

1. Problem 1

$$-r_A = kC_A$$

$$C_A = C_{A0}(1 - X)$$

CSTR:

$$V = \frac{XF_{A0}}{-r_A}$$

$$V = \frac{XF_{A0}}{kC_{A0}(1 - X)}$$

$$\gamma = \frac{F_{A0}}{kC_{A0}}$$

$$V = \frac{\gamma X}{(1 - X)}$$

$$\frac{d}{dX}(V) = \frac{d}{dX} \left( \frac{\gamma X}{(1 - X)} \right)$$

$$\frac{dV}{dX} = \frac{\gamma}{(1 - X)^2}$$

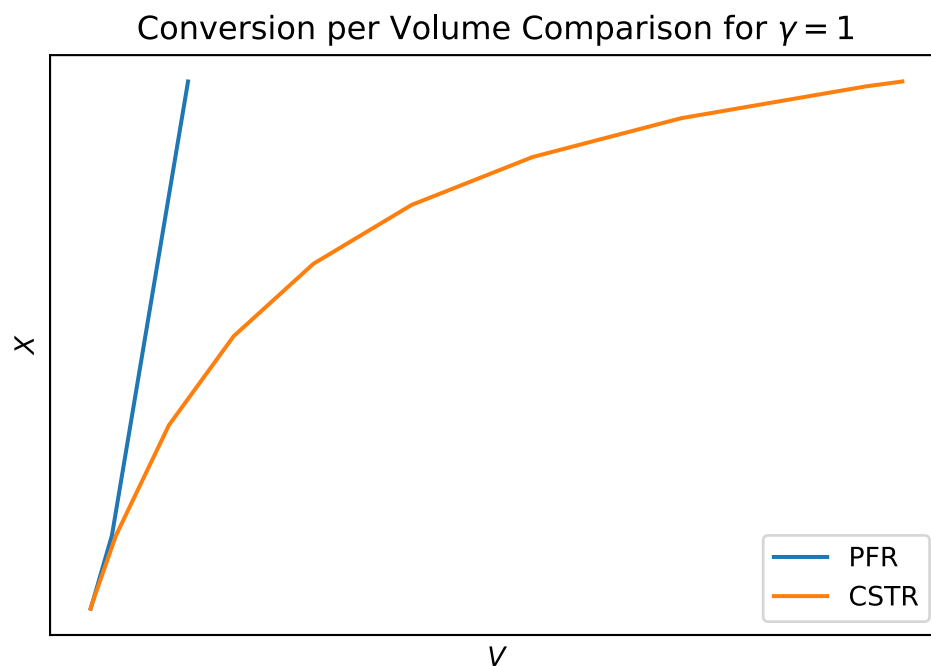
PFR:

$$F_{A0} \frac{dV}{dX} = -r_A$$

$$F_{A0} \frac{dV}{dX} = kC_{A0}(1 - X)$$

$$\frac{dV}{dX} = \frac{1 - X}{\gamma}$$

Compare conversion per volume between the CSTR and PFR:



For any given volume change, the PFR will have a much larger conversion change because it has a higher conversion per volume than the CSTR. Add more volume to the PFR because it will give more conversion.

## 2. Problem 2

(a)



$$F_{A0} \frac{dX}{dW} = -r'_A$$

$$-r'_A = k' C_A C_B$$

$$\epsilon = (1 - 1 - 1) \cdot \frac{1}{2} = -\frac{1}{2}$$

$$C_A = C_B = C_{A0} \left( \frac{1 - X}{1 - \frac{1}{2}X} \right) p$$

$$C_C = C_{A0} \left( \frac{X}{1 - \frac{1}{2}X} \right) p$$

$$\frac{dX}{dW} = \frac{k' C_{A0}^2 p^2}{F_{A0}} \left( \frac{1 - X}{1 - \frac{1}{2}X} \right)^2$$

$$C_{A0} = \frac{F_{A0}}{v_0}$$

$$\frac{dX}{dW} = \frac{k' F_{A0} p^2}{v_0^2} \left( \frac{1 - X}{1 - \frac{1}{2}X} \right)^2$$

$$\frac{dp}{dW} = -\frac{\alpha (1 - \frac{1}{2}X)}{2p}$$

Solve this system of ODEs:

$$\frac{dX}{dW} = \frac{k' F_{A0} p^2}{v_0^2} \left( \frac{1 - X}{1 - \frac{1}{2}X} \right)^2$$

$$\frac{dp}{dW} = -\frac{\alpha (1 - \frac{1}{2}X)}{2p}$$

$$k' = 0.004$$

$$F_{A0} = 2 \cdot 10^{-5}$$

$$v_0 = 2.83 \cdot 10^{-7}$$

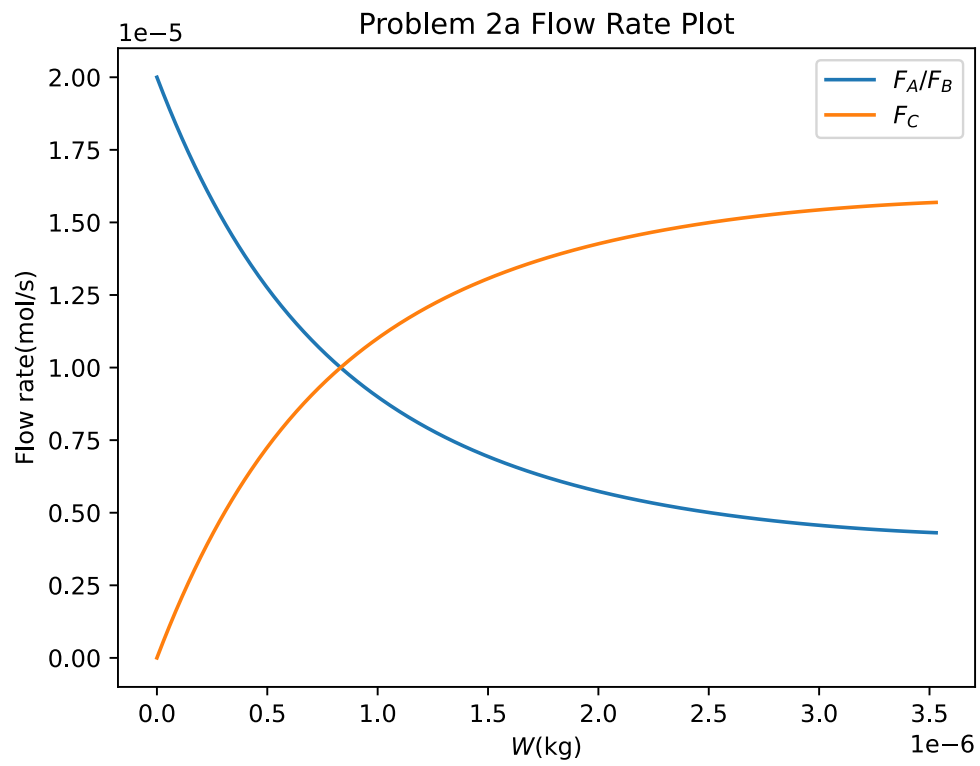
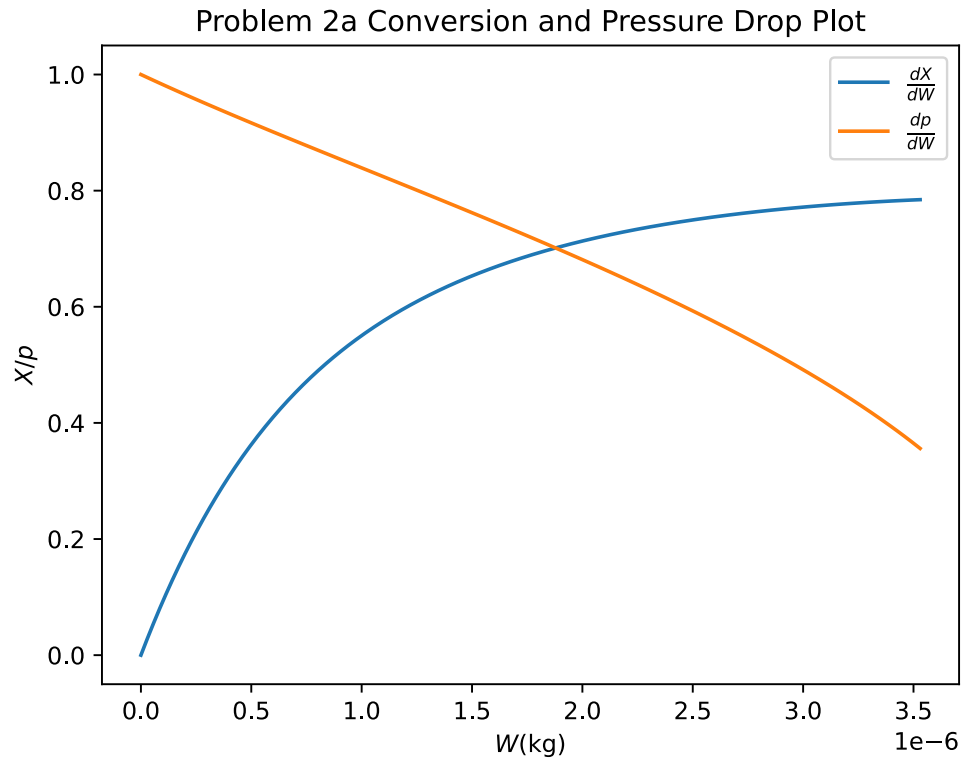
$$\alpha = 3.55 \cdot 10^5$$

For plotting flow rates over time:

$$F_A = F_B = F_{A0}(1 - X)$$

$$F_C = F_{A0}X$$

Output plots:



Code that creates the plots:

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

# ode system to solve dp/dW and dX/dW
def ode(t, y):
    X = y[0]
    p = y[1]
    # constants
    F_A0 = 2e-5
    v_0 = 2.83e-7
    alpha = 3.55e5
    k = 0.004

    f = np.zeros(len(y))

    f[0] = k * F_A0 * p**2 / v_0**2 * ((1 - X) / (1 - X/2)) **2
    f[1] = -alpha / 2 / p * (1 - X/2)

    return f

# arguments to pass to the ode solver
ode_kwargs = {
    'method': 'Radau',
    'atol': 1e-8,
    'rtol': 1e-8,
}

# solving the ode system
solution = solve_ivp(ode, [0, 3.53e-6], [0, 1], **ode_kwargs)

# plotting
plt.plot(solution.t, solution.y[0])
plt.plot(solution.t, solution.y[1])
plt.legend([r'$\frac{dX}{dW}$', r'$\frac{dp}{dW}$'])
plt.xlabel(r"$W$(kg)")
plt.ylabel(r"$X/p$")
plt.title("Problem 2a Conversion and Pressure Drop Plot")
plt.show()

# functions to calculate flow rates
F_A0 = 2e-5
F_A = lambda X: F_A0 * (1 - X)
F_C = lambda X: F_A0 * X
```

```
# plotting
plt.plot(solution.t, F_A(solution.y[0]))
plt.plot(solution.t, F_C(solution.y[0]))
plt.legend([r'$F_A/F_B$', r'$F_C$'])
plt.xlabel(r"$W$(kg)")
plt.ylabel(r"Flow rate(mol/s)")
plt.title("Problem 2a Flow Rate Plot")
plt.show()
```

- (b) The total conversion is 0.784.

Convert to kg/yr:

$$\begin{aligned}\dot{m}_C &= 0.784 \cdot 2 \cdot 10^{-5} \text{ mol/s} \cdot 99 \text{ g/mol} = 0.00155 \text{ g/s} \\ N &= \frac{10000 \text{ kg/yr} \cdot 1000 \text{ g/kg}}{0.00155 \text{ g/s} \cdot 3600 \text{ s/hr} \cdot 24 \text{ hr/day} \cdot 365 \text{ day/yr}} \\ N &= 204\end{aligned}$$

204 reactors are needed to achieve a production of 10,000 kg/yr.

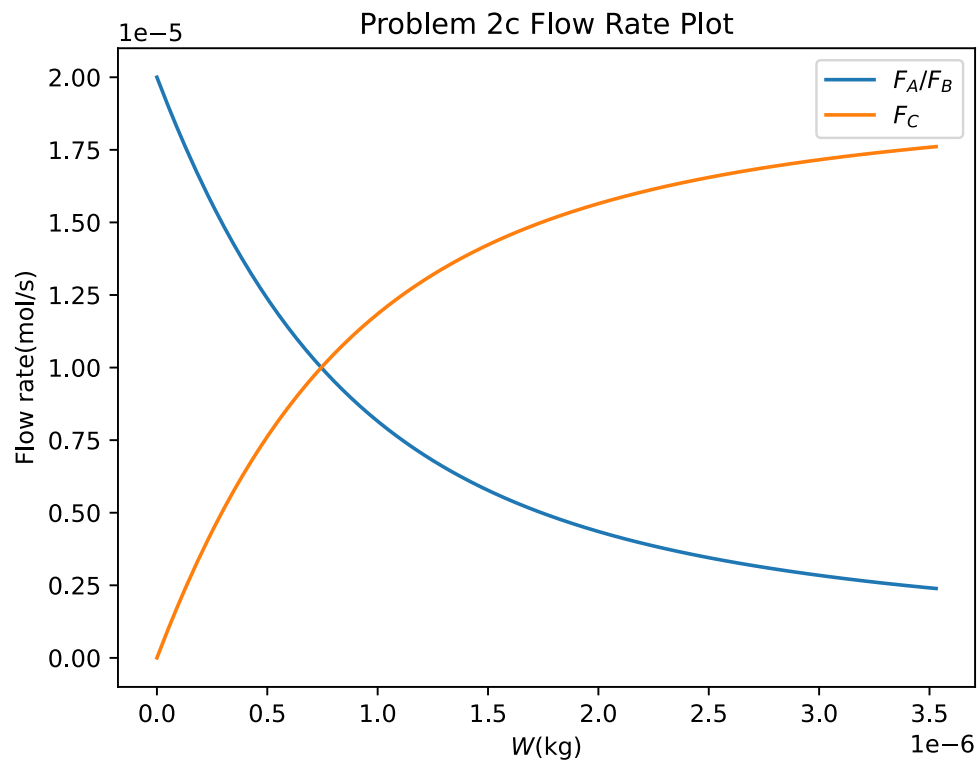
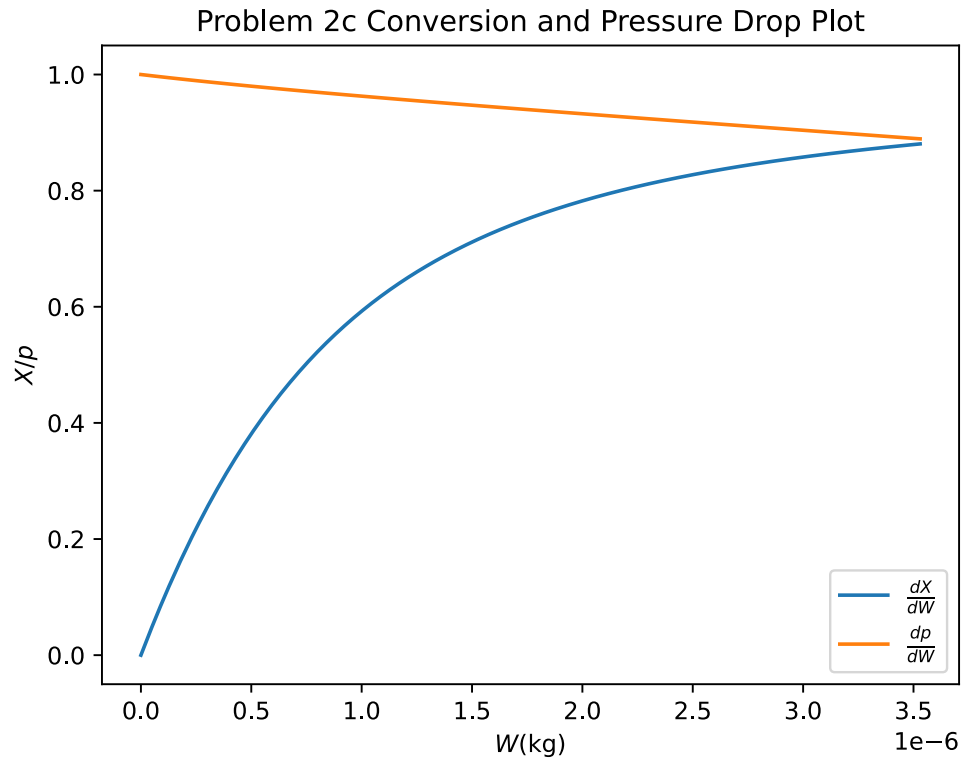
- (c)

$$\begin{aligned}\alpha &= \alpha_0 \frac{D_0^2}{D^2} \\ D &= 2D_0 \\ \alpha &= \alpha_0 \frac{D_0^2}{(2D_0)^2} \\ \alpha &= \frac{\alpha_0}{4}\end{aligned}$$

Solve the ODE system with the new alpha value.

$$\begin{aligned}\frac{dX}{dW} &= \frac{k' F_{A0} p^2}{v_0^2} \left( \frac{1-X}{1-\frac{1}{2}X} \right)^2 \\ \frac{dp}{dW} &= -\frac{\frac{\alpha_0}{4} (1-\frac{1}{2}X)}{2p}\end{aligned}$$

Output plots:



The conversion compared to part a is slightly higher, and the pressure drop is lower. Nothing unusual.

Code that creates the plots:

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp

def ode(t, y):
    X = y[0]
    p = y[1]

    F_A0 = 2e-5
    v_0 = 2.83e-7
    alpha = 3.55e5 / 4
    k = 0.004

    f = np.zeros(len(y))

    f[0] = k * F_A0 * p**2 / v_0**2 * ((1 - X) / (1 - X/2)) **2
    f[1] = -alpha / 2 / p * (1 - X/2)

    return f

ode_kwargs = {
    'method': 'Radau',
    'atol': 1e-8,
    'rtol': 1e-8,
}

solution = solve_ivp(ode, [0, 3.53e-6], [0, 1], **ode_kwargs)

plt.plot(solution.t, solution.y[0])
plt.plot(solution.t, solution.y[1])
plt.legend([r'$\frac{dX}{dW}$', r'$\frac{dp}{dW}$'])
plt.xlabel(r"$W$(kg)")
plt.ylabel(r"$X/p$")
plt.title("Problem 2c Conversion and Pressure Drop Plot")
plt.show()

# functions to calculate flow rates
F_A0 = 2e-5
F_A = lambda X: F_A0 * (1 - X)
F_C = lambda X: F_A0 * X

# plotting
plt.plot(solution.t, F_A(solution.y[0]))
plt.plot(solution.t, F_C(solution.y[0]))
plt.legend([r'$F_A/F_B$', r'$F_C$'])
plt.xlabel(r"$W$(kg)")

```



```
plt.ylabel(r"Flow rate(mol/s)")  
plt.title("Problem 2c Flow Rate Plot")  
plt.show()
```

### 3. Problem 3

(a) Suppose first order reaction:

$$-r_A = kC_A$$

Isothermal and isobaric

$$C_A = C_{A0} \left( \frac{1 - X}{1 + \epsilon X} \right)$$

$$\frac{dX}{dV} = -r_A$$

$$\frac{dX}{dV} = kC_{A0} \left( \frac{1 - X}{1 + \epsilon X} \right)$$

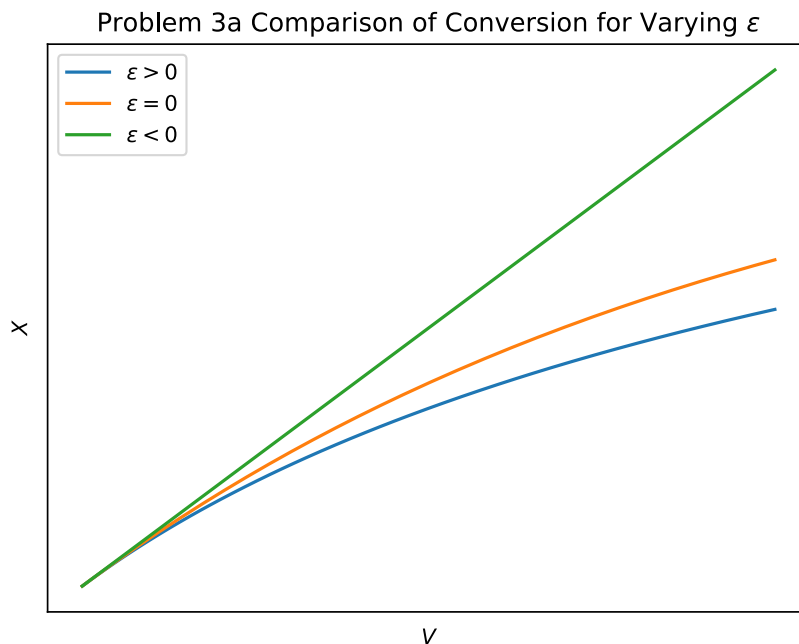
$$\gamma = kC_{A0}$$

General form:

$$\frac{dX}{dV} = \gamma \left( \frac{1 - X}{1 + \epsilon X} \right)$$

When  $\epsilon$  is less than zero, the numerator of the differential above becomes less than zero which increases  $\frac{dX}{dV}$ . The rate of change of conversion increases which increases conversion overall. When  $\epsilon$  is greater than zero, the numerator is greater than one which decreases  $\frac{dX}{dV}$ . The decrease causes the rate of change of conversion to slow which decreases conversion. The way  $\epsilon$  changes  $\frac{dX}{dV}$  changes the overall conversion. Negative  $\epsilon$  increases the rate which increases conversion.

The increase in conversion can be seen in the plot below which shows conversion with different  $\epsilon$  values.



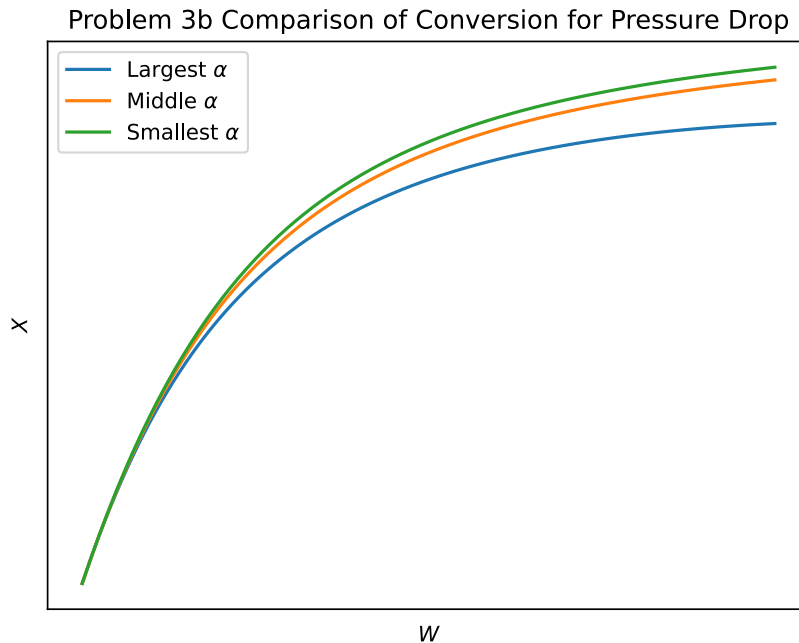
(b) Reorder the above differential for catalyst weight and pressure drop:

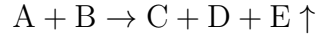
$$\frac{dX}{dW} = \gamma \left( \frac{1-X}{1+\epsilon X} \right) p$$

$$\frac{dp}{dW} = -\frac{\alpha(1+\epsilon X)}{p}$$

Smaller values of  $\alpha$  in  $\frac{dp}{dW}$  decrease the rate at which pressure drops in the reactor. From the differential for  $\frac{dX}{dW}$ , the pressure ratio,  $p$ , is proportional to the rate at which conversion increases. If the pressure ratio is decreasing at a slower rate, then the rate at which  $\frac{dX}{dW}$  decreases will be lower. If lower values for  $\alpha$  lower  $\frac{dp}{dW}$ , then lower values of  $\alpha$  should result in higher conversions. Essentially, a lower  $p$  results in the conversion increasing at a slower rate. The lower  $p$  is a result of a larger pressure drop.

The plot below demonstrates this relationship. Notice that the reactor with the lowest  $\alpha$  has the highest conversion. A lower  $\alpha$  corresponds to a lower pressure drop.





4. Problem 4

$$r = kC_A C_B$$

$$C_A = \frac{N_A}{V}$$

$$r = k \frac{N_A N_B}{V^2}$$

Design equation:

$$\frac{dN_A}{dt} = r_A$$

$$-r_A = -r_B = r_C = r_D = r_E = r$$

$$\frac{dN_A}{dt} = -k \frac{N_A N_B}{V^2}$$

$$\frac{dN_B}{dt} = F_{B0} - k \frac{N_A N_B}{V^2}$$

$$\frac{dN_C}{dt} = k \frac{N_A N_B}{V^2}$$

$$\frac{dN_D}{dt} = k \frac{N_A N_B}{V^2}$$

$$\frac{dV}{dt} = v_0 - v_{CO_2}$$

$$v_{CO_2} = \frac{r_E V M_{CO_2}}{\rho_{CO_2}}$$

$$\frac{dV}{dt} = v_0 - \frac{k \frac{N_A N_B}{V^2} V M_{CO_2}}{\rho_{CO_2}}$$

$$\frac{dV}{dt} = v_0 - \frac{k N_A N_B M_{CO_2}}{V \rho_{CO_2}}$$

$$v_0 = \frac{F_{B0}}{C_{B0}}$$

$$\frac{dV}{dt} = \frac{F_{B0}}{C_{B0}} - \frac{k N_A N_B M_{CO_2}}{V \rho_{CO_2}}$$

Solve the system of ODEs:

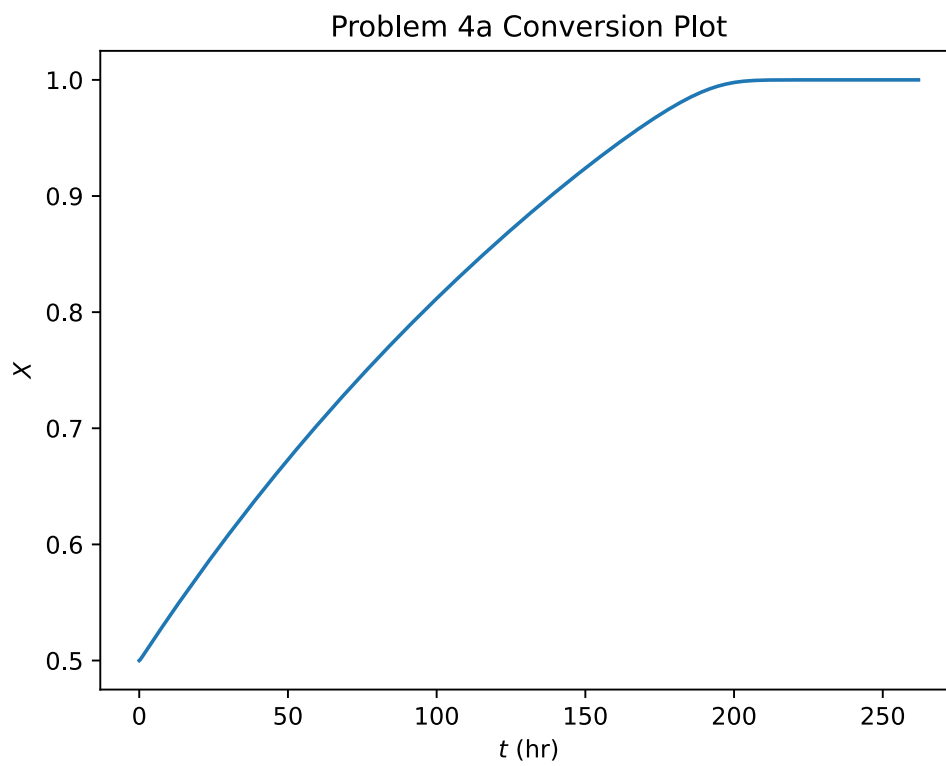
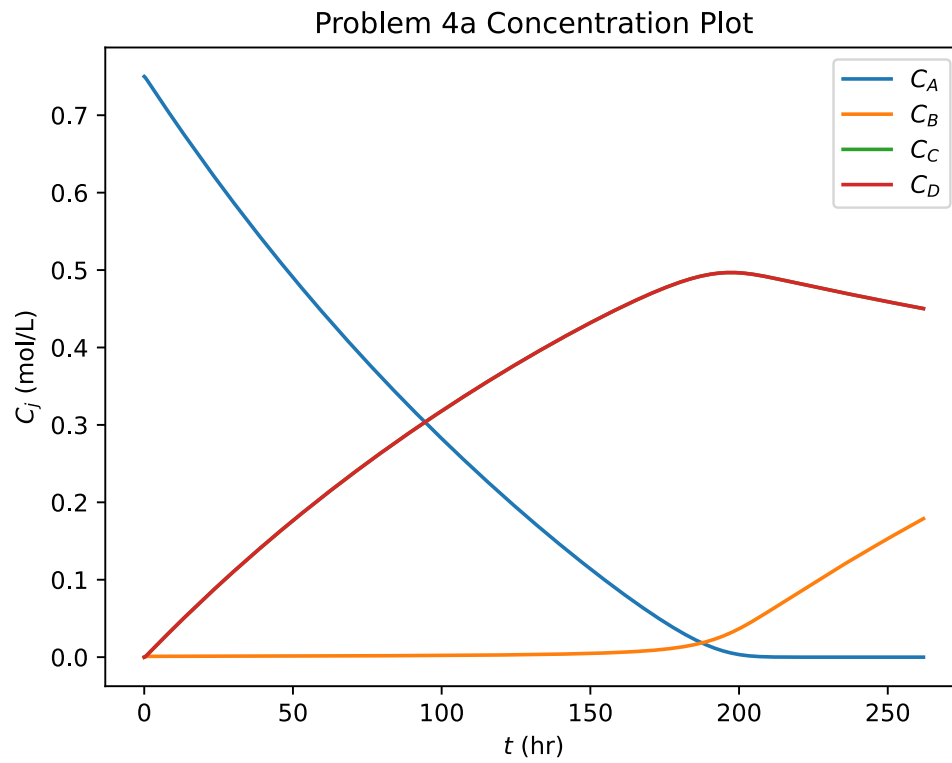
$$\begin{aligned}
\frac{dN_A}{dt} &= -k \frac{N_A N_B}{V^2} \\
\frac{dN_B}{dt} &= F_{B0} - k \frac{N_A N_B}{V^2} \\
\frac{dN_C}{dt} &= k \frac{N_A N_B}{V^2} \\
\frac{dN_D}{dt} &= k \frac{N_A N_B}{V^2} \\
\frac{dV}{dt} &= \frac{F_{B0}}{C_{B0}} - \frac{k N_A N_B M_{CO_2}}{V \rho_{CO_2}} \\
C_{B0} &= 1.5 \\
F_{B0} &= 60 \cdot 0.1 = 6 \\
k &= 5.1 \\
M_E &= 44 \\
\rho_E &= 1000 \\
v_0 &= \frac{F_{B0}}{C_{B0}} = \frac{6}{1.5} = 4 \\
N_{A0} &= 0.75 \cdot 1500 = 1125
\end{aligned}$$

Concentrations and conversion are calculated from solution arrays.

$$\begin{aligned}
C_j(t) &= \frac{N_j(t)}{V(t)} \\
C_A &= C_{A0}(1 - X) \\
X &= 1 - \frac{C_A}{C_{A0}} \\
X(t) &= 1 - \frac{\frac{C_A(t)}{V(t)}}{\frac{C_{A0}(t)}{V(t)}} \\
X(t) &= 1 - \frac{C_A(t)}{C_{A0}(t)}
\end{aligned}$$

Output plots:

$$C_C = C_D$$



Code that creates the plots:

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp

def ode(t, y):
    N_A = y[0]
    N_B = y[1]
    V = y[4]

    f = np.zeros(len(y))

    C_B0 = 1.5
    F_B0 = 6
    k = 5.1
    M_E = 44.0095
    rho_E = 1000
    v_0 = F_B0 / C_B0

    r = k * N_A * N_B / V**2

    f[0] = -r * V
    f[1] = F_B0 - r * V
    f[2] = r * V
    f[3] = r * V
    f[4] = v_0 - r * V * M_E / rho_E

    return f

ode_kwargs = {
    "method": 'Radau',
    'rtol': 1e-8,
    'atol': 1e-8,
}

N_A0 = 0.75*1500

initial_cond = [N_A0, 0, 0, 0, 1500]

solution = solve_ivp(ode, [0, 262], initial_cond, **ode_kwargs)

for i in range(4):
    plt.plot(solution.t, solution.y[i]/solution.y[4])

plt.legend([r'$C_A$', r'$C_B$', r'$C_C$', r'$C_D$'])
plt.xlabel(r"$t$ (hr)")
plt.ylabel(r"$C_j$ (mol/L)")

```

```
plt.title('Problem 4a Concentration Plot')
print(solution.y[4][-1])
plt.show()

plt.plot(solution.t, 1 - solution.y[0]/solution.y[4]/1.5)
plt.xlabel(r"$t$ (hr)")
plt.ylabel(r"$X$")
plt.title('Problem 4a Conversion Plot')
plt.show()
```