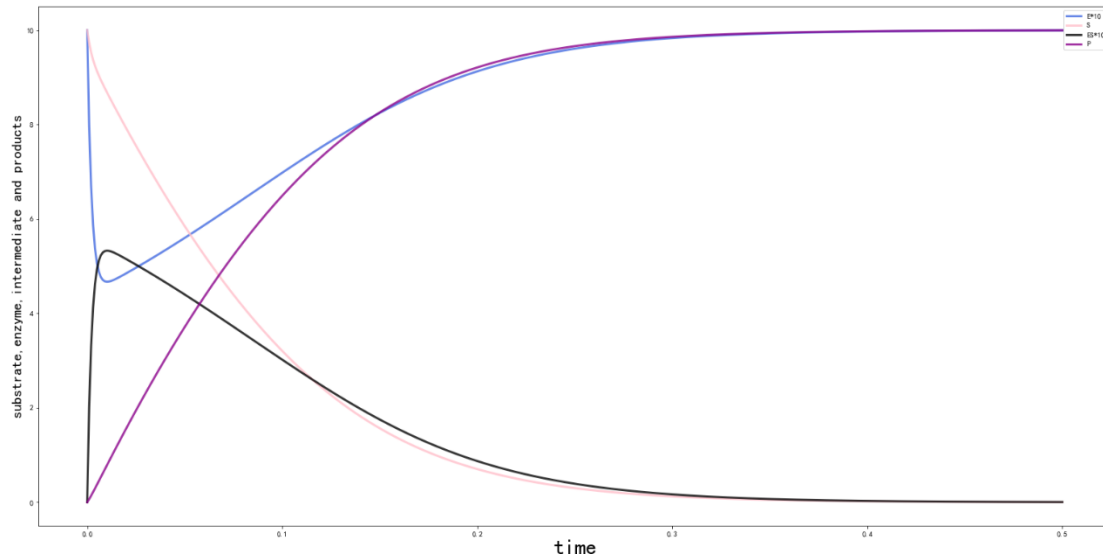


1. Using the law of mass action, we can derive the four equations for the rate of changes of the four species, E, S, ES, and P:

$$\begin{cases} \frac{dC(ES)}{dt} = k_1 C(E)C(S) - k_3 C(ES) - k_2 C(ES) \\ \frac{dC(P)}{dt} = k_3 C(ES) \\ \frac{dC(S)}{dt} = k_2 C(ES) - k_1 C(E)C(S) \\ \frac{dC(E)}{dt} = k_2 C(ES) + k_3 C(ES) - k_1 C(E)C(S) \end{cases}$$

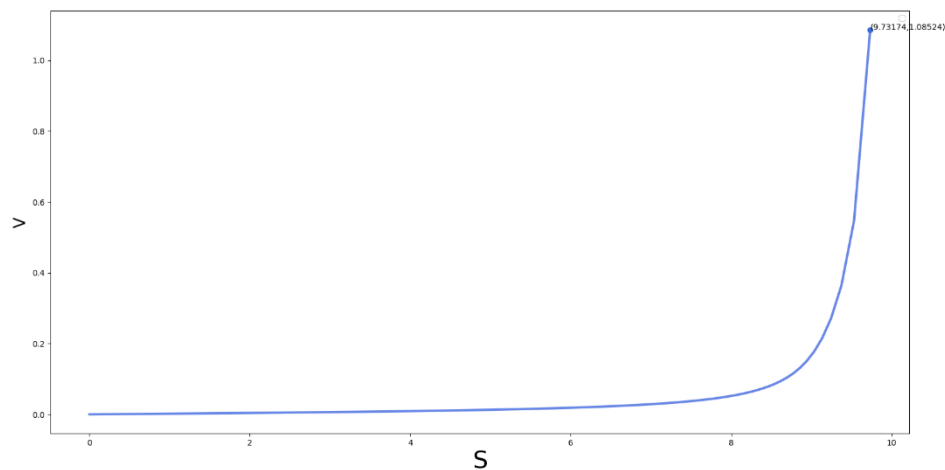
$(C(*) \text{ refers to the concentration of } (*))$

2. Using the forth-order Runge-Kutta method, we can calculate the changes of four species, E, S, ES, and P over time. The graph of changes is as follows:



(Code is at the end of this question)

3. The plot of the velocity V as a function of the concentration of the substrate S is as follows:



And when the concentration of S reaches to 9.73174, V reaches to 1.08524 which is the biggest V(Vm).

Code (Running under Python version 3.88) :

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

#four ode
def f(x):
    e,s,es,p=sy.symbols('e s es p')
    f1=150*es-100*e*s+600*es
    f2=600*es-100*e*s
    f3=100*e*s-150*es-600*es
    f4=150*es
    q1=f1.subs([(e, x[0]), (s, x[1]), (es, x[2])])
    q2=f2.subs([(e, x[0]), (s, x[1]), (es, x[2])])
    q3=f3.subs([(e, x[0]), (s, x[1]), (es, x[2])])
    q4=f4.subs(es, x[2])
    res=[q1,q2,q3,q4]
    return res

#forth-order Runge-Kutta
def R_K(x,t,h,i):
    t[i]=t[i-1]+h
    xt=np.array(x[i-1,:])
    k1=f(xt)
    for j in range(0,4):
        xt[j]=xt[j]+h*k1[j]/2
    k2=f(xt)
    for j in range(0,4):
        xt[j]=xt[j]+h*k2[j]/2
    k3=f(xt)
    for j in range(0,4):
        xt[j]=xt[j]+h*k3[j]
    k4=f(xt)
    k=np.array([k1,k2,k3,k4])
    for j in range(0,4):
        x[i,j]=x[i-1,j]+h*(k[0,j]+2*k[1,j]+2*k[2,j]+k[3,j])/6.0

m=4 #number of equations
```

```

h=0.001 #step
t0=0
t1=500
n=t1-t0+1# node number
m=np.zeros((n,m)) #variables
m[0,0]=1
m[0,1]=10
t=np.linspace(t0, t1, n)

for i in range(1,n):
    R_K(m,t,h,i)

V=np.zeros((500))
#calculate the change rate
for i in range(2,501):
    V[i-2]=(m[i,3]-m[i-1,3])/m[i-1,3]

#draw substrate,enzyme,intermediate and products—time plot
plt.subplot(2,1,2)
plt.plot(t,10*m[:,0], color='#4169E1', alpha=0.8,
linewidth=3,label="E*10")
plt.plot(t,m[:,1], color='pink', alpha=0.8, linewidth=3,label="S")
plt.plot(t,10*m[:,2], color='black', alpha=0.8,
linewidth=3,label="ES*10")
plt.plot(t,m[:,3], color='#8B008B', alpha=0.8, linewidth=3,label="P")
plt.legend(loc="upper right")
plt.ylabel('substrate,enzyme,intermediate and products',fontsize=20)
plt.xlabel('time',fontsize=30)
#draw S-V plot
plt.subplot(2,2,2)
plt.plot(m[1:501,1],V[0:500], color='#4169E1', alpha=0.8,
linewidth=3,marker='o',markevery=500)
plt.legend(loc="upper right")
plt.ylabel('V',fontsize=20)
plt.xlabel('S',fontsize=30)
plt.scatter(9.73174,1.08524)
plt.annotate('(9.73174,1.08524)', xy=(9.73174,1.08524),
xytext=(9.73174,1.08524))

plt.show()

```