8.1.

The law of mass action states that the rate of a chemical reaction is proportional to the product of the concentrations of the reactants. Using this principle, we can write the following four differential equations to describe the rate of change of each species in the two-step enzymatic reaction:

1. For the enzyme E: $d[E]/dt = -k1[E][S] + k2[ES] + k3[ES]$
2. For the substrate S: $d[S]/dt = -k1[E][S] + k2[ES]$
3. For the intermediate species ES: $d[ES]/dt = k1[E][S] - k2[ES] - k3[ES]$
4. For the product P: $d[P]/dt = k3[ES]$

In these equations, [E], [S], [ES], and [P] represent the concentrations of the enzyme, substrate, intermediate species, and product, respectively, and k1, k2, and k3 are the rate constants for the forward and reverse reactions and the breakdown of the intermediate species, respectively. The first equation describes the rate of change of the enzyme concentration, which is equal to the rate at which the enzyme is being formed by the reverse reaction minus the rate at which it is being consumed by the forward reaction and the breakdown of the intermediate species. The second equation describes the rate of change of the substrate concentration, which is equal to the rate at which the substrate is being consumed by the forward reaction minus the rate at which it is being produced by the reverse reaction. The third equation describes the rate of change of the intermediate species concentration, which is equal to the rate at which it is being formed by the forward reaction minus the rate at which it is being consumed by the reverse reaction and the breakdown of the intermediate species. The fourth equation describes the rate of change of the product concentration, which is equal to the rate at which the product is being produced by the breakdown of the intermediate species.

8.2.

To solve these differential equations numerically using the fourth-order Runge-Kutta method, we need to specify a time interval over which the solution will be computed and a step size for the time increments.

We can then use the following steps to solve the equations:

1. Initialize the time t to the starting time of the interval and the concentrations of the four species to their initial values.
2. Compute the four rate equations at the current time t using the current concentrations of the species.
3. Use the rates to compute estimates of the concentrations of the species at the midpoint of the current time step using the fourth-order Runge-Kutta method.
4. Compute the four rate equations at the midpoint of the current time step using the estimated concentrations from the previous step.
5. Use the rates at the midpoint of the time step and the rates at the beginning of the time step to compute the concentrations of the species at the end of the time step using the fourth-order Runge-Kutta method.
6. Increment the time by the time step size and repeat from step 2 until the end of the time interval is reached.

Python code to solve these four equations using the fourth-order Runge-Kutta method:

```python
print("Solver: Zhehan Qi\n")

# Set the initial concentrations
E0 = 1e-6
S0 = 10e-6
ES0 = 0
P0 = 0

# Set the rate constants
k1 = 100/1e-6/60
k2 = 600/60
k3 = 150/60

# Set the time interval and step size
t0 = 0
t1 = 10
h = 0.1

def four_equations(t, y):
```

```
    E, S, ES, P = y
    dE = -k1*E*S + k2*ES + k3*ES
    dS = -k1*E*S + k2*ES
    dES = k1*E*S - k2*ES - k3*ES
    dP = k3*ES
    return [dE, dS, dES, dP]

def rk4(t, y, h, f):
    k1 = f(t, y)
    k2 = f(t + h/2, [y[i] + [x * h/2 for x in k1][i] for i in range(4)])
    k3 = f(t + h/2, [y[i] + [x * h/2 for x in k2][i] for i in range(4)])
    k4 = f(t + h, [y[i] + [x * h for x in k3][i] for i in range(4)])
    return [y[i] + [x * h/6 for x in k1][i] + [x * h/3 for x in k2][i] + [x * h/3 for x in k3][i] + [x * h/6 for x in k4][i] for i in range(4)]

def solve_equations(t0, t1, h, y0, f):
    t = t0
    y = y0
    while t < t1:
        y = rk4(t, y, h, f)
        t = t + h
    return y

# Solve the equations
y0 = [E0, S0, ES0, P0]
E, S, ES, P = solve_equations(t0, t1, h, y0, four_equations)

print("E =", E*1e6, "μM")
print("S =", S*1e6, "μM")
print("ES =", ES*1e6, "μM")
print("P =", P*1e6, "μM")
```

Answers:

E = 0.8652316836373131 μM

S = 1.146738379330573 μM

ES = 0.1347683163626870 5 μM

P = 8.718493304306739 μM

8.3.

To plot the velocity of the enzymatic reaction as a function of the substrate concentration, we can use the differential equations that we derived earlier to compute the velocity at different substrate concentrations.

Python code to plot the velocity V as a function of the concentration of the substrate S:

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

def compute_velocity(S):
    y0 = [E0, S, ES0, P0]
    E, S, ES, P = solve_equations(t0, t1, h, y0, four_equation
s)
    return (P - P0)/(t1 - t0)

S_values = np.linspace(0, 10, 100) * 1e-6
V_values = [compute_velocity(S) for S in S_values]

plt.plot(S_values, V_values)
plt.xlabel('Substrate concentration')
plt.ylabel('Velocity')
plt.show()

Vm = max(V_values)
print(Vm)
```

When the concentrations of S are small, the velocity V increases approximately linearly.