1. Problem 1

$$\mathrm{A} \rightarrow \mathrm{B} + \mathrm{C}$$

(a)

$$\delta = 1$$

No pressure drop

$$C_A = C_{A0}\left(\frac{1-X}{1+X}\right)\frac{T_0}{T}$$
$$-r'_A = k(T)C_A$$

Design equation

$$\frac{dX}{dW} = \frac{-r'_A}{v_0 C_{A0}}$$
$$\frac{dX}{dW} = \frac{k(T)C_{A0}\left(\frac{1-X}{1+X}\right)\frac{T_0}{T}}{v_0 C_{A0}}$$
$$\frac{dX}{dW} = \frac{k(T)}{v_0}\left(\frac{1-X}{1+X}\right)\frac{T_0}{T}$$
$$k(T) = 0.133\exp\left[\frac{E}{R}\left(\frac{1}{450}-\frac{1}{T}\right)\right]$$
$$\frac{dX}{dW} = \frac{0.133\exp\left[\frac{E}{R}\left(\frac{1}{450}-\frac{1}{T}\right)\right]}{v_0}\left(\frac{1-X}{1+X}\right)\frac{T_0}{T}$$

Solve the differential equation with the following parameters

$$E = 31400$$
$$T_0 = 450$$
$$R = 8.314$$
$$v_0 = 20$$

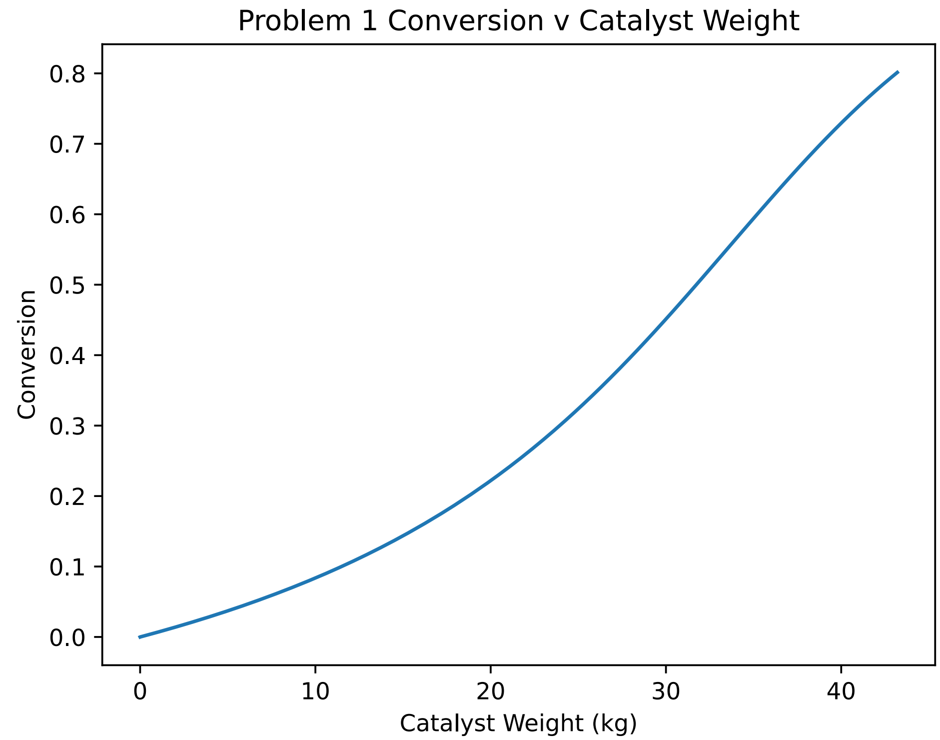
Temperature dependence

$$T = \frac{X\left[-\Delta H^\circ_{\mathrm{Rx}}(T_R)\right] + \sum \Theta_i C_{P_i} T_0 + X\Delta C_P T_R}{\left[\sum \Theta_i C_{P_i} + X\Delta C_P\right]}$$
$$\Delta C_P = 15 + 25 - 40 = 0$$
$$\sum \Theta_i C_{P_i} = 0\cdot 15 + 0\cdot 25 + 1\cdot 40 = 40$$
$$\Delta H^\circ_{\mathrm{Rx}}(T_R) = -40000 - 50000 + 70000 = -20000$$
$$T = \frac{20000X + 40\cdot 450}{40}$$

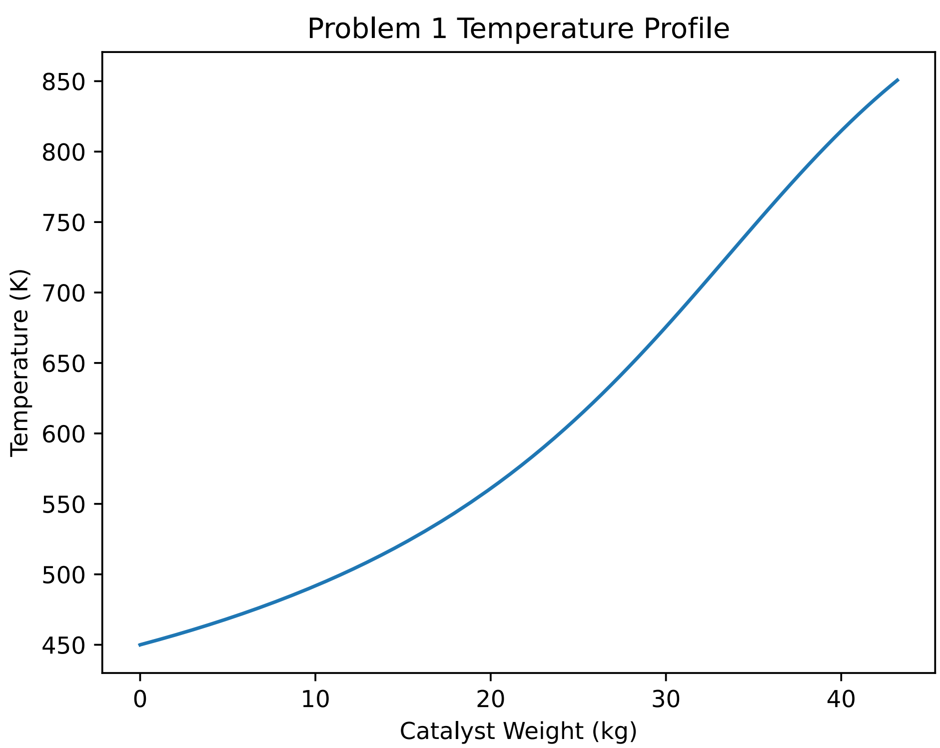
Use $T(X)$ to compue T in the differential equation

$$T = 450 + 500X$$

Conversion plot:



Temperature plot:



(b)

Heat analysis

$$\dot{Q} = F_{A0}\left(\sum \Theta_i C_{P_i}\left(T - T_{i0}\right) + X\left[H^{\circ}_{\text{Rx}}(T_R) + \Delta C_P(T - T_R)\right]\right)$$

$$\dot{Q} = F_{A0} X \Delta H^{\circ}_{\text{Rx}}(T_R)$$

$$F_{A0} = \frac{P_0 v_0}{RT_0}$$

$$P = 10$$

$$v_0 = 20$$

$$R = 0.08206$$

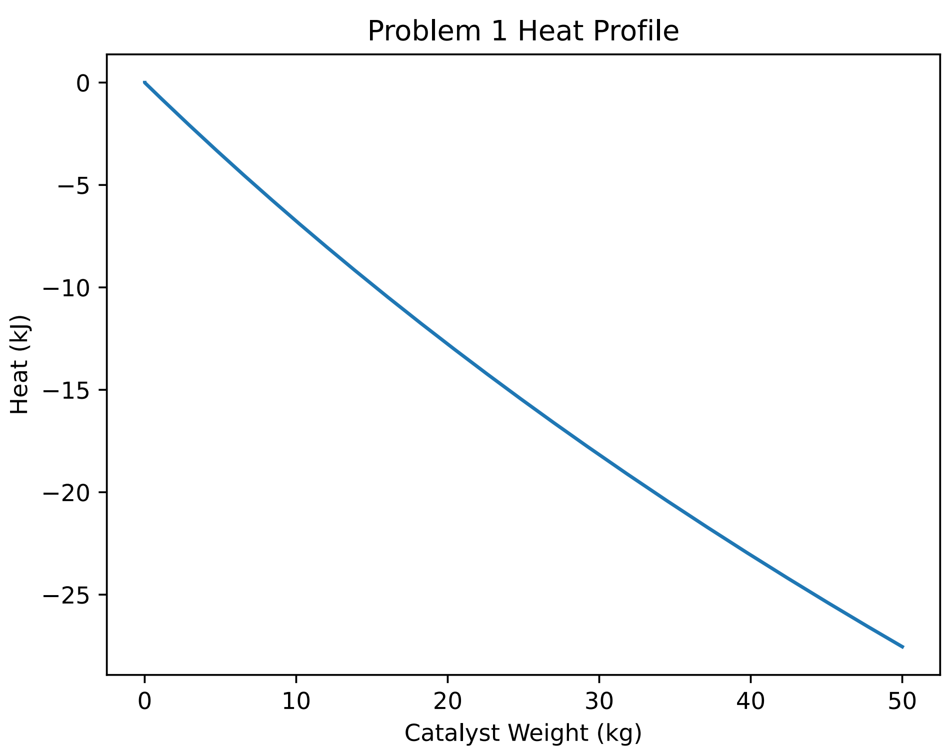
$$T_0 = 450$$

$$C_{P_A} = 40$$

$$\Delta H^{\circ}_{\text{Rx}}(T_R) = -20$$

$$\dot{Q} = -\frac{10 \cdot 20}{0.08206 \cdot 450} \cdot 20X$$

Recalculate conversion for isothermal operation.



Problem 1 code:

```
import numpy as np
import matplotlib.pyplot as plt

from scipy.integrate import solve_ivp

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

def ode_non_isothermal(t, y):
    f = y*0

    X = y[0]

    R = 8.314
    T_0 = 450
    F_A0 = 20
    E = 31.4e3

    T = T_0 + 500 * X
    k = 0.133 * np.exp(E / R * (1/450 - 1/T))

    f[0] = k * (1 - X) / (1 + X) * T_0 / T / F_A0

    return f

sol_non_isothermal = solve_ivp(ode_non_isothermal, [0, 43.2], [0], **ode_kwargs)

plt.plot(sol_non_isothermal.t, sol_non_isothermal.y[0])
plt.ylabel("Conversion")
plt.xlabel("Catalyst Weight (kg)")
plt.title("Problem 1 Conversion v Catalyst Weight")
plt.show()

plt.plot(sol_non_isothermal.t, 450 + 500 * sol_non_isothermal.y[0])
plt.ylabel("Temperature (K)")
plt.xlabel("Catalyst Weight (kg)")
plt.title("Problem 1 Temperature Profile")
plt.show()


def ode_non_isothermal(t, y):
    f = y*0
```

```
    X = y[0]

    R = 8.314
    T_0 = 450
    F_A0 = 20
    E = 31.4e3

    T = T_0
    k = 0.133 * np.exp(E / R * (1/450 - 1/T))

    f[0] = k * (1 - X) / (1 + X) * T_0 / T / F_A0

    return f

sol_isothermal = solve_ivp(ode_non_isothermal, [0, 50], [0], **ode_kwargs)

def Q(X):
    F_A0 = 10 * 20 / 0.08206 / 450
    return F_A0 * (-20 * X)

X = sol_isothermal.y[0]
plt.plot(sol_isothermal.t, Q(X))
plt.ylabel("Heat (kJ)")
plt.xlabel("Catalyst Weight (kg)")
plt.title("Problem 1 Heat Profile")
plt.show()
print(f"Isothermal conversion: {X[-1]}")
```

2. Problem 2

   The maximum possible conversion in this reaction is dictated by the equilibrium relationship. As temperature increases, the maximum possible conversion decreases. In order to maximize the amount of the reactants that are converted, the effluent of the reactor is cooled. The cooling steps bypass the physical limitations of euqilibrium conversion by maintaining lower temperatures such that total conversion increases over multiple reactor stages.

3. Problem 3

Equilibrium relationship

$$K_e = \frac{C_C C_D}{C_A C_B}$$
$$C_A = 1 - X$$
$$C_C = X$$
$$K_e = \frac{X^2}{(1-X)^2}$$
$$X = \frac{\sqrt{K_e}}{1+\sqrt{K_e}}$$

Equilibrium constant as a function of temperature

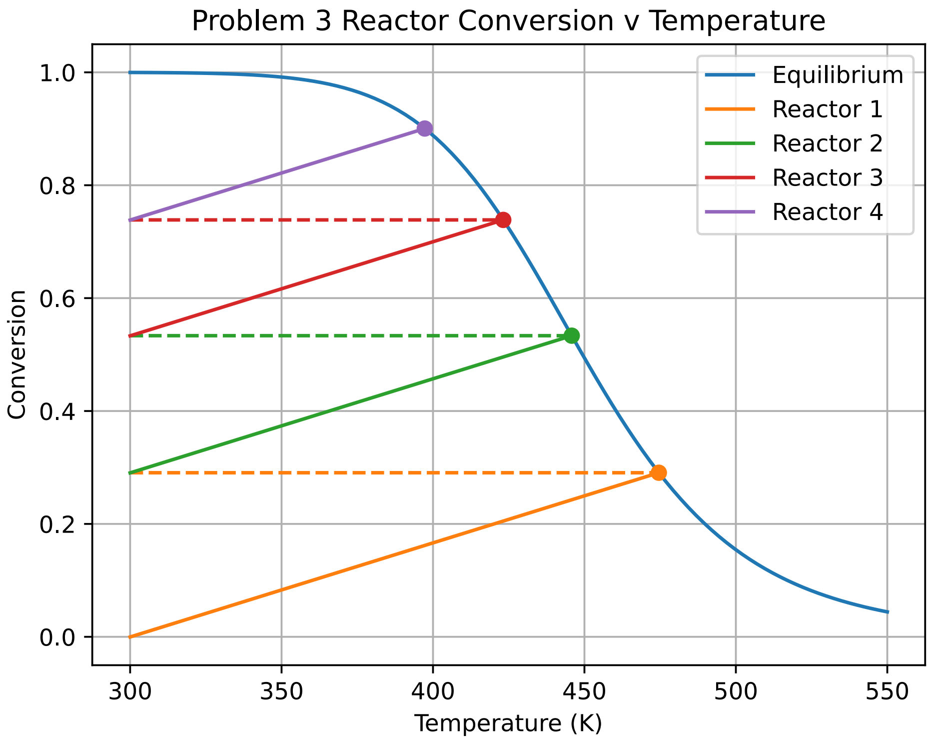
$$K_e(T) = K_e \exp\left[\frac{\Delta H^\circ_{Rx}}{RT}\left(\frac{1}{323.15} - \frac{1}{T}\right)\right]$$
$$K_e = 500,000$$
$$\Delta H^\circ_{Rx} = -30,000$$

Temperature as a function of conversion in the reactor

$$T = \frac{X\left[-\Delta H^\circ_{\mathrm{Rx}}(T_R)\right] + \sum \Theta_i C_{P_i} T_0 + X \Delta C_P T_R}{\left[\sum \Theta_i C_{P_i} + X\Delta C_P\right]}$$
$$\Delta C_P = 0$$
$$\sum \Theta_i C_{P_i} = 50$$
$$T = \frac{30000X + 50 \cdot 300.15}{50}$$
$$T = 600X + 300.15$$
$$X(T) = \frac{T - 300.15}{600}$$

Find intersections between conversion as a function of temperature in the reactor and the equilibrium relationship

Plotting the four reactors and three coolers.



The maximum conversion with the four reactors is $\boxed{X = 0.899}$.

Problem 3 code:

```
import numpy as np
import matplotlib.pyplot as plt

from scipy.optimize import fsolve

plt.grid(visible=True, which='both', axis='both')

# Definitions
X = lambda K: np.sqrt(K) / (1 + np.sqrt(K))
K = lambda T: 500000 * np.exp(-30000 * 4.184/8.314 * (1/(50+273.15) - 1/T))
T = lambda X: 600 * X + 300.15
X_T = lambda T: (T - 300.15) / 600

# Equilibrium plot
T_ran = np.linspace(300, 550, 100)
plt.plot(T_ran, X(K(T_ran)), label="Equilibrium")

# Reactor 1
def sys1(x):
    f = x*0

    f[0] = T(x[1]/0.999) - x[0]
    f[1] = X(K(x[0])) - x[1]/0.999

    return f

point_1 = fsolve(sys1, [300, 0.3])

T_r1 = np.linspace(300, point_1[0], 100)

plt.plot(T_r1, X_T(T_r1), color="tab:orange", label="Reactor 1")
plt.plot(point_1[0], point_1[1], 'o', color="tab:orange")
plt.plot(T_r1, np.ones(len(T_r1))*point_1[1], '--', color="tab:orange")

# Reactor 2
def sys2(x):
    f = x*0

    f[0] = T(x[1]/0.999 - point_1[1]) - x[0]
    f[1] = X(K(x[0])) - x[1]/0.999

    return f

point_2 = fsolve(sys2, [point_1[0], 0.5])
```

```python
T_r2 = np.linspace(300, point_2[0], 100)

plt.plot(T_r2, X_T(T_r2)+point_1[1], color="tab:green", label="Reactor 2")
plt.plot(point_2[0], point_2[1], 'o', color="tab:green")
plt.plot(T_r2, np.ones(len(T_r2))*point_2[1], '--', color="tab:green")

# Reactor 3
def sys3(x):
    f = x*0

    f[0] = T(x[1]/0.999 - point_2[1]) - x[0]
    f[1] = X(K(x[0])) - x[1]/0.999

    return f

point_3 = fsolve(sys3, [point_2[0], 0.5])

T_r3 = np.linspace(300, point_3[0], 100)

plt.plot(T_r3, X_T(T_r3)+point_2[1], color="tab:red", label="Reactor 3")
plt.plot(point_3[0], point_3[1], 'o', color="tab:red")
plt.plot(T_r3, np.ones(len(T_r3))*point_3[1], '--', color="tab:red")

# Reactor 4
def sys4(x):
    f = x*0

    f[0] = T(x[1]/0.999 - point_3[1]) - x[0]
    f[1] = X(K(x[0])) - x[1]/0.999

    return f

point_4 = fsolve(sys4, [point_3[0], 0.5])

T_r4 = np.linspace(300, point_4[0], 100)

plt.plot(T_r4, X_T(T_r4)+point_3[1], color="tab:purple", label="Reactor 4")
plt.plot(point_4[0], point_4[1], 'o', color="tab:purple")

plt.title("Problem 3 Reactor Conversion v Temperature")
plt.xlabel("Temperature (K)")
plt.ylabel("Conversion")
plt.legend()
plt.show()

print(f"Maximum conversion: {point_4[1]}")
```

4. Problem 1

(a)

Reaction constant relations:

$$\begin{aligned}
\frac{E}{R} &= 20000 \\
\Delta H^{\circ}_{Rx} &= -80000 \cdot 4.184 \\
k(T) &= 1 \cdot \exp\left[20000 \cdot \left(\frac{1}{400} - \frac{1}{T}\right)\right] \\
K_c(T) &= 100 \cdot \exp\left[\frac{-80000 \cdot 4.184}{8.314}\left(\frac{1}{400} - \frac{1}{T}\right)\right]
\end{aligned}$$

Raction rate law

$$\begin{aligned}
-r_A &= kC_A - \frac{k}{K_c} C_B \\
V &= \frac{XC_{A0}v_0}{-r_A} \\
C_A &= C_{A0}(1 - X) \\
C_B &= C_{A0}X \\
\tau &= \frac{Xv_0}{k(1-X) - \frac{k}{K_c}X} \\
X &= \frac{\tau k}{1 + \tau k\left(1 + K_c^{-1}\right)} \\
\tau &= 10
\end{aligned}$$

Conversion relationship with temperature

$$X(T) = \frac{\tau \exp\left[20000 \cdot \left(\frac{1}{400} - \frac{1}{T}\right)\right]}{1 + \tau \exp\left[20000 \cdot \left(\frac{1}{400} - \frac{1}{T}\right)\right]\left(1 + \left(100 \cdot \exp\left[\frac{-80000 \cdot 4.184}{8.314}\left(\frac{1}{400} - \frac{1}{T}\right)\right]\right)^{-1}\right)}$$

G and R

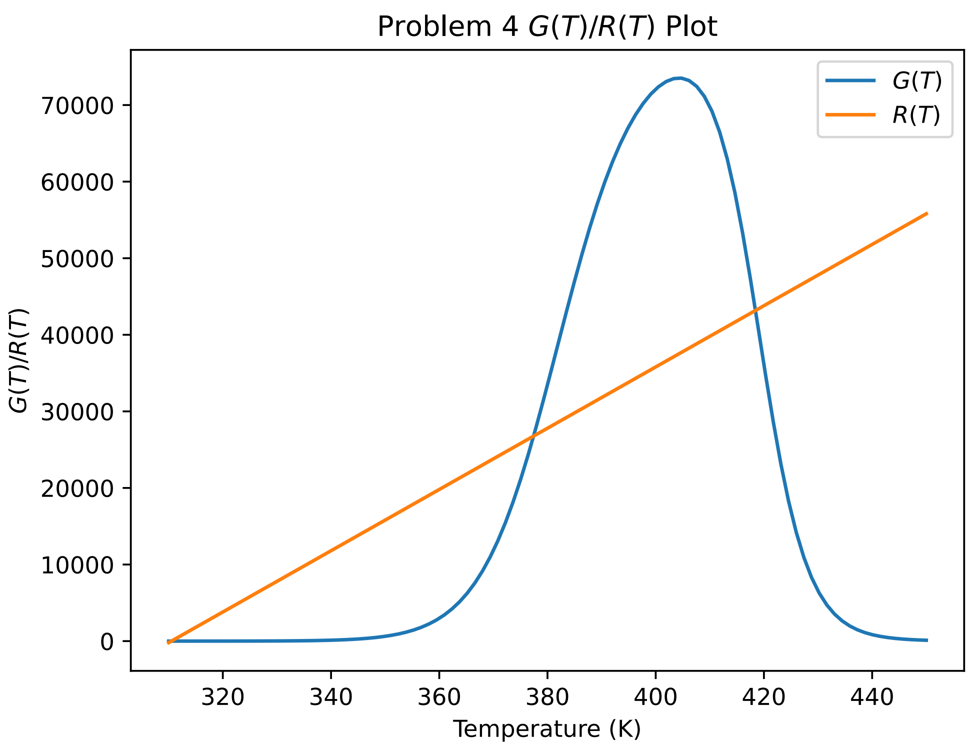
$$\begin{aligned}
G(T) &= 80000X(T) \\
R(T) &= C_{P0}(1 + \kappa)(T - T_c) \\
\kappa &= \frac{\mathrm{UA}}{C_{P0}F_{A0}} = \frac{3600}{40 \cdot 10} = 9 \\
T_c &= \frac{\kappa T_a + T_0}{\kappa + 1} = \frac{9 \cdot 310.15 + 310.15}{1 + 9} = 310.15 \\
R(T) &= 400(T - 310.15)
\end{aligned}$$

Plot R(T) and G(T)

Lower steady-state: 310.5 K

Middle steady-state: 377.4 k

Upper steady-state: 418.4 K



(b)

Find $X(418.4K)$. Upper steady state conversion $\boxed{X = 0.54}$.

Problem 4 code:

```
import numpy as np
import matplotlib.pyplot as plt

from scipy.optimize import fsolve

k = lambda T: np.exp(20000 * (1/400 - 1/T))
H = -80000 * 4.184
K_c = lambda T: 100 * np.exp(H/8.314 * (1/400 - 1/T))
tau = 10
X = lambda T: tau * k(T) / (1 + tau * k(T) * (1 + K_c(T)**-1))

G = lambda T: 80000 * X(T)
R = lambda T: 400 * (T - 310.5)

T_ran = np.linspace(310, 450, 100)

plt.plot(T_ran, G(T_ran), label=r"$G(T)$")
plt.plot(T_ran, R(T_ran), label=r"$R(T)$")
plt.xlabel("Temperature (K)")
plt.ylabel(r"$G(T)/R(T)$")
plt.title(r"Problem 4 $G(T)/R(T)$ Plot")
plt.legend()
plt.show()

point_1 = fsolve(lambda T: G(T) - R(T), [310])
point_2 = fsolve(lambda T: G(T) - R(T), [370])
point_3 = fsolve(lambda T: G(T) - R(T), [420])

print(f"Lower steady-state: {point_1[0]}\nMiddle steady-state: {point_2[0]}\nUpper
↪  steady-state: {point_3[0]}\n")
print(f"Upper steady state conversion: {X(point_3[0])}")
```