同濟大學

计算方法实验报告（一）

学院 机械与能源工程学院

专业 机械设计制造及其自动化

学号 1852951

姓名 李腾

指导教师 李梦茹、陈茂林

完成日期 2020 年 11 月 05 日

目录

一、 实验一

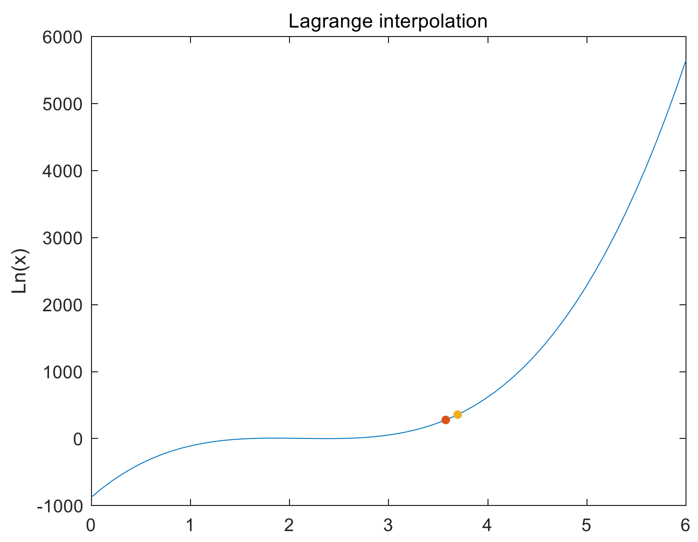
1.1 实验 1

1.1.1 实验结果

Lagrange 插值公式：$Ln(x) = 97.5x^3 - 618.2x^2 + 1286.3x - 873.9$

对应点坐标：$[x_i] = [x_1, x_2] = [3.58,3.70]$，$[y_i] = [y_1, y_2] = [278.73,357.97]$

1.1.2 曲线和数据点的绘制



1.1.3 程序数据截图

名称	值
a	[1.6263,1.7150,2.5057,2.2391]
ans	873.8595
b	[2.3851,5.6901,1.6762,1.4083]
i	4
k	4
l	*4x1 sym*
L	*1x1 sym*
L1	*4x4 double*
m	4
V	[-11.6790,68.2873,-130.3544,81.6208]
x	*1x60 double*
X	[1.6263,1.7150,2.5057,2.2391]
x0	[3.5757,3.6962]
y	*1x60 double*
Y	[2.3851,5.6901,1.6762,1.4083]
y0	[278.7301,357.9697]

1.1.4 程序源代码

文件 1：坐标生成与图像绘制

```
%1852951  李腾
clear;
rand(’seed’,1852951);
a = 1+5*rand(1,4);
b = 1+5*rand(1,4);
x = linspace(0,6,60);
y = lagrange_generate(a,b,x);
%曲线绘制
plot(x,y);
ylabel(’Ln(x)’);
title(’Lagrange interpolation’);
hold on
%描点
x0 = 2 + 3*rand(1,2);
y0 = lagrange_generate(a,b,x0);
for i = 1:2
    plot(x0(1,i),y0(1,i),’.’,’MarkerSize’,14);
end
```

文件 2：拉格朗日插值对应坐标计算

```
function y=lagrange_generate(x0,y0,x)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%函数功能：生成拉格朗日插值公式%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = length(x0);
m = length(x);
y = zeros(1,m);
for i=1:m
    z=x(i);s=0;
    for k=1:n
        L=1;
        for j=1:n
            if j~=k
                L=L*(z-x0(j))/(x0(k)-x0(j));
            end
        end
        s=s+L*y0(k);
    end
    y(i)=s;
end
```

文件 3：拉格朗日公式系数计算

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%该脚本用于求解拉格朗日多项式及基函数
%输入量:   n+1 个节点(x_i,y_i)(i = 1,2, ... , n+1)横坐标向量 X，纵坐标向量 Y
%输出量：n 次拉格朗日插值多项式 L 和基函数 l
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
X=input('请输入横坐标向量 X:\nX=');
Y=input('请输入纵坐标向量 Y:\nY=');
m = length(X);
```

```
L = ones(m,m);
for k = 1 : m
    V = 1;
    for i = 1 : m
        if k ~= i
            V = conv(V,poly(X(i))) / (X(k) - X(i));
        end
    end
    L1(k, :) = V;
    l(k, :) = poly2sym(V);
end
fprintf('基函数为：\n');
for k=1:m
    fprintf('q%d(x)=%s\n',k,l(k));
end
L = Y * l;
fprintf('拉格朗日多项式为:\nP(x)=%s\n',L);
```

1.2 实验 2

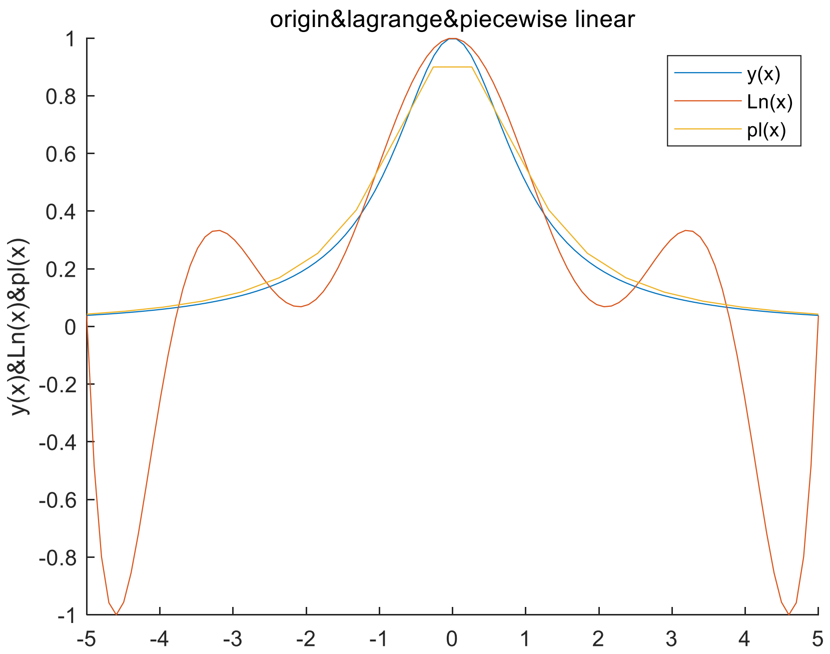
1.2.1 实验结果与分析

Lagrange 插值公式：

$$Ln(x) = 1.37e^{-4}\,x^8 - 7.75e^{-19}\,x^7 - 6.60e^{-3}\,x^6 - 5.37e^{-17}\,x^5 + 0.10\,x^4 - 6.66e^{-17}\,x^3 - 0.53\,x^2 + 1.33e^{-16}x - 873.9$$

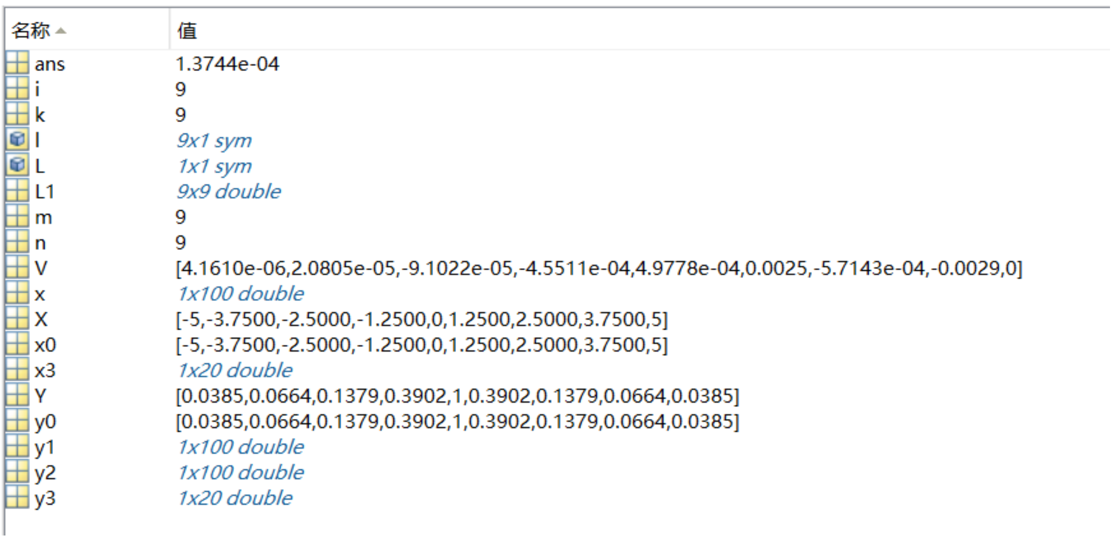
绘制曲线，发现在横坐标两端偏离较为严重，即产生了过拟合；因此进行分段插值，并绘制对应曲线$pl(x)$。

1.2.2 曲线绘制



分析：一般情况下，多项式的次数越多，需要的数据就越多，而预测也就越准确；插值次数越高，插值结果越偏离原函数。为了解决 Runge 现象，可采取分段线性插值，使得用分段线性函数进行拟合，避免了过拟合，但是函数失去了平滑性，即出现了不可导点。

1.2.3 程序结果截图



名称	值
ans	1.3744e-04
i	9
k	9
l	9x1 sym
L	1x1 sym
L1	9x9 double
m	9
n	9
V	[4.1610e-06,2.0805e-05,-9.1022e-05,-4.5511e-04,4.9778e-04,0.0025,-5.7143e-04,-0.0029,0]
x	1x100 double
X	[-5,-3.7500,-2.5000,-1.2500,0,1.2500,2.5000,3.7500,5]
x0	[-5,-3.7500,-2.5000,-1.2500,0,1.2500,2.5000,3.7500,5]
x3	1x20 double
Y	[0.0385,0.0664,0.1379,0.3902,1,0.3902,0.1379,0.0664,0.0385]
y0	[0.0385,0.0664,0.1379,0.3902,1,0.3902,0.1379,0.0664,0.0385]
y1	1x100 double
y2	1x100 double
y3	1x20 double

1.2.4 程序源代码

文件 1：坐标生成与图像绘制

```
clear;
```

```
rand('seed',1852951);
%construct coordinates
n = randi([8,20],1,1);
x0 = linspace(-5,5,n);
y0 = 1./(1+x0.^2);
x = linspace(-5,5,100);
y1 = 1./(1+x.^2);
y2 = lagrange_generate(x0,y0,x);
x3 = linspace(-5,5,20);
y3 = pl(20,-5,5);
%draw
plot(x,y1);
hold on
plot(x,y2);
hold on
plot(x3,y3);
ylabel('y(x)&Ln(x)&pl(x)');
legend('y(x)','Ln(x)','pl(x)');
title('origin&lagrange&piecewise linear');
%draw additional points
hold on
```

文件 2：分段线性插值坐标计算

```
function Sn=pl(L,b1,b2)
%当在区间内取 i 个等距节点时对应的小区间的中点值 Si 并绘制出图形
%b1 代表左边界，b2 代表右边界
%L 可以是一个数组，也可以是一个数字
%插值绘图
n=length(L);
for i=1:n
    s=L(i);
    L1=linspace(b1,b2,s+1);
    for j=2:s+1
```

```
        x(j-1)=(L1(j-1)+L1(j))/2; %寻找两端点中点值
        Sn(j-1) = (((x(j-1)-L1(j))/(L1(j-1)-L1(j)))/(1+L1(j-1)^2))+...
            (((x(j-1)-L1(j-1))/(L1(j)-L1(j-1)))/(1+L1(j)^2)); %中点值函数值
    end
    hold on
end
```

文件 3：拉格朗日公式系数计算

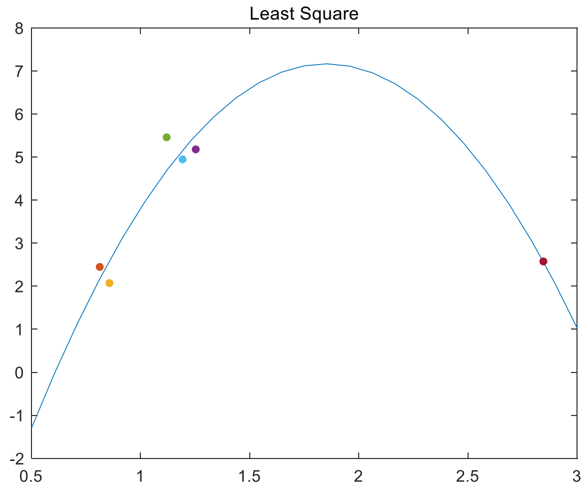
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%该脚本用于求解拉格朗日多项式及基函数
%输入量:   n+1 个节点(x_i,y_i)(i = 1,2, ... , n+1)横坐标向量 X，纵坐标向量 Y
%输出量：n 次拉格朗日插值多项式 L 和基函数 l
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

X=input('请输入横坐标向量 X:\nX=');
Y=input('请输入纵坐标向量 Y:\nY=');
m = length(X);
L = ones(m,m);
for k = 1 : m
    V = 1;
    for i = 1 : m
        if k ~= i
            V = conv(V,poly(X(i))) / (X(k) - X(i));
        end
    end
    L1(k, :) = V;
    l(k, :) = poly2sym(V);
end
fprintf('基函数为：\n');
for k=1:m
    fprintf('q%d(x)=%s\n',k,l(k));
end
L = Y * l;
```

```
fprintf('拉格朗日多项式为:\nP(x)=%s\n',L);
```

二、实验二

2.1 实验 1

2.1.1 曲线和数据点的绘制



2.2.2 程序数据截图

名称	值
a	[0.8131,0.8575,1.2528,1.1196,1.1926,2.8450]
b	[2.4467,2.0716,5.1766,5.4577,4.9457,2.5736]
i	6
p	[-4.6449,17.1883,-8.7364]
x	*1x25 double*
y	*1x25 double*

2.2.3 程序源代码

```
clear;

rand('seed',1852951);

%points

a = 0.5+2.5*rand(1,6);

b = 1.5+7*rand(1,6);
```

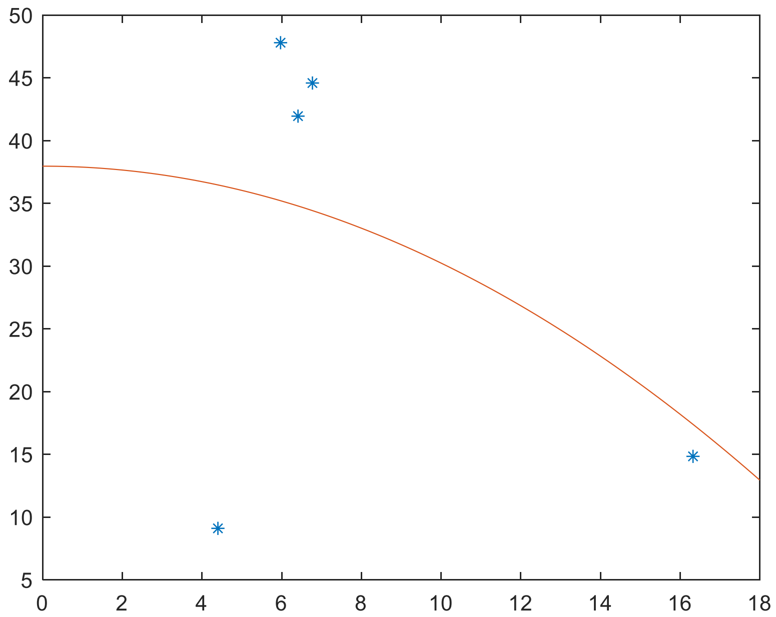
```
%fit
p = polyfit(a,b,2);
%construct coordinates
x = linspace(0.5,3,25);
y = polyval(p,x);
%draw
plot(x,y);
ylabel('');
title('Least Square');
hold on
%draw additional points
for i = 1:6
    plot(a(i),b(i),'.','MarkerSize',14);
    hold on
end
```

2.2 实验 2

2.2.1 实验结果

拟合经验公式：$y = 37.96 - 0.077x^2$

2.2.2 曲线绘制



3. 程序结果截图

名称	值
a	[4.3995,6.7717,5.9720,6.4099,16.3247]
b	[9.1028,44.5882,47.8008,41.9492,14.8389]
fun	*1x1 inline*
n	[37.9625,-0.0772]
n0	[1,1]
x	*1x100 double*

4. 程序源代码

```
clear;

rand(’seed’,1852951);

%init

a = 18*rand(1)+15*rand(1,5);

b = 19*rand(1)+80*rand(1,5);

x = linspace(0,18,100);

fun = inline(’n(1)+n(2)*x.^2’,’n’,’x’);

n0 = [1,1];

n=nlinfit(a,b,fun,n0);

plot(a,b,’*’);

hold on

plot(x,n(1)+n(2)*x.^2);
```

2.3 实验 3

2. 3. 1 实验结果

采用 0. 01 为 w 的最小单位精度，

矩形求积法误差：$loss\_rec = 2.01e^{-4}$

梯形求积法误差：$loss\_tra = 9.45e^{-4}$

因此，矩形求积法的精度更高。

2. 3. 2 程序数据截图

名称	值
a	2
b	6
f	@(t)exp(-0.5*t).*(t+pi/b)
loss_rec	2.0110e-04
loss_tra	9.4450e-04
s	4.2860
s_rec	4.2862
s_tra	4.2851

2. 3. 3 程序源代码

文件 1：坐标生成与误差计算

```
clear;
rand('seed',1852951);
%init
a = randi(10,1,1);
b = randi([5 15],1,1);
%calculate sum
% y = exp(-0.5*t)*(t+pi/b);
s_rec = rec_sum(0,a*pi,b);%rec
s_tra = tra_sum(0,a*pi,b);%tra
%calculate loss
f = @(t) exp(-0.5*t).*(t+pi/b);
s = integral(f,0,a*pi);
loss_rec = abs(s-s_rec);
```

```
loss_tra = abs(s-s_tra);
```

文件 2：矩形法求定积分

```
function rec_sum = rec_sum(floor,ceiling,b)
%init
s = 0;
i = 1;
w = 0.01; %矩形的宽度为0.01
n = (ceiling-floor)/w; %份数
t = floor;
y = exp(-0.5*t)*(t+pi/b);
while i < n
    s = s+w*y;
    t = t+w;
    y = exp(-0.5*t)*(t+pi/b);
    i = i+1;
end
rec_sum = s;
end
```

文件 3：梯形法求定积分

```
function tra_sum = tra_sum(floor,ceiling,b)
%init
s = 0;
i = 1;
w = 0.01; %梯形的宽度为0.01
n = (ceiling-floor)/w; %份数
t = floor;
y1 = exp(-0.5*t).*(t+pi/b);
y2 = exp(-0.5*(t+w)).*((t+w)+pi/b);
while i < n
    s = s+w*(y1+y2)/2;
    t = t+w;
    y1 = y2;
```

```
    y2 = exp(-0.5*(t+w)).*((t+w)+pi/b);
    i = i+1;
end
tra_sum = s;
end
```

2.4 实验 4

2.4.1 实验结果

积分值：0.1332

2.4.2 程序数据截图

名称	值
a	0.3131
b	0.4290
fun	@(x)x^b
s	0.1332

2.4.3 程序源代码

文件 1：数据初始化与求积计算

```
clear;
rand('seed',1852951);
%init
a = 2.5*rand(1,1);
b = 3*rand(1,1);
%calculate sum
fun = @(x) x^b;
[s,~,~] = romberg(fun,0,a,1e-5);
```

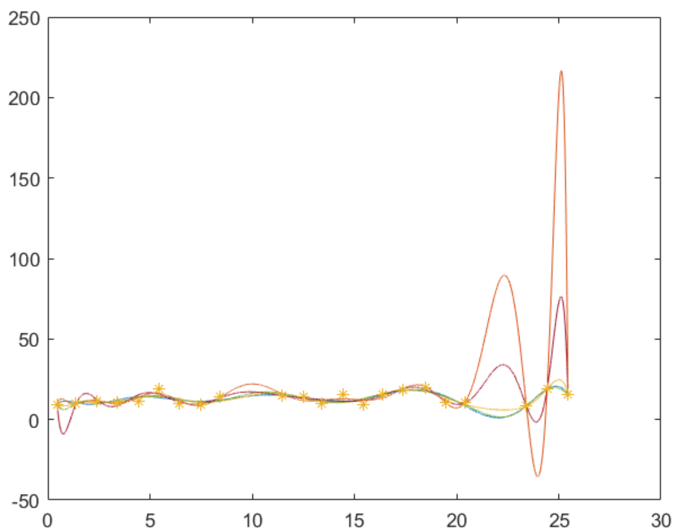
文件 2：龙贝格(Romberg 数值求解公式)

```
function [R,k,T]=romberg(fun,a,b,tol)
k=0; % 迭代次数
n=1; % 区间划分个数
h=b-a;
T=h/2*(fun(a)+fun(b));
```

```
err=1;
while err>=tol
    k=k+1;
    h=h/2;
    tmp=0;
    for i=1:n
        tmp=tmp+fun(a+(2*i-1)*h);
    end
    T(k+1,1)=T(k)/2+h*tmp;
    for j=1:k
        T(k+1,j+1)=T(k+1,j)+(T(k+1,j)-T(k,j))/(4^j-1);
    end
    n=n*2;
    err=abs(T(k+1,k+1)-T(k,k));
end
R=T(k+1,4);
```

2.5 实验 5

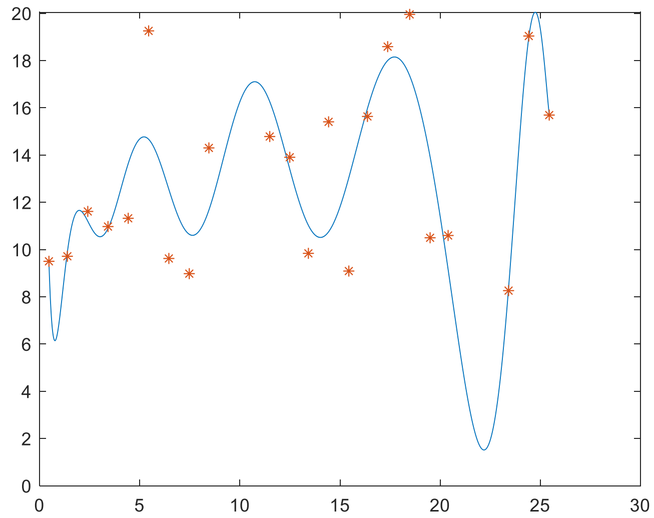
2.5.1 实验结果



以上为 12、13、14、15、16 阶时的拟合图像曲线；综合分析图像可知，在 14 阶时拟合图像最为理想，由此取 14 阶作为进一步分析。

绘制 14 阶图像，可求得 24 小时总用水量：$2.93 \times 10^5 m^3$。

2.5.2 拟合曲线绘制



2.5.3 程序数据截图

名称	值
a	22x1 double
ans	292.7086
b	1x15 double
d	14
h	22x1 double
i	15
k	1x1 sym
t	1x1 sym
x	1x24991 double
y	1x24991 double

2.5.4 程序源代码

```
clear;
rand('seed',1852951)
a=8+12*rand(22,1);d=14;    %改变 d 值即可改变最高次数
h =...
[0.46,1.38,2.40,3.41,4.42,5.44,6.45,7.47,8.45,11.49,12.49,13.42,14.43,15.44,...
    16.37,17.38,18.48,19.50,20.40,23.42,24.43,25.45]';
b=polyfit(h,a,d);
x=0.46:0.001:25.45;
```

```
y=0;k=0;

syms t

for i=1:(d+1)

   y=y+b(i).*x.^(d+1-i);

end

plot(x,y)

hold on

plot(h,a,'*')

for i=1:(d+1)

   k=k+b(i)*t^(d+1-i);

end

double(int(k,0.45,24.45))
```