for file in files:
        if file.endswith('.csv'):
            file_path = os.path.join(root, file)
            try:
                data = pd.read_csv(file_path)
                fossil_fuels_data_list.append(data)
                print(f"文件 {file} 的列名：{data.columns.tolist()}")
            except FileNotFoundError:
                print(f"警告：全球化石燃料数据文件 {file_path} 读取失败，请检查文件格式和内容。")
            except pd.errors.ParserError:
                print(f"警告：全球化石燃料数据文件 {file_path} 解析出错，请检查文件内容是否符合CSV格式规范。")

# 合并全球化石燃料CO₂排放数据
if fossil_fuels_data_list:
    fossil_fuels_data = pd.concat(fossil_fuels_data_list, ignore_index=True)
else:
    raise FileNotFoundError("未找到任何全球化石燃料CO₂排放数据文件。")

# 检查数据类型
print(fossil_fuels_data.dtypes)

# 转换数据类型为数值类型
fossil_fuels_data['Total carbon emissions from fossil-fuels (million metric tons of C)'] = pd.to_numeric(fossil_fuels_data['To
```

```
# 检查NaN值
print(fossil_fuels_data['Total carbon emissions from fossil-fuels (million metric tons of C)'].isnull().sum())

# 填充NaN值
fossil_fuels_data['Total carbon emissions from fossil-fuels (million metric tons of C)'].fillna(0, inplace=True)

# 计算每年的排放量变化
fossil_fuels_data['Annual_Emission'] = fossil_fuels_data['Total carbon emissions from fossil-fuels (million metric tons of C)'

# 重命名列，使其更具可读性
fossil_fuels_data.rename(columns={
    'Year': '年份',
    'Total carbon emissions from fossil-fuels (million metric tons of C)': '化石燃料总碳排放量',
    'Annual_Emission': '年排放量变化'
}, inplace=True)

# 检查数据是否成功读取和处理
print(fossil_fuels_data.head())
```

In [ ]:
```
import pandas as pd

# 尝试读取CSV文件，跳过文件开头的注释行，并跳过有问题的数据行
try:
    mauna_loa_data = pd.read_csv('E:/ese5023/homework/assignment5/co2_annmean_mlo.csv', parse_dates=['year'], skiprows=43, on_
    print(mauna_loa_data.head())
except Exception as e:
    print(f"警告：莫纳罗亚CO₂年平均数据文件读取失败，错误：{e}")
```

In [ ]:
```
#1.1
import numpy as np
import pandas as pd

# 读取全球化石燃料CO₂排放数据
data_folder = 'E:/ese5023/homework/assignment5/Global Fossil-Fuel CO2 Emissions/data/CSV-FILES/CSV-FILES'
fossil_fuels_data_list = []

for root, dirs, files in os.walk(data_folder):
    for file in files:
        if file.endswith('.csv'):
```

```
            file_path = os.path.join(root, file)
            try:
                data = pd.read_csv(file_path)
                fossil_fuels_data_list.append(data)
            except Exception as e:
                print(f"警告：文件 {file_path} 读取失败，错误：{e}")

# 合并全球化石燃料CO₂排放数据
if fossil_fuels_data_list:
    fossil_fuels_data = pd.concat(fossil_fuels_data_list, ignore_index=True)
else:
    raise FileNotFoundError("未找到任何全球化石燃料CO₂排放数据文件。")

# 检查数据类型并转换为数值类型
cols_to_convert = [
    '化石燃料总碳排放量',  # 正确的列名
    'carbon emissions from gas fuel consumption',
    'carbon emissions from liquid fuel consumption',
    'carbon emissions from solid fuel consumption',
    'carbon emissions from cement production',
    'carbon emissions from gas flaring',
    'Per capita carbon emissions (metric tons of carbon; after 1949 only)',
    'Total CO2 emissions from fossil-fuels (thousand metric tons of C)',
    'Emissions from solid fuel consumption',
    'Emissions from liquid fuel consumption',
    'Emissions from gas fuel consumption',
    'Emissions from cement production',
    'Emissions from gas flaring',
    'Per capita CO2 emissions (metric tons of carbon)',
    'Emissions from bunker fuels (not included in the totals)',
    'Emissions from gas fuel consum\nption',
    'Total CO2 emissions from fossil-fuels and cement production (thousand metric tons of C)'
]
for col in cols_to_convert:
    fossil_fuels_data[col] = pd.to_numeric(fossil_fuels_data[col], errors='coerce')

# 填充NaN值
fossil_fuels_data.fillna(0, inplace=True)

# 定义模型参数
k_oa = 0.1  # 大气与海洋之间的交换系数
```

```
C_o = 280  # 海洋中的初始CO₂浓度

# 初始化大气CO₂浓度数组
C_a_no_buffer = np.zeros(len(fossil_fuels_data))
C_a_no_buffer[0] = mauna_loa_data['mean'][0]  # 使用初始观测值作为起始浓度

# 使用欧拉法求解微分方程
for i in range(1, len(fossil_fuels_data)):
    E = fossil_fuels_data['化石燃料总碳排放量'][i]
    C_a_no_buffer[i] = C_a_no_buffer[i - 1] + (E - k_oa * (C_a_no_buffer[i - 1] - C_o)) * 1
```

In [ ]:

```
#1.2
# 定义计算缓冲系数的函数（根据Tomizuka 2009文中公式）
def buffer_coefficient(C_a):
    return k_oa / (1 + k_oa * C_a / (C_o * (1 + k_oa)))

# 初始化大气CO₂浓度数组
C_a_with_buffer = np.zeros(len(fossil_fuels_data))
C_a_with_buffer[0] = mauna_loa_data['CO2'][0]

# 使用欧拉法求解微分方程（考虑缓冲效应）
for i in range(1, len(fossil_fuels_data)):
    E = fossil_fuels_data['Annual_Emission'][i]
    k_oa_buffer = buffer_coefficient(C_a_with_buffer[i - 1])
    C_a_with_buffer[i] = C_a_with_buffer[i - 1] + (E - k_oa_buffer * (C_a_with_buffer[i - 1] - C_o)) * 1
```

In [ ]:

```
#1.3
from sklearn.metrics import mean_squared_error, mean_absolute_error

# 计算无缓冲效应模型的均方误差和平均绝对误差
mse_no_buffer = mean_squared_error(mauna_loa_data['CO2'], C_a_no_buffer)
mae_no_buffer = mean_absolute_error(mauna_loa_data['CO2'], C_a_no_buffer)

# 计算有缓冲效应模型的均方误差和平均绝对误差
mse_with_buffer = mean_squared_error(mauna_loa_data['CO2'], C_a_with_buffer)
mae_with_buffer = mean_absolute_error(mauna_loa_data['CO2'], C_a_with_buffer)

print(f"无缓冲效应模型 - MSE: {mse_no_buffer}, MAE: {mae_no_buffer}")
print(f"有缓冲效应模型 - MSE: {mse_with_buffer}, MAE: {mae_with_buffer}")
```

```
import matplotlib.pyplot as plt

# 绘制复现Figure 2的图表
plt.figure(figsize=(12, 6))
plt.plot(fossil_fuels_data['Date'], C_a_no_buffer, label='无缓冲效应模型')
plt.plot(fossil_fuels_data['Date'], C_a_with_buffer, label='有缓冲效应模型')
plt.plot(mauna_loa_data['Date'], mauna_loa_data['CO2'], label='莫纳罗亚观测数据', marker='o')
plt.xlabel('年份')
plt.ylabel('大气CO₂浓度 (ppm)')
plt.title('大气CO₂浓度随时间的变化（1987 - 2004）')
plt.legend()
plt.show()
```

In [ ]:
```
plt.figure(figsize=(12, 6))
plt.plot(fossil_fuels_data['Date'], C_a_extended, label='扩展模型')
plt.plot(fossil_fuels_data['Date'], C_a_no_buffer, label='无缓冲效应模型')
plt.plot(fossil_fuels_data['Date'], C_a_with_buffer, label='有缓冲效应模型')
plt.xlabel('年份')
plt.ylabel('大气CO₂浓度 (ppm)')
plt.title('不同模型下大气CO₂浓度随时间的变化（1987 - 2004）')
plt.legend()
plt.show()
```