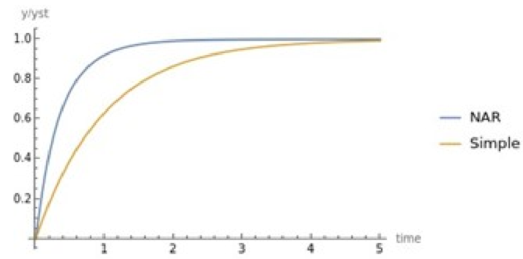
Assignment 2

A. Negative auto-regulation

a)

Response time for circuit with simple regulation is 0.693147 s.
Response time for circuit with negative autoregulation is 0.248914 s.

Note: Our goal is to have same steady state concentration for both NAR and simple regulatory circuit



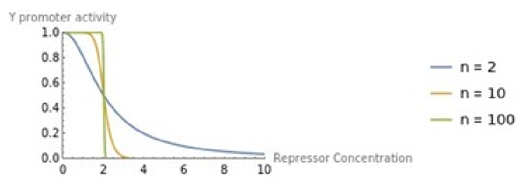

```
params = {β → 1, k → 0.3, α → 1, n → 2}
solNAR = NDSolve [{y '[t] == β / (1 + (y[t] / k)^ n) - α y[t], y[0] == 0.00} /. params , y, {t, 0, 5}]
ystNAR = y[t] /. NSolve[{0 == β / (1 + (y[t] / k)^ n) - α y[t] && y[t] > 0} /. params , y[t]]
params = Insert[params , βsimple → ystNAR⟦1⟧ , - 1]
solS = DSolve[{y '[t] == βsimple - α y[t], y[0] == 0} /. params , y, {t, 0, 5}] ystSimple = y[t] /.
NSolve[{0 == βsimple - α y[t] && y[t] > 0} /. params , y[t]]
p1 = Plot[{(y[t] /. solNAR) / ystNAR , (y[t] /. solS) / ystSimple }, {t, 0, 5}, PlotRange → All,
PlotLegends → {"NAR", "Simple "}, AxesLabel → {"time", "y/yst"}]
tSimp = t /. NSolve[{y[t] == ystSimple / 2 } /. solS , t] tNAR = t /. NSolve[{y[t] == ystNAR / 2} /.
solNAR , t]
```

```
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

def Simpleregulation(y,t):

    # The function returns dy/dt for the simple regulation circuit

    beta , k , alpha , n = 1 , 0.3 , 1 , 2

    dydt = beta - alpha * y

    return dydt

def NAR(y,t):

    # The function returns dy/dt for the negative regulatory circuit

    beta , k , alpha , n = 1 , 0.3 , 1 , 2

    dydt = beta * (1 / (1 + (y/k) ** n)) - alpha * y

    return dydt

# initial condition
y0 = 0

# time points
t = np.linspace(0,5)

# solve ODE for NAR
```

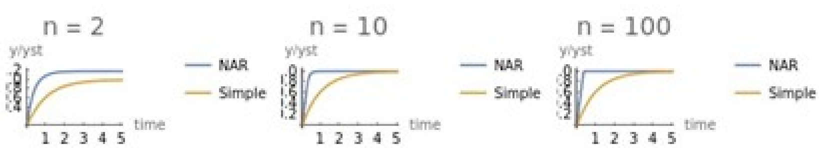
```
y1 = odeint(NAR,y0,t)

# solve ODE for Simple Regulation
y2 = odeint(Simpleregulation , y0 , t)

# plot results

plt.figure(dpi = 150)

plt.plot(t,y1 / y1[-1] , label = 'NAR')
plt.plot(t,y2 / y2[-1] , label = 'Simple Regulation')
plt.xlabel('time')
plt.ylabel('y(t)/Yst')

plt.legend()

plt.title('Dynamics for NAR and simple regulatory circuit')

plt.show()
```

b)

We will have to increase n (cooperativity) in hill function to reach the limit of step function of inhibition.



With the increase in n, we observe even faster initial growth and lower delay in auto repression to stop production at the desired steady-state



c)

Given, $\beta_{NAR} = 2, 20, 200\mu M$ and $n \to \infty$. Now since $y[t] << \beta/\alpha$, we can approximate $T_{1/2}^{NAR} = \frac{K}{2 * \beta_{NAR}}$. Response time of simple autoregulatory circuit is $T_{1/2}^{simple} = \frac{ln2}{\alpha}$

$$T_{1/2}^{NAR} = 0.3 \frac{\mu M}{2 * 2, 20, 200\mu M * h^{-1}} = 6.81 * 10^{-7} h$$

$$T_{1/2}^{simple} = \frac{ln2}{1} = 0.693h$$

d)

At the step function limit i.e $n \to \infty$ and $y[t] << \frac{\beta^{NAR}}{\alpha}$. We know $T_{1/2}^{NAR} = \frac{K}{2\beta}$ and $T_{1/2}^{simple} = \frac{ln2}{\alpha}$. Thus, $T_{1/2}^{NAR}$ does not depends on α while $T_{1/2}^{simple}$ is inversely proportional to α.

e)

Using the above mentioned formulas we can calculate $T_{1/2}^{NAR}$ and $T_{1/2}^{simple}$

α	$T_{1/2}^{NAR}$	$T_{1/2}^{simple}$
1	0.005	0.693
2	0.005	0.346
10	0.005	0.069
20	0.005	0.034

f)

Summary statement:

- Response time of NAR circuit depends on half − saturation constant K and maximal production rate β
- Response time of simple regulatory circuit depends on removal rate α
- Steady state concentration of NAR circuit depends on half − saturation constant K
- Steady state concentration of simple regulatory circuit depends on maximal production rate β simple and removal rate α

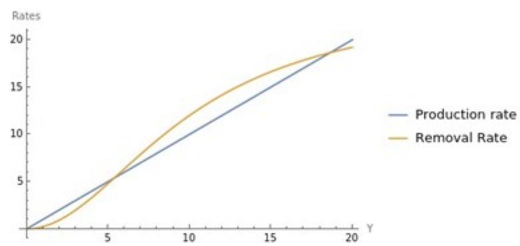
B. Positive Autoregulation

a)

Steady state concentration of y given $y[0] = 0\mu M$ is $0\mu M$.

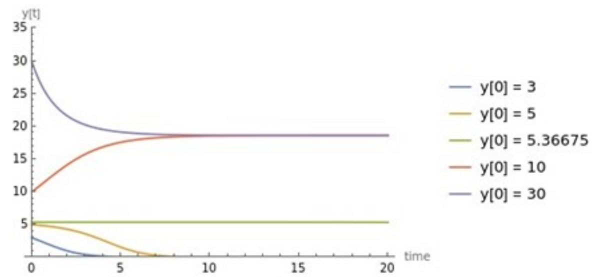
Steady state concentration of y given $y[0] = 30\mu M$ is $18.6332\mu M$

The given PAR circuit has three fixed points (y = 0 μM , y = 5.36675 μM *and* y = 18.6332 μM). The fixed point y = 5.36675 μM is unstable, thus steady state concentration of y is either 0 μM or 18.6332 μM which depends on initial conditions



b)

Critical concentration at which the systems switches from the OFF state to the ON state is y = 5.37775 μM.



```
params = { β → 24, k → 10, α → 1, n → 2}
solPAR1 = NDSolve [{y '[t] == β ((y[t] / k)^ n) / (1 + (y[t] / k)^ n) - α y[t], y[0] == 3} /. params , y, {t, 0, 20}]
solPAR2 = NDSolve [{y '[t] == β ((y[t] / k)^ n) / (1 + (y[t] / k)^ n) - α y[t], y[0] == 5} /. params , y, {t, 0, 20}]
solPAR3 = NDSolve [{y '[t] == β ((y[t] / k)^ n) / (1 + (y[t] / k)^ n) - α y[t], y[0] == 10} /. params
, y, {t, 0, 20}]
solPAR4 = NDSolve [{y '[t] == β ((y[t] / k)^ n) / (1 + (y[t] / k)^ n) - α y[t], y[0] == 30} /. params
, y, {t, 0, 20}] solPAR5 = NDSolve [ {y '[t] == β ((y[t] / k)^ n) / (1 + (y[t] / k)^ n) - α y[t], y[0] == 5.36675 } /.
params , y, {t, 0, 20}]
Plot[{y[t] /. solPAR1 , y[t] /. solPAR2 , y[t] /. solPAR5 , y[t] /. solPAR3 , y[t] /. solPAR4 }, {t, 0,
20}, PlotRange → {All, {0, 35}}, AxesLabel → {"time", "y[t]"} , PlotLegends → {"y[0] = 3" , "y[0] = 5" , "y[0] = 5.36675 " , "y[0] = 10" ,
```

```
# Part a

def PAR(y,t):

    # The function returns dy/dt for the positive autoregulatory circuit

    beta , k , alpha , n = 24 , 10 , 1 , 2

    dydt =  beta * ((y/k)**n)/(1+(y/k)**n) - alpha * y

    return dydt


# initial condition
y0_1 = 0
y0_2 = 30

# time points
t = np.linspace(0,10)

# solve ODE with first initial condition (y[0] = 0 uM)
y1 = odeint(PAR,y0_1,t)

# solve ODE with second initial condition (y[0] = 30 uM)
y2 = odeint(PAR , y0_2 , t)

# plot results

plt.figure(dpi = 150)

plt.plot(t , y1 , label = 'y[0] = 0 uM')

plt.plot(t,y2 , label = 'y[0] = 30 uM')
plt.plot(t, [y2[-1] for i in range(len(t))] , linestyle='dashed' , color = 'black')

plt.xlabel('time in seconds')
plt.ylabel('y(t)')

plt.title('Simulation of PAR circuit with different initial conditions')

plt.legend()

plt.show()

# Part b:

# In this question we will see how changing the intitial conditions will change the final steady state concentration

plt.figure(dpi = 150)

t = np.linspace(0,10)

# Plotting the curve for different concentration values
for conc in [3 , 5 , 5.36777 , 10 , 30]:

    plt.plot(t , odeint(PAR , conc , t) , label = 'y[0] = ' + str(conc) + ' uM')

plt.legend()

plt.xlabel('time in seconds')
plt.ylabel('y(t)')

plt.title('Simulation of PAR circuit with different initial conditions')
```

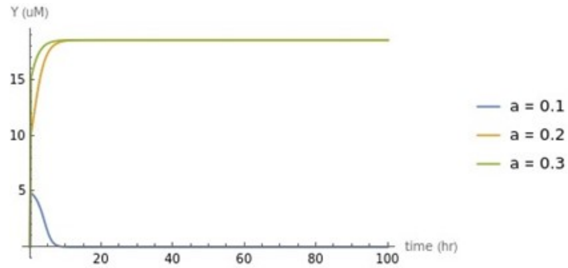
c)

System has three steady-states and only two of them are stable.

- y = 0 μM, *stable steady state*
- y = 5.36675 μM, *unstable steady state*
- y = 18.6332 μM, *stable steady state.*

d)

The steady-state concentration of Y after the pulse of production is gone depends on the duration of the pulse. If the pulse duration ≤ 0.10 hrs then the steady state concentration is 0 and if pulse duration ≥ 0.20 hrs then steady state concentration of Y is 18.6332 μM



Note: a is the duration of the pulse in hrs. Pulse duration to switch the steady state from 0 μM to 18.6332 μM has to be approximately greater than or equal to 0.2 hrs, which is evident from the above graph.

```
params = { β → 24, k → 10, α → 1, n → 2 }   p[t] = UnitStep [t] - UnitStep [t - r]
sol = solPAR = ParametricNDSolve [ {y '[t] == 50 * p[t] + β ((y[t] / k)^ n) / (1 + (y[t] / k)^ n) - α y[t], y[0] == 0} /. params , y, {t, 0,
Plot[Evaluate [Table[y[a][t] /. sol, {a, 0.1, 0.3, .1}]], {t, 0, 100}, PlotRange → All , PlotRange →
{All, {0, 50}} , AxesLabel → {"time (hr)" , "Y (uM)"} , PlotLegends → {"a = 0.1" , "a = 0.2" , "a = 0.3"} ]
```

```
# Part d:

def PAR(y , t , duration):

    # The function returns dy/dt for the new circuit
    # The new circuit has a additional production term which can be modelled using a step function

    def unit(t):

        # This is a unit step function

        if t <= duration and t >= 0:

            return 50

        else:

            return 0


    beta , k , alpha , n = 24 , 10 , 1 , 2

    dydt =  unit(t) + beta * ((y/k)**n)/(1+(y/k)**n) - alpha * y

    return dydt

plt.figure(dpi = 150)

t = np.linspace(0,50 , 1000)
y0 = 0

for duration in [0.05 , 0.1 , 0.15 , .2]:

    plt.plot(t , odeint(PAR , y0 , t , args = (duration , )) , label = 'Duration ' + str(duration) + ' s')
```

```
plt.legend()

plt.xlabel('time in seconds')
plt.ylabel('y(t)')

plt.title('Effect of production rate on final concentration')
```

C. Positive autoregulation without cooperativity

$$\frac{dy}{dt} = \beta\frac{y}{k+y} - \alpha y$$

a)

On fixed points, $\frac{dy}{dt} = 0$

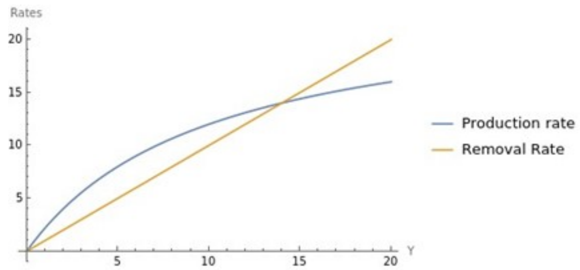
$$\frac{dy}{dt} = \beta\frac{y}{k+y} - \alpha y = 0$$

$$\implies y = 0, \frac{\beta}{\alpha} - k$$

Thus, system has two fixed points $y = 0, \frac{\beta}{\alpha} - k$

b)

System is monostable as the fixed point $y = \frac{\beta}{\alpha} - k$ is stable while the fixed point $y = 0$ is unstable. Fixed point $y = \frac{\beta}{\alpha} - k$ is stable because shifting y to either side (of $\frac{\beta}{\alpha} - k$) causes a return to fixed point




```
Plot[{24*y/(y+10) , y} , {y, 0 , 20} , PlotLegends->{"Production rate" , "Removal Rate"} , AxesLabel -> {"Y" , "Rates"}]
```

```
# Question C

# Part b

y = np.linspace(0 , 20 , 100)

beta = 24
k = 10
alpha = 1

production_rate = [beta * ele / (k + ele) for ele in y]
removal_rate = [ele * alpha for ele in y]

plt.figure(dpi = 150)
```

```
plt.plot(y , production_rate , label = "Production Rate")
plt.plot(y , removal_rate , label = "Removal Rate")

plt.xlabel("concentation of y")
plt.ylabel("Rates")

plt.legend()
```

Assignment 2

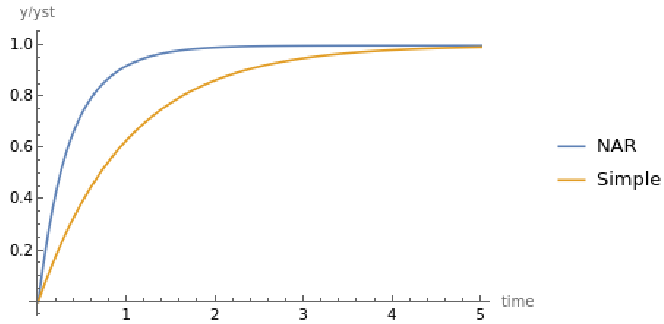
A)

a) Response time for circuit with simple regulation is 0.693147 s.
Response time for circuit with negative autoregulation is 0.248914 s.

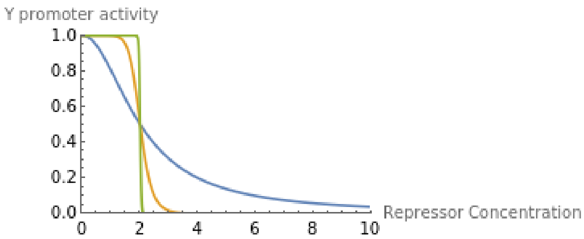
Note: Our goal is to have same steady state concentration for both NAR and simple regulatory circuit.

Code:

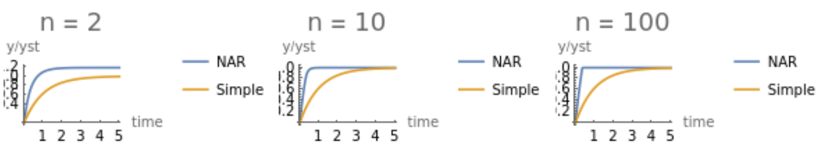
```
params = {β → 1, k → 0.3, α → 1, n → 2}
solNAR = NDSolve [{y '[t] == β / (1 + (y[t] / k)^ n) - α y[t], y[0] == 0.00} /. params , y, {t, 0, 5}]
ystNAR = y[t] /. NSolve[{0 == β / (1 + (y[t] / k)^ n) - α y[t] && y[t] > 0} /. params , y[t]]
params = Insert[params , βsimple → ystNAR ⟦1⟧ , - 1]
solS = DSolve[{y '[t] == βsimple - α y[t], y[0] == 0} /. params , y, {t, 0, 5}] ystSimple = y[t] /.
NSolve[{0 == βsimple - α y[t] && y[t] > 0} /. params , y[t]]
p1 = Plot[{(y[t] /. solNAR) / ystNAR , (y[t] /. solS) / ystSimple }, {t, 0, 5}, PlotRange → All,
PlotLegends → {"NAR", "Simple "}, AxesLabel → {"time", "y/yst"}]
tSimp = t /. NSolve[{y[t] == ystSimple / 2 } /. solS , t] tNAR = t /. NSolve[{y[t] == ystNAR / 2} /.
solNAR , t]
```



b) We will have to increase n (cooperativity) in hill function to reach the limit of step function of inhibition.



With the increase in n, we observe even faster initial growth and lower delay in auto repression to stop production at the desired steady-state.



c) Given, $\beta_{NAR} = 2{,}20{,}200$ μM/h and $n \to \infty$. Now since $y[t] \ll \beta_{NAR}/\alpha$, we can approximate $T_{1/2}^{NAR} = K/2\beta_{NAR}$. Response time of simple regulatory circuit is $T_{1/2}^{simple} = ln2/\alpha$

$$\Rightarrow T_{\frac{1}{2}}^{NAR} = 0.3 \frac{\mu M}{2 * 2{,}20{,}200\ \mu M\ h^{-1}} = 6.81 * 10^{-7}\ h$$

And,

$$T_{1/2}^{simple} = \frac{ln2}{1\ h^{-1}} = 0.693\ h$$

d) At the step function limit i.e. $n \to \infty$ and $y[t] \ll \beta_{NAR}/\alpha$, we know $T_{1/2}^{NAR} = K/2\beta_{NAR}$ and $T_{1/2}^{simple} = ln2/\alpha$. This means $T_{1/2}^{NAR}$ is does no depends on α, while $T_{1/2}^{simple}$ is inversely proportional to α.

e) Using the above mentioned formulas, we calculate $T_{1/2}^{NAR}$ and $T_{1/2}^{simple}$.

α	$T_{1/2}^{NAR}$ (hr)	$T_{1/2}^{simple}$ (hr)
1	0.005	0.693
2	0.005	0.346
10	0.005	0.069
20	0.005	0.034

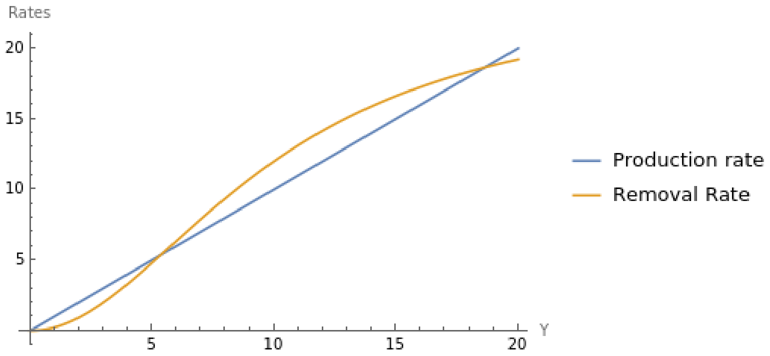
f) Summary statement:

- Response time of NAR circuit depends on half − saturation constant k and maximal production rate β**.**
- Response time of simple regulatory circuit depends on removal rate α.
- Steady state concentration of NAR circuit depends on half − saturation constant k.
- Steady state concentration of simple regulatory circuit depends on maximal production rate β_{simple} and removal rate α.

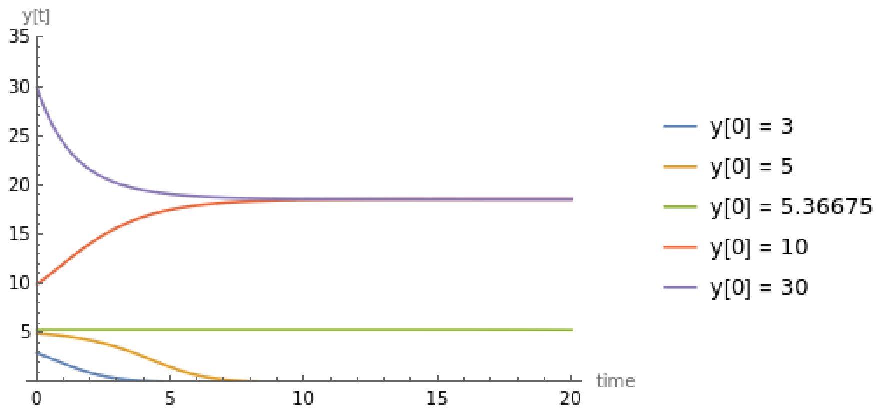
B)

a) Steady state concentration of y given $y[0] = 0\ \mu M$ is $0\ \mu M$.
Steady state concentration of y given $y[0] = 30\ \mu M$ is $18.6332\ \mu M$.

The given PAR circuit has three fixed points $(y = 0\ \mu M, y = 5.36675\ \mu M\ and\ y = 18.6332\ \mu M)$. The fixed point $y = 5.36675\ \mu M$ is unstable, thus steady state concentration of y is either $0\ \mu M$ or $18.6332\ \mu M$ which depends on initial conditions.



b) Critical concentration at which the systems switches from the OFF state to the ON state is $y = 5.37775\ \mu M$.



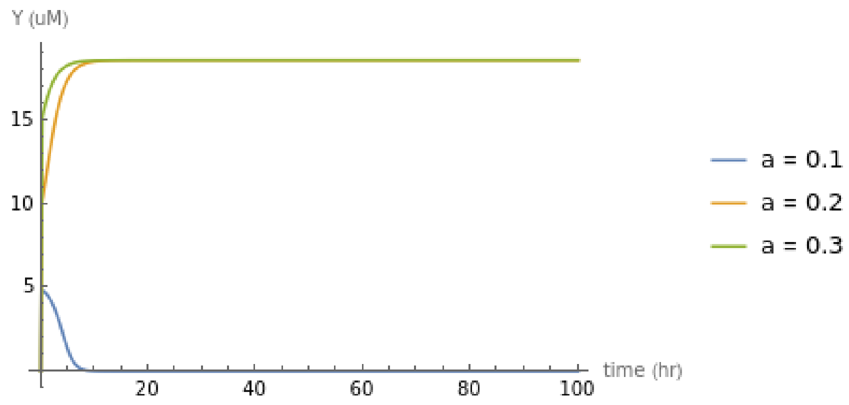
Code:

```
params = { β → 24, k → 10, α → 1, n → 2}
solPAR1 = NDSolve [{y '[t] == β ((y[t] / k)^ n) / (1 + (y[t] / k)^ n) - α y[t], y[0] == 3} /. params ,
y, {t, 0, 20}]
solPAR2 = NDSolve [{y '[t] == β ((y[t] / k)^ n) / (1 + (y[t] / k)^ n) - α y[t], y[0] == 5} /. params ,
y, {t, 0, 20}]
solPAR3 = NDSolve [{y '[t] == β ((y[t] / k)^ n) / (1 + (y[t] / k)^ n) - α y[t], y[0] == 10} /. params
, y, {t, 0, 20}]
solPAR4 = NDSolve [{y '[t] == β ((y[t] / k)^ n) / (1 + (y[t] / k)^ n) - α y[t], y[0] == 30} /. params
, y, {t, 0, 20}]
solPAR5 = NDSolve [ {y '[t] == β ((y[t] / k)^ n) / (1 + (y[t] / k)^ n) - α y[t], y[0] == 5.36675 } /.
params , y, {t, 0, 20}]
```

Plot[{y[t] /. solPAR1 , y[t] /. solPAR2 , y[t] /. solPAR5 , y[t] /. solPAR3 , y[t] /. solPAR4 }, {t, 0, 20}, PlotRange → {All, {0, 35}}, AxesLabel → {"time", "y[t]"} , PlotLegends → {"y[0] = 3" , "y[0] = 5" , "y[0] = 5.36675 " , "y[0] = 10" , "y[0] = 30"}]

c) System has three steady-states and only two of them are stable.
   - $y = 0\ \mu M, stable\ steady\ state.$
   - $y = 5.36675\ \mu M, unstable\ steady\ state.$
   - $y = 18.6332\ \mu M, stable\ steady\ state.$

d) The steady-state concentration of Y after the pulse of production is gone depends on the duration of the pulse. If the pulse duration ≤ 0.10 hr then the steady state concentration is 0 and if pulse duration ≥ 0.20 hr then steady state concentration of Y is 18.6332 μM.



Note: a is the duration of the pulse in hr.
Pulse duration to switch the steady state from 0 μM to 18.6332 μM has to be approximately greater than or equal to 0.2 hrs, which is evident from the above graph.

Code:

params = { β → 24, k → 10, α → 1, n → 2 }
p[t] = UnitStep [t] - UnitStep [t - r]

sol = solPAR = ParametricNDSolve [{y '[t] == 50 * p[t] + β ((y[t] / k)^ n) / (1 + (y[t] / k)^ n) - α y[t], y[0] == 0} /. params , y, {t, 0, 200} , {r}]
Plot[Evaluate [Table[y[a][t] /. sol, {a, 0.1, 0.3, .1}]], {t, 0, 100}, PlotRange → All , PlotRange → {All, {0, 50}} , AxesLabel → {"time (hr)" , "Y (uM)"} , PlotLegends → {"a = 0.1" , "a = 0.2" , "a = 0.3"}]

C)

$$\frac{dy}{dt} = \beta \frac{y}{k + y} - \alpha * y$$

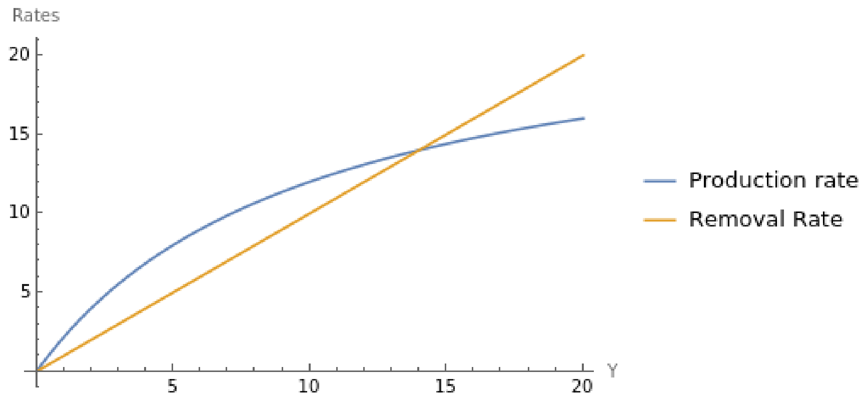
a) On fixed points $\frac{dy}{dt} = 0$.

$$\frac{dy}{dt} = \beta \frac{y}{k + y} - \alpha * y = 0$$

$$\Rightarrow y = 0 \, , y = \frac{\beta}{\alpha} - k$$

Systems has two fixed points $y = 0 \; and \; y = \frac{\beta}{\alpha} - k.$

b) System is monostable as fixed point $y = \frac{\beta}{\alpha} - k$ is stable while fixed point $y = 0$ is unstable. Fixed point $y = \frac{\beta}{\alpha} - k$ is stable, because shifting y to either side $(of \; \frac{\beta}{\alpha} - k)$ causes a return to the fixed point.



Code:

```
Plot[{24*y/(y+10) , y} , {y, 0 , 20} , PlotLegends->{"Production rate" , "Removal Rate"} ,
AxesLabel -> {"Y" , "Rates"}]
```