8.1

$d[ES]/dt=k_1*[E]*[S]-k_2*[ES]-K_3[ES]$

$d[P]/dt=k_3*[ES]$

$d[S]/dt=k_2*[ES]-k_1*[E][S]$

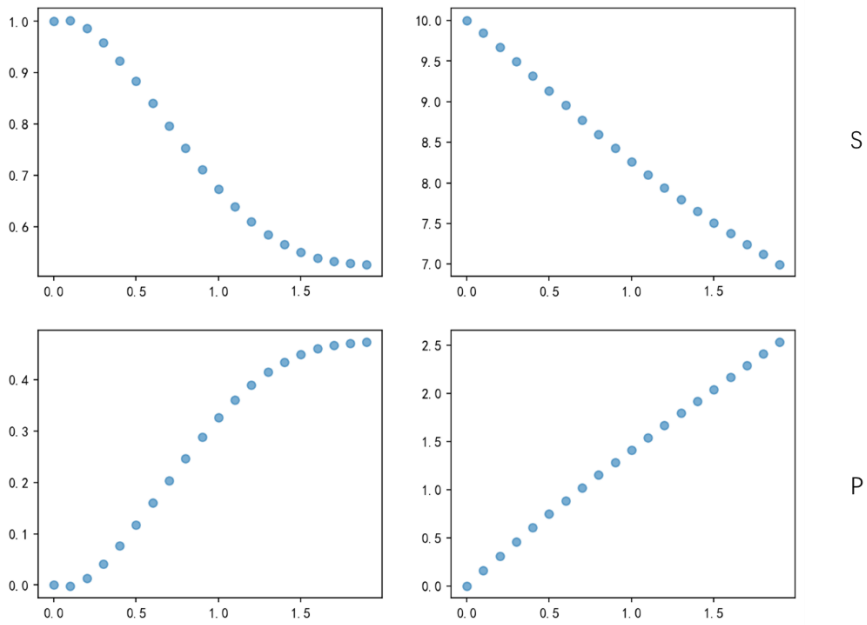
$d[E]/dt=k_3*[ES]+k_2*[ES]-k_1[E][S]$

```
8.2
'''
@Author: ZizhuoLiao
@Date: 2022-12-17 12:22:36
@LastEditTime: 2022-12-17 13:07:20
'''
import numpy as np
import math as m
from sympy import *
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus']=False

xarray=[] #E
yarray=[] #S
marray=[] #ES
narray=[] #P
array=[]

def f(x,y,m,n):
    a=2.5*m+10*m-100*x*y/60
    return a

def g(x,y,m,n):
    a=10*m-100*x*y/60
    return a

def c(x,y,m,n):
    a=100*x*y/60-10*m-2.5*m
    return a

def v(x,y,m,n):
    a=2.5*m
    return a

def RK4():
    h=0.1
    a=0
    x=1
    y=10
    m=n=0
    while a<=2:
        array.append(a)
        xarray.append(x)
```

```
        yarray.append(y)
        marray.append(m)
        narray.append(n)
        a+=h

        f1=f(x,y,m,n) #Step1
        x1=x+f1*h/2
        g1=g(x,y,m,n)
        y1=y+g1*h/2
        c1=c(x,y,m,n)
        m1=m+c1*h/2
        v1=v(x,y,m,n)
        n1=n+v1*h/2

        f2=f(x1,y1,m1,n1) #Step2
        x2=x+f2*h/2
        g2=g(x1,y1,m1,n1)
        y2=y+g2*h/2
        c2=c(x1,y1,m1,n1)
        m2=m+c2*h/2
        v2=v(x1,y1,m1,n1)
        n2=n+v2*h/2

        f3=f(x2,y2,m2,n2) #Step3
        x3=x+f3*h
        g3=g(x2,y2,m2,n2)
        y3=y+g3*h
        c3=c(x2,y2,m2,n2)
        m3=m+c3*h
        v3=v(x2,y2,m2,n2)
        n3=n+v3*h


        f4=f(x3,y3,m3,n3) #Step4
        g4=g(x3,y3,m3,n3)
        c4=c(x3,y3,m3,n3)
        v4=v(x3,y3,m3,n3)

        x=x+(f1+2*f2+2*f3+f4)*h/6
        y=y+(g1+2*g2+2*g3+g4)*h/6
        m=m+(c1+2*c2+2*c3+c4)*h/6
        n=n+(v1+2*v2+2*v3+v4)*h/6

def main():
```

```
    RK4()
    for i in xarray:
        print(i)
    print("----------")
    for i in yarray:
        print(i)
    print("----------")
    for i in marray:
        print(i)
    print("----------")
    for i in narray:
        print(i)
    plt.figure(12)
    plt.figure(figsize=(10,8),dpi=150)
    plt.subplot(221)
    plt.scatter(array, xarray, alpha=0.6)
    plt.subplot(222)
    plt.scatter(array, yarray, alpha=0.6)
    plt.subplot(223)
    plt.scatter(array, marray, alpha=0.6)
    plt.subplot(224)
    plt.scatter(array, narray, alpha=0.6)
    plt.show()

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

E



S

ES

P

```
8.3
'''
@Author: ZizhuoLiao
@Date: 2022-12-17 12:22:36
@LastEditTime: 2022-12-17 13:07:20
'''
import numpy as np
import math as m
from sympy import *
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus']=False

xarray=[] #E
yarray=[] #S
marray=[] #ES
narray=[] #P
varray=[]
array=[]

def f(x,y,m,n):
    a=2.5*m+10*m-100*x*y/60
    return a

def g(x,y,m,n):
    a=10*m-100*x*y/60
    return a

def c(x,y,m,n):
    a=100*x*y/60-10*m-2.5*m
    return a

def v(x,y,m,n):
    a=2.5*m
    return a

def RK4():
    h=0.01
    a=0
    x=1
    y=200
    m=n=0
    vm=0
    Et=0
```

```
while a<=300:
    array.append(a)
    xarray.append(x)
    yarray.append(y)
    marray.append(m)
    narray.append(n)
    varray.append(vm)
    a+=h

    f1=f(x,y,m,n) #Step1
    x1=x+f1*h/2
    g1=g(x,y,m,n)
    y1=y+g1*h/2
    c1=c(x,y,m,n)
    m1=m+c1*h/2
    v1=v(x,y,m,n)
    n1=n+v1*h/2

    f2=f(x1,y1,m1,n1) #Step2
    x2=x+f2*h/2
    g2=g(x1,y1,m1,n1)
    y2=y+g2*h/2
    c2=c(x1,y1,m1,n1)
    m2=m+c2*h/2
    v2=v(x1,y1,m1,n1)
    n2=n+v2*h/2

    f3=f(x2,y2,m2,n2) #Step3
    x3=x+f3*h
    g3=g(x2,y2,m2,n2)
    y3=y+g3*h
    c3=c(x2,y2,m2,n2)
    m3=m+c3*h
    v3=v(x2,y2,m2,n2)
    n3=n+v3*h


    f4=f(x3,y3,m3,n3) #Step4
    g4=g(x3,y3,m3,n3)
    c4=c(x3,y3,m3,n3)
    v4=v(x3,y3,m3,n3)

    x=x+(f1+2*f2+2*f3+f4)*h/6
    y=y+(g1+2*g2+2*g3+g4)*h/6
```

```
            m=m+(c1+2*c2+2*c3+c4)*h/6
            n=n+(v1+2*v2+2*v3+v4)*h/6
            Et=x+m
            vm=2.5*Et*y/(7.5+y)

def main():
    RK4()
    plt.figure(figsize=(10,8),dpi=150)
    plt.subplot(221)
    plt.scatter(yarray, varray, alpha=0.6,s=0.1,marker='o')
    plt.axhline(2.42,0,0.91,color="red",ls="--",lw=1)
    plt.axvline(200,0,0.95,color="red",ls="--",lw=1)
    plt.axvline(7.5,0,0.5,color="red",ls="--",lw=1)
    plt.axhline(1.21,0,0.08,color="red",ls="--",lw=1)
    plt.show()

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