第四章数值分析作业

2430534 杨赵山

2024 年 10 月 25 日

1 题目 1

1.1 题目描述

已知常用对数 lg2=0.3010,1g3=0.4771, 求出 5,6,8,9 的常用对数值并用这四个值进行三次插值估算 lg7, 给出误差估计。你有其他方法仅凭这两个给定的对数值计算 lg7 的方法吗?

1.2 解答

(1) 求解 5，6，8，9 的常用对数值

$$lg5 = lg(\frac{10}{2}) = lg10 - lg2 = 1 - 0.3010 = 0.6990$$
$$lg6 = lg(2 * 3) = lg2 + lg3 = 0.3010 + 0.4771 = 0.8780$$
$$lg8 = lg(2^3) = 3 * lg2 = 3 * 0.0.3010 = 0.9030$$
$$lg9 = lg(3^2) = 2 * lg3 = 2 * 0.4771 = 0.9542$$

(2) 利用三次插值估算 lg7, 采用拉格朗日插值法进行进行估算。首先计算拉格朗日基函数：设已知四个点 $(x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3)$, 其中 x_i 是自变量对应的 $\lg x_i$ 值。拉格朗日插值公式表示为:

$$P(x) = \sum_{i=0}^{3} y_i \cdot l_i(x)$$

其中 $l_i(x)$ 为第 i 个拉格朗日基函数

$$l_i(x) = \prod_{0 \le j \le 3} \frac{x - x_j}{x_i - x_j}$$

通过计算可以得到，

$$l_0(7) = -\frac{1}{6} \qquad l_1(7) = \frac{2}{3}$$

$$l_2(7) = \frac{2}{3} \qquad l_3(7) = -\frac{1}{6}$$

从而可以得到 lg7 的值如下：

$$lg7 = l_0 * lg5 + l_1 * lg6 + l_2 * lg8 + l_3 * lg9 = 0.8453$$

(3) 误差计算。利用拉格朗日余项进行误差估计，计算如下：

$$R_3(7) = \frac{f^4(\xi)}{4!}\omega_4(7) = \frac{-6}{4! * 7^4}(7-5)(7-6)(7-8)(7-9) = 0.00042$$

因此，得到误差估计值为 0.00042.

2 题目 2

2.1 题目描述

将区间 $[0, \pi]$ 作 6 等分, 函数 $cosx$ 在这 7 个等分点上的 6 次插值多项式函数 $p(x)$，则该插值的误差界是多少?

2.2 解答

(1) 确定插值点：区间 $[0, \pi]$ 被 6 等分，插值点为 $x_0 = 0, x_1 = \frac{\pi}{6}, x_2 = \frac{2\pi}{6}, \ldots, x_6 = \pi$，总共有 7 个插值点，即 $n = 6$。

(2) 计算导数项：由于插值的函数是 $\cos(x)$，其 $n + 1 = 7$ 阶导数是：

$$f^{(7)}(x) = \frac{d^7}{dx^7}(\cos(x)) = \sin(x)$$

(3) 计算插值误差：

$$R_6(x) = \frac{\sin(\xi)}{7!}(x-0)(x-\frac{\pi}{6})(x-\frac{2\pi}{6})\ldots(x-\pi)$$

(4) 计算结果：在区间中，多次选取 x 的值，分别计算误差值，从而得出误差的最大值。通过运行 MATLAB 程序，得到最终的计算结果为：

$$R_6(x) = 4.5064 * 10^{-5}$$

(5) matlab 代码如下：

```
n = 6;
x_nodes = linspace(0, pi, n+1);
f = @(x) cos(x);

f_derivative = @(x) sin(x);

factorial_n1 = factorial(n+1);  % (n+1)!
max_error = 0;

x_vals = linspace(0, pi, 1000);  % 1000个点用于估算误差
for x = x_vals
    % 计算误差项中的乘积 (x - x0)(x - x1)...(x - xn)
    product_term = 1;
    for i = 1:length(x_nodes)
        product_term = product_term * (x - x_nodes(i));
    end

    error = abs(f_derivative(x) / factorial_n1 * product_term);

    if error > max_error
        max_error = error;
    end
end

fprintf('插值误差的上界为: %.10f\n', max_error);
```

3 题目 3

3.1 题目描述

若函数 $f(x)$ 在 $[a,b]$ 上有五阶连续导数，且已知函数 $f(x)$ 在 $[a,b]$ 的互异的节点 x_0,x_1,x_2 上的函数值以及节点 x_0,x_1 的一阶导数值如表 2. 求一个四次埃尔米特插值多项式 $H(x)$, 使其满足

$$\begin{cases} H(x_i) = f(x_i), & i = 0,1,2 \\ H'(x_i) = f'(x_i), & i = 0,1. \end{cases}$$

并估计余项.

3.2 解答

(1) 构建埃尔米特插值多项式。可假设构建插值多项式如下:

$$H(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

(2) 计算余项。

$$R(x) = \frac{f^{(5)}(\xi)}{5!}\prod_{i=0}^{2}(x - x_i)^2$$

(3) 计算结果: 通过运行 matlab 程序求解插值多项式系数, 得到的四次插值多项式为

$$H(x) = x^4 - 2x^3 + 3x^2 - 4x + 5$$

计算得到的余项为:

$$R(x) = \frac{f^{(5)}(\xi)x^2(x-1)^2(x-2)}{120}$$

(4) matlab 代码如下:

```
x = [0, 1, 2];
f = [5, 3, 9];
f_prime = [-4, 0];

A = [
    x(1)^4, x(1)^3, x(1)^2, x(1), 1;      % H(x0) = f(x0)
    x(2)^4, x(2)^3, x(2)^2, x(2), 1;      % H(x1) = f(x1)
    x(3)^4, x(3)^3, x(3)^2, x(3), 1;      % H(x2) = f(x2)
    4*x(1)^3, 3*x(1)^2, 2*x(1), 1, 0;     % H'(x0) = f'(x0)
    4*x(2)^3, 3*x(2)^2, 2*x(2), 1, 0;     % H'(x1) = f'(x1)
];

b = [f(1); f(2); f(3); f_prime(1); f_prime(2)];

% 求解系数向量 [a4; a3; a2; a1; a0]
coeffs = A\b;

disp('插值多项式的系数 [a4, a3, a2, a1, a0]:');
```

```
disp(coeffs');

% 定义插值多项式 H(x)
syms x_sym
H_x = coeffs(1)*x_sym^4 + coeffs(2)*x_sym^3 + ...
coeffs(3)*x_sym^2 + coeffs(4)*x_sym + coeffs(5);

disp('插值多项式 H(x):');
disp(expand(H_x));

% 计算余项 R(x) 的估计
syms xi fifth_derivative
R_x = (fifth_derivative / factorial(5)) * ...
(x_sym - x(1))^2 * (x_sym - x(2))^2 * (x_sym - x(3));
disp('余项估计 R(x):');
disp(R_x);
```

4 题目 4

4.1 题目描述

使用区间 [-5,5] 上的 21 个等距节点，找出函数 $f(x) = (x^2 + 1)^-1$ 的 20 阶插值多项式 $p(x)$. 打印出 $f(x)$ 和 $p(x)$ 的图形，观察 $f(x)$ 和 $p(x)$ 的最大偏差. 若使用切比雪夫节点 $x_i = 5\cos(i\pi/20), 0 \leq i \leq 20$, 结果如何? 如果想让你找一个使用同样这两组节点插值效果差别很大的被插函数 $f(x)$(可微的函数), 你的候选函数有哪些? 验证你的结论.

4.2 解答

(1) 思路说明：通过定义插值点，定义函数，利用 plotfit 函数求解插值多项式的系数，进行绘图并计算最大的误差。

(2) 等距节点：使用等距节点计算 20 阶插值多项式，得到最大的误差值为 59.7684。绘制的图像如图 1 所示：

(3) 切比雪夫节点：使用切比雪夫节点计算 20 阶插值多项式，得到最大的误差值为 0.0177。绘制的图像如图 2 所示：

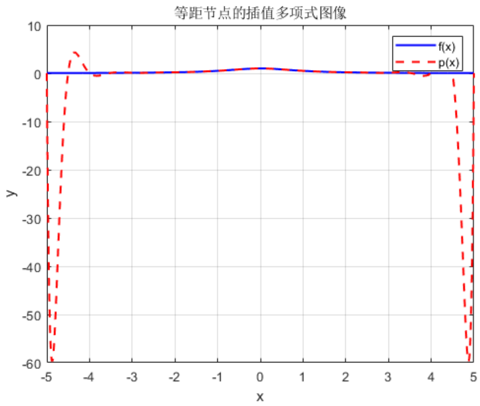

图 1: 等距节点计算插值多项式

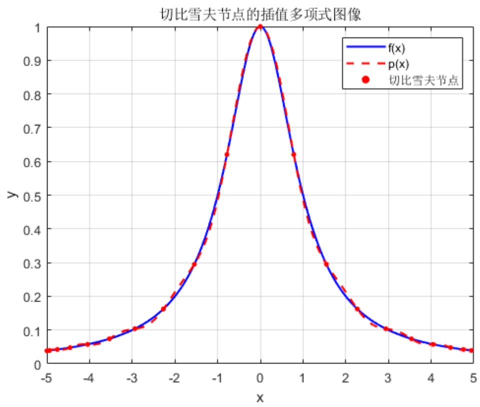

图 2: 切比雪夫节点计算插值多项式

(4) 其他函数举例：选择的函数为 $f(x) = sin(3x)$, 绘制出两种方法的插值多项式图像如图 3 所示。通过计算得到等距节点多项式插值的最大误差为 2.4949，切比雪夫节点多项式插值的最大误差为 0.0141。

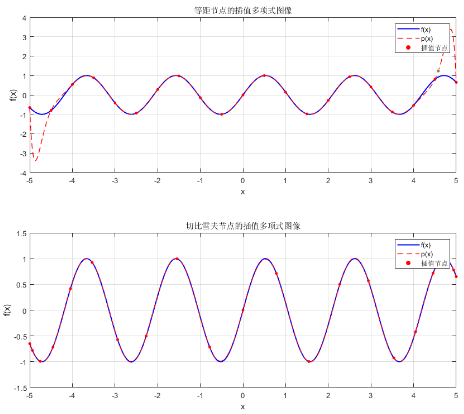

图 3: f(x)=sin(3x) 多项式插值分析

(5) 结论：在处理具有快速变化特性的函数时，使用相同数量的节点，切比雪夫节点的插值效果显著优于等距节点。

(6) matlab 代码如下：

```
f = @(x) sin(3.*x);

n = 20;  % 20阶插值多项式，对应21个节点

% 等距节点
x_uniform = linspace(-5, 5, n + 1);
y_uniform = f(x_uniform);
p_uniform = polyfit(x_uniform, y_uniform, n);

% 切比雪夫节点
x_chebyshev = 5.*cos((0:n) * pi / n);
y_chebyshev = f(x_chebyshev);
p_chebyshev = polyfit(x_chebyshev, y_chebyshev, n);

x_fine = linspace(-5, 5, 1000);
y_fine = f(x_fine);
y_interp_uniform = polyval(p_uniform, x_fine);
y_interp_chebyshev = polyval(p_chebyshev, x_fine);
```

```
% 创建两个子图
figure;

% 子图1: 等距节点插值
subplot(2,1,1);
plot(x_fine, y_fine, 'b-', 'LineWidth', 1.5);
hold on;
plot(x_fine, y_interp_uniform, 'r--', 'LineWidth', 1);
scatter(x_uniform, y_uniform,15, 'ro', 'filled');
legend('f(x)', 'p(x)', '插值节点');
title('等距节点的插值多项式图像');
xlabel('x');
ylabel('f(x)');
grid on;
hold off;

% 子图2: 切比雪夫节点插值
subplot(2,1,2);
plot(x_fine, y_fine, 'b-', 'LineWidth', 1.5);
hold on;
plot(x_fine, y_interp_chebyshev, 'r--', 'LineWidth', 1);
scatter(x_chebyshev, y_chebyshev, 15,'ro', 'filled');
legend('f(x)', 'p(x)', '插值节点');
title('切比雪夫节点的插值多项式图像');
xlabel('x');
ylabel('f(x)');
grid on;
hold off;

% 计算最大偏差
max_deviation_uniform = max(abs(y_interp_uniform - y_fine));
max_deviation_chebyshev = max(abs(y_interp_chebyshev - y_fine));
```

5 题目 5

5.1 题目描述

输入向量 c, 请给出下面方程组的更好的数值求解方法，使得 n 较大时要么计算时间更短，要么计算结果更准确，你可以和左除相比较：

$$\begin{cases} x_1 & +2x_2 & +\cdots & +2^{n-1}x_n & = c_1 \\ x_1 & +3x_2 & \cdots & +3^{n-1}x_n & = c_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_1 & +(n+1)x_2 & +\cdots & +(n+1)^{n-1}x_n & = c_n \end{cases}$$

当 $c_i = \frac{1}{i}((i+1)^n - 1)$ 时，该方程组有精确解，所有解分量均为 1.

5.2 解答

(1) QR 分解法。QR 分解将系数矩阵分解为一个正交矩阵和一个上三角矩阵，求解线性方差组。

$$Q^T\mathbf{c} = \mathbf{y}$$

$$R\mathbf{x} = \mathbf{y}$$

(2) 最小二乘法。最小而成的目标函数为 $\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|^2$，即通过构建残差平方和，以求解正太方程的方式进行方程组求解。残差 P 平方计算公式为：

$$S(\mathbf{x}) = \|\mathbf{r}\|^2 = (A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b})$$

为了找到最小值，对 $S(\mathbf{x})$ 求导并设其为零

$$\nabla S(\mathbf{x}) = 2A^T(A\mathbf{x} - \mathbf{b}) = 0$$

得到正太矩阵为：

$$A^TA\mathbf{x} = A^T\mathbf{b}$$

最终得到的求解公式为：

$$\mathbf{x} = (A^TA)^{-1}A^T\mathbf{b}$$

(3) 结果分析：通过改变 n 的值，分别绘制出两种方法的运行时间对比和求解精度的比较。

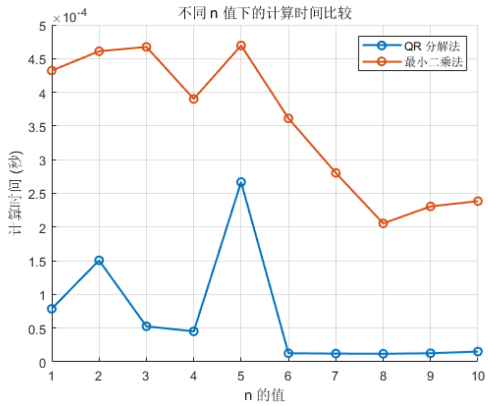

图 4: 计算时间的比较分析

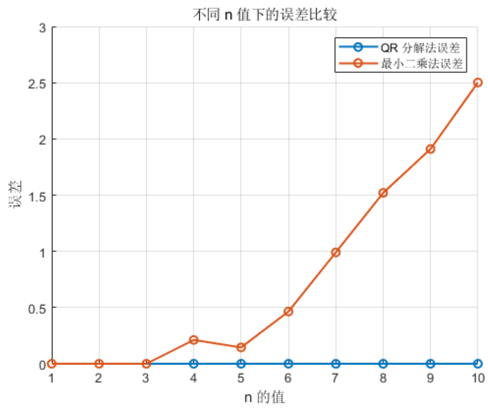

图 5: 计算精度的比较分析

(4) matlab 代码:

```
n_values = 1:1:10;
num_trials = length(n_values);
time_left_div = zeros(num_trials, 1);
time_qr = zeros(num_trials, 1);
time_lsqr = zeros(num_trials, 1);
error_left_div = zeros(num_trials, 1);
error_qr = zeros(num_trials, 1);
error_lsqr = zeros(num_trials, 1);

% 循环处理每个 n 的值
```

```
for idx = 1:num_trials
    n = n_values(idx);

    % 构建 Vandermonde 矩阵 A
    A = zeros(n, n);
    for i = 1:n
        for j = 1:n
            A(i, j) = (i + 1)^(j - 1);
        end
    end

    % 构建向量 c
    c = zeros(n, 1);
    for i = 1:n
        c(i) = (1 / i) * ((i + 1)^n - 1);
    end

    % 精确解
    exact_solution = ones(n, 1);

    % 左除法求解
    tic;
    x_left_div = A \ c;
    time_left_div(idx) = toc;

    % QR 分解法求解
    tic;
    [Q, R] = qr(A);
    y = Q' * c;
    x_qr = R \ y;
    time_qr(idx) = toc;

    % 最小二乘法求解
    tic;
    x_lsqr = lsqr(A, c);
    time_lsqr(idx) = toc;
```

```
    % 计算误差
    error_left_div(idx) = norm(x_left_div - exact_solution);
    error_qr(idx) = norm(x_qr - exact_solution);
    error_lsqr(idx) = norm(x_lsqr - exact_solution);
end

% 可视化计算时间
figure;
hold on;
%plot(n_values, time_left_div, '-o', 'DisplayName', '左除法');
plot(n_values, time_qr, '-o', 'DisplayName', 'QR 分解法');
plot(n_values, time_lsqr, '-o', 'DisplayName', '最小二乘法');
hold off;
xlabel('n 的值');
ylabel('计算时间 (秒)');
title('不同 n 值下的计算时间比较');
legend;
grid on;

% 可视化误差
figure;
hold on;
%plot(n_values, error_left_div, '-o', 'DisplayName', '左除法误差');
plot(n_values, error_qr, '-o', 'DisplayName', 'QR 分解法误差');
plot(n_values, error_lsqr, '-o', 'DisplayName', '最小二乘法误差');
hold off;
xlabel('n 的值');
ylabel('误差');
title('不同 n 值下的误差比较');
legend;
grid on;
```