# 1 Equations

- According to the law of mass action, the kinetic equation of the enzymatic reaction is:

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

$$\frac{d[S]}{dt} = -k_1[E][S] + k_2[ES] \tag{2}$$

$$\frac{d[ES]}{dt} = k_1[E][S] - k_2[ES] - k_3[ES] \tag{3}$$

$$\frac{d[P]}{dt} = k_3[ES] \tag{4}$$

# 2 Code to numerically solve equations using RK4 method

- The numerical solution stored in the .csv file has been attached. Code is as follows:

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
import pandas as pd
```

```
# E+S===ES-->E+P

# Define some constants
E=1
S=10
ES=0
P=0

k1=100
k2=600
k3=150

n=0
t=0

h=0.00002

# Define few lists to record the result
nlist=[]
tlist=[]
elist=[]
slist=[]
eslist=[]
plist=[]
vlist=[]
```

```
#define the function to calculate the fourth-order Runge- Kutta method'
                                    s result
def KL4(Et,St,t,h,n):
    K1=func1(t,Et,St)
    G1=func2(t,Et,St)
    K2=func1(t+(1/2)*h,Et+(1/2)*h*K1,St+(1/2)*h*G1)
    G2=func2(t+(1/2)*h,Et+(1/2)*h*K1,St+(1/2)*h*G1)
    K3=func1(t+(1/2)*h,Et+(1/2)*h*K2,St+(1/2)*h*G2)
    G3=func2(t+(1/2)*h,Et+(1/2)*h*K2,St+(1/2)*h*G2)
    K4=func1(t+h,Et+h*K3,St+h*G3)
    G4=func2(t+h,Et+h*K3,St+h*G3)

    e=Et+(1/6)*h*(K1+2*K2+2*K3+K4)
    s=St+(1/6)*h*(G1+2*G2+2*G3+G4)

    elist.append(e)
    slist.append(s)
```

```
#define the function to describe Et changes by t
def func1(t,Et,St):
    return k2*(1-Et)-k1*Et*St+k3*(1-Et)
```

```
#define the function to describe St changes by t
def func2(t,Et,St):
    return k2*(1-Et)-k1*Et*St
```

```
#define the main program to improve the robustness
def main():
    nlist.append(n)
    tlist.append(t)
    elist.append(E)
    slist.append(S)
    eslist.append(ES)
    plist.append(P)
    i=0
    N=n
    T=t
    while 1:
        KL4(elist[i],slist[i],tlist[i],h,nlist[i])
        N=N+1
        nlist.append(N)
        T=T+h
        tlist.append(T)
        ESt=1-elist[i+1]
        eslist.append(ESt)
        #Pt=10-slist[i+1]-eslist[i+1]
        Pt=plist[i]+h*k3*eslist[i+1]
        plist.append(Pt)


        #if(k3*eslist[i+1]<0.00000000000000001):
        if(elist[i+1]-elist[i]<0.00000000000000001 and slist[i+1]-slist
                                            [i]<0.00000000000000001 and
                                             eslist[i+1]-eslist[i]<0.
                                            00000000000000001 and plist
```

```
                                                    [i+1]-plist[i]<0.
                                                    00000000000000001):
        #if(i==2080):
            break

        i=i+1

    for pp in range (0,len(eslist)):
        vlist.append(eslist[pp]*k3)

    totallist=[]
    totallist.append(nlist)
    totallist.append(tlist)
    totallist.append(elist)
    totallist.append(slist)
    totallist.append(eslist)
    totallist.append(plist)
    totallist.append(vlist)

    df=pd.DataFrame(totallist)
    df_T=pd.DataFrame(df.values.T,columns=['n','time','[E]','[S]','[ES]
                                    ','[P]','v'])

    print(df_T)

    df_T.to_csv("/Users/hanweiyu/Desktop/bmds.csv")
```

```
if __name__ == '__main__':
    main()
```

# 3 Figure and result

- Using the equation (4) to calculate the rate of change of the product P in a quite short period, the figure shows the relationship between V, the velocity of the enzymatic reaction and [S]. The figure illustrates that, when the concentrations of S are small, the velocity V increases approximately linearly. At large concentrations of S, however, the velocity V saturates to a maximum value 82.6479.

```
fig=plt.figure(dpi=1200)
p1=fig.add_subplot(111)
p1.spines['bottom'].set_position(('data',0))
p1.spines['left'].set_position(('data',0))
p1.plot(slist,vlist)
#plt.legend(loc='best')
plt.title('v changes by [S]')
#plt.xlable('[S]',fontsize=14)
#plt.ylable('v',fontsize=14)

for i in range(1,len(slist)):
    if vlist[i]>vlist[i+1]:
        p1.text(slist[i],vlist[i],(slist[i],vlist[i]),c='red')
```

```
        break

plt.savefig('/Users/hanweiyu/Desktop/bmds.png')
plt.show()
```

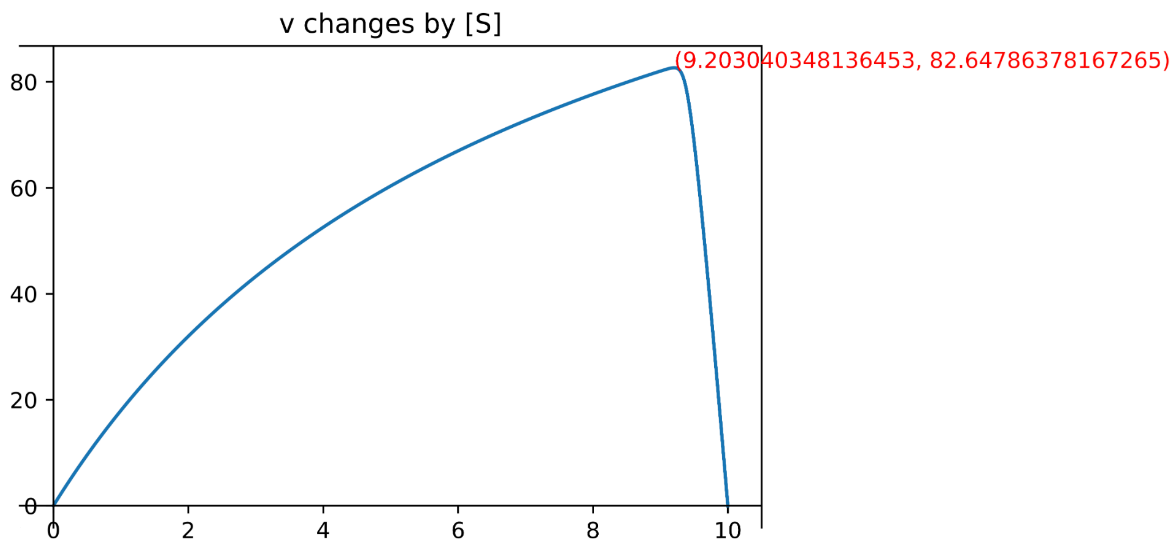


Figure 1: Plot V changes by [ES] and find the max V