Question 2

8.1

$$d[E]/dt = -k1 * [E][S] + k2[ES] + k3 * [ES]$$

$$d[S]/dt = -k1 * [E][S] + k2[ES]$$

$$\frac{d[ES]}{dt} = k1 * [E][S] - k2[ES] - k3 * [ES]$$

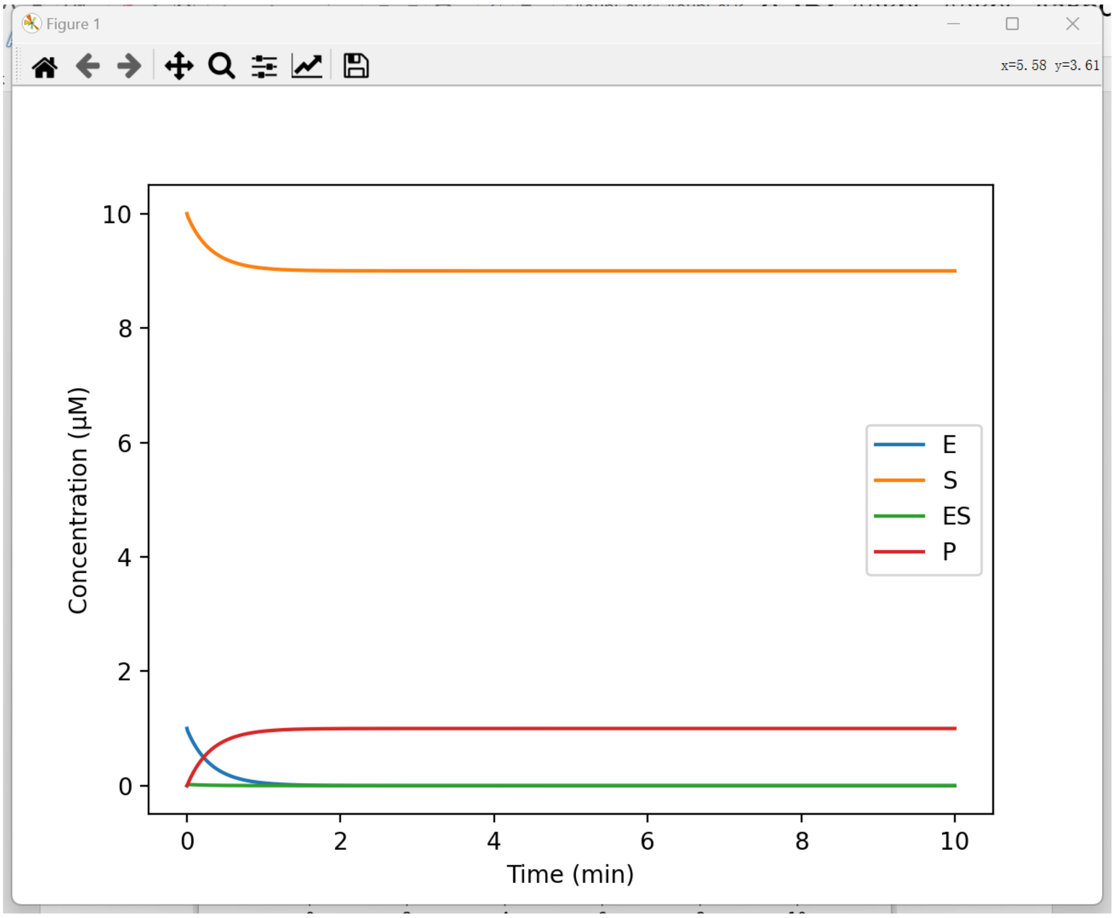
$$d[P]/dt = k3 * [ES]$$

[E]: concentrations of the enzyme
[S]: concentrations of the substrate
[ES]: concentrations of the intermediate
[P]: concentrations of the product
k1, k2, and k3 are the rate constants for the forward and reverse reactions of the enzyme-substrate complex and the breakdown of the intermediate, respectively.

8.2

```
import numpy as np
from scipy.integrate import odeint

# Define the system of differential equations
def model(y, t, k1, k2, k3):
    E, S, ES, P = y
    dEdt = -k1*E*S + k2*ES
    dSdt = -k1*E*S + k2*ES
    dESdt = k1*E*S - k2*ES - k3*ES
    dPdt = k3*ES
    return [dEdt, dSdt, dESdt, dPdt]

# Define the initial conditions, time span, and rate constants
y0 = [1, 10, 0, 0]    # initial concentrations of E, S, ES, P in µM
t = np.linspace(0, 10, 1000)    # time span in min
k1 = 100.0/60.0    # forward rate constant in µM^-1 min^-1
k2 = 600.0    # reverse rate constant in min^-1
k3 = 150.0    # product formation rate constant in min^-1

# Solve the system of differential equations using the fourth-order Runge-Kutta method
sol = odeint(model, y0, t, args=(k1, k2, k3))

# Plot the results
import matplotlib.pyplot as plt

plt.plot(t, sol[:, 0], label='E')
plt.plot(t, sol[:, 1], label='S')
plt.plot(t, sol[:, 2], label='ES')
plt.plot(t, sol[:, 3], label='P')
plt.xlabel('Time (min)')
plt.ylabel('Concentration (µM)')
plt.legend()
plt.show()
```

The code produces a plot showing the time-dependent concentrations of the enzyme, substrate, intermediate species, and product.



8.3

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

# Define the system of differential equations
def model(y, t, k1, k2, k3):
    E, S, ES, P = y
    dEdt = -k1*E*S + k2*ES
    dSdt = -k1*E*S + k2*ES
    dESdt = k1*E*S - k2*ES - k3*ES
    dPdt = k3*ES
    return [dEdt, dSdt, dESdt, dPdt]

# Define the initial conditions, time span, and rate constants
y0 = [1, 10, 0, 0]    # initial concentrations of E, S, ES, P in µM
t = np.linspace(0, 10, 1000)    # time span in min
k1 = 100.0/60.0    # forward rate constant in µM^-1 min^-1
k2 = 600.0    # reverse rate constant in min^-1
k3 = 150.0    # product formation rate constant in min^-1

# Solve the system of differential equations using the fourth-order Runge-Kutta method
sol = odeint(model, y0, t, args=(k1, k2, k3))

# Compute the velocity of the enzymatic reaction
V = k3*sol[:, 2]

# Plot the velocity as a function of the substrate concentration
plt.plot(sol[:, 1], V)
plt.xlabel('Substrate concentration (µM)')
plt.ylabel('Enzyme velocity (µM/min)')
plt.show()

# Find the maximum velocity of the enzymatic reaction
Vm = np.max(V)
print('Maximum velocity of the enzymatic reaction:', Vm, 'µM/min')
```

The code produces a plot showing the velocity V as a function of the substrate concentration S. As it can see from the graph, $Vm = 3.2e^{-12}$ M/s.