Question 2: Enzyme Kinetics

Yingdong Zhang

<u>8.1</u>

[E]: concentration of enzyme E.

[S]: concentration of substrate S.

[ES]: concentration of intermediate species ES.

[P]: concentration of the product P.

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

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

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

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

8.2^[1]

Answer (The result retains 5 decimal places.)

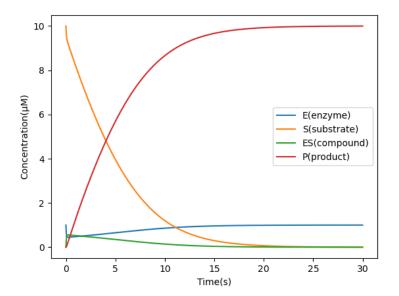
The final concentration of enzyme E is $0.99954 \mu M$.

The final concentration of substrate S is 0.00336 μM.

The final concentration of intermediate species ES is $0.00046 \mu M$.

The final concentration of product P is $9.99618 \mu M$.

Picture



Code

```
import numpy as np
import matplotlib.pyplot as plt
# define initial value
t = 0.01 \# s
T = 30 \# s
T_ls = np.arange(0, T, t)
E ls = np.zeros(T ls.shape, float)
S ls = np.zeros(T ls.shape, float)
ES_ls = np.zeros(T_ls.shape, float)
P_ls = np.zeros(T_ls.shape, float)
E ls[0] = 1 # \mu M
S ls[0] = 10 \# \mu M
ES_ls[0] = 0 \# \mu M
P_ls[0] = 0 \# \mu M
k1 = 100/60 \# \mu M/s
k2 = 600/60 \# \mu M/s
k3 = 150/60 \# \mu M/s
# Define the equations
def func E(E, S, ES):
   fE = -k1*E*S + k2*ES + k3*ES
   return fE
```

```
def func S(E, S, ES):
   fS = -k1*E*S + k2*ES
   return fS
def func ES(E, S, ES):
   fES = k1*E*S - k2*ES - k3*ES
   return fES
def func P(ES):
   fP = k3*ES
   return fP
# The fourth-order RungeKutta method
for i in range(T ls.size-1):
   E1 = func E(E ls[i], S ls[i], ES ls[i])
   S1 = func S(E ls[i], S ls[i], ES ls[i])
   ES1 = func ES(E ls[i], S ls[i], ES ls[i])
   P1 = func P(ES ls[i])
   E2 = func E(E ls[i]+E1*t/2, S ls[i]+S1*t/2, ES ls[i]+ES1*t/2)
   S2 = func S(E ls[i]+E1*t/2, S ls[i]+S1*t/2, ES ls[i]+ES1*t/2)
   ES2 = func ES(E ls[i]+E1*t/2, S ls[i]+S1*t/2, ES ls[i]+ES1*t/2)
   P2 = func P(ES ls[i]+ES1*t/2)
   E3 = func E(E ls[i]+E2*t/2, S ls[i]+S2*t/2, ES ls[i]+ES2*t/2)
   S3 = func S(E ls[i]+E2*t/2, S ls[i]+S2*t/2, ES ls[i]+ES2*t/2)
   ES3 = func ES(E ls[i]+E2*t/2, S ls[i]+S2*t/2, ES ls[i]+ES2*t/2)
   P3 = func P(ES ls[i]+ES2*t/2)
   E4 = func_E(E_ls[i]+E3*t/2, S_ls[i]+S3*t/2, ES_ls[i]+ES3*t/2)
   S4 = func S(E ls[i]+E3*t/2, S ls[i]+S3*t/2, ES ls[i]+ES3*t/2)
   ES4 = func ES(E ls[i]+E3*t/2, S ls[i]+S3*t/2, ES ls[i]+ES3*t/2)
   P4 = func P(ES ls[i]+ES3*t/2)
   E ls[i+1] = E ls[i] + t*(E1+2*E2+2*E3+E4)/6
   S ls[i+1] = S ls[i] + t*(S1+2*S2+2*S3+S4)/6
   ES ls[i+1] = ES ls[i] + t*(ES1+2*ES2+2*ES3+ES4)/6
   P ls[i+1] = P ls[i] + t*(P1+2*P2+2*P3+P4)/6
# Print the answer
print('The final concentration of enzyme E is %.5f µM.' %E ls[-1])
print('The final concentration of substrate S is %.5f µM.' %S ls[-1])
print('The final concentration of intermediate species ES is %.5f
```

```
pM.' %ES_ls[-1])
print('The final concentration of product P is %.5f µM.' %P_ls[-1])

# Draw the picture
plt.figure()
plt.plot(T_ls, E_ls, label='E(enzyme)')
plt.plot(T_ls, S_ls, label='S(substrate)')
plt.plot(T_ls, ES_ls, label='ES(compound)')
plt.plot(T_ls, P_ls, label='P(product)')
plt.xlabel('Time(s)')
plt.ylabel('Concentration(µM)')
plt.legend()
plt.show()
```

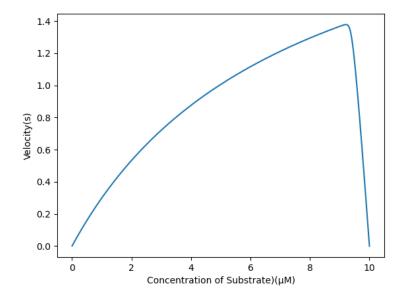
<u>8.3</u>

Answer (The result retains 5 decimal places.)

When the substrate concentration is $9.20447~\mu M$, the reaction rate reaches its maximum value.

The maximum speed Vm is $82.67835 \mu M/min$.

Picture



Code

```
V_ls = k3*ES_ls

Vm = max(V_ls)*60

index = np.argmax(V_ls)

Sm = S_ls[index]

#Print the answer

print('8.3')

print('When the substrate concentration is %.5f µM, the reaction rate

reaches its maximum value.'%Sm)

print ('The maximum speed Vm is %.5f µM/min.'%Vm)

#Draw the picture

plt.figure()

plt.plot(S_ls, V_ls)

plt.xlabel('Concentration of Substrate)(µM)')

plt.ylabel('Velocity(s)')

plt.show()
```

References

[1] HAO, B. (n.d.). 四阶 Runge-Kutta 法. 知乎专栏. Retrieved February 27, 2023, from https://zhuanlan.zhihu.com/p/146771778