In [10]:
```
# Assignment 05 by Yi Yang
import scipy
import pandas as pd
import numpy as np
from scipy import constants
from scipy.interpolate import UnivariateSpline
from numpy import exp
from scipy import integrate
from scipy import optimize
import numpy as np
import matplotlib.pyplot as plt
import numpy as np
import xlrd
%matplotlib inline
from scipy.integrate import odeint
```

In [12]:
```
# 1.1 Build a two-box model to compute the atmospheric CO2 level in ppm (parts p
df1=pd.read_csv('global.1751_2014.csv')
df1
```

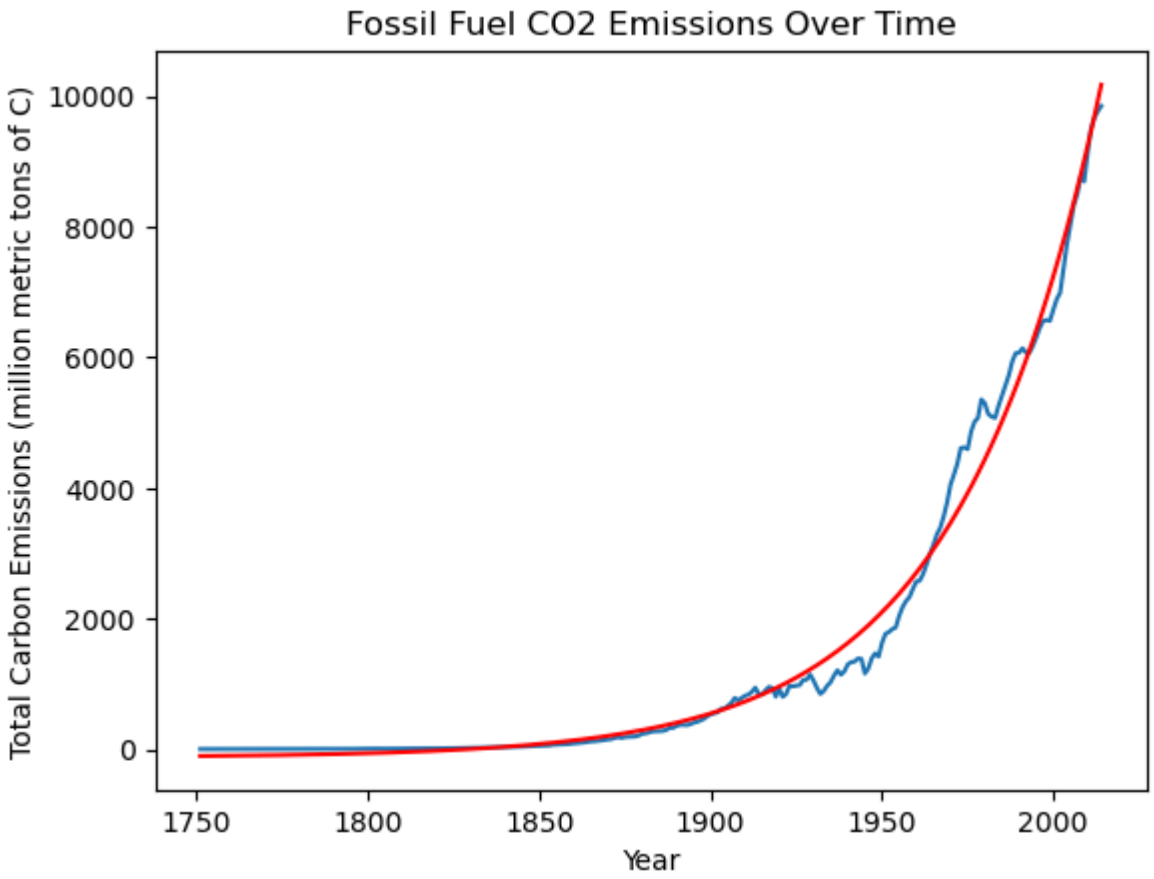
Out[12]:

	Year	Total carbon emissions from fossil fuel consumption and cement production (million metric tons of C)	Carbon emissions from gas fuel consumption	Carbon emissions from liquid fuel consumption	Carbon emissions from solid fuel consumption	Carbon emissions from cement production	e
0	1751	3	0	0	3	0	
1	1752	3	0	0	3	0	
2	1753	3	0	0	3	0	
3	1754	3	0	0	3	0	
4	1755	3	0	0	3	0	
...	...	...	...	...	...	...	
259	2010	9128	1696	3107	3812	446	
260	2011	9503	1756	3134	4055	494	
261	2012	9673	1783	3200	4106	519	
262	2013	9773	1806	3220	4126	554	
263	2014	9855	1823	3280	4117	568	

264 rows × 8 columns

In [18]:
```
# 本题我请教了我的师姐龙师倩
# 定义拟合函数，使用指数模型拟合化石燃料排放的二氧化碳随时间变化的曲线
def fossil_emiss(x, a1, a2, a3):
    """
    拟合函数，形式为 y = exp(a1*x + a2) + a3
```

```
    :param x: 年份
    :param a1: 指数模型的系数
    :param a2: 指数模型的指数
    :param a3: 指数模型的偏移量
    :return: 拟合的二氧化碳排放量
    """
    return np.exp(a1 * x + a2) + a3

# 设置拟合的初始值
a1 = 0.1  # 指数系数的初始值
a2 = 0.1  # 指数的初始值
a3 = 0     # 偏移量的初始值
p0 = [a1, a2, a3]  # 初始参数值列表

# 调用拟合函数，确保数据列的数据类型为整数
df1['Year'].astype(int)
df1['Total carbon emissions from fossil fuel consumption and cement production (

# 使用curve_fit进行曲线拟合
para,cov = optimize.curve_fit(fossil_emiss, df1['Year'], df1['Total carbon emiss

# 绘制真实数据点
plt.plot(df1['Year'],df1['Total carbon emissions from fossil fuel consumption an
# 绘制拟合曲线
plt.plot(df1['Year'], fossil_emiss(df1['Year'], *para), '-', label='Fitted Curve

# 添加图表标题和标签
plt.title('Fossil Fuel CO2 Emissions Over Time')
plt.xlabel('Year')
plt.ylabel('Total Carbon Emissions (million metric tons of C)')

plt.show()
print(para)
```

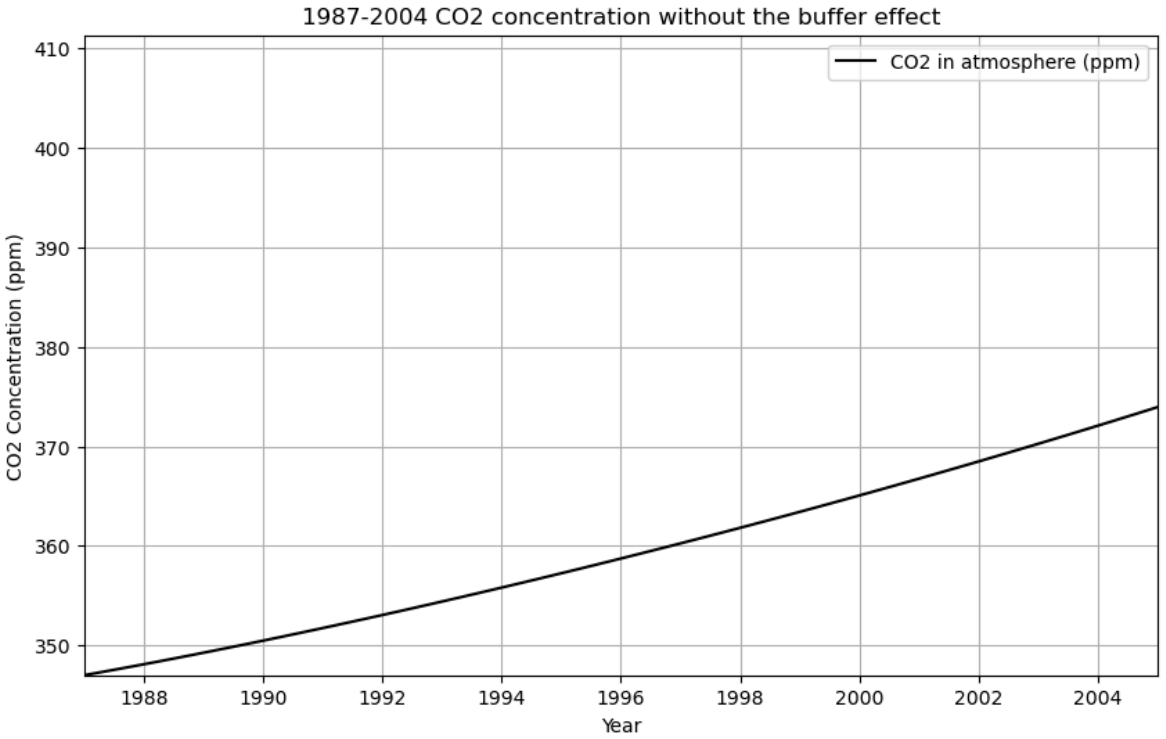



```
[ 2.39596800e-02 -3.90147674e+01 -1.22229282e+02]
```

In [40]:

```python
# https://zhuanlan.zhihu.com/p/484167038
# 定义没有buffer效应的碳循环函数
def model1(y,t,k12,k21):
    """
    碳循环模型，不考虑buffer效应。
    :param y: 状态变量数组，包含大气中的CO2浓度(N1)，海洋中的CO2浓度(N2)，以及γ(a
    :param t: 时间变量。
    :param k12: 从大气到海洋的CO2转移率。
    :param k21: 从海洋到大气的CO2转移率。
    :return: 状态变量的导数数组。
    """
    N1,N2,a = y #a是γ
    dydt=[-k12*N1+k21*N2+a,k12*N1-k21*N2,2.39596800e-02*a+2.39596800e-02*1.22229
    return dydt

# 设置初始值
t1=np.linspace(1987, 2005)
a=fossil_emiss(t1,*para)  # 使用之前拟合的函数计算γ值
k12=105/740   # 从大气到海洋的CO2转移率
k21=102/900   # 从海洋到大气的CO2转移率
N1=740*1000   # 大气中的CO2浓度初始值
N2=900*1000   # 海洋中的CO2浓度初始值
y0=[N1,N2,a[0]]

# 解决常微分方程
sol = odeint(model1, y0, t1, args=(k12,k21))/1000/740*347  # 转换单位

plt.figure(figsize=(10, 6))  # 设置图表大小
plt.plot(t1, sol[:, 0], 'k', label='CO2 in atmosphere (ppm)')  # 绘制大气中的CO2
# 添加图例
plt.legend(loc='best')
```

```
# 设置横坐标轴刻度为整数
plt.gca().xaxis.set_major_locator(MaxNLocator(integer=True))
# 设置图表标题和坐标轴标签
plt.xlabel('Year')
plt.ylabel('CO2 Concentration (ppm)')
# 设置横坐标轴的范围
plt.xlim([t1[0], t1[-1]])
# 设置纵坐标轴的范围
plt.ylim([sol[:, 0].min(), sol[:, 0].max() * 1.1])  # 纵坐标轴范围为最小值到最大
plt.grid(True)
plt.title('1987-2004 CO2 concentration without the buffer effect')
plt.show()
```



In [42]:

```
# 1.2 Build a two-box model to compute the atmospheric CO2 level in ppm from 198
# 定义有buffer效应的函数
def model2(y,t,k12,k21,N0):
    """
    碳循环模型，考虑buffer效应。
    :param y: 状态变量数组，包含大气中的CO2浓度(N1)，海洋中的CO2浓度(N2)，以及γ(a
    :param t: 时间变量。
    :param k12: 从大气到海洋的CO2转移率。
    :param k21: 从海洋到大气的CO2转移率。
    :param N0: 海洋中的CO2平衡浓度。
    :return: 状态变量的导数数组。
    """
    N1,N2,a2 = y
    bf=3.69+1.86e-2*(N1/740/1000*347)-1.8e-6*((N1/740/1000*347)**2) #buffer fact
    dydt=[-k12*N1+k21*(N0+bf*(N2-N0))+a2,
          k12*N1-k21*(N0+bf*(N2-N0)),
          2.39596800e-02*a2+2.39596800e-02*1.22229282e+02]
    return dydt
# 设置初始值
t2=np.linspace(0, 253, 253)+1751
a2=fossil_emiss(t2,*para)

k12=105/740
k21=102/900
```

```
N0=821*1000
N1=618*1000
N2=821*1000
y0=[N1,N2,a2[0]]

# 解决常微分方程，并且转化为ppm的单位
sol2 = odeint(model2, y0, t2, args=(k12,k21,N0))/740/1000*347 #计算微分方程，并且
sol2
```

Out[42]:
```
array([[ 2.89791892e+02,  3.84982432e+02, -4.84577482e-02],
       [ 2.91238045e+02,  3.83487737e+02, -4.82420907e-02],
       [ 2.91660746e+02,  3.83016713e+02, -4.80211828e-02],
       [ 2.91763603e+02,  3.82865757e+02, -4.77948966e-02],
       [ 2.91766902e+02,  3.82814589e+02, -4.75631011e-02],
       [ 2.91739337e+02,  3.82794521e+02, -4.73256623e-02],
       [ 2.91702311e+02,  3.82784156e+02, -4.70824426e-02],
       [ 2.91662489e+02,  3.82776834e+02, -4.68333014e-02],
       [ 2.91621951e+02,  3.82770480e+02, -4.65780946e-02],
       [ 2.91581350e+02,  3.82764449e+02, -4.63166743e-02],
       [ 2.91540891e+02,  3.82758540e+02, -4.60488894e-02],
       [ 2.91500644e+02,  3.82752693e+02, -4.57745849e-02],
       [ 2.91460633e+02,  3.82746889e+02, -4.54936021e-02],
       [ 2.91420870e+02,  3.82741121e+02, -4.52057783e-02],
       [ 2.91381363e+02,  3.82735390e+02, -4.49109471e-02],
       [ 2.91342120e+02,  3.82729695e+02, -4.46089378e-02],
       [ 2.91303146e+02,  3.82724038e+02, -4.42995755e-02],
       [ 2.91264449e+02,  3.82718418e+02, -4.39826815e-02],
       [ 2.91226034e+02,  3.82712838e+02, -4.36580724e-02],
       [ 2.91187909e+02,  3.82707298e+02, -4.33255603e-02],
       [ 2.91150082e+02,  3.82701798e+02, -4.29849528e-02],
       [ 2.91112558e+02,  3.82696341e+02, -4.26360526e-02],
       [ 2.91075346e+02,  3.82690926e+02, -4.22786576e-02],
       [ 2.91038453e+02,  3.82685556e+02, -4.19125612e-02],
       [ 2.91001886e+02,  3.82680231e+02, -4.15375517e-02],
       [ 2.90965654e+02,  3.82674953e+02, -4.11534120e-02],
       [ 2.90929765e+02,  3.82669722e+02, -4.07599200e-02],
       [ 2.90894227e+02,  3.82664540e+02, -4.03568479e-02],
       [ 2.90859048e+02,  3.82659408e+02, -3.99439623e-02],
       [ 2.90824238e+02,  3.82654327e+02, -3.95210246e-02],
       [ 2.90789805e+02,  3.82649299e+02, -3.90877897e-02],
       [ 2.90755758e+02,  3.82644325e+02, -3.86440073e-02],
       [ 2.90722107e+02,  3.82639406e+02, -3.81894202e-02],
       [ 2.90688861e+02,  3.82634544e+02, -3.77237657e-02],
       [ 2.90656031e+02,  3.82629739e+02, -3.72467743e-02],
       [ 2.90623626e+02,  3.82624994e+02, -3.67581699e-02],
       [ 2.90591656e+02,  3.82620310e+02, -3.62576699e-02],
       [ 2.90560133e+02,  3.82615688e+02, -3.57449845e-02],
       [ 2.90529066e+02,  3.82611130e+02, -3.52198171e-02],
       [ 2.90498468e+02,  3.82606638e+02, -3.46818636e-02],
       [ 2.90468349e+02,  3.82602213e+02, -3.41308129e-02],
       [ 2.90438721e+02,  3.82597857e+02, -3.35663460e-02],
       [ 2.90409596e+02,  3.82593572e+02, -3.29881364e-02],
       [ 2.90380986e+02,  3.82589358e+02, -3.23958494e-02],
       [ 2.90352904e+02,  3.82585219e+02, -3.17891425e-02],
       [ 2.90325363e+02,  3.82581156e+02, -3.11676644e-02],
       [ 2.90298375e+02,  3.82577171e+02, -3.05310555e-02],
       [ 2.90271954e+02,  3.82573266e+02, -2.98789475e-02],
       [ 2.90246114e+02,  3.82569442e+02, -2.92109630e-02],
       [ 2.90220869e+02,  3.82565702e+02, -2.85267155e-02],
       [ 2.90196233e+02,  3.82562048e+02, -2.78258090e-02],
       [ 2.90172222e+02,  3.82558482e+02, -2.71078379e-02],
       [ 2.90148850e+02,  3.82555007e+02, -2.63723868e-02],
       [ 2.90126133e+02,  3.82551623e+02, -2.56190302e-02],
       [ 2.90104087e+02,  3.82548334e+02, -2.48473320e-02],
       [ 2.90082729e+02,  3.82545142e+02, -2.40568457e-02],
       [ 2.90062074e+02,  3.82542050e+02, -2.32471139e-02],
       [ 2.90042140e+02,  3.82539059e+02, -2.24176681e-02],
       [ 2.90022944e+02,  3.82536173e+02, -2.15680282e-02],
       [ 2.90004505e+02,  3.82533393e+02, -2.06977026e-02],
```

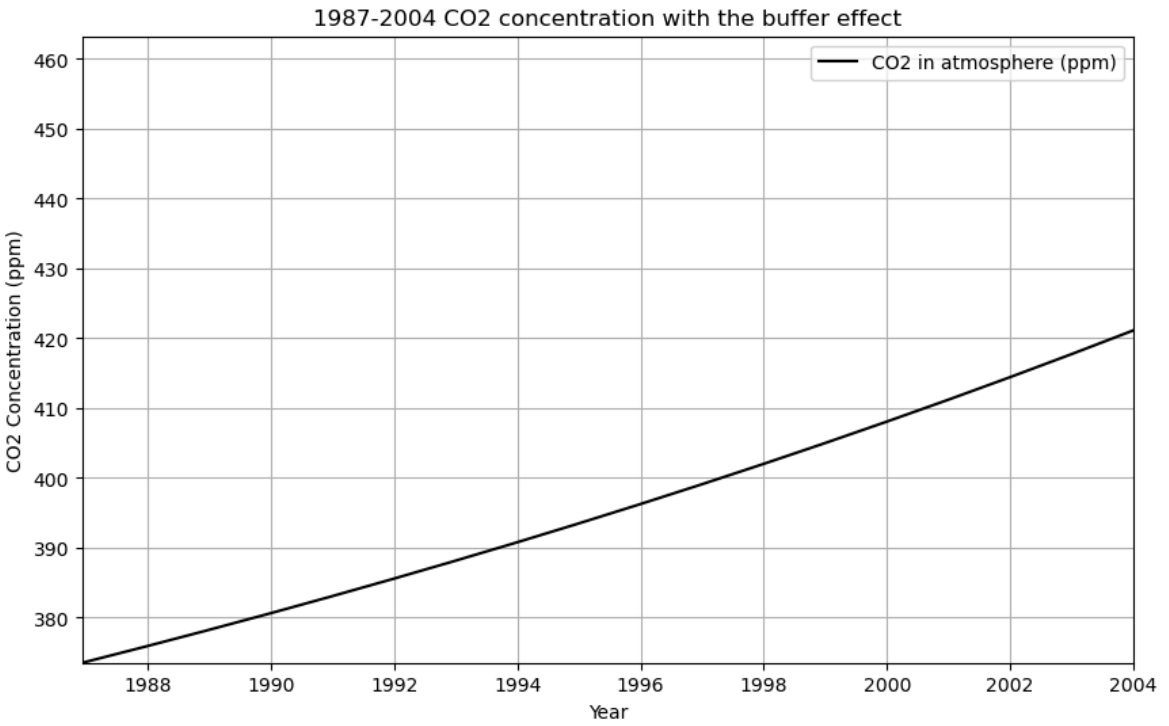
```
[ 2.89986841e+02,  3.82530723e+02, -1.98061878e-02],
[ 2.89969971e+02,  3.82528166e+02, -1.88929677e-02],
[ 2.89953913e+02,  3.82525723e+02, -1.79575140e-02],
[ 2.89938688e+02,  3.82523398e+02, -1.69992854e-02],
[ 2.89924317e+02,  3.82521194e+02, -1.60177274e-02],
[ 2.89910819e+02,  3.82519113e+02, -1.50122719e-02],
[ 2.89898216e+02,  3.82517159e+02, -1.39823372e-02],
[ 2.89886530e+02,  3.82515335e+02, -1.29273273e-02],
[ 2.89875783e+02,  3.82513643e+02, -1.18466317e-02],
[ 2.89865998e+02,  3.82512088e+02, -1.07396250e-02],
[ 2.89857198e+02,  3.82510673e+02, -9.60566673e-03],
[ 2.89849407e+02,  3.82509401e+02, -8.44410065e-03],
[ 2.89842650e+02,  3.82508275e+02, -7.25425462e-03],
[ 2.89836953e+02,  3.82507299e+02, -6.03544012e-03],
[ 2.89832340e+02,  3.82506476e+02, -4.78695188e-03],
[ 2.89828838e+02,  3.82505811e+02, -3.50806742e-03],
[ 2.89826475e+02,  3.82505308e+02, -2.19804672e-03],
[ 2.89825277e+02,  3.82504969e+02, -8.56131730e-04],
[ 2.89825274e+02,  3.82504800e+02,  5.18454059e-04],
[ 2.89826494e+02,  3.82504805e+02,  1.92650607e-03],
[ 2.89828968e+02,  3.82504987e+02,  3.36883909e-03],
[ 2.89832725e+02,  3.82505350e+02,  4.84628774e-03],
[ 2.89837797e+02,  3.82505901e+02,  6.35970695e-03],
[ 2.89844216e+02,  3.82506642e+02,  7.90997249e-03],
[ 2.89852014e+02,  3.82507578e+02,  9.49798143e-03],
[ 2.89861226e+02,  3.82508715e+02,  1.11246527e-02],
[ 2.89871886e+02,  3.82510057e+02,  1.27909275e-02],
[ 2.89884029e+02,  3.82511609e+02,  1.44977702e-02],
[ 2.89897691e+02,  3.82513377e+02,  1.62461684e-02],
[ 2.89912910e+02,  3.82515364e+02,  1.80371338e-02],
[ 2.89929722e+02,  3.82517578e+02,  1.98717027e-02],
[ 2.89948168e+02,  3.82520022e+02,  2.17509368e-02],
[ 2.89968286e+02,  3.82522704e+02,  2.36759236e-02],
[ 2.89990118e+02,  3.82525628e+02,  2.56477768e-02],
[ 2.90013705e+02,  3.82528800e+02,  2.76676376e-02],
[ 2.90039090e+02,  3.82532227e+02,  2.97366747e-02],
[ 2.90066317e+02,  3.82535915e+02,  3.18560855e-02],
[ 2.90095430e+02,  3.82539869e+02,  3.40270963e-02],
[ 2.90126477e+02,  3.82544096e+02,  3.62509634e-02],
[ 2.90159504e+02,  3.82548603e+02,  3.85289738e-02],
[ 2.90194558e+02,  3.82553397e+02,  4.08624455e-02],
[ 2.90231691e+02,  3.82558484e+02,  4.32527288e-02],
[ 2.90270952e+02,  3.82563872e+02,  4.57012070e-02],
[ 2.90312393e+02,  3.82569567e+02,  4.82092968e-02],
[ 2.90356067e+02,  3.82575578e+02,  5.07784496e-02],
[ 2.90402030e+02,  3.82581911e+02,  5.34101520e-02],
[ 2.90450336e+02,  3.82588575e+02,  5.61059269e-02],
[ 2.90501043e+02,  3.82595577e+02,  5.88673343e-02],
[ 2.90554210e+02,  3.82602925e+02,  6.16959720e-02],
[ 2.90609897e+02,  3.82610628e+02,  6.45934769e-02],
[ 2.90668164e+02,  3.82618694e+02,  6.75615257e-02],
[ 2.90729076e+02,  3.82627132e+02,  7.06018357e-02],
[ 2.90792696e+02,  3.82635951e+02,  7.37161664e-02],
[ 2.90859091e+02,  3.82645160e+02,  7.69063199e-02],
[ 2.90928329e+02,  3.82654767e+02,  8.01741421e-02],
[ 2.91000479e+02,  3.82664783e+02,  8.35215241e-02],
[ 2.91075612e+02,  3.82675217e+02,  8.69504027e-02],
[ 2.91153801e+02,  3.82686080e+02,  9.04627623e-02],
[ 2.91235121e+02,  3.82697381e+02,  9.40606353e-02],
[ 2.91319648e+02,  3.82709130e+02,  9.77461036e-02],
```

```
[ 2.91407460e+02,  3.82721339e+02,  1.01521300e-01],
[ 2.91498639e+02,  3.82734018e+02,  1.05388408e-01],
[ 2.91593266e+02,  3.82747179e+02,  1.09349667e-01],
[ 2.91691425e+02,  3.82760832e+02,  1.13407369e-01],
[ 2.91793203e+02,  3.82774990e+02,  1.17563860e-01],
[ 2.91898688e+02,  3.82789663e+02,  1.21821548e-01],
[ 2.92007972e+02,  3.82804865e+02,  1.26182895e-01],
[ 2.92121146e+02,  3.82820608e+02,  1.30650425e-01],
[ 2.92238307e+02,  3.82836904e+02,  1.35226723e-01],
[ 2.92359551e+02,  3.82853767e+02,  1.39914438e-01],
[ 2.92484980e+02,  3.82871209e+02,  1.44716281e-01],
[ 2.92614694e+02,  3.82889244e+02,  1.49635033e-01],
[ 2.92748799e+02,  3.82907887e+02,  1.54673538e-01],
[ 2.92887403e+02,  3.82927151e+02,  1.59834713e-01],
[ 2.93030616e+02,  3.82947050e+02,  1.65121544e-01],
[ 2.93178550e+02,  3.82967600e+02,  1.70537090e-01],
[ 2.93331322e+02,  3.82988816e+02,  1.76084485e-01],
[ 2.93489050e+02,  3.83010713e+02,  1.81766940e-01],
[ 2.93651855e+02,  3.83033306e+02,  1.87587742e-01],
[ 2.93819862e+02,  3.83056613e+02,  1.93550259e-01],
[ 2.93993198e+02,  3.83080648e+02,  1.99657942e-01],
[ 2.94171994e+02,  3.83105430e+02,  2.05914326e-01],
[ 2.94356384e+02,  3.83130976e+02,  2.12323030e-01],
[ 2.94546506e+02,  3.83157302e+02,  2.18887763e-01],
[ 2.94742499e+02,  3.83184427e+02,  2.25612323e-01],
[ 2.94944509e+02,  3.83212369e+02,  2.32500603e-01],
[ 2.95152682e+02,  3.83241147e+02,  2.39556587e-01],
[ 2.95367170e+02,  3.83270780e+02,  2.46784360e-01],
[ 2.95588129e+02,  3.83301287e+02,  2.54188102e-01],
[ 2.95815716e+02,  3.83332688e+02,  2.61772099e-01],
[ 2.96050095e+02,  3.83365003e+02,  2.69540739e-01],
[ 2.96291434e+02,  3.83398254e+02,  2.77498518e-01],
[ 2.96539902e+02,  3.83432461e+02,  2.85650040e-01],
[ 2.96795676e+02,  3.83467646e+02,  2.94000022e-01],
[ 2.97058934e+02,  3.83503830e+02,  3.02553297e-01],
[ 2.97329862e+02,  3.83541037e+02,  3.11314813e-01],
[ 2.97608648e+02,  3.83579289e+02,  3.20289640e-01],
[ 2.97895485e+02,  3.83618609e+02,  3.29482972e-01],
[ 2.98190571e+02,  3.83659021e+02,  3.38900129e-01],
[ 2.98494110e+02,  3.83700550e+02,  3.48546560e-01],
[ 2.98806311e+02,  3.83743220e+02,  3.58427847e-01],
[ 2.99127385e+02,  3.83787056e+02,  3.68549707e-01],
[ 2.99457553e+02,  3.83832084e+02,  3.78917999e-01],
[ 2.99797038e+02,  3.83878331e+02,  3.89538721e-01],
[ 3.00146070e+02,  3.83925822e+02,  4.00418019e-01],
[ 3.00504885e+02,  3.83974586e+02,  4.11562190e-01],
[ 3.00873724e+02,  3.84024650e+02,  4.22977681e-01],
[ 3.01252835e+02,  3.84076042e+02,  4.34671098e-01],
[ 3.01642471e+02,  3.84128790e+02,  4.46649208e-01],
[ 3.02042893e+02,  3.84182925e+02,  4.58918942e-01],
[ 3.02454366e+02,  3.84238476e+02,  4.71487400e-01],
[ 3.02877164e+02,  3.84295473e+02,  4.84361855e-01],
[ 3.03311568e+02,  3.84353947e+02,  4.97549756e-01],
[ 3.03757863e+02,  3.84413929e+02,  5.11058735e-01],
[ 3.04216346e+02,  3.84475453e+02,  5.24896610e-01],
[ 3.04687316e+02,  3.84538549e+02,  5.39071387e-01],
[ 3.05171084e+02,  3.84603251e+02,  5.53591269e-01],
[ 3.05667966e+02,  3.84669593e+02,  5.68464658e-01],
[ 3.06178288e+02,  3.84737609e+02,  5.83700161e-01],
[ 3.06702383e+02,  3.84807334e+02,  5.99306593e-01],
```

```
[ 3.07240592e+02,  3.84878802e+02,  6.15292986e-01],
[ 3.07793265e+02,  3.84952051e+02,  6.31668591e-01],
[ 3.08360762e+02,  3.85027115e+02,  6.48442882e-01],
[ 3.08943451e+02,  3.85104034e+02,  6.65625567e-01],
[ 3.09541709e+02,  3.85182843e+02,  6.83226589e-01],
[ 3.10155923e+02,  3.85263581e+02,  7.01256133e-01],
[ 3.10786490e+02,  3.85346286e+02,  7.19724631e-01],
[ 3.11433817e+02,  3.85430999e+02,  7.38642771e-01],
[ 3.12098320e+02,  3.85517759e+02,  7.58021500e-01],
[ 3.12780427e+02,  3.85606606e+02,  7.77872030e-01],
[ 3.13480577e+02,  3.85697581e+02,  7.98205851e-01],
[ 3.14199220e+02,  3.85790726e+02,  8.19034726e-01],
[ 3.14936815e+02,  3.85886082e+02,  8.40370711e-01],
[ 3.15693837e+02,  3.85983693e+02,  8.62226149e-01],
[ 3.16470770e+02,  3.86083601e+02,  8.84613689e-01],
[ 3.17268111e+02,  3.86185850e+02,  9.07546286e-01],
[ 3.18086369e+02,  3.86290484e+02,  9.31037208e-01],
[ 3.18926068e+02,  3.86397548e+02,  9.55100051e-01],
[ 3.19787743e+02,  3.86507087e+02,  9.79748736e-01],
[ 3.20671944e+02,  3.86619146e+02,  1.00499753e+00],
[ 3.21579235e+02,  3.86733772e+02,  1.03086104e+00],
[ 3.22510194e+02,  3.86851011e+02,  1.05735423e+00],
[ 3.23465413e+02,  3.86970910e+02,  1.08449244e+00],
[ 3.24445501e+02,  3.87093516e+02,  1.11229137e+00],
[ 3.25451082e+02,  3.87218878e+02,  1.14076710e+00],
[ 3.26482795e+02,  3.87347043e+02,  1.16993611e+00],
[ 3.27541295e+02,  3.87478060e+02,  1.19981528e+00],
[ 3.28627257e+02,  3.87611978e+02,  1.23042191e+00],
[ 3.29741369e+02,  3.87748845e+02,  1.26177369e+00],
[ 3.30884339e+02,  3.87888713e+02,  1.29388878e+00],
[ 3.32056893e+02,  3.88031629e+02,  1.32678575e+00],
[ 3.33259776e+02,  3.88177645e+02,  1.36048365e+00],
[ 3.34493751e+02,  3.88326810e+02,  1.39500197e+00],
[ 3.35759602e+02,  3.88479175e+02,  1.43036069e+00],
[ 3.37058132e+02,  3.88634791e+02,  1.46658027e+00],
[ 3.38390165e+02,  3.88793708e+02,  1.50368166e+00],
[ 3.39756547e+02,  3.88955976e+02,  1.54168634e+00],
[ 3.41158144e+02,  3.89121647e+02,  1.58061629e+00],
[ 3.42595846e+02,  3.89290772e+02,  1.62049406e+00],
[ 3.44070565e+02,  3.89463400e+02,  1.66134270e+00],
[ 3.45583238e+02,  3.89639584e+02,  1.70318585e+00],
[ 3.47134823e+02,  3.89819372e+02,  1.74604774e+00],
[ 3.48726307e+02,  3.90002817e+02,  1.78995317e+00],
[ 3.50358699e+02,  3.90189967e+02,  1.83492753e+00],
[ 3.52033036e+02,  3.90380872e+02,  1.88099685e+00],
[ 3.53750381e+02,  3.90575583e+02,  1.92818780e+00],
[ 3.55511824e+02,  3.90774148e+02,  1.97652768e+00],
[ 3.57318484e+02,  3.90976616e+02,  2.02604446e+00],
[ 3.59171509e+02,  3.91183035e+02,  2.07676680e+00],
[ 3.61072076e+02,  3.91393453e+02,  2.12872404e+00],
[ 3.63021394e+02,  3.91607917e+02,  2.18194626e+00],
[ 3.65020700e+02,  3.91826473e+02,  2.23646424e+00],
[ 3.67071267e+02,  3.92049167e+02,  2.29230954e+00],
[ 3.69174397e+02,  3.92276043e+02,  2.34951448e+00],
[ 3.71331429e+02,  3.92507146e+02,  2.40811215e+00],
[ 3.73543735e+02,  3.92742519e+02,  2.46813646e+00],
[ 3.75812723e+02,  3.92982203e+02,  2.52962215e+00],
[ 3.78139837e+02,  3.93226239e+02,  2.59260480e+00],
[ 3.80526558e+02,  3.93474667e+02,  2.65712084e+00],
[ 3.82974406e+02,  3.93727526e+02,  2.72320762e+00],
```

```
       [ 3.85484939e+02,  3.93984853e+02,  2.79090338e+00],
       [ 3.88059758e+02,  3.94246682e+02,  2.86024728e+00],
       [ 3.90700502e+02,  3.94513050e+02,  2.93127947e+00],
       [ 3.93408855e+02,  3.94783987e+02,  3.00404102e+00],
       [ 3.96186543e+02,  3.95059526e+02,  3.07857406e+00],
       [ 3.99035335e+02,  3.95339696e+02,  3.15492172e+00],
       [ 4.01957050e+02,  3.95624524e+02,  3.23312816e+00],
       [ 4.04953549e+02,  3.95914036e+02,  3.31323865e+00],
       [ 4.08026744e+02,  3.96208256e+02,  3.39529954e+00],
       [ 4.11178595e+02,  3.96507206e+02,  3.47935833e+00],
       [ 4.14411112e+02,  3.96810904e+02,  3.56546364e+00],
       [ 4.17726357e+02,  3.97119369e+02,  3.65366530e+00],
       [ 4.21126447e+02,  3.97432616e+02,  3.74401436e+00]])
```

In [52]:
```
# 画图
plt.figure(figsize=(10, 6))  # 设置图表大小
# 绘制1987年到2004年间大气中CO2浓度的变化
plt.plot(t2[235:253], sol2[235:253, 0], 'k', label='CO2 in atmosphere (ppm)')
# 添加图例
plt.legend(loc='best')
# 设置横坐标轴刻度为整数
plt.gca().xaxis.set_major_locator(MaxNLocator(integer=True))
# 设置图表标题和坐标轴标签
plt.xlabel('Year')
plt.ylabel('CO2 Concentration (ppm)')
plt.xlim([t2[235], t2[252]])  # 设置横坐标轴的范围为1987年到2004年
plt.ylim([sol2[235:253, 0].min(), sol2[235:253, 0].max() * 1.1])  # 设置纵坐标轴
plt.grid(True)
plt.title('1987-2004 CO2 concentration with the buffer effect')
plt.show()
```



In [28]:
```
# 1.3 Reproduce Figure 2 in Tomizuka (2009).
# 导入观测值数据
df2=pd.read_csv('co2_annmean_mlo.csv')
df2
```

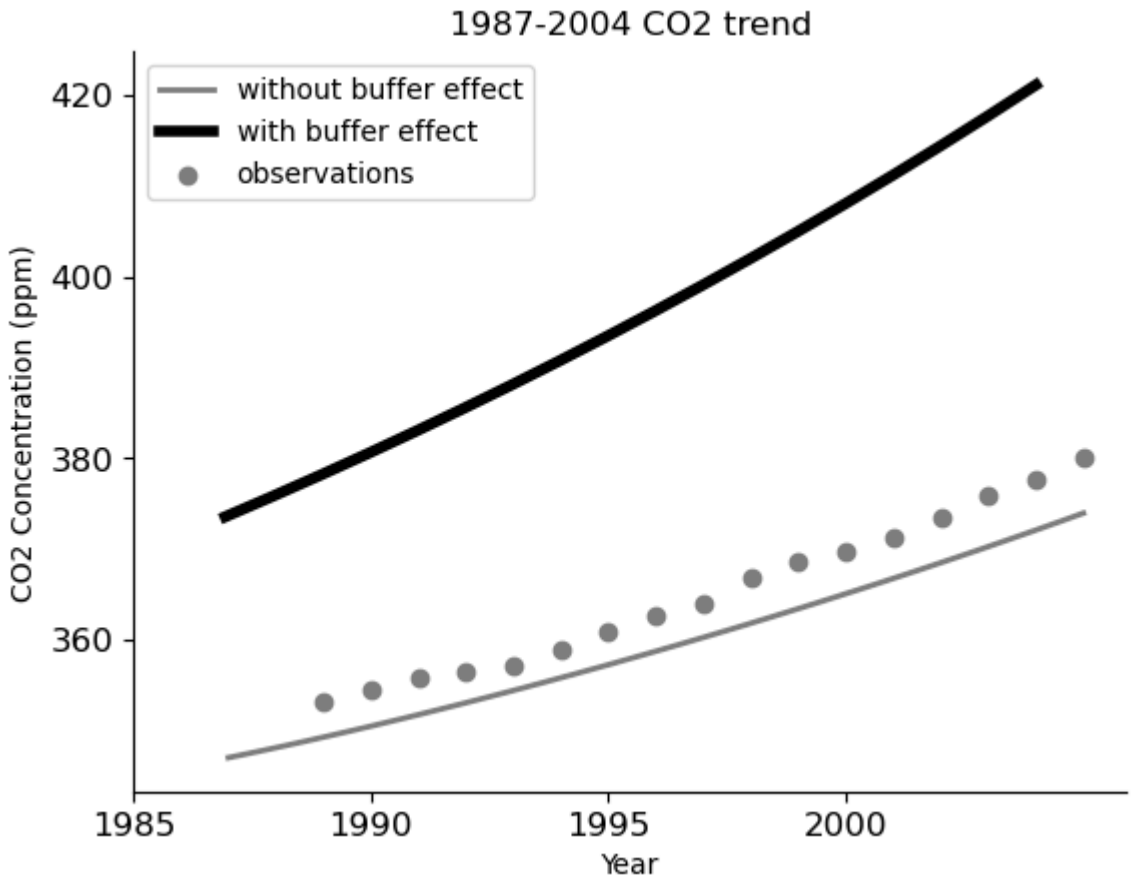
Out[28]:

	year	mean	unc
0	1959	315.98	0.12
1	1960	316.91	0.12
2	1961	317.64	0.12
3	1962	318.45	0.12
4	1963	318.99	0.12
...	...	...	...
59	2018	408.72	0.12
60	2019	411.65	0.12
61	2020	414.21	0.12
62	2021	416.41	0.12
63	2022	418.53	0.12

64 rows × 3 columns

In [54]:

```
# 画图
plt.plot(t1, sol[:, 0], linewidth=2, color='grey') # without buffer effect
plt.plot(t2[235:253], sol2[235:253,0], linewidth=4, color='black') # with buffer
plt.scatter(df2['year'][30:47],df2['mean'][30:47], color='gray') # observed data
# 设置图例
plt.legend(['without buffer effect','with buffer effect','observations'],loc='be
plt.xlabel('Year')
plt.ylabel('CO2 Concentration (ppm)')
# 设置横纵坐标
plt.gca().xaxis.set_major_locator(MaxNLocator(integer=True))
plt.xticks([1985, 1990, 1995, 2000], fontsize=12)
plt.yticks([360, 380, 400, 420], fontsize=12)
# 隐藏右边和上边的边框
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['top'].set_visible(False)
plt.title('1987-2004 CO2 trend')
plt.show()
# 不考虑buffer effect的曲线与文献中图2略有区别，可能是函数拟合不够好
```



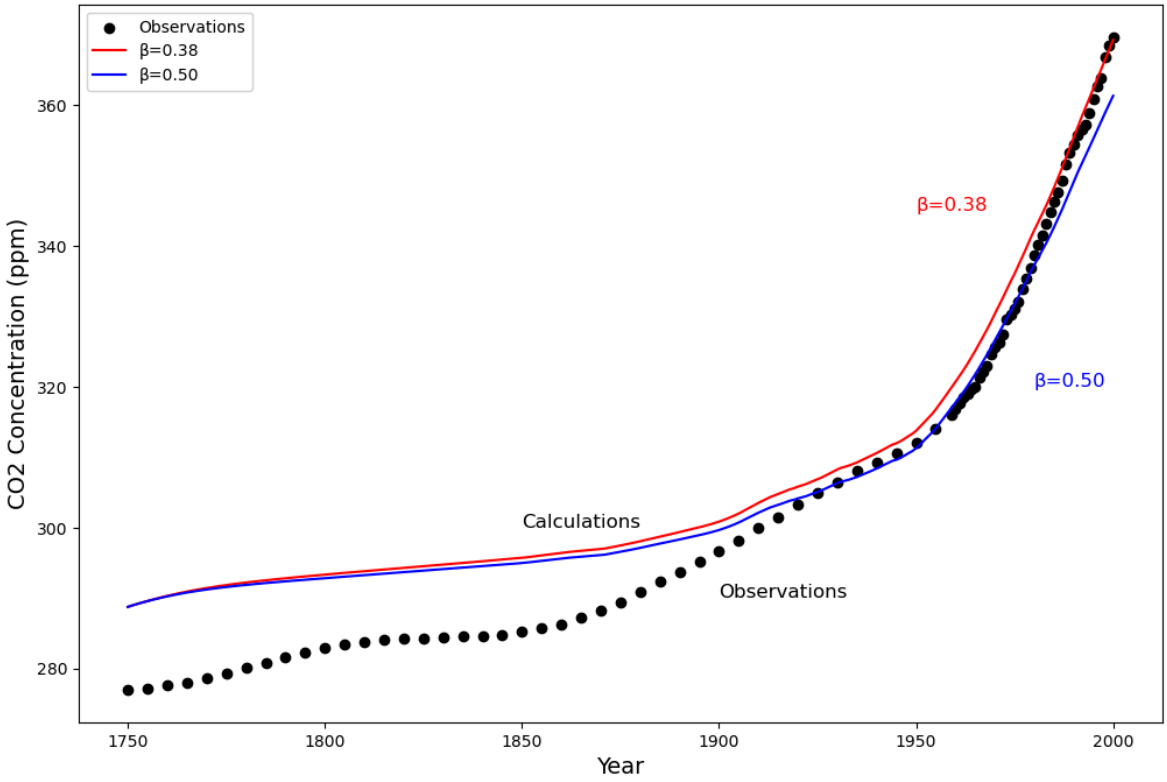
In [62]:
```
# [Bonus] Compute the atmospheric CO2 level in ppm and reproduce Figure 4 in Tom
# 本题我向我的师姐龙师倩寻求了帮助
co2_observations = pd.read_csv('1750-2000CO2.csv')
land_use_data = pd.read_excel('Global_land-use_flux-1750_2005.xls')
ff_emissions = pd.read_csv('global.1751_2014.csv')

# 选择需要的列，并计算土地利用变化导致的CO2排放量
land_use_data = land_use_data[['Year', 'Global']]
land_use_data['LandUseChange'] = land_use_data['Global'] / (1000 * 2.13)

# 选择需要的列，并计算化石燃料燃烧导致的CO2排放量
ff_emissions = ff_emissions[['Year', 'Total carbon emissions from fossil fuel co
ff_emissions['FossilFuelEmissions'] = ff_emissions.iloc[:, 1] - ff_emissions.ilo
ff_emissions['EmissionFactor'] = ff_emissions['FossilFuelEmissions'] / (1000 * 2
```

In [64]:
```
# 定义碳循环模型中的参数
k12, k21, k23, k24, k32, k34, k43, k45, k51, k67, k71 = [60 / 615, 60 / 842, 9 /
N2_0 = 842 / 2.13

# 定义初始条件
initial_conditions = [615 / 2.13, 842 / 2.13, 9744 / 2.13, 26280 / 2.13, 9000000
f0 = 62 / 2.13
P0 = 615 / 2.13

# 探索Beta值
beta_values = [0.38, 0.5]
results = []

for beta in beta_values:
    # 复制初始条件
    N1, N2, N3, N4, N5, N6, N7 = initial_conditions.copy()
    atmosphere = [N1]
```

```
    for year in range(1751, 2001):
        gamma = ff_emissions[ff_emissions['Year'] == year]['EmissionFactor'].val
        delta = land_use_data[land_use_data['Year'] == year]['LandUseChange'].va

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

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

         # 计算每个部分的变化率
        dN1_dt = -k12 * N1 + k21 * (N2_0 + xi * (N2 - N2_0)) + gamma - f + delta
        dN2_dt = k12 * N1 - k21 * (N2_0 + xi * (N2 - N2_0)) - k23 * N2 + k32 * N
        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

        # 更新每个部分的值
        N1 += dN1_dt
        N2 += dN2_dt
        N3 += dN3_dt
        N4 += dN4_dt
        N5 += dN5_dt
        N6 += dN6_dt
        N7 += dN7_dt

        atmosphere.append(N1)

    # 将atmosphere列表添加到results列表中
    results.append(atmosphere)
```

In [66]:

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

plt.scatter(co2_observations['year'], co2_observations['mean'], label='Observati

plt.text(1850, 300, 'Calculations', fontsize=12)
plt.text(1900, 290, 'Observations', fontsize=12)
plt.text(1950, 345, 'β=0.38', fontsize=12, color='red')
plt.text(1980, 320, 'β=0.50', fontsize=12, color='blue')

# 直接画出每个beta值的结果，用红色和蓝色表示
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')

plt.xlabel('Year', fontsize=14)
plt.ylabel('CO2 Concentration (ppm)', fontsize=14)

plt.legend()

plt.show()
```


Observations
β=0.38
β=0.50
CO2 Concentration (ppm)
360
340
320
300
280
1750
1800
1850
1900
1950
2000
Year
β=0.38
β=0.50
Calculations
Observations