Homework #2

CEIE 450/550
Environmental Systems
Nikolas Hawley
12 February 2024

For each of the systems described below, complete the following items. Submit your solutions as a single PDF file. Submit only sketched figures, numerical or text answers to each problem, and computer-generated plots (do not submit code or spreadsheets). Each of the following is worth 10 points.

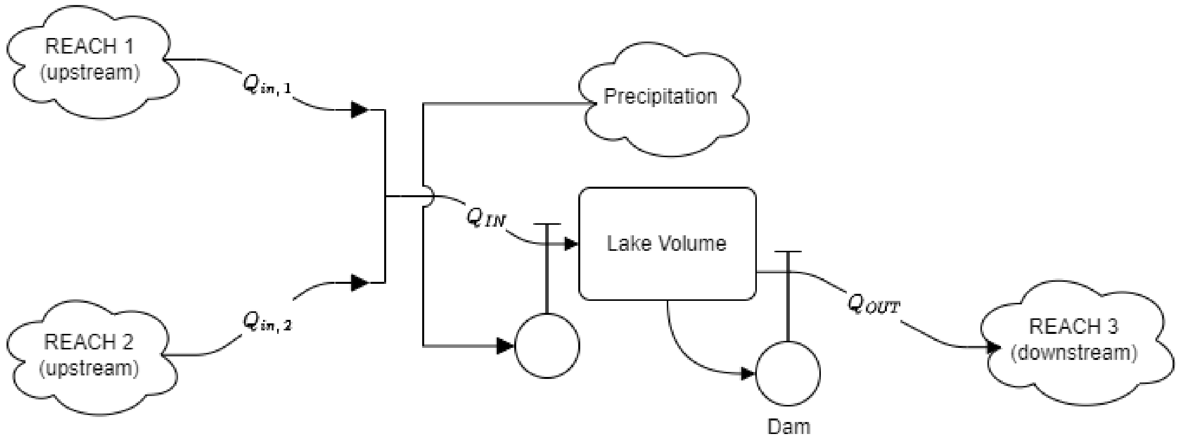
a) Sketch the conceptual model in a stock-and-flow diagram
b) Write a differential equation describing the model and solve it
c) What is an appropriate time step (including units)? Hint: convert all input data to consistent units in this time step
d) Plot* important flows and stocks over time
e) Describe insights from the results based on prompts listed below
f) (550 students only) Perform one sensitivity analysis (e.g., vary in the inputs/output flows over time, introduce a feedback loop, etc., and create time series plots). Describe your sensitivity analysis and interpret the plots.

System #1: A Lake

A lake with two input rivers and a dam: River #1 has a typical flow of 1×10^9 ft^3 per second (cfs—about as much as the Potomac) and River #2 flows at 1×10^8 cfs. The lake's dam operator typically releases 1.1×10^9 cfs, but she increases the release using a spillway to 1.5×10^9 cfs if the lake volume exceeds 90% of its capacity. Assume the lake begins at 80% of its capacity, and that once a week there is a rainstorm that increases flow by 40% in both inflow rivers for 24 hours. The capacity of the lake is 1×10^{15} cubic feet. Address the following questions for part (e):

- What is the maximum water volume in the lake over 100 days? What percent of the capacity is this?
- What is the first day that the plant operator will need to open the spillway?
- How many days in the first 100 days does the dam operator need to open the spillway?

a) Sketch the conceptual model in a stock-and-flow diagram



Given:

$$Q_{in,\,1\,base} = 1 \bullet 10^9 \text{ cfs}$$
$$Q_{in,\,2\,base} = 1 \bullet 10^8 \text{ cfs}$$

$$Q_{out} = \begin{cases} 1.1 \bullet 10^9 \text{ cfs} & V \leq 0.9V_{max}, \text{ (de facto spill rate)} \\ 1.5 \bullet 10^9 \text{ cfs} & V > 0.9V_{max}, \text{ (overflow spill rate)} \end{cases}$$

$$V_0 = 0.8V_{max}$$
$$V_{max} = 1 \bullet 10^{15} \text{ft}^3$$

---

b) Write a differential equation describing the model and solve it

$$V_t = V_0 + \Delta V = V_0 + \frac{dV}{dt}$$
$$\Delta V = \frac{dV}{dt} = \ ?$$

$$\frac{dV}{dt} = Q_{in,1} + Q_{in,2} - Q_{out}$$
$$\int_{t_0}^{t} \frac{dV}{dt} dt = \int_{t_0}^{t} \left(Q_{in,1} + Q_{in,2} - Q_{out}\right) dt$$

$$dV \Big|_{t_0}^{t} = \left[Q_{in,1} + Q_{in,2} - Q_{out}\right] \Big|_{t_0}^{t}$$
$$V_t - V_{t_0} = \left(Q_{in,1} + Q_{in,2} - Q_{out}\right)\left(t - t_0\right)$$

$$\implies \Delta V = \left(Q_{in,1} + Q_{in,2} - Q_{out}\right) \Delta t$$

---

c) What is an appropriate time step (including units)? Hint: convert all input data to consistent units in this time step

$$\Delta t = 1 \text{ day}$$

An appropriate time step can be case specific for the motive behind the question at hand or the scenario being modeled. In this scenario, the largest time interval unit, 1 day, is suitable for mass and flow estimations for the volume V of the lake at any given time step t.

d) Plot important flows and stocks over time

With the equation,

$$V_t = V_0 + \left(Q_{IN,\,t} - Q_{OUT,\,t}\right) \Delta t$$

where,

$$\Delta t = 1 \text{ day}$$

then, by order of operations, the method for the lake simulation model is defined as:

$$\text{For each time step } (\Delta\ t):$$
$$V_t = V_{t-1} + \left(Q_{IN,\,t} - Q_{OUT,\,t}\right)\Delta t$$

1) Step 1

$$\text{Find } Q_{IN,\,t}$$

$$Q_{IN,\,t} = Q_{in,\,1} + Q_{in,\,2}$$

$$Q_{IN,\,t} = \begin{cases} Q_{in,\,base\,1} + Q_{in,\,base\,2} & \textit{baseflow} \\ 1.4\left(Q_{in,\,base\,1} + Q_{in,\,base\,2}\right) & \textit{precipitation 24 hr/week} \end{cases}$$

$$\text{where,}$$
$$Q_{in,\,base\,1} = 1 \bullet 10^9 \text{ cfs}$$
$$Q_{in,\,base\,2} = 1 \bullet 10^8 \text{ cfs}$$

$$\text{then,}$$
$$Q_{IN,\,t} = \begin{cases} 1 \bullet 10^9 + 1 \bullet 10^8 \text{ cfs 6 day/week} & \textit{baseflow} \\ 1.4\left(1 \bullet 10^9 + 1 \bullet 10^8\right) \text{ cfs 1 day/week} & \textit{precipitation 24 hr/week} \end{cases}$$

2) Step 2

$$\text{Find } Q_{OUT,\,t}$$

$$\text{Given } V_{t-1},$$
$$Q_{OUT,\,t} = \begin{cases} 1.1 \bullet 10^9 \text{ cfs} & V_{t-1} < 0.9V_{max} \textit{ (de facto spill rate)} \\ 1.5 \bullet 10^9 \text{ cfs} & V_{t-1} >= 0.9V_{max} \textit{ (overflow spill rate)} \end{cases}$$

3) Step 3

$$\text{Finally,}$$
$$V_t = V_{t-1} + \left(Q_{IN,\,t} - Q_{OUT,\,t}\right)\Delta t$$

Initial Value Problem:

$$V_{t=1} = V_0 + \left(Q_{IN,\,t=1} - Q_{OUT,\,t=1}\right)(1)\ \text{day}$$

$$Q_{IN,\,t=1} = \ ?$$
$$Q_{OUT,\,t=1} = \ ?$$

1) Step 1

$$Q_{IN,\,t=1} = \ ?$$

$$\text{Given,}$$
$$Q_{IN,\,t} = \begin{cases} 1\bullet10^9 + 1\bullet10^8 \text{ cfs } 6 \text{ day/week} & \textit{baseflow} \\ 1.4\left(1\bullet10^9 + 1\bullet10^8\right) \text{ cfs } 1 \text{ day/week} & \textit{precipitation 24 hr/week} \end{cases}$$
$$\text{with the storm modeled on the last day of each week, then,}$$
$$Q_{IN,\,t=1} = \ 1\bullet10^9 + 1\bullet10^8$$

$$\Longrightarrow Q_{IN,\,t=1} = \ 1.1\bullet10^9$$

2) Step 2

$$Q_{OUT,\,t=1} = \ ?$$

$$\text{Given,}$$
$$V_0 = 0.8V_{max}$$
$$= 0.8(1\bullet10^{15})$$
$$V_0 = \ 8\bullet10^{14}$$

$$\text{and,}$$
$$Q_{OUT,\,t} = \begin{cases} 1.1\bullet10^9 \text{ cfs} & V_0 < 0.9V_{max}\ \textit{(de facto spill rate)} \\ 1.5\bullet10^9 \text{ cfs} & V_0 >= 0.9V_{max}\ \textit{(overflow spill rate)} \end{cases}$$

$$\text{with,}$$
$$V_0 = \ 8\bullet10^{14} < \ 9\bullet10^{14} = 0.9V_{max}$$

$$\text{then,}$$

$$\implies Q_{OUT,\,t=1} = \ 1.1 \bullet 10^9 \text{ cfs}$$

3) Step 3

$$\text{Finally,}$$

$$V_{t=1} = \ ?$$

$$V_{t=1} = V_0 + \left(Q_{IN,\,t=1} - Q_{OUT,\,t=1}\right) \Delta t$$

$$= 8 \bullet 10^{14} \ (\text{ft}^3) \ + (1.1 \bullet 10^9 - 1.1 \bullet 10^9) \ (\text{cfs}) \ 86400 \ (\text{s}/\text{c}$$

$$\implies V_{t=1} = 8 \bullet 10^{14} \text{ ft}^3$$

In [1]:
```
%matplotlib inline
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
pd.options.mode.chained_assignment = None  # default='warn'

# river 1 base flow rate (cfs)
Q_in1base = 1e9
# river 2 base flow rate (cfs)
Q_in2base = 1e8

# setting x-coordinates, x=t, Dt=1 (day)
xl = range(0,100)

# setting river 1 flow vs time (cfs)
Q_in1 = [Q_in1base if i % 7 != 6 else 1.4*Q_in1base for i in range(len(xl))]
# setting river 2 flow vs time (cfs)
Q_in2 = [Q_in2base if i % 7 != 6 else 1.4*Q_in2base for i in range(len(xl))]
# setting total flow in vs time (cfs)
Q_IN = [x + y for x, y in zip(Q_in1, Q_in2)]
```

In [2]:
```
### Simulated Lake Volume & Spillway Out-Flow Rate vs Time ###

## Data table prep
# df dict
df_dict = {
    "Q_IN":Q_IN,
    "V_t_i":0,
    "Q_OUT":0,
    "V_t":0
}
```

```
# convert to df
lake_sim = pd.DataFrame(df_dict)
df = pd.DataFrame(lake_sim)
df1 = pd.DataFrame(lake_sim)
```

In [3]:

```
## setting parameters, variables, and constants...

# time delta = 1 day = 86400 s/day
Dt = 86400
# max lake Volume (ft^3)
V_max = 1e15
# initial lake Volume (ft^3)
V_0 = 0.8*V_max

# dam spillway flow rate, typ. < 90% V_max (cfs)
Q_out1 = 1.1e9
# dam spillway flow rate, overflow >= 90% V_max (cfs)
Q_out2 = 1.5e9


## solving Initial Value Problem using given boundary conditions...

### df ###
# 1)   Vt_i = V_t-1 + Q_IN*Dt
df["V_t_i"][0] = V_0 + df["Q_IN"][0]*Dt

# 2)  Q_OUT = ? ...
if(df["V_t_i"][0] < 0.9*V_max):
    df["Q_OUT"][0] = Q_out1      # de-facto spillrate
else:
    df["Q_OUT"][0] = Q_out2      # overflow spillrate

# 3)   Vt = Vt_i - Q_OUT*Dt
df["V_t"][0] = df["V_t_i"][0] - df["Q_OUT"][0]*Dt


### df1 ###
# drop V_t_i column... V_t-1 = V_t of i-1
df1.drop("V_t_i", axis=1, inplace=True)

# 1)  Q_IN = ...premodeled

# 2)  Q_OUT = ? ...
if(V_0 < 0.9*V_max):
    df1["Q_OUT"][0] = Q_out1    # de-facto spillrate
else:
    df1["Q_OUT"][0] = Q_out2    # overflow spillrate

# 3)  V_t = ? ...
df1["V_t"][0] = V_0 + (df1["Q_IN"][0] - df1["Q_OUT"][0])*Dt
```

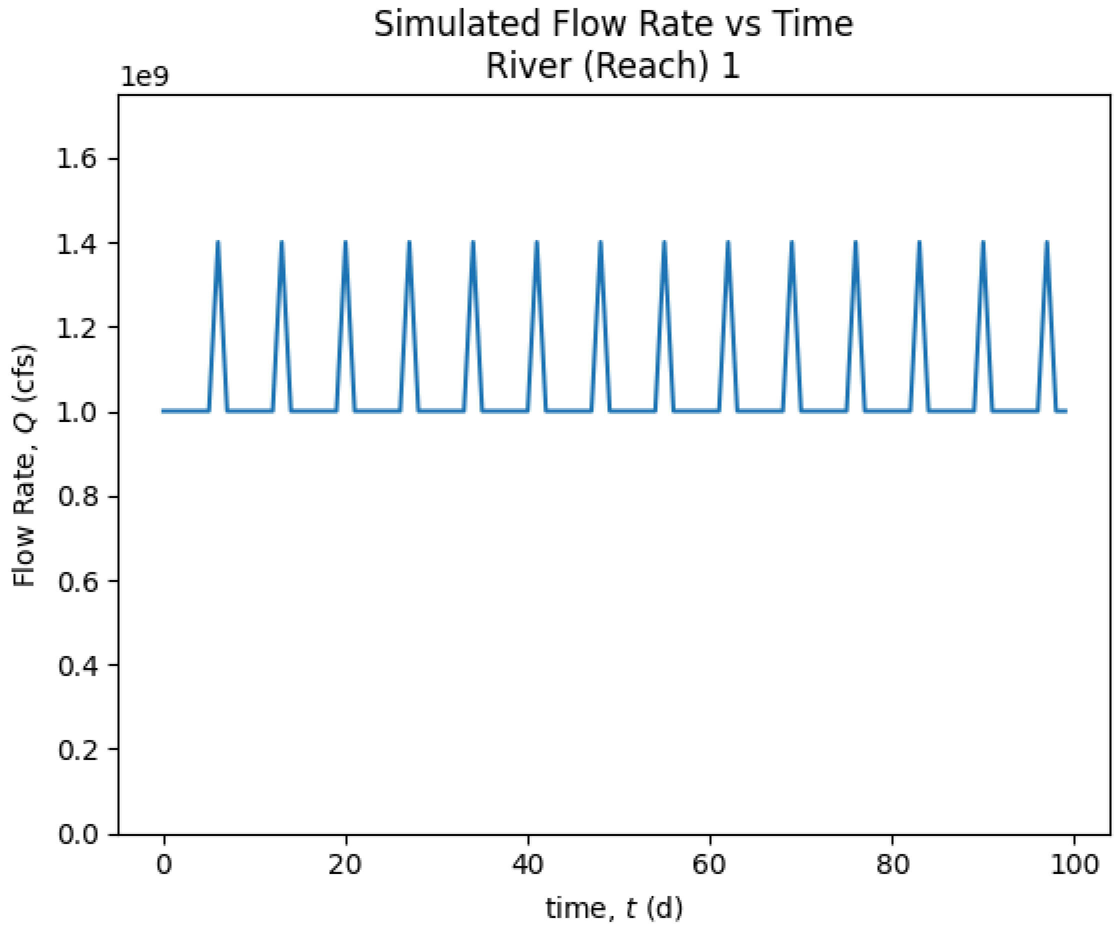
In [4]:

```
## simmulating...

### df ###
for t in range(1, len(df)):

    # setting functional variables...
    j = t - 1
```

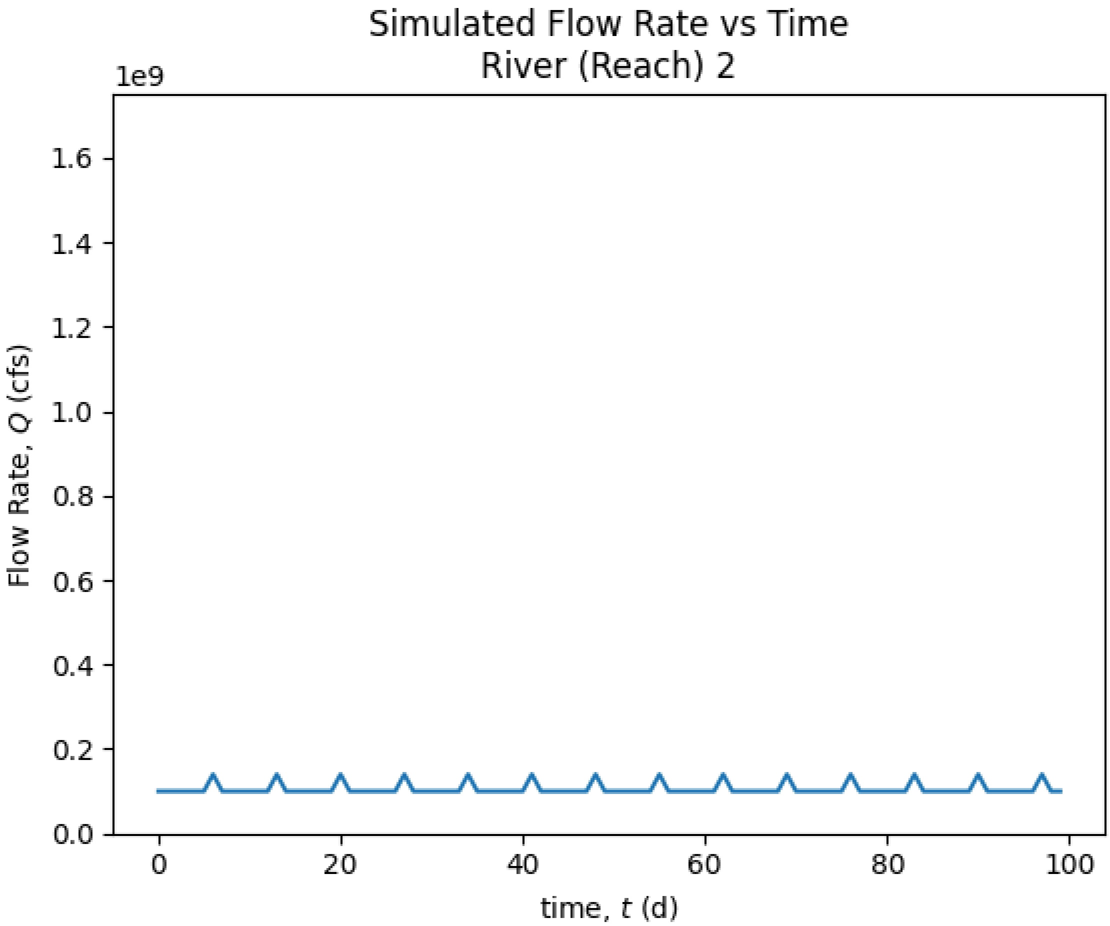
```
    # 1)   Vt_i = V_t-1 + Q_IN*Dt
    df["V_t_i"][t] = df["V_t"][j] + df["Q_IN"][t]*Dt

    # 2)   Q_OUT = ? ...
    if(df["V_t_i"][t] < 0.9*V_max):
        df["Q_OUT"][t] = Q_out1    # de-facto spillrate
    else:
        df["Q_OUT"][t] = Q_out2    # overflow spillrate

    # 3)   Vt = Vt_i - Q_OUT*Dt
    df["V_t"][t] = df["V_t_i"][t] - df["Q_OUT"][t]*Dt


### df1 ###
for t in range(1, len(df1)):
    # setting functional variables...
    j = t - 1

    # 1)   Q_IN = ...premodeled

    # 2)   Q_OUT = ? ...
    if(df1["V_t"][j] < 0.9*V_max):
        df1["Q_OUT"][t] = Q_out1    # de-facto spillrate
    else:
        df1["Q_OUT"][t] = Q_out2    # overflow spillrate

    # 3)   Vt = Vt_i - Q_OUT*Dt
    df1["V_t"][t] = df1["V_t"][j] + (df1["Q_IN"][t] - df1["Q_OUT"][t])*Dt
```

In [5]:

```
### Simulated Flow Rate vs Time - River 1
plt.plot(xl, Q_in1)                        #create plot
plt.ylim(0, 1.75*Q_in1base)      #set the range of y-values displayed
plt.title('Simulated Flow Rate vs Time\nRiver (Reach) 1')          #set the plot titl
plt.xlabel('time, $t$ (d)')     #set the label of the x-axis
plt.ylabel('Flow Rate, $Q$ (cfs)')     #set the label of the y-axis
plt.show()                               #show plot - not necessary in the inline mode
```



In [6]:
```
### Simulated Flow Rate vs Time - River 2
plt.plot(xl, Q_in2)      #create plot
plt.ylim(0, 1.75*Q_in1base)      #set the range of y-values displayed
plt.title('Simulated Flow Rate vs Time\nRiver (Reach) 2')      #set the plot title
plt.xlabel('time, $t$ (d)')     #set the label of the x-axis
plt.ylabel('Flow Rate, $Q$ (cfs)')      #set the label of the y-axis
plt.show()                      #show plot - not necessary in the inline mode
```

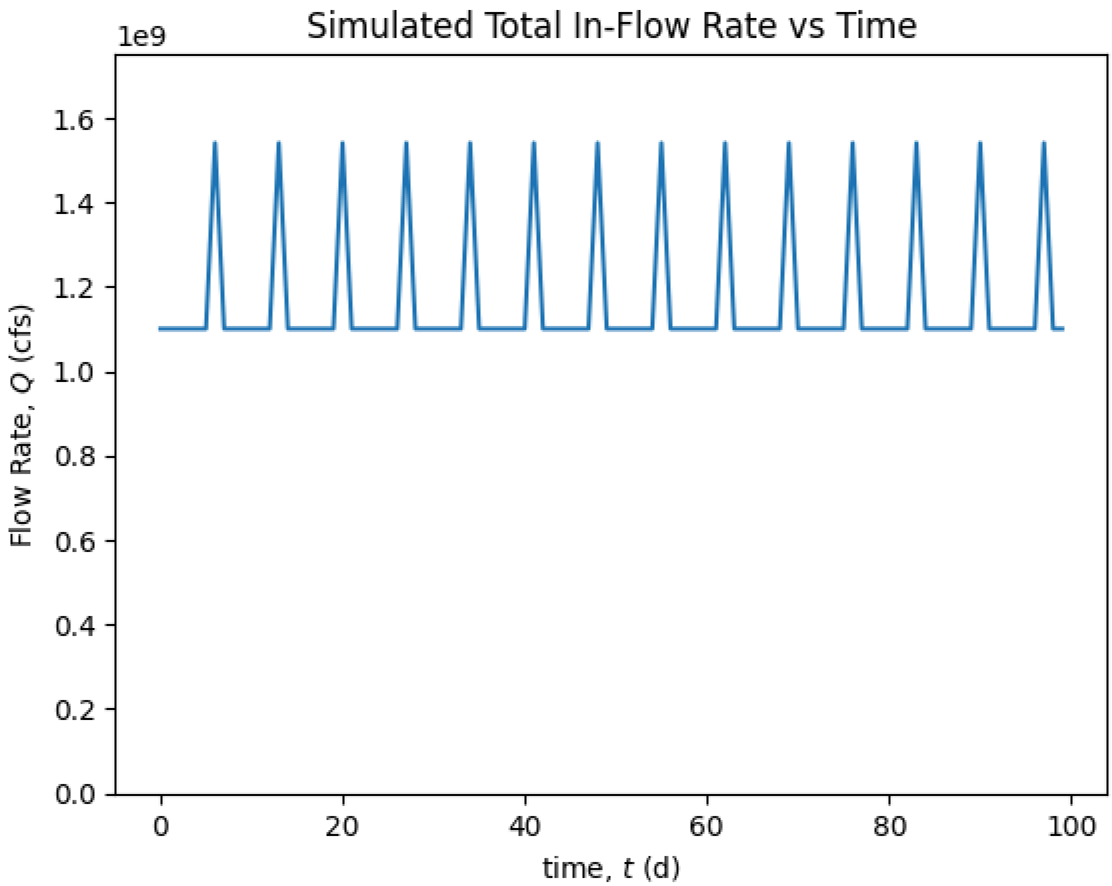


In [7]:

```
### Simulated Total In-Flow Rate vs Time
plt.plot(xl, Q_IN)                    #create plot
plt.ylim(0, 1.75*Q_in1base)      #set the range of y-values displayed
plt.title('Simulated Total In-Flow Rate vs Time')          #set the plot title
plt.xlabel('time, $t$ (d)')     #set the label of the x-axis
plt.ylabel('Flow Rate, $Q$ (cfs)')     #set the label of the y-axis
```

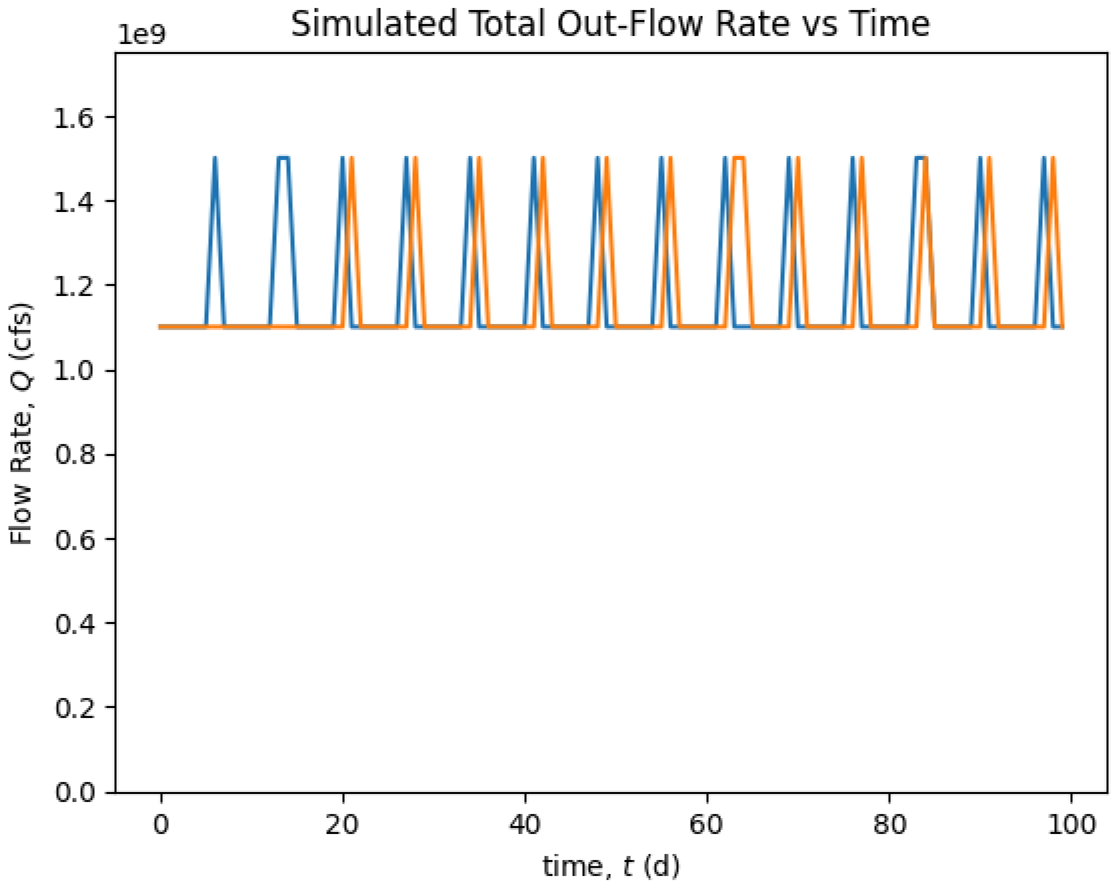
Out[7]:

```
Text(0, 0.5, 'Flow Rate, $Q$ (cfs)')
```



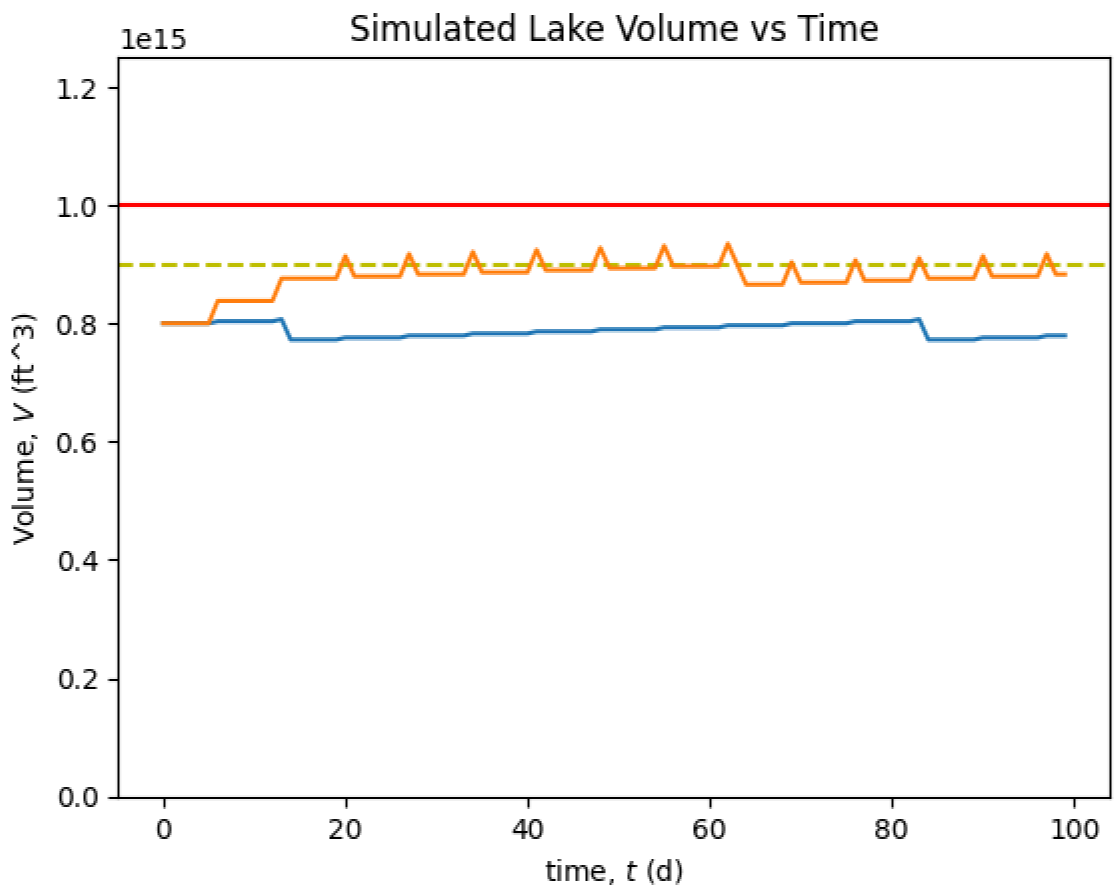
In [8]:
```
### Simulated Total Out-Flow Rate vs Time

### df ###
Q_OUT = df["Q_OUT"]              #y-coordinates, total in-flow (cfs), blank list

plt.plot(xl, Q_OUT)              #create plot
plt.ylim(0, 1.75*Q_in1base)      #set the range of y-values displayed
plt.title('Simulated Total Out-Flow Rate vs Time')           #set the plot title
plt.xlabel('time, $t$ (d)')        #set the label of the x-axis
plt.ylabel('Flow Rate, $Q$ (cfs)')     #set the label of the y-axis


### df1 ###
Q_OUT = df1["Q_OUT"]              #y-coordinates, total in-flow (cfs), blank list

plt.plot(xl, Q_OUT)               #create plot
```

Out[8]:
```
[<matplotlib.lines.Line2D at 0x178428562c0>]
```



In [9]:
```
### Simulated Lake Volume vs Time

### df ###
Vt = df["V_t"]             #y-coordinates, total volume (ft^3)

plt.plot(xl, Vt)                                                #create plot
plt.axhline(y=V_max, color='r', linestyle='-', label='$$V_{max}$$')          #plot max
plt.axhline(y=0.9*V_max, color='y', linestyle='--')          #plot max volume
plt.ylim(0, 1.25*V_max)      #set the range of y-values displayed
plt.title('Simulated Lake Volume vs Time')           #set the plot title
plt.xlabel('time, $t$ (d)')     #set the label of the x-axis
plt.ylabel('Volume, $V$ (ft^3)')     #set the label of the y-axis



### df1 ###
Vt = df1["V_t"]             #y-coordinates, total volume (ft^3)

plt.plot(xl, Vt)                                                #create plot
```

Out[9]:
```
[<matplotlib.lines.Line2D at 0x178425a5ea0>]
```



e) Describe insights from the results based on prompts listed below

1) What is the maximum water volume in the lake over 100 days? What percent of the capacity is this?
2) What is the first day that the plant operator will need to open the spillway?
3) How many days in the first 100 days does the dam operator need to open the spillway?

In [10]:
```
# 1)  get max lake volume during 100 day simulation period
max_vol = df["V_t"].max(0)
max_vol1 = df1["V_t"].max(0)

# 2)  get day number when spillway first opened
day = df[df["Q_OUT"].gt(Q_out1)].index[0] + 1
day1 = df1[df1["Q_OUT"].gt(Q_out1)].index[0] + 1

# 3)  Get count of values greater than 20 in the column 'C'
count = df["Q_OUT"][df["Q_OUT"] >= Q_out2].count()
count1 = df1["Q_OUT"][df1["Q_OUT"] >= Q_out2].count()

######

# 1)  print max water volume in the lake over time period & percentage of maximum capa
print("\n1) Maximum Lake Volume over time period:\n",
      "sim 1: ", "{:.2E}".format(max_vol), " ft^3  (", "{:.1%}".format(max_vol/V_max),
      "sim 2: ", "{:.2E}".format(max_vol1), " ft^3  (", "{:.1%}".format(max_vol1/V_max

# 2)  print first day plant operator will need to open spillway
print("2) First day spillway opened:\n",
```

```
      "sim 1: day ", day, "\n",
      "sim 2: day ", day1, "\n")

# 3)  print how many days in the first 100 days the dam operator needs to open the spi
print("3) Number of days spillway opened:\n",
      "sim 1: ", count, " days\n",
      "sim 2: ", count1, " days\n")
```

```
1) Maximum Lake Volume over time period:
 sim 1:  8.07E+14  ft^3  ( 80.7%  V_max)
 sim 2:  9.35E+14  ft^3  ( 93.5%  V_max)

2) First day spillway opened:
 sim 1: day  7
 sim 2: day  22

3) Number of days spillway opened:
 sim 1:  16  days
 sim 2:  13  days
```

A large variable in the model is how the precipitation is modeled. In the current model (consistent in both simulations 1 & 2), the precipitation is seen every 7 days (the end of each week). For some geographic areas at certain times, such predictability could be feasable. In other cases, it may be more appropriate to add some degree of randomness into the precipitation variable.

In any case, each of the two simulations were modeled differently based on a conceptual difference of parameters. In the first attempt at developing code for the model, a logical error existed in the lines which determined the outflow value for any given time step Δt. This became apparent when examining the initial model's order of operations within the algorithm conceptually.

In Simulation 1's model, Outflow for each time step, Δt is determined by an intermediate volume calculation, $V_{t,\,i}$ (" `V_t_i` ***"), from the sum of the current time step's predetermined inflow Q_{IN} plus V_{t-1} (*** `V_t_i = df["Q_IN"][i] + df["V_t"][j]` ***, where*** `j -= i` ***). In Simulation 2's model however, Outflow for each time step, Δt is determined by the parameters set in the computational model initially defined mathematically; Outflow is determined soley by*** `df["V_t"][j]` ***, V_{t-1} .***

The fundamental difference in the two simulations is that in Simulation 1, the intermediate volume calculation, $V_{t,\,i}$, which adds predermined modeled Inflow over the course of the day to V_{t-1} , serves as of predictive determination for the Outflow. Whereas, Simulation 2 determines the outflow for each day solely by V_{t-1} , the Volume of the lake at the end of the previous day and not anticipating any potential change in Inflow throughout the next day.

Observing the outcomes of the plots for Lake Volume over time for each of the simulations, it could be argued to be useful in some scenarios to use an intermediate volume calculation, $V_{t,\,i}$, to potentially stablize the lake volume over time as well as the outflow. This calculation could be viewed as a predictive volume value, as if there were only inflow

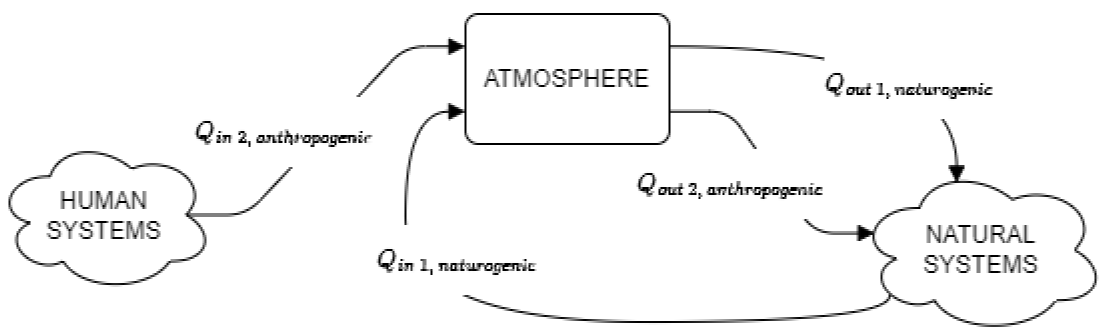
throughout the day into and on top of the existing lake volume. This could be useful or potentially damaging, based on the confidence in the weather modeling as opposed to a deterministic method.

f) (550 students only) Perform one sensitivity analysis (e.g., vary in the inputs/output flows over time, introduce a feedback loop, etc., and create time series plots). Describe your sensitivity analysis and interpret the plots.

System #2: The Carbon Cycle

Consider the stock of carbon in the atmosphere. The mass of carbon in the atmosphere at time t=0 is M_0 = 800 GtC. Natural CO_2 inputs to the system from plant respiration, microbial respiration, and decomposition on land and sea combine to emit 210 gigatons carbon per year (GtC/yr) into the atmosphere. Without human activity, these natural systems would remove an equivalent amount of carbon from the atmosphere (210 GtC/yr). Human activity emits 9 GtC/yr. Feedback loops result in the absorption of 5/9 of human-emitted carbon into the ocean and biomass. Consider the concentration of CO_2 [in parts per million] in the atmosphere as the stock of interest.

a) Sketch the conceptual model in a stock-and-flow diagram



Given:

$$Q_{in\ 1,\ naturogenic} = Q_{out\ 1,\ naturogenic} = 210 \text{ Gt C/yr}$$
$$Q_{in\ 2,\ anthropogenic} = 9 \text{ Gt C/yr}$$
$$Q_{out\ 2,\ anthropogenic} = \left(\frac{5}{9}\right)\ Q_{in\ 2,\ anthropogenic} = 5 \text{ Gt C/yr}$$

Relavent Constants:

$M_0 = 800$ Gt C, mass of Carbon in atmosphere at time $t = 0$

$M_{Atm} = 5.137 \cdot 10^{21}$ g , mass of atmosphere

$Mol_{CO_2} = 44 \ \frac{\text{g}}{\text{mol}}$, molar mass of Carbon Dioxide

$Mol_C = 12 \ \frac{\text{g}}{\text{mol}}$, molar mass of Carbon

$Mol_{Atm} = 29 \ \frac{\text{g}}{\text{mol}}$, molar mass of atmosphere

$1 \ Gt = 1 \cdot 10^{15}$ g , grams per Gigaton

$$C_{CO_2} = \text{ppm} = \frac{M_C \ (\text{g})}{M_{atm} \ (\text{g})} \cdot \frac{Mol_{atm} \left(\frac{\text{g}}{\text{mol}}\right)}{Mol_C \left(\frac{\text{g}}{\text{mol}}\right)} \cdot 10^6 \ \left(\frac{\text{mol}}{\text{mil. mol}}\right)$$

In [11]:
```
### Simulated Atmospheric CO_2 Concentration vs Time ###

## setting parameters, variables, and constants...

# mass of Carbon in atmosphere at time t=0 (g)
M_0 = 800e15
# mass of atmosphere (g)
M_atm = 5.137e21
# molar mass of Carbon Dioxide (g/mol)
Mol_CO2 = 44
# molar mass of Carbon (g/mol)
Mol_C = 12
# molar mass of atmosphere (g/mol)
Mol_atm = 29


# CO2 in, naturogenic (g/yr)
Q_in1 = 210e15
# CO2 in, anthropogenic (g/yr)
Q_in2 = 9e15
# CO2 out, naturogenic (g/yr)
Q_out1 = 210e15
# CO2 out, anthropogenic (g/yr)
Q_out2 = 5e15
```

In [12]:
```
## setting data frame

## Data table prep
# df dict
df_dict = {
    "t":range(0,101),
    "M_t":0,  # atmospheric Carbon mass M(t) (g)
    "C_t":0,  # atmospheric Carbon concentration C(t) (ppm)
}

# converting to df
atm_sim = pd.DataFrame(df_dict)
df = pd.DataFrame(atm_sim)
```

b) Write a differential equation describing the model and solve it

$$M_t = M_0 + \Delta M = M_0 + \frac{dM}{dt}$$

$$\Delta M = \frac{dM}{dt} = \ ?$$

$$\frac{dM}{dt} = Q_{in\,1} + Q_{in\,2} - Q_{out\,1} - Q_{out\,2}$$

$$\text{with,}$$

$$Q_{in\,1} = Q_{out\,1}$$

$$Q_{out\,2} = \frac{5}{9}\,Q_{in\,2}$$

$$\text{then,}$$

$$\frac{dM}{dt} = Q_{in\,2} - \frac{5}{9}\,Q_{in\,2}$$

$$\frac{dM}{dt} = \frac{4}{9}\,Q_{in\,2}$$

$$\int_{t_0}^{t} \frac{dM}{dt} dt = \frac{4}{9} \int_{t_0}^{t} Q_{in\,2}\,dt$$

$$dM \,\Big|_{t_0}^{t} = \frac{4}{9} [Q_{in\,2}] \,\Big|_{t_0}^{t}$$

$$M_t - M_{t_0} = \frac{4}{9}\,Q_{in\,2}\,(t - t_0)$$

$$\implies \Delta M = \frac{4}{9}\,Q_{in\,2}\,\Delta t$$

c) What is an appropriate time step (including units)? Hint: convert all input data to consistent units in this time step

$$\Delta t = 1 \text{ year}$$

From the given boundary conditions, the largest time interval unit, 1 year, is suitable for mass and flow estimations for the concentration C_{CO_2} of Carbon Dioxide in the atmosphere at any given time step t .

d) Plot important flows and stocks over time

$$C_{CO_2} = \text{ppm} = \frac{M_C\ (\text{g})}{M_{atm}\ (\text{g})} \cdot \frac{Mol_{atm}\left(\frac{\text{g}}{\text{mol}}\right)}{Mol_C\left(\frac{\text{g}}{\text{mol}}\right)} \cdot 10^6 \ \left(\frac{\text{mol}}{\text{mil. mol}}\right)$$

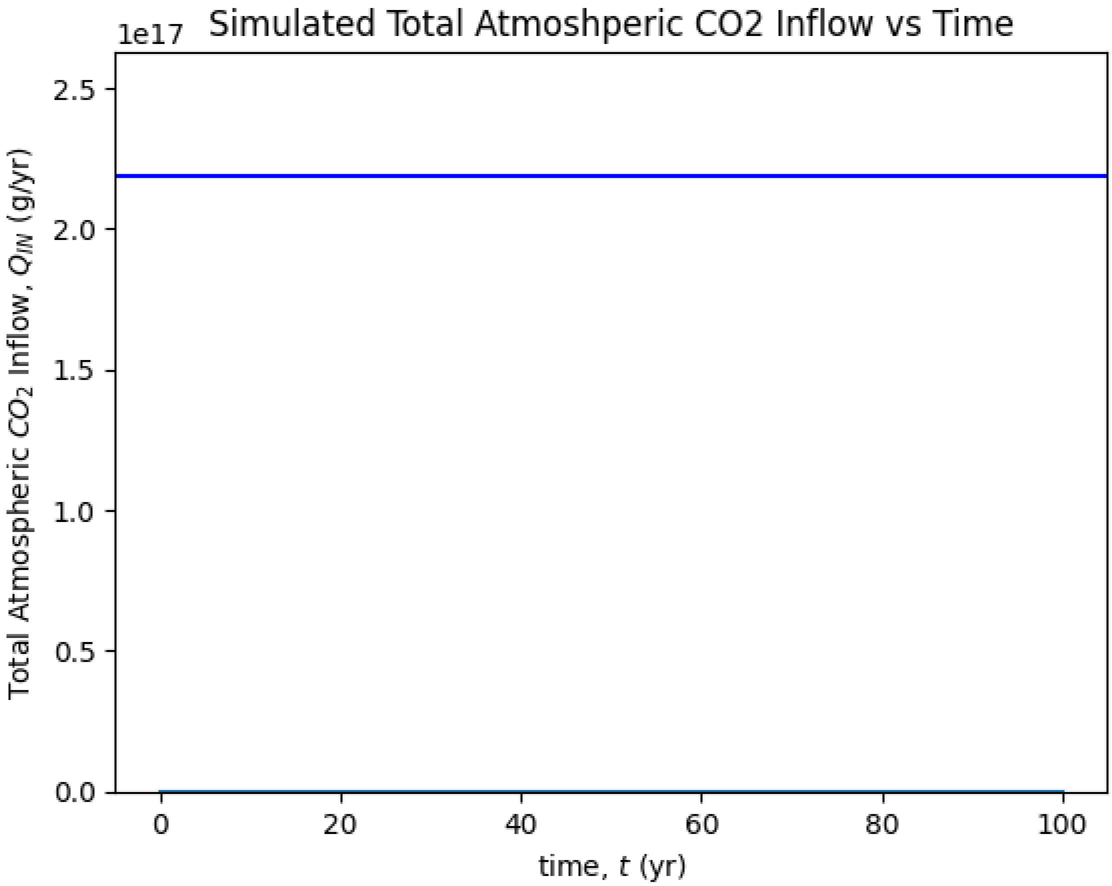
In [13]:
```
## solving Initial Value Problem with given boundary conditions...

df["M_t"][0] = M_0
df["C_t"][0] = (M_0/M_atm)*(Mol_atm/Mol_C)*(10^6)
print("C_(CO2)(0):", df["C_t"][0])
```

```
C_(CO2)(0): 0.004516254623321005
```

In [14]:
```
## simmulating...

for t in range(1, len(df)):

    # setting functional variables...
    j = t - 1

    # 1)  compute atmospheric CO2 mass M(t)
    df["M_t"][t] = df["M_t"][j] + (4/9)*Q_in2

    # 2)  compute atmospheric CO2 concentration C(t)
    df["C_t"][t] = (df["M_t"][t]/M_atm)*(Mol_atm/Mol_C)*(10^6)


# setting x-coordinates, x=t, Dt=1 (yr)
xl = range(0,len(df["t"]))
```

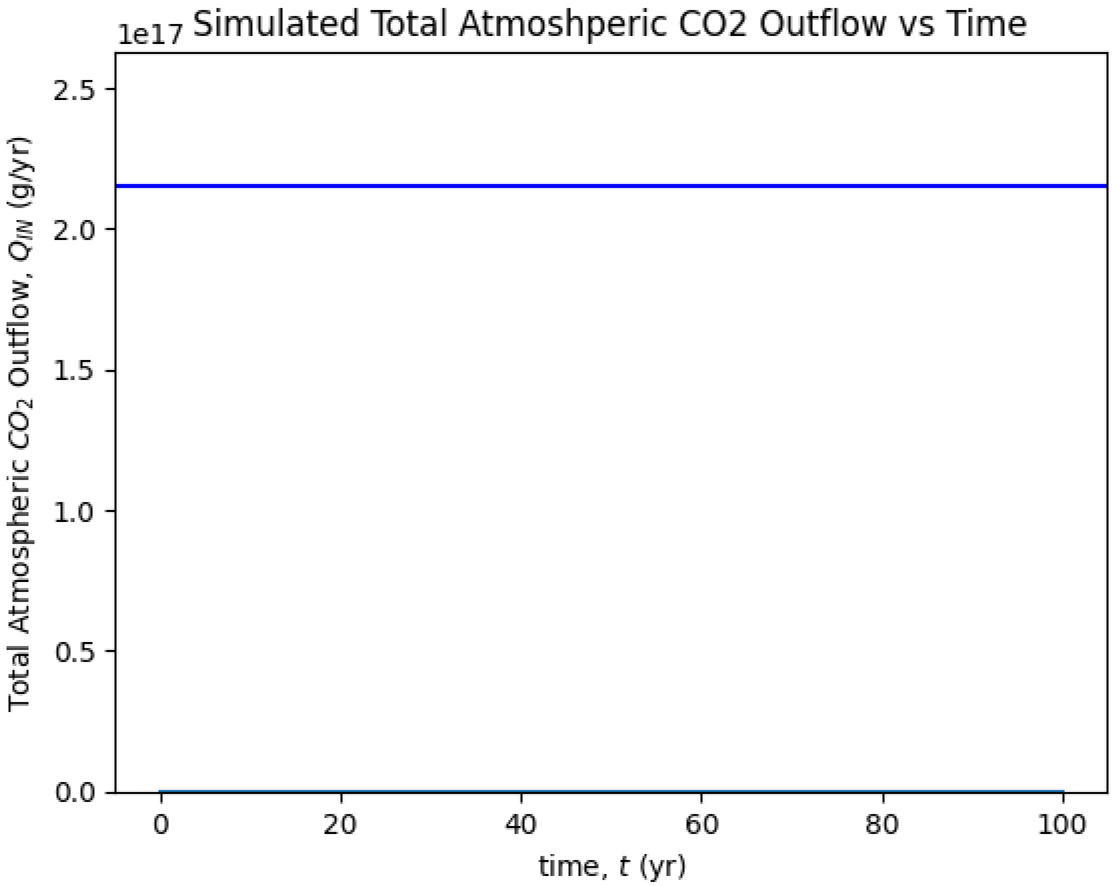
In [15]:
```
### Simulated Total Atmoshperic CO2 Inflow vs Time

Q_IN = Q_in1 + Q_in2

plt.plot(xl)                        #create plot
plt.axhline(y=Q_IN, color='b', linestyle='-', label='$$V_{max}$$')          #plot Q_IN
plt.ylim(0, 1.2*Q_IN)      #set the range of y-values displayed
plt.title('Simulated Total Atmoshperic CO2 Inflow vs Time')          #set the plot ti
plt.xlabel('time, $t$ (yr)')     #set the label of the x-axis
plt.ylabel('Total Atmospheric $CO_2$ Inflow, $Q_{IN}$ (g/yr)')     #set the label of t
```

Out[15]:
```
Text(0, 0.5, 'Total Atmospheric $CO_2$ Inflow, $Q_{IN}$ (g/yr)')
```



In [16]:
```
### Simulated Total Atmoshperic CO2 Outflow vs Time

Q_OUT = Q_out1 + Q_out2

plt.plot(xl)                        #create plot
plt.axhline(y=Q_OUT, color='b', linestyle='-', label='')          #plot Q_OUT
plt.ylim(0, 1.2*Q_IN)      #set the range of y-values displayed
plt.title('Simulated Total Atmoshperic CO2 Outflow vs Time')          #set the plot t
plt.xlabel('time, $t$ (yr)')     #set the label of the x-axis
plt.ylabel('Total Atmospheric $CO_2$ Outflow, $Q_{IN}$ (g/yr)')     #set the label of
```

Out[16]:
```
Text(0, 0.5, 'Total Atmospheric $CO_2$ Outflow, $Q_{IN}$ (g/yr)')
```

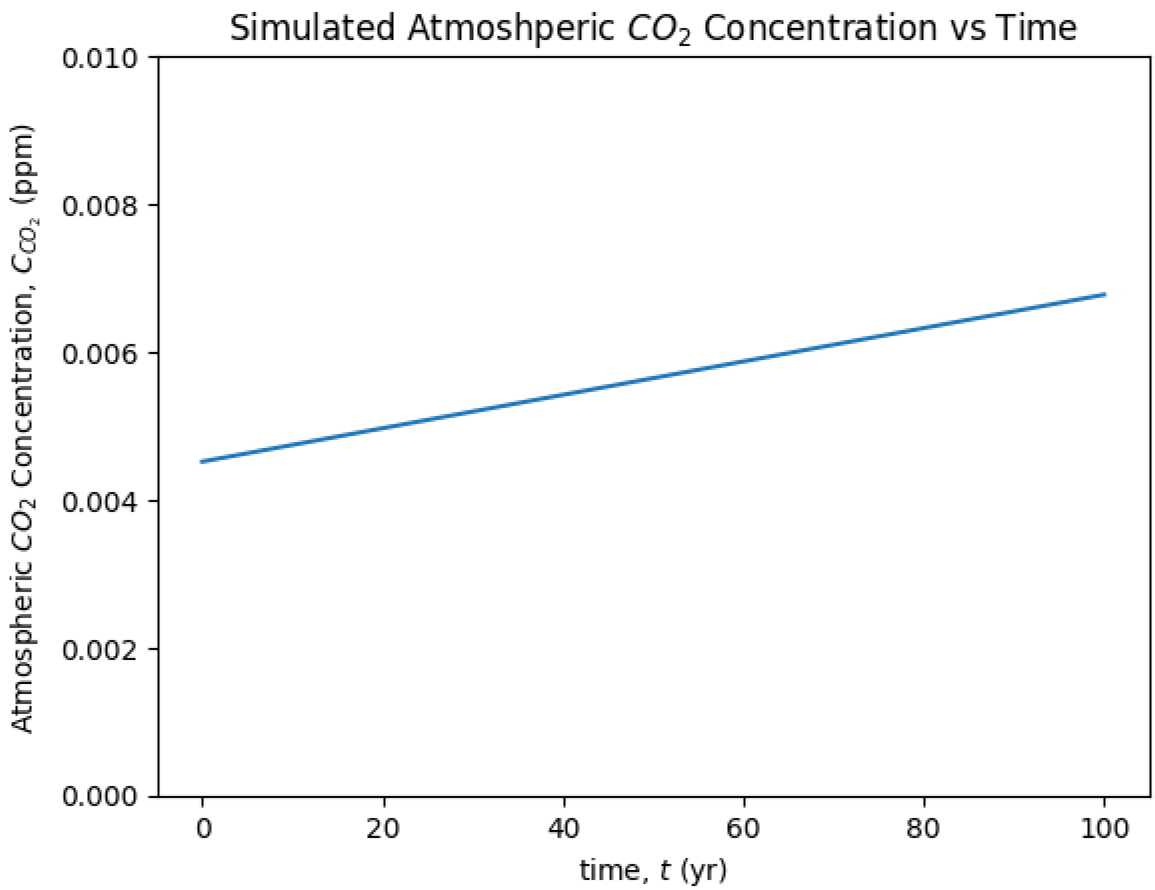


In [17]:

```
### Simulated Atmoshperic CO2 Concentration vs Time

Ct = df["C_t"]

plt.plot(xl, Ct)                    #create plot
plt.ylim(0, 0.01)      #set the range of y-values displayed
plt.title('Simulated Atmoshperic $CO_2$ Concentration vs Time')          #set the plo
plt.xlabel('time, $t$ (yr)')     #set the label of the x-axis
plt.ylabel('Atmospheric $CO_2$ Concentration, $C_{CO_{2}}$ (ppm)')     #set the label
```

Out[17]:

```
Text(0, 0.5, 'Atmospheric $CO_2$ Concentration, $C_{CO_{2}}$ (ppm)')
```



e) Describe insights from the results based on prompts listed below

1) At the current emissions rate with the feedback loop, what CO_2 concentration is expected in the atmosphere in 50 years? 100 years?

(550 students) For the sensitivity analysis, consider a scenario in which human emissions decrease by 5% each year. How different are concentrations in 50 and 100 years from the base case?

In [18]:
```
# get atmospheric CO2 concentration, t=50, t=100
C_50 = df["C_t"][51]
C_100 = df["C_t"][100]

# print results
print("atmoshperic CO2 concentration C(50):", "{:.2E}".format(C_50), " ppm",
      "\natmoshperic CO2 concentration C(100):", "{:.2E}".format(C_100), " ppm")
```

```
atmoshperic CO2 concentration C(50): 5.67E-03  ppm
atmoshperic CO2 concentration C(100): 6.77E-03  ppm
```

Although natural Carbon sinks allow for the full absorption of naturogenic emissions as well as partial amounts of anthropogenic emissions, human activities produce more Carbon emissions than naturogenic sinks are able to accomidate. The differential equation derived from the given data is a linear model, with a constant rate of increase in atmospheric Carbon concentration. This may be a suitable model for short term predictive estimates, however, for a more accurate estimate, anthropogenic emission rates (

$Q_{out\ 2,\ anthropogenic}$***) would most likely be better modeled as increasing a non-linearly, unless major cultural and societal changes occur throughout the world to both reduce anthropogenic Carbon emissions as well as implement more anthropogenic Carbon sinks.***