1、学号平均值483，根据公式 lnb=lnB+nln10，我们计算 lnb 约为 ln4.83+2×ln10。

为了使用抛物线插值法计算 ln(4.83)，我们从表中选取了三个最接近4.83的 *b* 值及其对应的ln(*b*) 值。这三个 *b* 值分别是4.7、4.8和4.9，对应的ln(*b*) 值分别是1.5476、1.5686和1.5892。

使用这三个点，我们可以拟合一个抛物线（二次函数），并在 *b*=4.83 处插值来

估计 ln(4.83) 的值。

插值公式如下：

**P(b)=a⋅b^2+b⋅b+c**

其中*a*，*b*，和 *c* 是通过二次拟合找到的系数。

抛物线插值公式的系数是通过二次多项式拟合来确定的。

具体来说，我们使用了三个点 (*b*0​,ln(*b*0​)), (*b*1​,ln(*b*1​)), (*b*2​,ln(*b*2​)) 来拟合然后，我们在 *b*=4.83 处计算 *P*(*b*) 的值。

a=−0.02

b=0.4

c=0.1094

通过插值得到的 ln(4.83) 的估计值约为 1.5748。带入ln4.83+2×ln10得6.18

2、对于*b*=440: ln(*b*)=6.0868

对于 *b*=450: ln(*b*)=6.1092

对于 *b*=460: ln(*b*)=6.1312

对于 *b*=470: ln(*b*)=6.1527

对于*b*=480: ln(*b*)=6.1738

对于 *b*=490: ln(*b*)=6.1944

对于*b*=500: ln(*b*)=6.2146

对于 *b*=510: ln(*b*)=6.2344

对于*b*=520: ln(*b*)=6.2538

对于*b*=530: ln(*b*)=6.2729

代码：

% 定义b值和对应的ln(b)值

b\_values = [440, 450, 460, 470, 480, 490, 500, 510, 520, 530];

lnb\_values = arrayfun(@compute\_ln, b\_values);

%lnb\_values = [6.0868, 6.1092, 6.1312, 6.1527, 6.1738, 6.1944, 6.2146, 6.2344, 6.2538, 6.2729]

% 定义插值范围

b\_interp = linspace(min(b\_values), max(b\_values), 1000);

% 进行拉格朗日插值

lnb\_interp = zeros(size(b\_interp));

for i = 1:length(b\_interp)

lnb\_interp(i) = lagrange\_interpolation(b\_values, lnb\_values, b\_interp(i));

end

% 绘制插值图形

plot(b\_interp, lnb\_interp);

hold on;

% 标记学号平均值的点

average\_b = 483;

average\_lnb = lagrange\_interpolation(b\_values, lnb\_values, average\_b);

plot(average\_b, average\_lnb, 'ro'); % 使用红色圆圈标记

text(average\_b, average\_lnb, sprintf('(%.2f, %.2f)', average\_b, average\_lnb));

% 设置图形的标题和轴标签

title('拉格朗日多项式插值');

xlabel('b');

ylabel('ln(b)');

hold off;

% 定义拉格朗日插值函数

function result = lagrange\_interpolation(x, y, x\_new)

total = 0;

n = length(x);

for i = 1:n

xi = x(i);

yi = y(i);

prod = yi;

for j = 1:n

if i ~= j

xj = x(j);

prod = prod \* (x\_new - xj) / (xi - xj);

end

end

total = total + prod;

end

result = total;

end

% 定义一个函数，根据b的值计算ln(b)

function ln\_val = compute\_ln(b)

if b > 10

n = floor(log10(b)); % 获取10的幂次

B = b / 10^n; % 获取B的值

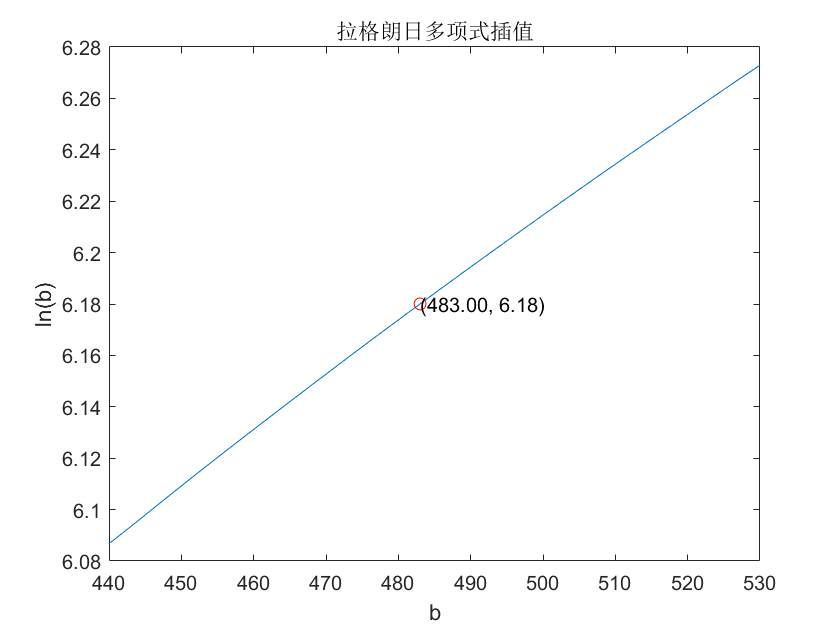
ln\_val = log(B) + n \* log(10);

else

ln\_val = log(b);

end

end



3、分段线性插值代码：

% 数据集中的b值，确保范围涵盖了学号值

b\_values = linspace(1, 600, 1000); % 创建一个充分密集的b值范围

% 计算数据集中b值的ln(b)值

lnb\_values = arrayfun(@compute\_ln, b\_values);

% 插值范围

b\_interp\_range = linspace(250, 600, 100);

% 插值计算

interp\_values = interp1(b\_values, lnb\_values, b\_interp\_range, 'linear');

% 学号数据

student\_numbers = [259, 595, 596];

student\_ln\_values = arrayfun(@compute\_ln, student\_numbers);

student\_points = [student\_numbers; student\_ln\_values];

student\_points

% 绘制插值图形

figure;

plot(b\_interp\_range, interp\_values, 'b-', 'LineWidth', 2, 'DisplayName', 'Segmented Linear Interpolation');

hold on;

scatter(student\_numbers, student\_ln\_values, 50, 'r', 'filled', 'DisplayName', 'Student Numbers');

for i = 1:numel(student\_numbers)

text(student\_numbers(i), student\_ln\_values(i), sprintf('(%d, %.2f)', student\_numbers(i), student\_ln\_values(i)), 'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'right', 'Color', 'r');

end

hold off;

xlabel('b');

ylabel('ln(b)');

title('Segmented Linear Interpolation of Logarithmic Values');

legend('Location', 'Best');

grid on;

% 定义一个函数，根据b的值计算ln(b)

function ln\_val = compute\_ln(b)

if b > 10

n = floor(log10(b)); % 获取10的幂次

B = b / 10^n; % 获取B的值

ln\_val = log(B) + n \* log(10);

else

ln\_val = log(b);

end

end

图表, 折线图

描述已自动生成

文本

描述已自动生成

三次样条插值代码：

% 数据集中的b值，确保范围涵盖了学号值

b\_values = linspace(1, 600, 6000); % 创建一个更加密集的b值范围以便更平滑

% 计算数据集中b值的ln(b)值

lnb\_values = arrayfun(@compute\_ln, b\_values);

% 插值范围

b\_interp\_range = linspace(250, 600, 350); % 选择一个适当的插值范围

% 插值计算，使用三次样条插值

interp\_values = spline(b\_values, lnb\_values, b\_interp\_range);

% 学号数据

student\_numbers = [259, 595, 596];

student\_ln\_values = arrayfun(@compute\_ln, student\_numbers);

student\_points = [student\_numbers; student\_ln\_values];

student\_points

% 绘制插值图形

figure;

plot(b\_interp\_range, interp\_values, 'g-', 'LineWidth', 2, 'DisplayName', 'Cubic Spline Interpolation');

hold on;

scatter(student\_numbers, student\_ln\_values, 50, 'r', 'filled', 'DisplayName', 'Student Numbers');

for i = 1:numel(student\_numbers)

text(student\_numbers(i), student\_ln\_values(i), sprintf('(%d, %.2f)', student\_numbers(i), student\_ln\_values(i)), 'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'right', 'Color', 'r');

end

hold off;

xlabel('b');

ylabel('ln(b)');

title('Cubic Spline Interpolation of Logarithmic Values');

legend('Location', 'Best');

grid on;

% 定义一个函数，根据b的值计算ln(b)

function ln\_val = compute\_ln(b)

if b > 10

n = floor(log10(b)); % 获取10的幂次

B = b / 10^n; % 获取B的值

ln\_val = log(B) + n \* log(10);

else

ln\_val = log(b);

end

end

图表, 折线图

描述已自动生成

文本

描述已自动生成