

二氧化硫绝热反应器的优化

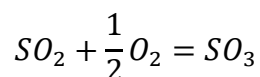
化研 2201 骆忱 2022200008

目录

一、题目及要求.....	2
1、化学反应方程式.....	2
2、基础数据.....	2
3、基本要求.....	2
4、讨论.....	3
二、T-X 图绘制平衡线与等反应速率线	3
1、平衡线的绘制.....	3
2、等反应速率线的绘制.....	4
三、反应具体参数计算.....	6
四、讨论.....	8
1、要求的最终转化率从 98%变化到 99%对催化剂用量的影响.....	8
2、 $y_{O_2}+y_{SO_2}=21\%$ ， SO_2 进口浓度在 7—9%之间变化，对催化剂装量的影响	8
五、附录.....	9
1、平衡线的 C 语言源代码	9
2、等反应速率线的 C 语言源代码	9
3、反应具体参数进出口 C 语言源代码	10

一、题目及要求

1、化学反应方程式



本反应是放热反应,采用四段绝热反应器,级间间接换热,并在常压下反应。

2、基础数据

混合物恒压热容 $C_p = 0.2549 \text{ kcal/kg} \cdot \text{K}$

$-\Delta H = 23135 \text{ kcal/kmol}$

催化剂堆密度 $\rho_b = 554 \text{ kg/m}^3$

进口 SO_2 浓度 8.0mol%, O_2 浓度 9.0mol%, 其余为氮气

处理量 $131 \text{ kmol } SO_2 / \text{hr}$, 要求最终转化率 98%

动力学方程:

$$-R_{SO_2} = k_{eff} P_{O_2} \frac{K P_{SO_2} / P_{SO_3} (1 - \xi^2)}{\left(\sqrt{B + (B-1) P_{SO_2} / P_{SO_3}} + \sqrt{K P_{SO_2} / P_{SO_3}} \right)^2} [\text{mol/gcat} \cdot \text{sec}]$$

式中:

$$k_{eff} = 7.6915 \times 10^{18} \exp\left(\frac{-76062}{RT}\right), (420 - 475^\circ\text{C})$$

$$k_{eff} = 1.5128 \times 10^7 \exp\left(\frac{-35992}{RT}\right), (475 - 600^\circ\text{C})$$

$$B = 48148 \exp\left(\frac{-7355.5}{T}\right)$$

$$K = 2.3 \times 10^{-8} \exp\left(\frac{27200}{RT}\right)$$

$$\xi = P_{SO_3} / (K P_{SO_2} P_{O_2}^{1/2})$$

$$K_p = 2.26203 \times 10^{-5} \exp\left(\frac{11295.3}{T}\right), R = 1.987$$

3、基本要求

- ① 在 T-X 图上, 做出平衡线; 至少有四条等速率线。
- ② 以一维拟均相平推流模型为基础, 在催化剂用量最少的前提下, 求总的及各段的催化剂装量、进出口温度以及转化率, 并在 T-X 图上标出折线。
- ③ 程序用 C、Fortran、BASIC 语言之一编制。

④ 12月1日左右交（电子版），独立编制程序。

4、讨论

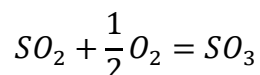
① 要求的最终转化率从 98%变化到 99%对催化剂用量的影响

② $y_{O_2} + y_{SO_2} = 21\%$ ， SO_2 进口浓度在 7-9%之间变化，对催化剂装量的影响。

二、T-X 图绘制平衡线与等反应速率线

1、平衡线的绘制

由动力学方程可知二氧化硫反应温度在 420°C-600°C之间，即 693.15-873.15K。平衡线是在一定温度下，可逆反应所能达到的最大转化率，在平衡线上的点，反应速率为 0，达到了反应平衡。化学反应式如下所示：



为方便命名，可以用 $A + \frac{1}{2}B = C$ 代替，膨胀因子 $\delta_A = -0.5$ ，当反应平衡时， $-R_{SO_2} = 0$ ，所以 $(1 - \xi^2) = 0$ ， $\xi = 1 = P_C / (K_p P_A P_B^{1/2})$ 。

$$\text{所以 } K_p = 2.26203 \times 10^{-5} \exp\left(\frac{11295.3}{T}\right) = \frac{P_C}{P_A \cdot P_B^{0.5}}$$

$$P_A = \frac{y_{A0} P (1 - X_A)}{1 + \delta_A y_{A0} X_A}$$

$$P_B = \frac{y_{B0} P - \frac{1}{2} y_{A0} P (1 - X_A)}{1 + \delta_A y_{A0} X_A}$$

$$P_C = \frac{y_{A0} P X_A}{1 + \delta_A y_{A0} X_A}$$

根据题目要求可知： $y_a = 0.08$ $y_b = 0.09$ ，本次绘制平衡线求得 X 的方法是运用 C 语言的嵌套循环法，温度的步长为 10K，求得对应的反应转化率，详细代码见附录 1。得到 T-X 的值如下表所示

表 1 平衡线上的温度和转化率的值

温度/K	转化率
693.15	0.9881
703.15	0.9850
713.15	0.9813
723.15	0.9768
733.15	0.9714
743.15	0.9650
753.15	0.9574
763.15	0.9485
773.15	0.9381
783.15	0.9261
793.15	0.9124
803.15	0.8967
813.15	0.8790
823.15	0.8592
833.15	0.8371
843.15	0.8129
853.15	0.7865
863.15	0.7579
873.15	0.7274

根据上表的数据，用 origin 画图得：

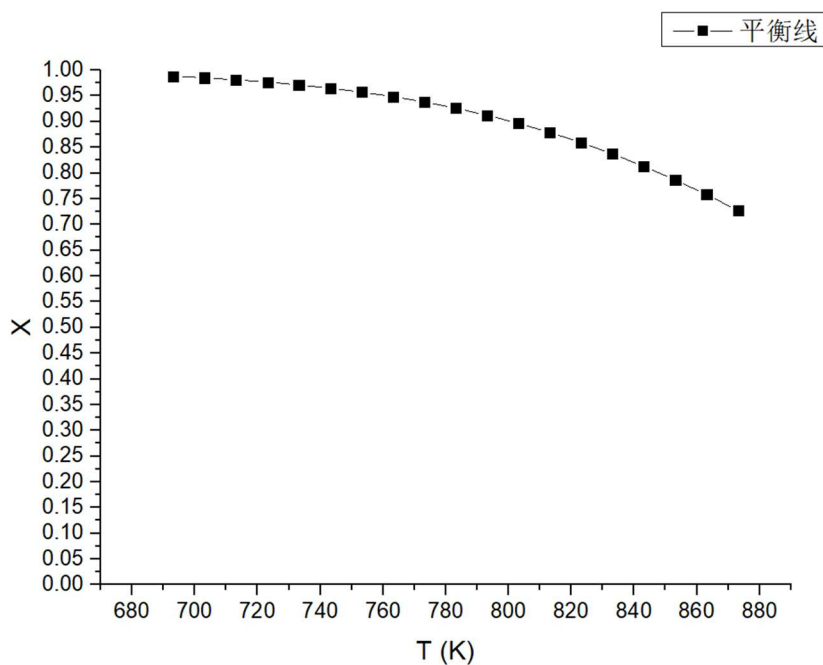


图 1 二氧化硫在反应温度下的平衡线

2、等反应速率线的绘制

平衡线相当于 $r=0$ 时的等反应速率线，选定 r 等于特定的值后，输入动力学

参数，迭代 x 的值使计算得到的 r 等于特定的值后跳出循环即可。其他四条等反应速率线选定 $r=2.5\times 10^{-6}$ 、 $r=5\times 10^{-6}$ 、 $r=7.5\times 10^{-6}$ 、 $r=1\times 10^{-5}$ mol/gcat·sec 源代码见附录二（以 $r=1\times 10^{-5}$ mol/gcat·s 为例），同理只要改变 r 的值，即可得到其他等反应速率线。具体数值如下表 2 所示

表 2 等速率线的 T-X 数值

	$r=2.5\times 10^{-6}$	$r=5\times 10^{-6}$	$r=7.5\times 10^{-6}$	$r=1\times 10^{-5}$
温度/K	mol/gcat·s 时的 转化率	mol/gcat·s 时的 转化率	mol/gcat·s 时的 转化率	mol/gcat·s 时的 转化率
693.15	0.1632	0.0427	0.0191	0.0108
703.15	0.4035	0.1293	0.0588	0.0333
713.15	0.6405	0.3265	0.1671	0.0972
723.15	0.7732	0.5605	0.3755	0.2478
733.15	0.8392	0.7125	0.5839	0.4643
743.15	0.8719	0.7937	0.7141	0.6343
753.15	0.8771	0.8160	0.7553	0.6946
763.15	0.8676	0.8094	0.7527	0.6967
773.15	0.8562	0.8001	0.7466	0.6944
783.15	0.8429	0.7885	0.7376	0.6884
793.15	0.8276	0.7746	0.7258	0.6792
803.15	0.8104	0.7585	0.7114	0.6670
813.15	0.7912	0.7402	0.6948	0.6523
823.15	0.7701	0.7199	0.6759	0.6352
833.15	0.7470	0.6977	0.6550	0.6159
843.15	0.7221	0.6736	0.6323	0.5948
853.15	0.6954	0.6480	0.6080	0.5720
863.15	0.6672	0.6209	0.5823	0.5479
873.15	0.6241	0.5718	0.5285	0.4903

用 origin 把上面等速率线的数据和平衡线的数据合并，可以得到平衡线与等反应速率曲线。如下图所示：

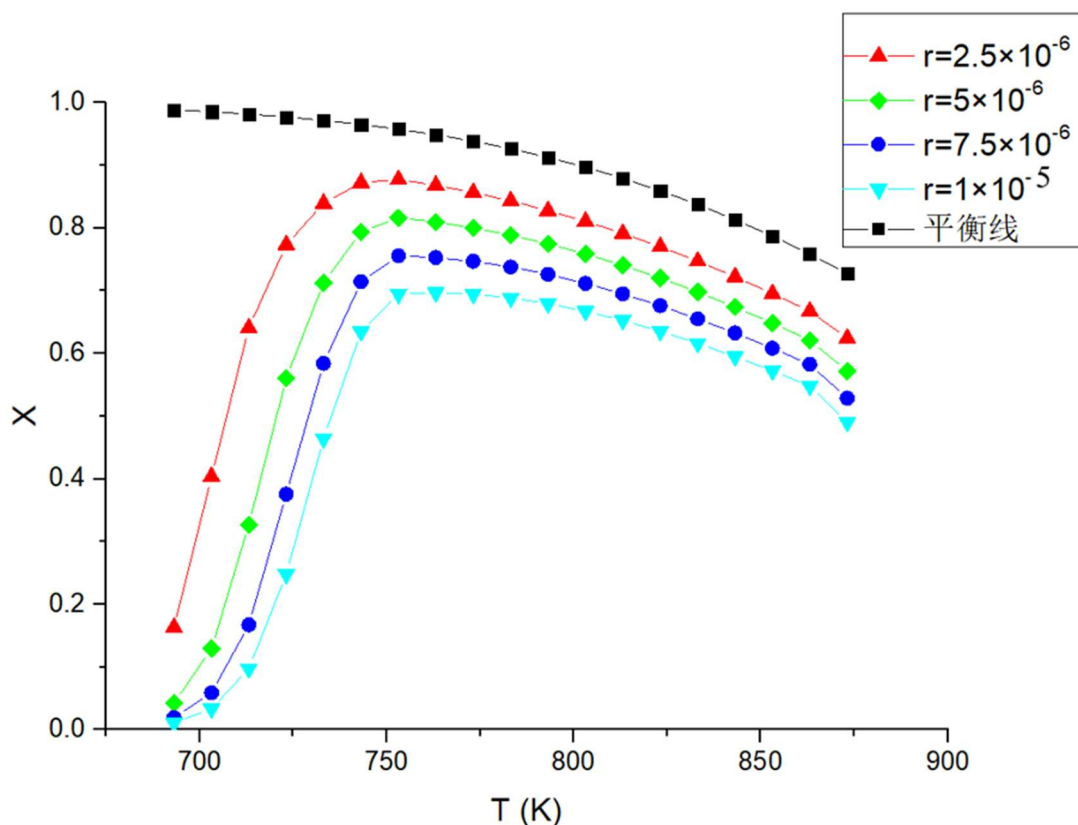


图2 二氧化硫在反应温度下的平衡线与等反应速率曲线

三、反应具体参数计算

本反应为放热反应，当温度升高时，转化率会下降。但温度升高可以使反应加快，添加催化剂同样也可以使反应速度加快。本题目要求在催化剂用量最少的情况下求总的及各段的催化剂装量，进出口温度、转化率，催化剂用量最少，化学反应器的体积也就最小。由于本设计要求四级反应器，若使反应所需催化剂用量最少。催化剂总的用量可以由下式计算：

$$W_{\text{cat}} = F_{\text{SO}_2} \sum_{i=1}^4 \int_{x_{i-1}}^{x_i} \frac{dx}{-r}$$

把催化剂的用量求对 x 和 T 的偏导，使其等于 0，即可得到最小值的催化剂，化学反应速率 r 与 x 和 T 都有关。

$$\frac{\partial W_{\text{cat}}}{\partial T} = F_{\text{SO}_2} \sum_{i=1}^4 \int_{x_{i-1}}^{x_i} \frac{dx}{-r} \frac{\partial}{\partial T} = 0$$

$$\frac{\partial W_{\text{cat}}}{\partial x} = F_{\text{SO}_2} \sum_1^4 \int_{x_{i-1}}^{x_i} \frac{dx}{-r} \frac{\partial}{\partial x} = 0$$

在上面的公式中，只有 $X_{\text{in}}=0$ 已知，因此以步长为 0.0001K 循环得到一个 T_{in} ，得到反应转化率 X_{out} ，然后根据 $n\text{SO}_2 \cdot \Delta x_{\text{SO}_2} \cdot (-\Delta H) = M \cdot c_p \cdot \Delta T$ 即可得到 T_{out} ，上一段出口的反应速率等于下一段的进口反应速率，根据相同的反应速率可以得到下一段进口的温度。最后判断 $X_{4\text{out}}$ 大于 0.98 即可跳出循环。根据上述条件可以得到具体的数值，源代码见附录 3。最后得出的反应器数据见下表所示

表 3 最终反应器的具体数据

	X_{in}	X_{out}	T_{in}/K	T_{out}/K	W_{cat}/kg	总质量
1	0.00	0.66	732.30	873.15	9311.43	48405.24
2	0.66	0.90	723.60	788.24	6765.34	
3	0.90	0.97	698.95	738.17	9758.86	
4	0.97	0.98	702.49	703.01	22569.62	

然后将上面计算出的数据绘制在 T-X 图中得到如下折线图。

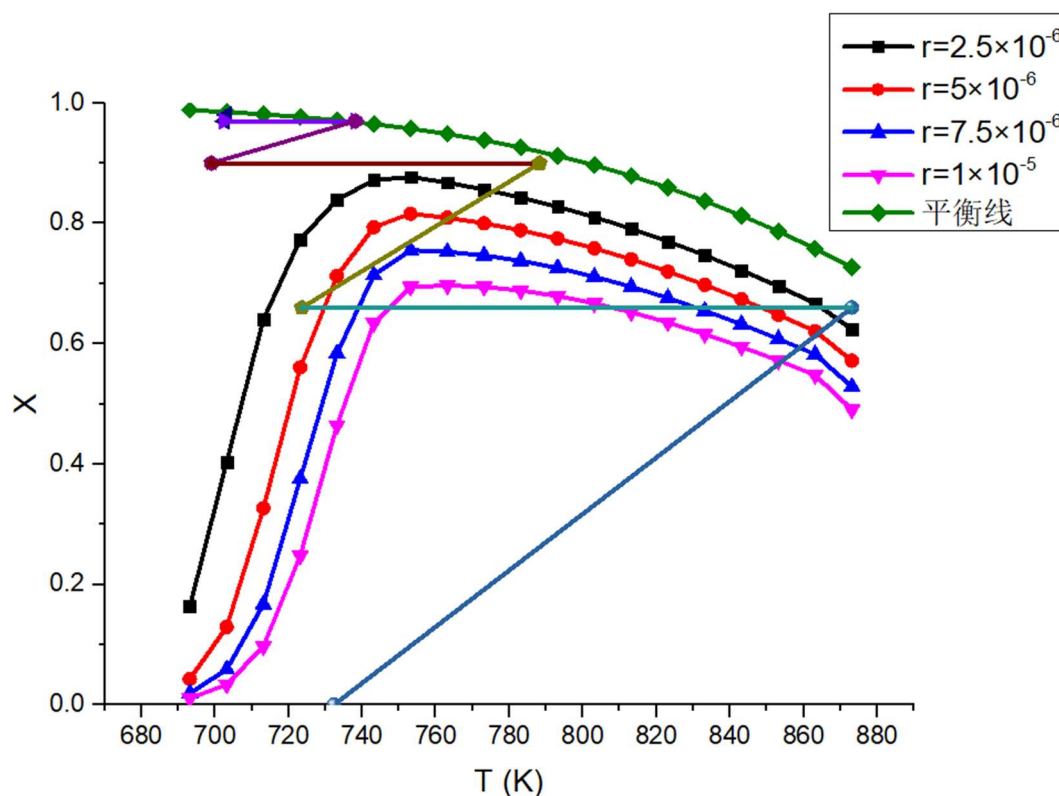


图 3 T-X 下的操作折线图

四、讨论

1、要求的最终转化率从 98%变化到 99%对催化剂用量的影响

根据动力学反应式，当平衡转化率 $X=0.99$ 时， $r=322\text{mol/gcat}\cdot\text{s}$ 对应的平衡温度 $T_e=686\text{--}687\text{K}$ ，而催化剂适用范围为 $420\text{--}600^\circ\text{C}$ 也就是 $693.15\text{--}873.15\text{K}$ ，平衡温度不在此温度范围内，因此无论如何改变催化剂用量都达不到 99%的转化率。

2、 $y_{\text{O}_2}+y_{\text{SO}_2}=21\%$ ， SO_2 进口浓度在 7—9%之间变化，对催化剂装量的影响

改变附录 3 代码里的 y_a 、 y_b 的初值进行计算催化剂装量。分别计算 $y_a=0.07$ 、 0.08 、 0.09 时的催化剂装量，数据由下表所示：

表 4 不同进口浓度二氧化硫对催化剂总量的关系

y_a	y_b	Wcat/kg
0.07	0.14	18465.83
0.08	0.13	21456.22
0.09	0.12	27814.49

由上表可知，催化剂装量随 y_a 的增大而增大，说明二氧化硫为本反应的关键组分。当二氧化硫的物质的量增大时，所需反应催化剂的量也随之升高。

五、附录

1、平衡线的 C 语言源代码

```
#include <stdio.h>
#include <math.h>
#define R 1.987f
const float ya0 =0.08;
const float yb0 =0.09;
const float p =1.01325;
float x,T;

int main(void)
{
    for (float T=693.15; T<=883.15; T+=10)
    {
        printf("%.2f ",T);
        float Kp = 2.26203 * pow(10, -5) * exp(11295.3 / T);
        //printf("%.2f ",Kp);
        for (float x=0; x<=1; x+= 0.000001)
        {
            float pa = p * (ya0 - ya0 * x) / (1 - ya0 * x / 2);
            float pc = p * ya0 * x / (1 - ya0 * x / 2);
            float pb = p * (yb0 - ya0 * (1-x) / 2) / (1 - ya0 * x / 2);
            float pe=pc/(pa*( sqrt (pb)));
            float cha=Kp-pe;
            if ( fabs(cha) <= 0.1 && fabs(cha) >= 0)
            {
                printf("%.4f\r\n",x);
                break;}
        }

    }
}
```

2、等反应速率线的 C 语言源代码

```
#include <stdio.h>
#include <math.h>
#define R 1.987f
const float ya0 =0.08;
const float yb0 =0.09;
const float p =1.01325;
```

```

float x,T, keff, B ,k ,epson,r;

int main(void)
{
    for (float T=693.15; T<=883.15; T+=10)
    {
        printf("%.2f ",T);
        float Kp = 2.26203 * pow(10, -5) * exp(11295.3 / T);
        //printf("%.2f ",Kp);
        if (T>=693.15 && T<= 748.15)
            keff= 7.6915 * pow(10, 18)* exp(-76062 / (T* R));
        else if (T>=748.15 && T<= 873.15)
            keff= 1.5128 * pow(10, 7)* exp(-35992 / (T* R));
        B=48148 *exp(-7355.5/T);
        k=2.3*pow(10, -8)*exp(27200/ (T* R));
        for (float x=0; x<=1; x+= 0.000001)
        {
            float pa = p * (ya0 - ya0 * x) / (1 - ya0 * x / 2);
            float pc = p * ya0 * x / (1 - ya0 * x / 2);
            float pb = p * (yb0 - ya0 * (1-x) / 2) / (1 - ya0 * x / 2);
            epson=pc/(Kp*pa*sqrt (pb));
            r=keff* pb*k *pa* (1-epson)*(1-epson) /pc/(sqrt(B+(B-
1)*pa/pc)+sqrt(k*pa/pc));
            //printf("%f",r);
            float cha= r-0.00001;
            if ( fabs(cha) <= 0.0000001 && fabs(cha) >= 0)
            {
                printf("%.4f\r\n",x);
                break;}}
        }

    }

}

```

3、反应具体参数进出口 C 语言源代码

```

#include<math.h>
#include<stdio.h>
#define R 1.987
#define h 0.0001
const float p =1.01325;
const float yb0 =0.09;
double r(double x, double t, double ya)
{

```

```

double r, A1, A2, A3, B, ke, K, Kp, Pa, Pc, Pb;
if (t >= 693.15 && t < 748.15)
    ke = 7.6915 * pow(10, 18) * exp(-76062 / (R * t));
if (t >= 748.15 && t <= 873.15)
    ke = 1.5128 * pow(10, 7) * exp(-35992 / (R * t));
K = 2.3 * pow(10, -8) * exp(27200 / (R * t));
Kp = 2.26203 * pow(10, -5) * exp(11295.3 / t);
Pa = p * (ya - ya * x) / (1 - ya * x / 2);
Pc = p * ya * x / (1 - ya * x / 2);
Pb = p * (yb0 - ya * (1-x) / 2) / (1 - ya * x / 2);
A1 = Pb * Pa / Pc;
A2 = Pc / (Pa * sqrt(Pb) * Kp);
B = 48148 * exp(-7355.5 / t);
A3 = sqrt(B + (B - 1) * (1 - x) / x) + sqrt(K * (1 - x) / x);
r = ke * K * A1 * (1 - A2 * A2) / (A3 * A3);
return(r);
}

double dr(double x, double t, double ya)
{
    double y;
    y = (r(x, t + h, ya) - r(x, t - h, ya)) / (2 * h);
    return(y);
}

double t(double t0, double x0, double x)
{
    double y, lamda, H = -23135, Cp = 254.9, rou = 0.500, c = 1.282;
    lamda = -H * c / (rou * Cp);
    y = t0 + lamda * (x - x0);
    return(y);
}

double fun1(double x, double t, double ya)
{
    double y;
    y = -dr(x, t, ya) / (r(x, t, ya) * r(x, t, ya));
    return(y);
}

double jifen(double x0, double t0, double ya)
{
    double sum = 0.0, x1 = x0, x2, t1, t2 = 693.15, xout;
    do {
        t1 = t(t0, x0, x1);
        x2 = x1 + h / 10;
        t2 = t(t0, x0, x2);
        if (t2 > 873.15)

```

```

    {
        xout = x1;
        goto end;
    }
    sum = sum + h * (fun1(x1, t1, ya) + fun1(x2, t2, ya)) / 20;
    x1 = x2;
}
while (sum < 0);
    xout = x1 - h / 10;
end:return(xout);
}
double wjifen(double xin, double xout, double tin, double ya)
{
    double y, x1 = xin, x2, t1, t2, sum = 0.0, wcat;
    do {
        t1 = t(tin, xin, x1);
        x2 = x1 + h;
        t2 = t(tin, xin, x2);
        if (t2 >= 873.15)
            goto end;
        sum = sum + (1 / r(x1, t1, ya) + 1 / r(x2, t2, ya)) * h / 1000;
        x1 = x2;
    }
    while (x2 <= xout);
    end:wcat = sum * 131 * 1000 / 3600;
    return(wcat);
}
void main()
{
    double ya = 0.08, xout, tout, x0 = 0.0001, t0, t00 = 719, t1, wcat,
wsum; //T=719
    int i, j;
    loop1:wsum = 0.0;
    x0 = 0.00001;
    t0 = t00;
    //printf("1 tin=%.2f xin=%e\n", t00, x0);
    for (i = 0; i <= 3; i++) //4次循环
    {
        xout = jifen(x0, t0, ya);
        tout = t(t0, x0, xout);
        printf("%d tout=%.2f,", i + 1, tout);
        printf("xout=%.2f,", xout);
        wcat = wjifen(x0, xout, t0, ya);
        printf("wcat=%.2f,", wcat);
    }
}

```

```

        wsum = wsum + wcat;
        t1 = 693.15;
        do {
            t1 = t1 + 0.01;
        } while (fabs(pow(10, 5) * r(xout, t1, ya) - pow(10, 5) *
r(xout, tout, ya)) > h);
        x0 = xout;
        t0 = xout;
        t0 = t1;
        printf(" tin=%f,xin=%f.\n", t0, xout);
    }
    printf("wsum = %.2f ", wsum);
    t00 = t00 - 0.1;
    if (x0 <= 0.98)
        goto loop1;
}

```