In [1]:
```python
# independently by Shao Shi
# discussed with Yanchen Li to find out one problem in the paper(see # 3.)
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt
from math import e
```

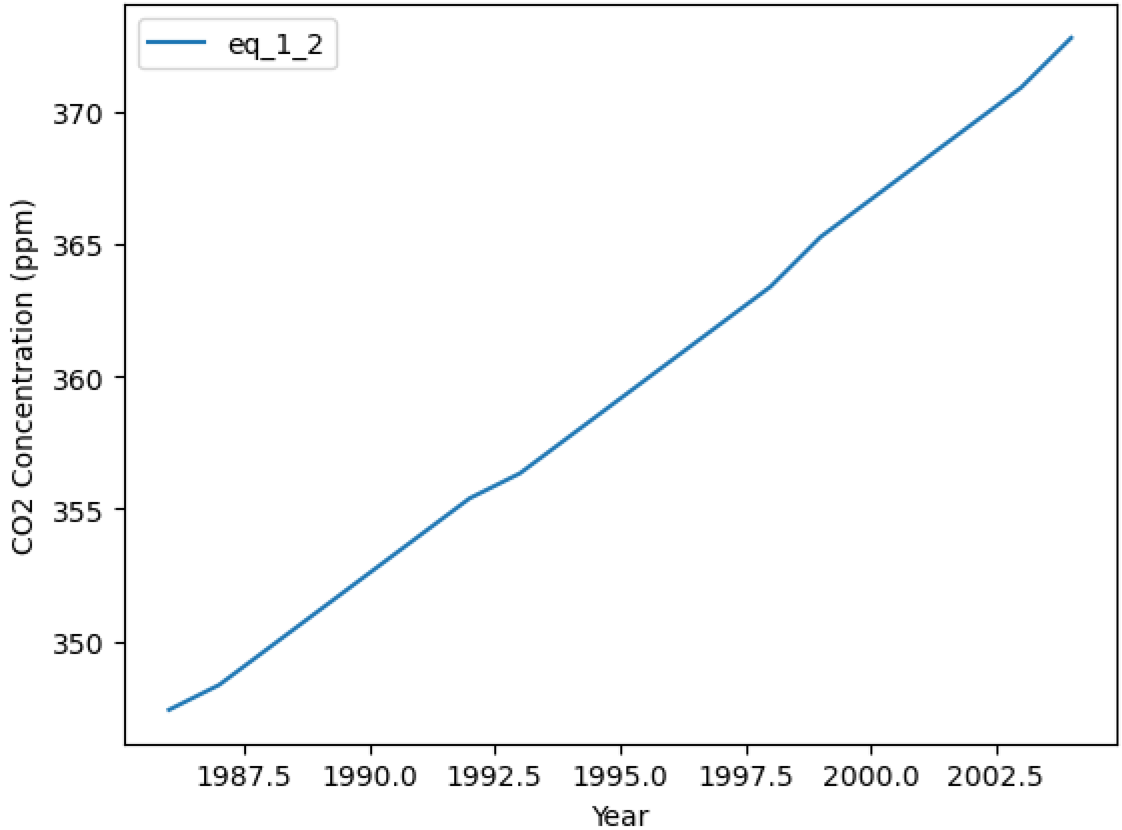
In [2]:
```python
# obtain the gama value, i.e., the CO2 emission data
gama = np.loadtxt("global.1751_2008.csv", delimiter=",", skiprows = 237)[0:-4,1]/10**3
# declare the vars
yearCount = 2004 - 1986 + 1
t = np.linspace(1986,2004,yearCount,dtype='int')
N1_eq_1_2 = np.empty_like(t)
N2_eq_1_2 = np.empty_like(t)
t
```

Out[2]:
```
array([1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996,
       1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004])
```

In [3]:
```python
# 1.
# ODE creation of eq.1 and eq.2
def eq_1_2(N,t,gama):
    k12 = 105/740
    k21 = 102/900
    N1,N2 = N
    dN1dt = -k12 * N1 + k21 * N2 + gama
    dN2dt = k12 * N1 - k21 * N2
    return [dN1dt, dN2dt]
```

In [4]:
```python
N0 = [740, 900]
N1_eq_1_2[0],N2_eq_1_2[0] = N0
# slove the ODE year by year
for i in range(1,yearCount):
    #set the setp size
    dt = [0,1]
    # numericly solve the eq.s
    N = odeint(eq_1_2,N0,dt,args=(gama[i-1],))
    # store the result
    N1_eq_1_2[i] = N[1][0]
    N2_eq_1_2[i] = N[1][1]
    # initiation of the next step
    N0 = N[1]
```

In [5]:
```python
#plt.plot(t,gama,'g:',label='u(t)')
plt.plot(t,N1_eq_1_2/2.13,label='eq_1_2')
#plt.plot(t,y,'r--',label='y(t)')
plt.ylabel('CO2 Concentration (ppm)')
plt.xlabel('Year')
plt.legend(loc='best')
plt.show()
```



In [6]:
```
# 2.
annual_CO2_Obs = np.loadtxt("co2_annmean_mlo.csv", delimiter=",", skiprows = 83 )[0:-17,
```

In [7]:
```
N1_eq_3_4 = np.empty_like(t)
N2_eq_3_4 = np.empty_like(t)
```

In [8]:
```
def compute_xi(CO2_ppm):
    xi = 3.69 + 1.86 * 10**-2 * CO2_ppm - 1.80 * 10**-6 * CO2_ppm**2
    return xi
```

In [9]:
```
# ODE creation of eq.3 and eq.4
def eq_3_4(N,t,gama,xi):
    N02 = 821
    k12 = 105/740
    k21 = 102/900
    N1,N2 = N
    dN1dt = -k12 * N1 + k21 * (N02 + xi*(N2 - N02)) + gama
    dN2dt = k12 * N1 - k21 * (N02 + xi*(N2 - N02))
    return [dN1dt, dN2dt]
```

In [10]:
```
N0 = [740, 900]
N1_eq_3_4[0],N2_eq_3_4[0] = N0
CO2_ppm = N1_eq_3_4[0]/2.13
xi = compute_xi(CO2_ppm)
# slove the ODE year by year
for i in range(1,yearCount):
    #set the setp size
    dt = [0,1]
    # numericly solve the eq.s
    N = odeint(eq_3_4,N0,dt,args=(gama[i-1],xi))
    # store the result
    N1_eq_3_4[i] = N[1][0]
    N2_eq_3_4[i] = N[1][1]
    # initiation of the next step
    CO2_ppm = N1_eq_3_4[i]/2.13
    N0 = N[1]
```

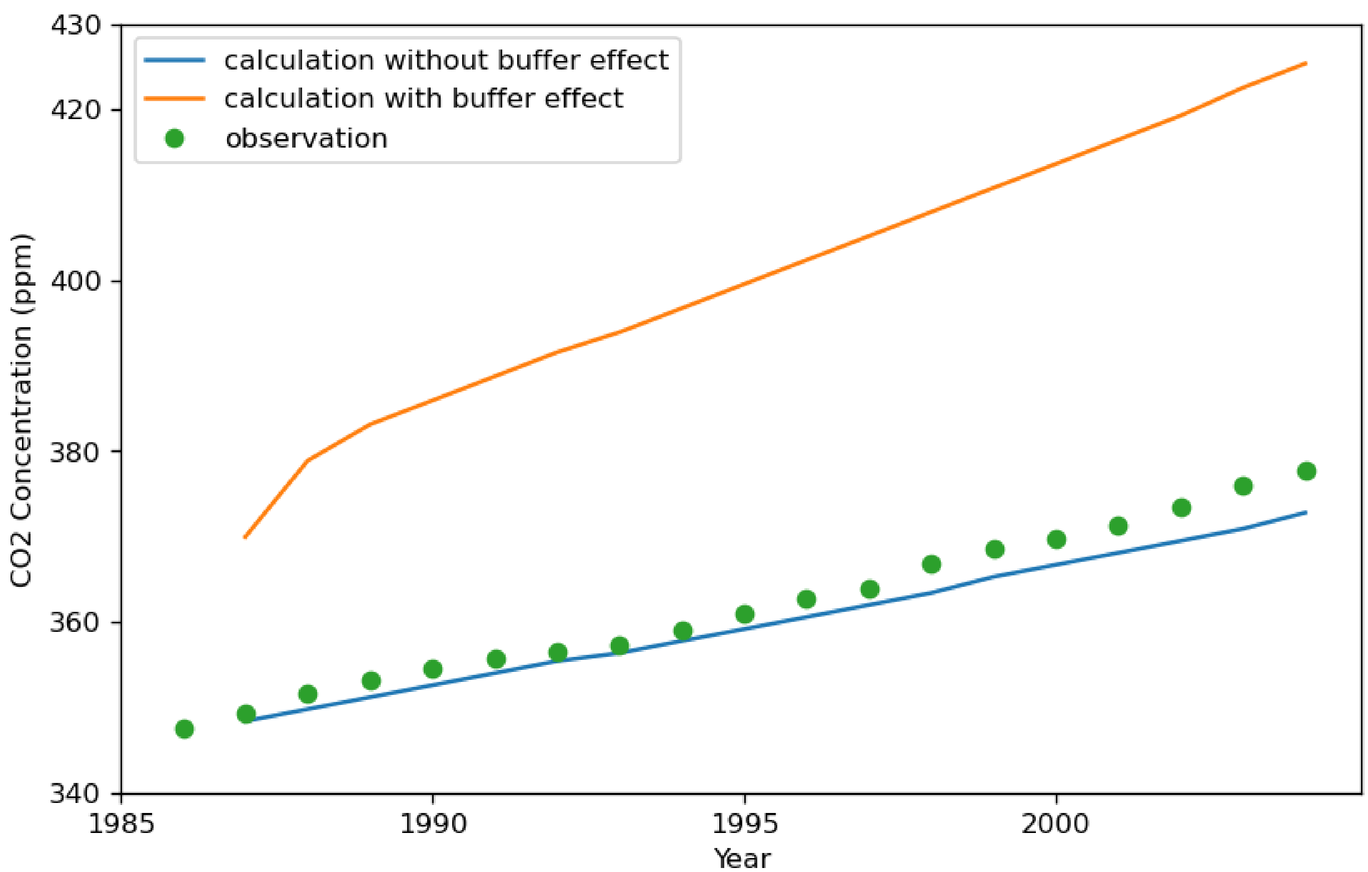
```
    xi = compute_xi(CO2_ppm)

N1_eq_3_4[0:]/2.13
```

Out[10]:
```
array([347.41784038, 369.95305164, 378.87323944, 383.09859155,
       385.91549296, 388.73239437, 391.54929577, 393.89671362,
       396.71361502, 399.53051643, 402.34741784, 405.16431925,
       407.98122066, 410.79812207, 413.61502347, 416.43192488,
       419.24882629, 422.53521127, 425.35211268])
```

In [11]:
```
# 3.
plt.figure(figsize=(8,5),dpi=120)
plt.plot(t[1:],N1_eq_1_2[1:]/2.13,label='calculation without buffer effect')
plt.plot(t[1:],N1_eq_3_4[1:]/2.13,label='calculation with buffer effect')
plt.plot(t,annual_CO2_Obs,'o',label='observation')
plt.ylabel('CO2 Concentration (ppm)')
plt.xlabel('Year')
plt.legend(loc='best')
plt.xticks([1985,1990,1995,2000])
plt.yticks([340,360,380,400,420,430])
plt.show()

# you may find that my value of 1987 in plot is lower from the one in the paper
# that is because the paper wrongly plotted the data in 1988 as 1987, 1989 as 1988 ...
# for this reason, my data in 1988 is the same of the 1987 value in the paper
# this problem may caused by the initial condition, the author of the paper wrongly
# used the initial condition (740,900) of 1986 to solve the ODE in 1985
# the conclusion is: my plot is correct, the plot in the paper is wrong
```



In [12]:
```
# Bonus
import math
yearCount = 2008 - 1751 + 1
# read the data of gama
gama = np.loadtxt("global.1751_2008.csv", delimiter=",", skiprows = 2, usecols = 1)/10**
# create year vector
t = np.linspace(1751,1751+yearCount-1,yearCount,dtype='int')
# read CO2 observation data 1
```

```
co2_1010 = np.loadtxt("co2_1010.csv", delimiter=",", skiprows = 148)
# read CO2 observation data 2
annual_CO2_Obs = np.loadtxt("co2_annmean_mlo.csv", delimiter=",", skiprows = 56 )[:,1]
annual_CO2_Obs_t = np.loadtxt("co2_annmean_mlo.csv", delimiter=",", skiprows = 56 )[:,0]
# read the data of delta
delta = np.empty_like(gama)
delta[0:100] = np.linspace(0.2,0.5,100)
delta[99:-3]= np.loadtxt("Global_land-use_flux-1850_2005.CSV", delimiter=",", skiprows =
delta[-3:] = delta[-4]
```

In [13]:
```
def compute_f(P,beta):
    f0 = 62
    P0=290.21
    f = f0 * (1 + beta * math.log(P/P0))
    return f
```

In [14]:
```
# ODE creation of eq.6 to eq.12
def eq_6_12(N,t,gama,xi,f,delta):
    N02 = 821
    k12 = 60/615
    k21 = 60/842
    k23 = 9/842
    k24 = 43/842
    k32 = 52/9744
    k34 = 162/9744
    k43 = 205/26280
    k45 = 0.2/26280
    k51 = 0.2/90000000
    k67 = 62/731
    k71 = 62/1328
    N1,N2,N3,N4,N5,N6,N7 = N
    dN1dt = -k12 * N1 + k21 * (N02 + xi*(N2 - N02)) + gama - f + delta + k51 * N5 + k71
    dN2dt = k12 * N1 - k21 * (N02 + xi*(N2 - N02)) - k23 * N2 + k32 * N3 - k24 * N2
    dN3dt = k23*N2 - k32*N3 - k34*N3 +k43*N4
    dN4dt = k34*N3 - k43*N4 + k24*N2 -k45*N4
    dN5dt = k45*N4 - k51*N5
    dN6dt = f - k67*N6 - 2*delta
    dN7dt = k67*N6 - k71*N7 + delta
    return [dN1dt, dN2dt, dN3dt, dN4dt, dN5dt, dN6dt, dN7dt]
```

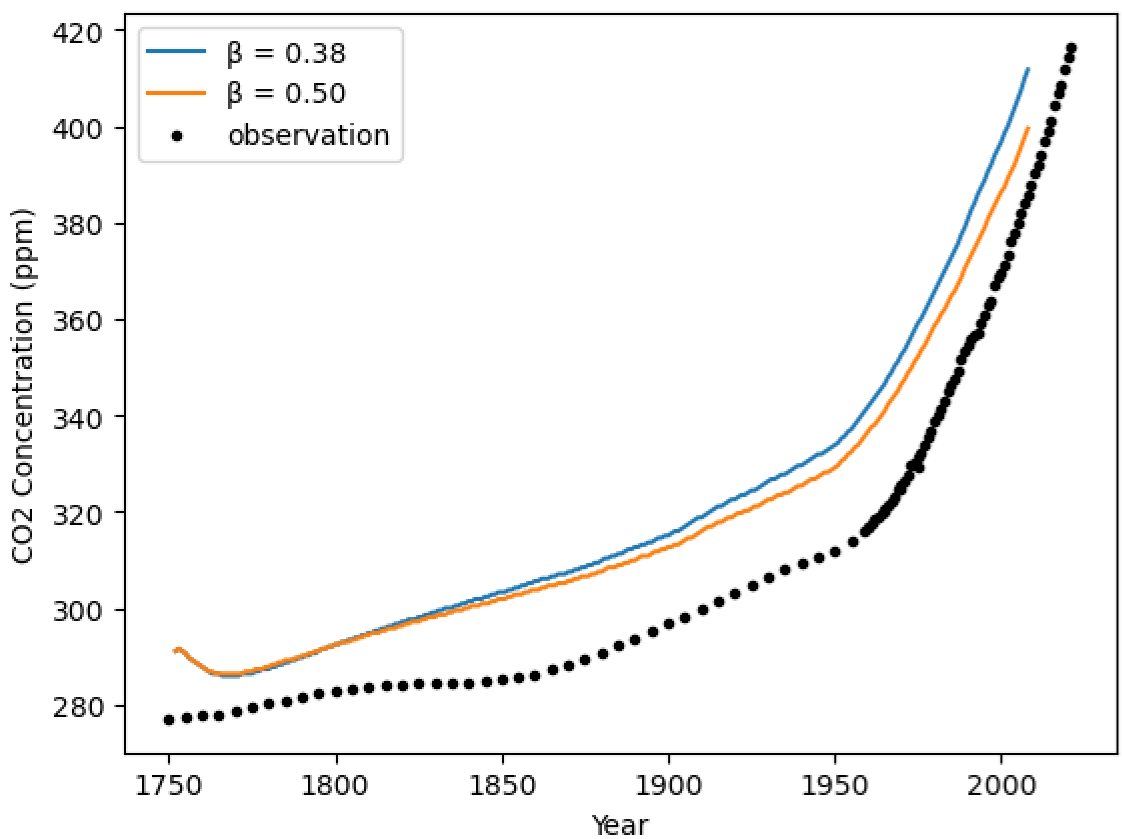
In [15]:
```
N1_eq_6_12_beta1 = np.empty_like(t)
N1_eq_6_12_beta2 = np.empty_like(t)
```

In [16]:
```
# beta = 0.38
beta = 0.38
N0 = [615, 842, 9744, 26280, 90000000, 731, 1238]
N1_eq_6_12_beta1[0] = N0[0]
CO2_ppm = N1_eq_6_12_beta1[0]/2.13
xi = compute_xi(CO2_ppm)
f = compute_f(CO2_ppm,beta)
# slove the ODE year by year
for i in range(1,yearCount):
    #set the setp size
    dt = [0,1]
    # numericly solve the eq.s
    N = odeint(eq_6_12,N0,dt,args=(gama[i-1],xi,f,delta[i-1]))
    # store the result
    N1_eq_6_12_beta1[i] = N[1][0]
    # initiation of the next step
    CO2_ppm = N1_eq_6_12_beta1[i]/2.13
    N0 = N[1]
    xi = compute_xi(CO2_ppm)
    f = compute_f(CO2_ppm,beta)
```

In [17]:
```
# beta = 0.50
beta = 0.5
N0 = [615, 842, 9744, 26280, 90000000, 731, 1238]
N1_eq_6_12_beta2[0] = N0[0]
CO2_ppm = N1_eq_6_12_beta2[0]/2.13
xi = compute_xi(CO2_ppm)
f = compute_f(CO2_ppm,beta)
# slove the ODE year by year
for i in range(1,yearCount):
    #set the setp size
    dt = [0,1]
    # numericly solve the eq.s
    N = odeint(eq_6_12,N0,dt,args=(gama[i-1],xi,f,delta[i-1]))
    # store the result
    N1_eq_6_12_beta2[i] = N[1][0]
    # initiation of the next step
    CO2_ppm = N1_eq_6_12_beta2[i]/2.13
    N0 = N[1]

    xi = compute_xi(CO2_ppm)
    f = compute_f(CO2_ppm,beta)
```

In [18]:
```
plt.plot(t[1:],N1_eq_6_12_beta1[1:]/2.13,label='β = 0.38')
plt.plot(t[1:],N1_eq_6_12_beta2[1:]/2.13,label='β = 0.50')
plt.plot(co2_1010[:,0], co2_1010[:,1],'k.')
plt.plot(annual_CO2_Obs_t,annual_CO2_Obs,'k.',label='observation')
plt.ylabel('CO2 Concentration (ppm)')
plt.xlabel('Year')
plt.legend(loc='best')
my_x_ticks = np.arange(1985,2004,3)
plt.show()
```



In [ ]: