1. Problem 1

   (a)

$$k = Ae^{\frac{-E_a}{RT}}$$

Becomes linear equation:

$$\ln k = \ln A - \frac{E_a}{RT}$$
$$-r_A = kC_A^2 C_B$$
$$k = \frac{-r_A}{C_A^2 C_B}$$
$$\text{Fit } \ln k \text{ v } \frac{1}{T}$$

Code for fitting parameters to data:

```
import numpy as np
from scipy.optimize import curve_fit, fsolve
from sklearn.metrics import r2_score
# data
r_A = np.array([0.002, 0.046, 0.73, 8.33])
T_K = np.array([300, 320, 340, 360])
# reaction concentration
C_A = 2
C_B = 1.5
# convert r_A to k; elementary reaction
# 2A + B -> C
k = r_A / C_A**2 / C_B
# linearize data
ln_k = np.log(k)
inverse_T = 1 / T_K
# function to fit data to
def obj_func(T, E, A):
  return -E / 8.314 * T + np.log(A)
# fit data
fit_covs, _ = curve_fit(obj_func, inverse_T, ln_k)
# check fit against data
print("r^2:" ,r2_score(ln_k, obj_func(inverse_T, *fit_covs)))
# Output results
print(f"Activation energy: {round(fit_covs[0]/1000, 3)} kJ")
print(f"Frequency factor: {round(fit_covs[1], 3)}")
# Aggie Honor Code: An Aggie does not lie, cheat, or steal or tolerate
# those who do.
# I certify that this work is my own and not the work of another.
#
# Name: Mark Levchenko
# Assignment #: HW 3
# Question #: 1
```

Output:

Activation energy: $\boxed{124.77 \text{ kJ}}$

(b) Code also outputs the frequency factor:

Frequency factor: $\boxed{1.78 \cdot 10^{18}}$

2. Problem 2

   (a) PFR:

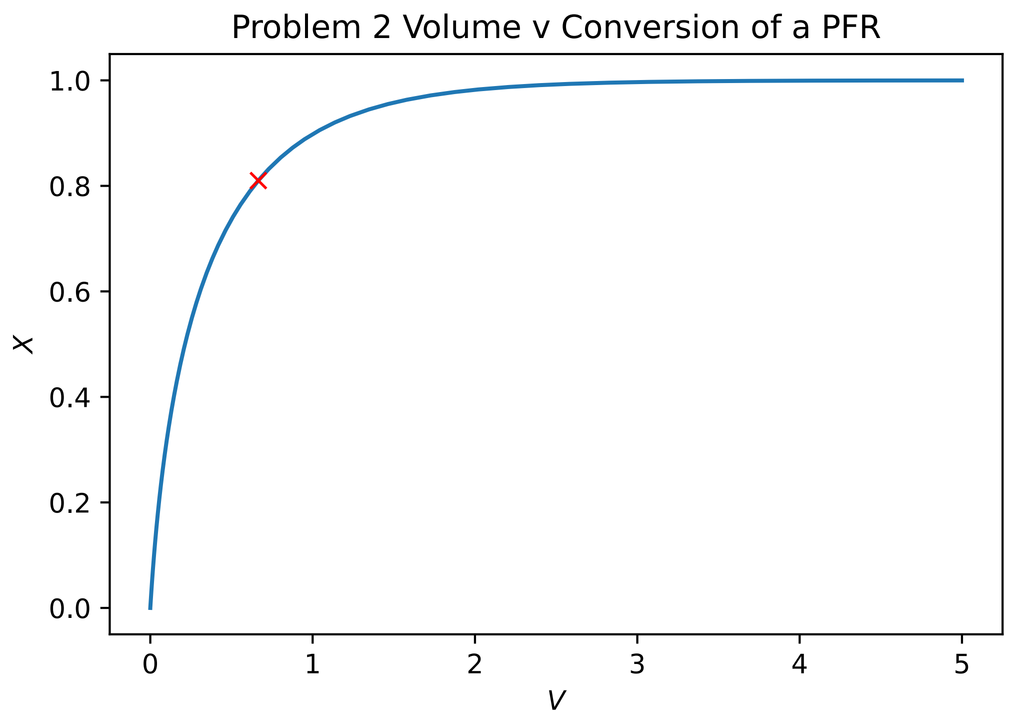
$$F_{A0}\frac{dX}{dV} = -r_A$$
$$-r_A = kC_A$$
$$C_A = C_{A0}\left(\frac{1-X}{1+2X}\right)$$
$$k = k_0 \exp\left[\frac{E_a}{R}\left(\frac{1}{T_0}-\frac{1}{T}\right)\right]$$
$$\frac{dX}{dV} = k_0 \exp\left[\frac{E_a}{R}\left(\frac{1}{T_0}-\frac{1}{T}\right)\right]\frac{P}{RT}\left(\frac{1-X}{1+2X}\right)/F_{A0}$$
$$E_a = 87000$$
$$T_0 = 323.15$$
$$T = 398.15$$
$$k_0 = 10^{-4}$$
$$F_{A0} = 2.7$$
$$P = 1013250$$

Solve the ODE with the following code:

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp, trapezoid
from scipy.optimize import curve_fit, fsolve
import time, math
# ODE for pfr
def HW_3_P_2_PFR(t, x):
    # constants
    k = 10**-4
    E = 87000
    R = 8.314
    T_1 = 50 + 273.15
    T_2 = 125 + 273.15
    P = 10 * 101325
    C_A0 = P / T_2 / R
    F_A0 = 2.7
    # get conversion for independent var array
    X = x[0]
    return [k * np.exp(E / R *(1/T_1 - 1/T_2)) * C_A0 * (1 - X)/(1 + 2*X) / F_A0]
# solve ODE implicitly
HW_3_P_2_solution_pfr = solve_ivp(HW_3_P_2_PFR, [0, 5], [0],  method="Radau",
↪   rtol=1e-6, atol=1e-6)
# plot solution
V_81 = np.interp(0.81, HW_3_P_2_solution_pfr.y[0], HW_3_P_2_solution_pfr.t)
```

```
plt.plot(HW_3_P_2_solution_pfr.t, HW_3_P_2_solution_pfr.y[0])
plt.plot(V_81, 0.81, 'rx')
plt.xlabel(r"$V$")
plt.ylabel(r"$X$")
plt.title("Problem 2 Volume v Conversion of a PFR")
# interpolate for X=0.81
print(round(V_81, 3))
# Aggie Honor Code: An Aggie does not lie, cheat, or steal or tolerate
# those who do.
# I certify that this work is my own and not the work of another.
#
# Name: Mark Levchenko
# Assignment #: HW 3
# Question #: 1a
```

The volume for $X = 0.81$ is $\boxed{666 \text{ L}}$.
Output plot:



(b) CSTR:

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

$-r_A$ is the same as the PFR.

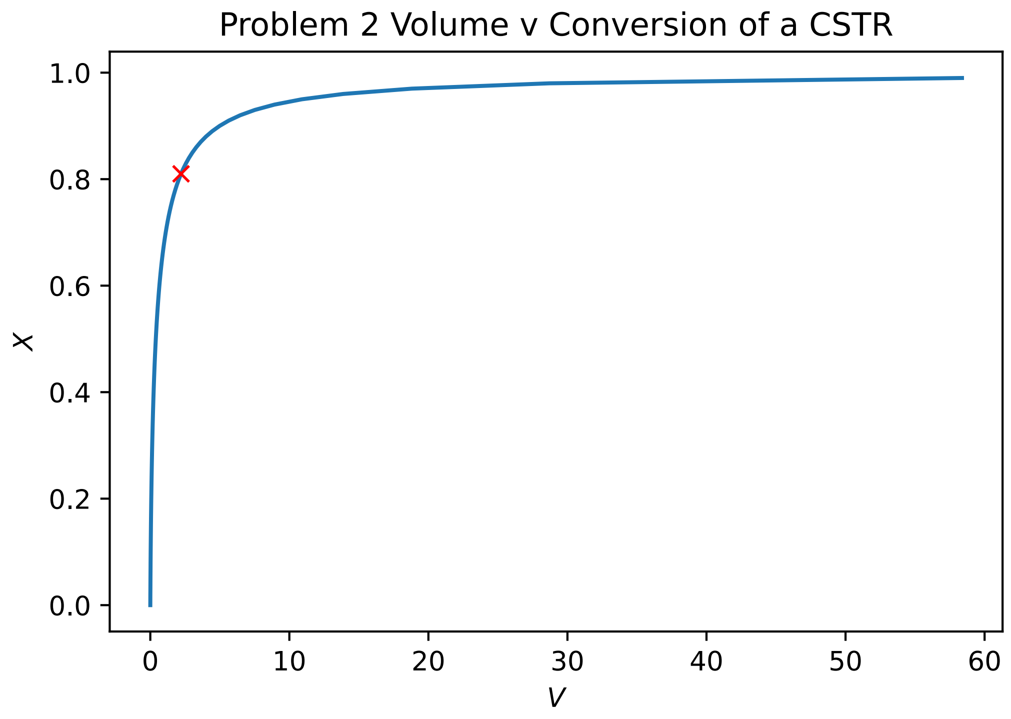
$$V = \frac{XF_{A0}}{k_0 \exp\left[\frac{E_a}{R}\left(\frac{1}{T_0} - \frac{1}{T}\right)\right] \frac{P}{RT}\left(\frac{1-X}{1+2X}\right)}$$

The following code creates a plot and outputs the appropriate volume:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp, trapezoid
from scipy.optimize import curve_fit, fsolve
import time, math
# constants
k = 10**-4
E = 87000
R = 8.314
T_1 = 50 + 273.15
T_2 = 125 + 273.15
P = 10 * 101325
C_A0 = P / T_2 / R
F_A0 = 2.7
# CSTR design equation
HW_3_P_2_CSTR = lambda X: X * F_A0 / (k * np.exp(E / R *(1/T_1 - 1/T_2)) * C_A0 * (1
↪  - X)/(1 + 2*X))

X = np.linspace(0, 0.99, 100)
# plot CSTR conversion
V_81 = HW_3_P_2_CSTR(0.81)
plt.plot(HW_3_P_2_CSTR(X), X)
plt.plot(V_81, 0.81, 'rx')
plt.xlabel(r"$V$")
plt.ylabel(r"$X$")
plt.title("Problem 2 Volume v Conversion of a CSTR")
# V at X=0.81
print(round(V_81, 3))
# Aggie Honor Code: An Aggie does not lie, cheat, or steal or tolerate
# those who do.
# I certify that this work is my own and not the work of another.
#
# Name: Mark Levchenko
# Assignment #: HW 3
# Question #: 2b
```

The volume for $X = 0.81$ is $\boxed{2210 \text{ L}}$.
Output plot:



(c)

$$V_{\text{CSTR}} = \frac{XF_{A0}}{k_0 \exp\left[\frac{E_a}{R}\left(\frac{1}{T_0} - \frac{1}{T}\right)\right] \frac{P}{RT}\left(\frac{1-X}{1+2X}\right)}$$

Solve for X where $V_{\text{CSTR}} = 1000$ L. The conversion for the CSTR becomes the initial condition for the ODE that corresponds to the PFR. The last value of the solution array for the PFR ODE is the total conversion.

The code below builds on the code from the PFR and CSTR:

```
# solve for CSTR conversion at 1000L = 1m^3
cstr_conversion = fsolve(lambda x: HW_3_P_2_CSTR(x) - 1, [0.5])
# find PFR conversion at 400L = 0.4m^3
# CSTR conversion is initial condition for PFR ODE
total_conversion = solve_ivp(HW_3_P_2_PFR, [0, 0.4], cstr_conversion,
↪  method="Radau", rtol=1e-6, atol=1e-6)
# final conversion is last value in PFR ODE solution
print(round(total_conversion.y[0][-1], 3))
# Aggie Honor Code: An Aggie does not lie, cheat, or steal or tolerate
# those who do.
# I certify that this work is my own and not the work of another.
#
# Name: Mark Levchenko
# Assignment #: HW 3
# Question #: 2c
```

The output conversion is $\boxed{X = 0.866}$.

3. Problem 3

(f)

$$\begin{aligned}\epsilon &= 0\\ \delta &= 0\end{aligned}$$

Elementary rate law:

$$\begin{aligned}
-r_A &= k_f P_A^2 - k_b P_B P_C\\
K_e &= \frac{k_f}{k_b}\\
k_b &= \frac{k_f}{K_e}\\
-r_A &= k_f P_A^2 - \frac{k_f}{K_e} P_B P_C\\
C_A &= C_{A0}\frac{(1-X)}{(1+\epsilon X)}\frac{PT_0}{P_0T}\\
C_A &= C_{A0}(1-X)\\
C_B = C_C &= C_{A0}\frac{\left(\Theta + \frac{1}{2}X\right)}{(1+\epsilon X)}\frac{PT_0}{P_0T}\\
C_B = C_C &= \frac{C_{A0}X}{2}\\
C &= \frac{P}{RT}\\
C_A &= \frac{P_A}{RT}\\
\frac{P_A}{RT} &= \frac{P_{A0}}{RT}(1-X)\\
P_A &= P_{A0}(1-X)\\
P_B = P_C &= \frac{P_{A0}X}{2}\\
-r_A &= k_f(P_{A0}(1-X))^2 - \frac{k_f}{K_e}\frac{P_{A0}X}{2}\frac{P_{A0}X}{2}\\
-r_A &= k_f P_{A0}^2\left((1-X)^2 - \frac{X^2}{2K_e}\right)
\end{aligned}$$

At equilibrium:

$$\begin{aligned}
-r_A &= 0\\
0 &= k_f P_{A0}^2\left((1-X)^2 - \frac{X^2}{2K_e}\right)\\
k_f &= 4.25\cdot 10^{-3}\\
P_{A0} &= 18\\
K_e &= 2.5
\end{aligned}$$

Solve for X:

$$\boxed{X = 0.760}$$

(h)

$$\begin{aligned}
F_{A0}\frac{dX}{dW} &= -r_A \\
\frac{dX}{dW} &= k_f P_{A0}^2\left((1-X)^2 - \frac{X^2}{2K_e}\right)/F_{A0} \\
F_{A0} &= 23.6
\end{aligned}$$

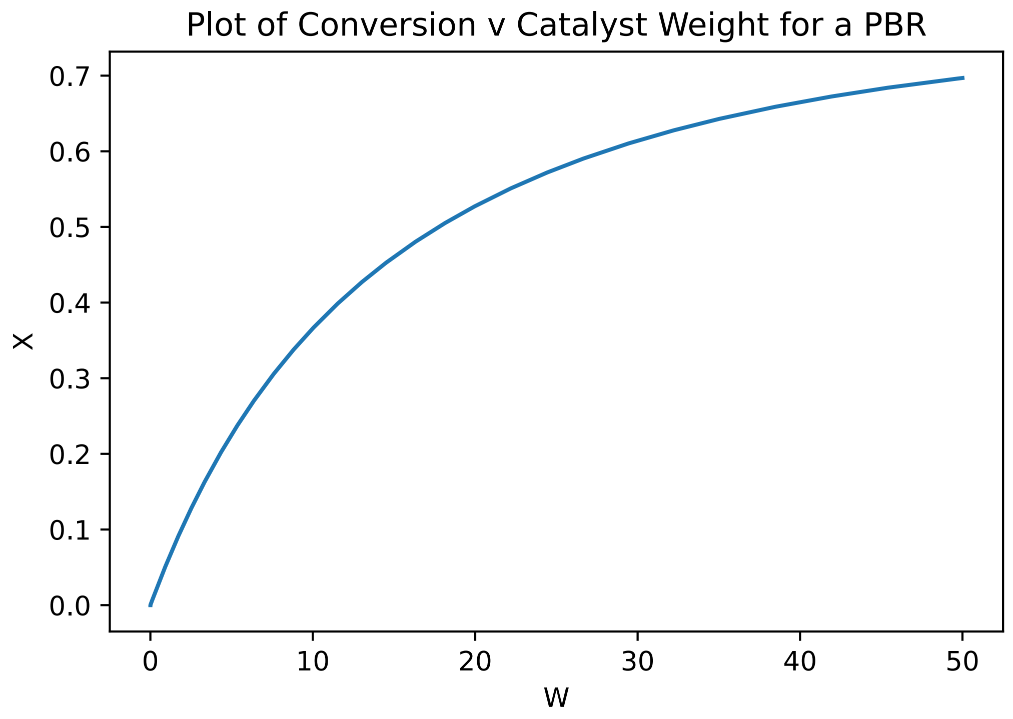
The code below solves the ODE:

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp, trapezoid
from scipy.optimize import curve_fit, fsolve
# PFR ODE
def HW_3_P_3_ODE(t, y):
    X = y[0]
    k_f = 4.25e-3
    k_e = 2.5
    k_b = k_f/k_e
    P_A0 = 18
    F_A0 = 23.6
    return [ (k_f * (1 - X)**2 - k_b * X**2 / 4) * P_A0**2 / F_A0 ]

# solve ODE implicitly
p_3_solution = solve_ivp(HW_3_P_3_ODE, [0, 50], [0], method="Radau", rtol=1e-6,
↪   atol=1e-6)

# plotting
plt.plot(p_3_solution.t, p_3_solution.y[0])
plt.xlabel(r"W")
plt.ylabel(r"X")
plt.title("Plot of Conversion v Catalyst Weight for a PBR")
# Aggie Honor Code: An Aggie does not lie, cheat, or steal or tolerate
# those who do.
# I certify that this work is my own and not the work of another.
#
# Name: Mark Levchenko
# Assignment #: HW 3
# Question #: 3h
```

Output plot:


Plot of Conversion v Catalyst Weight for a PBR
X
W
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
10
20
30
40
50