

# 熔解热实验参考代码

本代码适用于熔解热实验的处理，仅供参考，请勿照搬代码或数据。以老师的要求和实际情况为准。

作者：TIsolving047

版本：v3

时间：2025-03-07

- [熔解热实验参考代码](#)
- [第-1步：下载并打开matlab](#)
- [如何使用](#)
- ▼ [代码介绍](#)
  - [第0步：数据](#)
  - [第1步：拟合曲线](#)
  - [第2步：找到温度分界点](#)
  - [第3步：各种计算](#)
  - [最小二乘法函数](#)
  - [作图结果](#)
- [附：完整代码](#)

## 第-1步：下载并打开matlab

- 阅读 [关于正版化平台提供Matlab软件下载服务的通知\\_兰州大学新闻网](#) 按提示操作。不要下载R2024b版本，显示界面有大bug。
- 打开matlab，选择主页-新建实时脚本，新建.mlx格式的文件，然后将以下代码复制到代码框内。.mlx文件的输出结果与代码并排显示，支持在主代码后添加自定义函数（而.m格式的文件输出结果与代码分离，自定义函数需要另存为新文件后引用）。

## 如何使用

- 一段文本框成为“一节”。本文档共分两节，第一节为主代码，第二节为自定义函数（类似c语言的主函数和main之前的自定义函数）。选中第一节，实时编辑器-运行节，得到图像。
- 选中图像右上角在图窗窗口中打开，调整窗口大小使图表美观（不要全屏，会很丑）。
- 左上角文件-另存为，保存为.eps格式（为了清楚，以及利用latex排版页面），或.png格式（插入到word中排版页面）。或者直接保存为.pdf格式。
- 排版后打印。

# 代码介绍

## 第0步：数据

思路：输入原始数据，手动分成三段，放冰前、放冰后升温前、升温后。

t1--放冰--t2--温度到达最低点--t3，以下记作第一二三阶段，用下标的第一位数字123表示。640s-650s放冰，为一二阶段分界；960s-1080s一直都是13.8度，选择最后一个13.8度作为第三阶段的开始点。本代码使用t表示时间数组，T表示温度数组（注意三组数据不能有重复，否则代码会爆掉）。

```
clear;clf;clc; % 初始化
```

```
% 数据,行末加三点表示换行
```

```
% t=[0 120 240 360 480 600 640      650 660 670 680 690 700 710 720 730 740 ...  
% 750 760 770 780 790 800 810 820 830 840 850 860 870 880 890 900 910 920 930 940 950 ...  
% 960      1080 1165 1285 1405 1525 1645 1705];  
% T=[34.3 34.2 34.0 33.6 33.4 33.2 33.2 33.2 27.5 28.2 27.8 26.0 24.1 22.3 ...  
% 20.7 19.9 19.1 18.3 17.8 17.6 17.1 16.8 16.6 16.4 16.1 15.9 15.7 15.5 15.3 15.1 ...  
% 14.9 14.7 14.5 14.3 14.2 13.9 13.9 13.8 13.8 13.8 13.8 13.9 14.3 14.4 14.6 14.8 15.0];  
t1=[0 120 240 360 480 600 640];  
T1=[34.3 34.2 34.0 33.6 33.4 33.2 33.2];  
t2=[650 660 670 680 690 700 710 720 730 740 750 760 770 780 790 800 810 820 830 ...  
      840 850 860 870 880 890 900 910 920 930 940 950 960];  
T2=[27.5 28.2 27.8 26.0 24.1 22.3 20.7 19.9 19.1 18.3 17.8 17.6 17.1 16.8 16.6 ...  
      16.4 16.1 15.9 15.7 15.5 15.3 15.1 14.9 14.7 14.5 14.3 14.2 13.9 13.9 13.8 ...  
      13.8 13.8];  
t3=[1080 1165 1285 1405 1525 1645 1705];  
T3=[13.8 13.9 14.3 14.4 14.6 14.8 15.0];
```

```
% 从t2中舍去(650,27.5)，因为距离整条曲线太远，可能是记错了。数据好的话不用这样。
```

```
t21=[660 670 680 690 700 710 720 730 740 750 760 770 780 790 800 810 820 830 840 ...  
      850 860 870 880 890 900 910 920 930 940 950 960];  
T21=[28.2 27.8 26.0 24.1 22.3 20.7 19.9 19.1 18.3 17.8 17.6 17.1 16.8 16.6 16.4 16.1 ...  
      15.9 15.7 15.5 15.3 15.1 14.9 14.7 14.5 14.3 14.2 13.9 13.9 13.8 13.8 13.8];
```

# 第1步：拟合曲线

思路：作出原数据点以及拟合曲线。第一和第三阶段为线性拟合（最小二乘法），第二阶段为曲线拟合（三次样条法）。

给定若干个点 $(x_0, y_0)$ 可得拟合函数 $f(x)$ ，再给定一组 $x_1$ 求得 $y_1=f(x_1)$ ，则拟合曲线为 $(x_1, y_1)$ 。

使用Lagrange多项式法可能会出现Runge现象（在边缘处误差变得非常大），因此采用更为准确的三次样条spline法。

```
% 对第一段和第三段做最小二乘法并画图
[k1,b1,r1,sse1]=zuixiaoercheng(t1,T1);
[k3,b3,r3,sse3]=zuixiaoercheng(t3,T3);
figure(1);
f1=plot(t1,k1*t1+b1,'r-',t1,T1,'b+');
hold on % 在同一画布上继续做图
f3=plot(t3,k3*t3+b3,'r-',t3,T3,'b+');

% 对第二阶段的曲线进行拟合

% x0=t211,y0=T211,x1=t20,y1=T22
ta=640;tb=1080;t211=[ta t21 tb];% 第二阶段以及其两端（边界）ta和tb
T211=[ta*k1+b1 T21 tb*k3+b3];% t211对应的温度
t20=ta:0.01:tb;% x1

% spline法
pp=spline(t211,[k1,T211,k3]);%拟合函数=spline(x0,[左边界斜率,y0,右边界斜率]);
T22=ppval(pp,t20);%y1=ppval(拟合函数,x1);
% T22=spline(t211,[k1,T211,k3],t20); %若不需要拟合函数可以这样写

% 作出第二阶段的拟合曲线，标出原始数据
f2=plot(t20,T22,'r-',t21,T21,'b+',650,27.5,'b+');

% 美化图表,需要人工确定各种参数
axis([-10 1740 12 35]); % 坐标范围axis([xmin xmax ymin ymax]);
grid on % 背景网格
grid minor % 背景小网格
set(gca,'Xtick',0:60:1740)% x轴刻度
set(gca,'Ytick',12:1:35)% y轴刻度
xlabel('时间t(s)');%x轴标题
ylabel('温度T(^{\circ}C)');%y轴标题(^{\circ}C)是摄氏度
title('熔解热曲线');
```

## 第2步：找到温度分界点

思路：利用matlab提供的polyarea()函数计算 $A1=S1+S2$ 和 $A2=S3$ ，通过for循环，找到绝对值最小的 $\Delta A=A2-A1$ ，据此不断用while循环缩小范围，确定时间 $tx0$ （和对应的 $\Delta A$ ）。

例如第一次循环， $t$ 为640,650,...1080，找到绝对值最小的 $\Delta A$ 在 $t=720$ 处；第二次循环的 $t$ 则为710,711,...730，以此类推。

```
% 初始变量设定
format long; % 设定数据显示方式为long型，显示的小数位数更多
ii=1;% 循环变量
deltat=0.01;% 面积精度
deltatx=10;% 用来找tx,先设定为10
txleft=ta;txright=tb;n=ceil((txright-txleft)/deltatx)+1;
% 循环结果为记录在数组A中，A的第一列为tx值，第二列为对应的deltaA
A(:,1)=txleft:deltatx:txright;
A(:,2)=zeros(n,1);% 设置为全0数组相当于初始化，提升运行速度

while(deltatx>1e-4)% 小数点后太多位数没用，拟合曲线与实际曲线的误差远比这点位数大
    for tx=txleft:deltatx:txright
        % 使用求得的pp()函数确定A1和A2的形状
        A1x=ta:deltat:tx; A1y=ppval(pp,A1x);
        A2x=tx:deltat:tb; A2y=ppval(pp,A2x);
        % 使用polyarea()计算A1和A2
        A1 = polyarea([A1x tx],[A1y tx*k1+b1]);
        A2 = polyarea([A2x tx],[A2y tx*k3+b3]);
        A(ii,2)=A2-A1;
        ii=ii+1;
    end
    % 找到abs(deltaA)最小的位置，确定下一次循环的参数，[最小值，最小值的位置]=min(一维列向量)
    % deltax除以10后，以tx0加减deltatx作为下一次的范围，直接分成20份
    [~,i0]=min(abs(A(:,2)));tx0=A(i0,1);deltaA=A(i0,2);
    txleft=tx0-deltatx;txright=tx0+deltatx;
    deltax=deltatx/10;n=21;
    clear A
    A(:,1)=txleft:deltatx:txright;A(:,2)=zeros(n,1);
    ii=1;
end
% tx0,deltaA
```

## 第3步：各种计算

最困难的步骤已经结束了，接下来就是各种计算。

思路：首先计算修正后的面积差 $\Delta A_{fix}$ ，修正后（在 $t_{x0}$ 处）的温度 $T_2'$ 和 $T_3'$ ；

然后是代入各种数据计算修正前和修正后的熔解热；

最后是添加辅助线，标注 $S_1 \sim S_5$ 等。根据需求自行标记 $T_2, T_2'$ 等

```
%计算修正后的面积差deltaAfix，不要求
theta=24.5;%室温
A1fix=polyarea([ta ta tx0 tx0],[ta*k1+b1 theta theta tx0*k1+b1]);% S1+S4
A2fix=polyarea([tx0 tx0 tb tb],[theta tx0*k3+b3 tb*k3+b3 theta]);% S3+S5
deltaAfix=A2fix-A1fix;

c_water=4.18;c_container=0.389;c_ice=1.80;% 比热容(J/(g*K))
m_water=147.370;m_container=139.43;m_ice=34.935;%质量(g)
T_0=0;T_1=-4;T_2=k1*ta+b1;T_3=k3*tb+b3;% 书中的T0,T1,T2和T3
T_2fix=tx0*k1+b1;T_3fix=tx0*k3+b3;% 修正后的T2和T3

Lorigin=c_ice*(T_1-T_0)-(c_water*m_ice*(T_3-T_0)+ ...
(c_water*m_water+c_container*m_container)*(T_3-T_2))/m_ice;% 原始熔解热
Lfix=c_ice*(T_1-T_0)-(c_water*m_ice*(T_3fix-T_0)+ ...
(c_water*m_water+c_container*m_container)*(T_3fix-T_2fix))/m_ice;% 修正熔解热
% K=Lfixed*m_ice/deltaAfix % 系数K,不要求

% 整理结果，根据需求增减
result=[k1,b1,k3,b3,tx0,deltaA,deltaAfix,T_2,T_3,T_2fix,T_3fix,Lorigin,Lfix];

%辅助线
ftheta=plot([-10 1750],theta.*[1 1],'k--');
f11=plot(ta*[1 1],[T_2,theta],'k--');
f41=plot(tb*[1 1],[theta,13.8],'k--');
fleft=plot([ta,tx0],[ta,tx0].*k1+b1,'r--');
fright=plot([tx0,tb],[tx0,tb].*k3+b3,'r--');
fx=plot(tx0*[1 1],[T_2fix,T_3fix],'r--');

% 标注S1-S5,坐标需要手动确定
text(670,30,'S1');
text(690,23.5,'S2');
text(780,15,'S3');
text(645,26,'S4');
text(900,20,'S5');
```

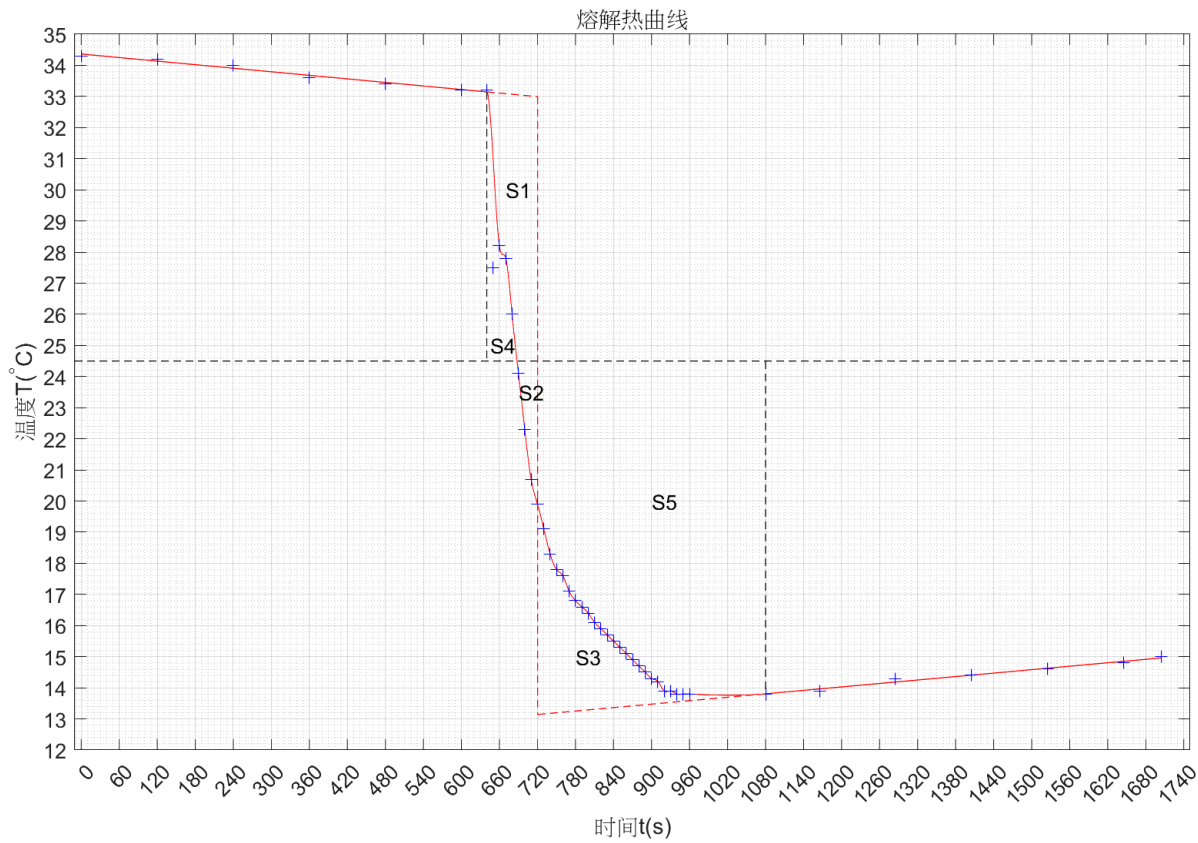
# 最小二乘法函数

```
function[k,b,r,sse]=zuixiaoercheng(x,y)
% x y 输入的x和y数组
% k 斜率
% b 截距
% r 相关系数，绝对值越接近1说明越线性
% sse 均方误差，预测值与真实值之差的平方和,(Sum of Squares of Errors, SSE)

if numel(x)==numel(y)
    n=numel(x);
else
    error('Please check the number of x and y')
end
xbar=sum(x)/n;
ybar=sum(y)/n;

k=(sum(x.*y)-n*xbar*ybar)/(sum(x.*x)-n*xbar*xbar);
b=ybar-k*xbar;
y_0=k.*x+b;
r=(sum(x.*y)-n*xbar*ybar)/sqrt(sum(x.*x)-n*xbar*xbar)/sqrt(sum(y.*y)-n*ybar*ybar);
sse=sum((y-y_0).^2);
end
```

# 作图结果



# 附：完整代码

```
clear;clf;clc;

%-----step0-----

% 数据,行末加三点表示换行
% t=[0 120 240 360 480 600 640      650 660 670 680 690 700 710 720 730 740 ...
% 750 760 770 780 790 800 810 820 830 840 850 860 870 880 890 900 910 920 930 940 950 ...
% 960      1080 1165 1285 1405 1525 1645 1705];
% T=[34.3 34.2 34.0 33.6 33.4 33.2 33.2 33.2 27.5 28.2 27.8 26.0 24.1 22.3 ...
% 20.7 19.9 19.1 18.3 17.8 17.6 17.1 16.8 16.6 16.4 16.1 15.9 15.7 15.5 15.3 15.1 ...
% 14.9 14.7 14.5 14.3 14.2 13.9 13.9 13.8 13.8 13.8 13.8 13.9 14.3 14.4 14.6 14.8 15.0];
t1=[0 120 240 360 480 600 640];
T1=[34.3 34.2 34.0 33.6 33.4 33.2 33.2];
t2=[650 660 670 680 690 700 710 720 730 740 750 760 770 780 790 800 810 820 830 ...
      840 850 860 870 880 890 900 910 920 930 940 950 960];
T2=[27.5 28.2 27.8 26.0 24.1 22.3 20.7 19.9 19.1 18.3 17.8 17.6 17.1 16.8 16.6 ...
      16.4 16.1 15.9 15.7 15.5 15.3 15.1 14.9 14.7 14.5 14.3 14.2 13.9 13.9 13.8 ...
      13.8 13.8];
t3=[1080 1165 1285 1405 1525 1645 1705];
T3=[13.8 13.9 14.3 14.4 14.6 14.8 15.0];

% 从t2中舍去(650,27.5),因为距离整条曲线太远,可能是记错了。数据好的话不用这样。
t21=[660 670 680 690 700 710 720 730 740 750 760 770 780 790 800 810 820 830 840 ...
      850 860 870 880 890 900 910 920 930 940 950 960];
T21=[28.2 27.8 26.0 24.1 22.3 20.7 19.9 19.1 18.3 17.8 17.6 17.1 16.8 16.6 16.4 16.1 ...
      15.9 15.7 15.5 15.3 15.1 14.9 14.7 14.5 14.3 14.2 13.9 13.9 13.8 13.8 13.8];

%-----step1-----

% 对第一段和第三段做最小二乘法并画图
[k1,b1,r1,sse1]=zuixiaoercheng(t1,T1);
[k3,b3,r3,sse3]=zuixiaoercheng(t3,T3);
figure(1);
f1=plot(t1,k1*t1+b1,'r-',t1,T1,'b+');
hold on % 在同一画布上继续做图
f3=plot(t3,k3*t3+b3,'r-',t3,T3,'b+');

% 对第二阶段的曲线进行拟合

% x0=t211,y0=T211,x1=t20,y1=T22
ta=640;tb=1080;t211=[ta t21 tb];% 第二阶段以及其两端(边界)ta和tb
```



```

T211=[ta*k1+b1 T21 tb*k3+b3];% t211对应的温度
t20=ta:0.01:tb;% x1

% spline法
pp=spline(t211,[k1,T211,k3]);%拟合函数=spline(x0,[左边界斜率,y0,右边界斜率]);
T22=ppval(pp,t20);%y1=ppval(拟合函数,x1);
% T22=spline(t211,[k1,T211,k3],t20); %若不需要拟合函数可以这样写

% 作出第二阶段的拟合曲线，标出原始数据
f2=plot(t20,T22,'r-',t21,T21,'b+',650,27.5,'b+');

% 美化图表,需要人工确定各种参数
axis([-10 1740 12 35]); % 坐标范围axis([xmin xmax ymin ymax])
grid on % 背景网格
grid minor % 背景小网格
set(gca,'Xtick',0:60:1740)% x轴刻度
set(gca,'Ytick',12:1:35)% y轴刻度
xlabel('时间t(s)');%x轴标题
ylabel('温度T(^{\circ}C)');%y轴标题(^{\circ}C)是摄氏度
title('熔解热曲线');

%-----step2-----

% 初始变量设定
format long; % 设定数据显示方式为long型，显示的小数位数更多
ii=1;% 循环变量
deltat=0.01;% 面积精度
deltatx=10;% 用来找tx,先设定为10
txleft=ta;txright=tb;n=ceil((txright-txleft)/deltatx)+1;
% 循环结果为记录在数组A中，A的第一列为tx值，第二列为对应的deltaA
A(:,1)=txleft:deltatx:txright;
A(:,2)=zeros(n,1);% 设置为全0数组相当于初始化，提升运行速度

while(deltatx>1e-4)% 小数点后太多位数没用，拟合曲线与实际曲线的误差远比这点位数大
    for tx=txleft:deltatx:txright
        % 使用求得的pp()函数确定A1和A2的形状
        A1x=ta:deltat:tx; A1y=ppval(pp,A1x);
        A2x=tx:deltat:tb; A2y=ppval(pp,A2x);
        % 使用polyarea()计算A1和A2
        A1 = polyarea([A1x tx],[A1y tx*k1+b1]);
        A2 = polyarea([A2x tx],[A2y tx*k3+b3]);
        A(ii,2)=A2-A1;
        ii=ii+1;
    end
    % 找到abs(deltaA)最小的位置，确定下一次循环的参数
    % deltatx除以10后，以tx0加减deltatx作为下一次的范围，直接分成20份

```

```

[~,i0]=min(abs(A(:,2)));tx0=A(i0,1);deltaA=A(i0,2);
txleft=tx0-deltatx;txright=tx0+deltatx;
deltatx=deltatx/10;n=21;
clear A
A(:,1)=txleft:deltatx:txright;A(:,2)=zeros(n,1);
ii=1;
end
% tx0,deltaA

%-----step3-----

%计算修正后的面积差deltaAfix, 不要求
theta=24.5;%室温
A1fix=polyarea([ta ta tx0 tx0],[ta*k1+b1 theta theta tx0*k1+b1]);% S1+S4
A2fix=polyarea([tx0 tx0 tb tb],[theta tx0*k3+b3 tb*k3+b3 theta]);% S3+S5
deltaAfix=A2fix-A1fix;

c_water=4.18;c_container=0.389;c_ice=1.80;% 比热容(J/(g*K))
m_water=147.370;m_container=139.43;m_ice=34.935;%质量(g)
T_0=0;T_1=-4;T_2=k1*ta+b1;T_3=k3*tb+b3;% 书中的T0,T1,T2和T3
T_2fix=tx0*k1+b1;T_3fix=tx0*k3+b3;% 修正后的T2和T3

Lorigin=c_ice*(T_1-T_0)-(c_water*m_ice*(T_3-T_0)+ ...
(c_water*m_water+c_container*m_container)*(T_3-T_2))/m_ice;% 原始熔解热
Lfix=c_ice*(T_1-T_0)-(c_water*m_ice*(T_3fix-T_0)+ ...
(c_water*m_water+c_container*m_container)*(T_3fix-T_2fix))/m_ice;% 修正熔解热
% K=Lfixed*m_ice/deltaAfix % 系数K,不要求

% 整理结果, 根据需求增减
result=[k1,b1,k3,b3,tx0,deltaA,deltaAfix,T_2,T_3,T_2fix,T_3fix,Lorigin,Lfix];

%辅助线
ftheta=plot([-10 1750],theta.*[1 1],'k--');
f11=plot(ta*[1 1],[T_2,theta],'k--');
f41=plot(tb*[1 1],[theta,13.8],'k--');
fleft=plot([ta,tx0],[ta,tx0].*k1+b1,'r--');
fright=plot([tx0,tb],[tx0,tb].*k3+b3,'r--');
fx=plot(tx0*[1 1],[T_2fix,T_3fix],'r--');

% 标注S1-S5,坐标需要手动确定
text(670,30,'S1');
text(690,23.5,'S2');
text(780,15,'S3');
text(645,26,'S4');
text(900,20,'S5');

```

% 此处应分节

```
function[k,b,r,sse]=zuixiaoercheng(x,y)
% x y 输入的x和y数组
% k 斜率
% b 截距
% r 相关系数，绝对值越接近1说明越线性
% sse 均方误差，预测值与真实值之差的平方和,(Sum of Squares of Errors, SSE)

if numel(x)==numel(y)
    n=numel(x);
else
    error('Please check the number of x and y')
end
xbar=sum(x)/n;
ybar=sum(y)/n;

k=(sum(x.*y)-n*xbar*ybar)/(sum(x.*x)-n*xbar*xbar);
b=ybar-k*xbar;
y_0=k.*x+b;
r=(sum(x.*y)-n*xbar*ybar)/sqrt(sum(x.*x)-n*xbar*xbar)/sqrt(sum(y.*y)-n*ybar*ybar);
sse=sum((y-y_0).^2);
end
```