$$\frac{ds}{dt} = k_2 * es - k_1 * s * e$$

$$\frac{de}{dt} = (k_2 + k_3) * es - k_1 * s * e$$

$$\frac{des}{dt} = k_1 * s * e - (k_2 + k_3) * es$$

$$\frac{dp}{dt} = k_3 * es$$

python:

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

def funcSt(s,e,es,p,k1,k2,k3):
    return k2 * es - k1 * s * e
def funcEt(s,e,es,p,k1,k2,k3):
    return (k2 + k3) * es - k1 * s * e
def funcESt(s,e,es,p,k1,k2,k3):
    return k1 * s * e - (k2 + k3) * es
def funcPt(s,e,es,p,k1,k2,k3):
    return k3 * es

# RK4
t_ini = 0                   # tmin
t_end = 0.5                 # tmax
t_h = 1e-5                  # 步进长度
```

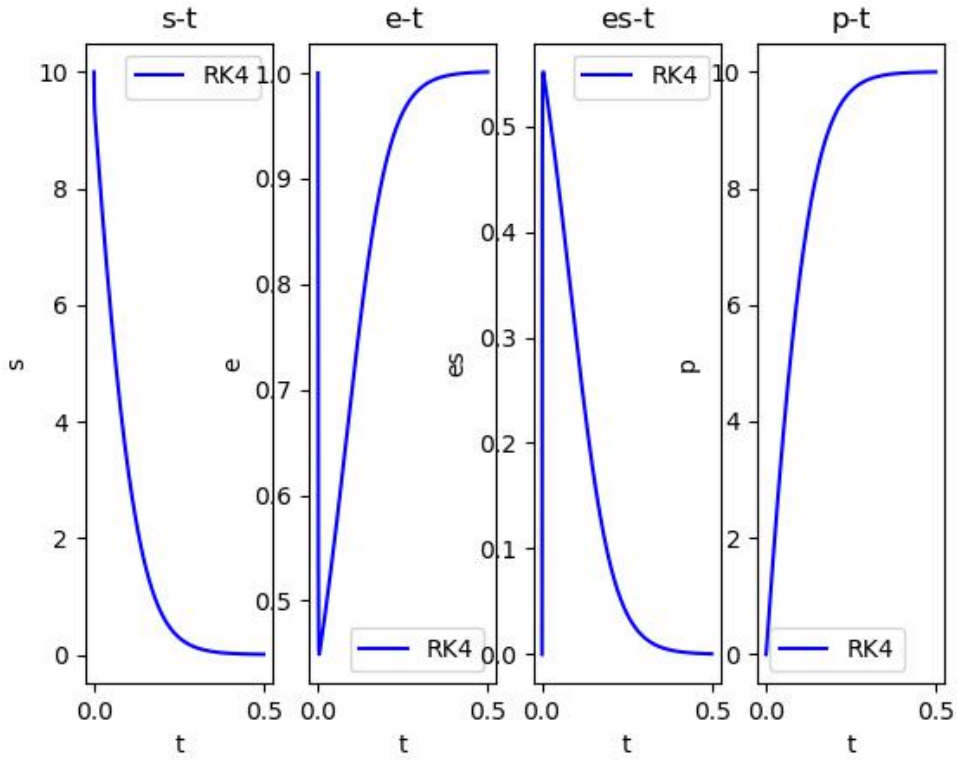
```
17.
18. t = np.linspace(t_ini, t_end, int((t_end-t_ini)/t_h+1))
19. s = t.copy()
20. e = t.copy()
21. es = t.copy()
22. p = t.copy()
23.
24. pt = t.copy()
25.
26. s[0] = 10.0
27. e[0] = 1.0
28. es[0] = 0
29. p[0] = 0
30.
31. k1 = 100
32. k2 = 600
33. k3 = 150
34.
35. for i in range(t.shape[0]-1):
36.     h_i = t[i+1] - t[i]
37.
38.     k1_s = funcSt(s[i],e[i],es[i],p[i],k1,k2,k3)
```

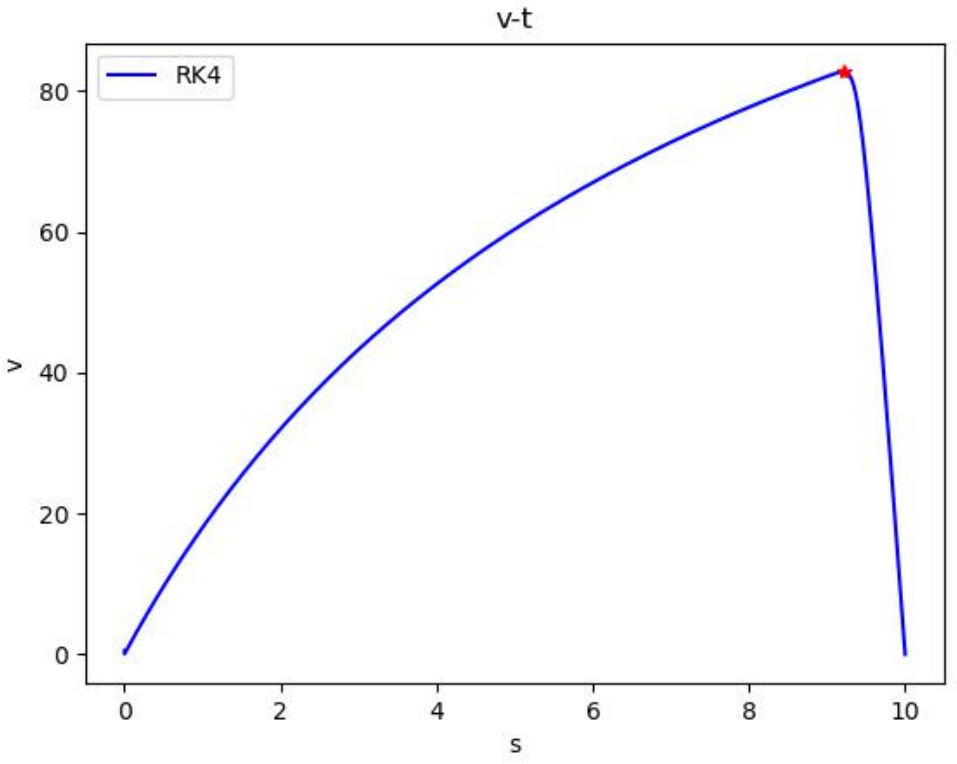
```
k1_e = funcEt(s[i],e[i],es[i],p[i],k1,k2,k3)
k1_es = funcESt(s[i],e[i],es[i],p[i],k1,k2,k3)
k1_p = funcPt(s[i],e[i],es[i],p[i],k1,k2,k3)

k2_s = funcSt(s[i] + h_i/2.0 * k1_s,e[i],es[i],p[i],k1,k2,k3)
k2_e = funcEt(s[i],e[i] + h_i/2.0 * k1_e,es[i],p[i],k1,k2,k3)
k2_es = funcESt(s[i],e[i],es[i] + h_i/2.0 * k1_es,p[i],k1,k2,k3)
k2_p = funcPt(s[i],e[i],es[i],p[i] + h_i/2.0 * k1_p,k1,k2,k3)

k3_s = funcSt(s[i] + h_i/2.0 * k2_s,e[i],es[i],p[i],k1,k2,k3)
k3_e = funcEt(s[i],e[i] + h_i/2.0 * k2_e,es[i],p[i],k1,k2,k3)
k3_es = funcESt(s[i],e[i],es[i] + h_i/2.0 * k2_es,p[i],k1,k2,k3)
k3_p = funcPt(s[i],e[i],es[i],p[i] + h_i/2.0 * k2_p,k1,k2,k3)


k4_s = funcSt(s[i] + h_i/2.0 * k3_s,e[i],es[i],p[i],k1,k2,k3)
k4_e = funcEt(s[i],e[i] + h_i/2.0 * k3_e,es[i],p[i],k1,k2,k3)
k4_es = funcESt(s[i],e[i],es[i] + h_i/2.0 * k3_es,p[i],k1,k2,k3)
k4_p = funcPt(s[i],e[i],es[i],p[i] + h_i/2.0 * k3_p,k1,k2,k3)

s[i+1] = s[i] + h_i/6.0*(k1_s + 2.0 * k2_s + 2.0* k3_s + k4_s)
e[i+1] = e[i] + h_i/6.0*(k1_e + 2.0 * k2_e + 2.0* k3_e + k4_e)
```

```
    es[i+1] = es[i] + h_i/6.0*(k1_es + 2.0 * k2_es + 2.0* k3_es + k4_es)
    p[i+1] = p[i] + h_i/6.0*(k1_p + 2.0 * k2_p + 2.0* k3_p + k4_p)

    pt[i] = k1_p

pt =np.abs(pt)
### 画图
plt.subplot(1, 4, 1)
plt.plot(t, s, 'b', label='RK4')
plt.legend()
plt.xlabel('t')
plt.ylabel('s')
plt.title('s-t')

plt.subplot(1, 4, 2)
plt.plot(t, e, 'b', label='RK4')
plt.legend()
plt.xlabel('t')
plt.ylabel('e')
plt.title('e-t')

plt.subplot(1, 4, 3)
```

```
plt.plot(t, es, 'b', label='RK4')
plt.legend()
plt.xlabel('t')
plt.ylabel('es')
plt.title('es-t')

plt.subplot(1, 4, 4)
plt.plot(t, p, 'b', label='RK4')
plt.legend()
plt.xlabel('t')
plt.ylabel('p')
plt.title('p-t')
plt.show()

plt.plot(s, pt, 'b', label='RK4')
plt.legend()
plt.xlabel('s')
    plt.ylabel('v')
    plt.title('v-t')
    x = s[pt==np.max(pt)]
    y = np.max(pt)
    plt.plot(x,y,'r',marker='*')
```

```
105.    plt.show()
```





The reason why there is a sudden drop to 0 later in the curve is, I think, to consider that the reaction starts with the maximum concentration, but the production rate increases from 0