1 8.1

$$\frac{d[ES]}{dt} = k_1[E][S] - k_2[ES] - k_3[ES] \tag{1}$$
$$\frac{d[E]}{dt} = k_2[ES] + k_3[ES] - k_1[E][S] \tag{2}$$
$$\frac{d[S]}{dt} = k_2[ES] - k_1[E][S] \tag{3}$$
$$\frac{d[P]}{dt} = k_3[ES] \tag{4}$$

2 8.2

Final concentration of E = 0.9999999442422705 µm

Final concentration of ES = 5.575771576830324e-08 µm

Final concentration of P = 9.999999536112117 µm

Final concentration of S = 4.0813006299449143e-07 µm

3 8.3

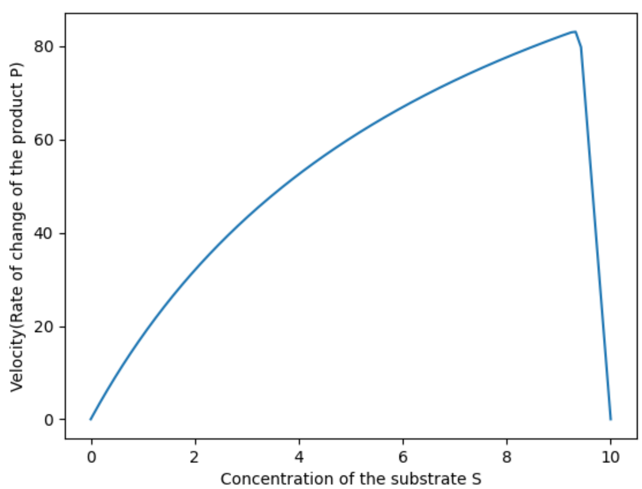


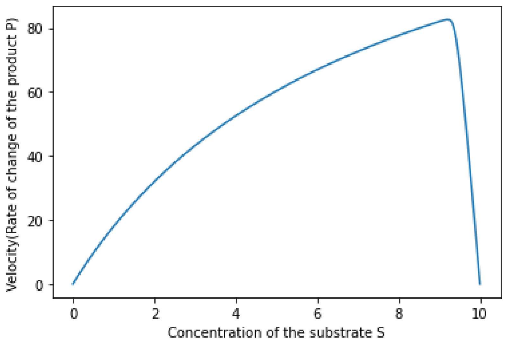
Figure 1: Graph generated by Python

According to the graph, Vm= 82.64953649378555 µm/min, Appendix:

In [1]:
```
import math
import matplotlib.pyplot as plt

#8.2
#Define initial concetration
E0=1
S0=10
ES0=0
P0=0

#Define rate constants
k1=100
k2=600
k3=150

#Define time interval and step size
t0=0
n=1
h=0.00001

#Define functions of each speice's rate of change
def dES(E,S,ES,P):
    return k1*E*S-(k2+k3)*ES
def dE(E,S,ES,P):
    return (k2+k3)*ES-k1*E*S
def dS(E,S,ES,P):
    return k2*ES-k1*E*S
def dP(E,S,ES,P):
    return k3*ES

#Define a function for Runge Kutta Fourth Order Method
def rk4(E,S,ES,P,t):
    #Record concentrations of each specie at each time interval (these records are useful for 8.3)
    time_record=[0]
    E_record=[E0]
    S_record=[S0]
    ES_record=[ES0]
    P_record=[P0]
    while t<=n:
        t+=h
        time_record.append(t)
        E1=dE(E, S, ES, P)
        ES1 = dES(E, S, ES, P)
        S1 = dS(E, S, ES, P)
        P1 = dP(E, S, ES, P)
        E2 = dE(E + E1 * h / 2, S + S1 * h / 2, ES + ES1 * h / 2, P + P1 * h / 2)
        ES2 = dES(E + E1 * h / 2, S + S1 * h / 2, ES + ES1 * h / 2, P + P1 * h / 2)
        S2 = dS(E + E1 * h / 2, S + S1 * h / 2, ES + ES1 * h / 2, P + P1 * h / 2)
        P2 = dP(E + E1 * h / 2, S + S1 * h / 2, ES + ES1 * h / 2, P + P1 * h / 2)
        E3 = dE(E + E2 * h / 2, S + S2 * h / 2, ES + ES2 * h / 2, P + P2 * h / 2)
        ES3 = dES(E + E2 * h / 2, S + S2 * h / 2, ES + ES2 * h / 2, P + P2 * h / 2)
        S3 = dS(E + E2 * h / 2, S + S2 * h / 2, ES + ES2 * h / 2, P + P2 * h / 2)
        P3 = dP(E + E2 * h / 2, S + S2 * h / 2, ES + ES2 * h / 2, P + P2 * h / 2)
        E4 = dE(E + E3 * h / 2, S + S3 * h / 2, ES + ES3 * h / 2, P + P3 * h / 2)
        ES4 = dES(E + E3 * h / 2, S + S3 * h / 2, ES + ES3 * h / 2, P + P3 * h / 2)
        S4 = dS(E + E3 * h / 2, S + S3 * h / 2, ES + ES3 * h / 2, P + P3 * h / 2)
        P4 = dP(E + E3 * h / 2, S + S3 * h / 2, ES + ES3 * h / 2, P + P3 * h / 2)
        E += (E1 + 2 * E2 + 2 * E3 + E4) * h / 6
        ES += (ES1 + 2 * ES2 + 2 * ES3 + ES4) * h / 6
        S += (S1 + 2 * S2 + 2 * S3 + S4) * h / 6
        P += (P1 + 2 * P2 + 2 * P3 + P4) * h / 6
        E_record.append(E)
        ES_record.append(ES)
        S_record.append(S)
        P_record.append(P)
    return E_record, ES_record, S_record, P_record, time_record

E_data,ES_data,S_data,P_data,time=rk4(E0,S0,ES0,P0,t0)

#Take and print the value at the last index of all species' records because they are the final concentrations of each
# specie at the end of the reaction.
print("Final concentration of E =",E_data[-1],"μm")
print("Final concentration of ES =",ES_data[-1],"μm")
print("Final concentration of P =",P_data[-1],"μm")
print("Final concentration of S =",S_data[-1],"μm")
```

```
Final concentration of E = 0.9999999442422705 μm
Final concentration of ES = 5.575771576830324e-08 μm
Final concentration of P = 9.999999536112117 μm
Final concentration of S = 4.0813006299449143e-07 μm
```

In [2]:
```
#8.3
#Since time intervals and the concentrations of ES at each time intervals are recorded in 8.2, rate of change of
#the product p at each time interval could be caculated according to the equation defined.

v_record=[]
maximum=0
for v in ES_data:
    v_record.append(v*k3)
    if v*k3>maximum:
        maximum=v*k3
plt.plot(S_data,v_record)
plt.xlabel("Concentration of the substrate S")
plt.ylabel("Velocity(Rate of change of the product P)")
plt.show()
print('Vm=',maximum,"μm/min")
```




```
Vm= 82.64953649378555 μm/min
```

In [ ]: