数值分析实验报告 - Code 2

Chase Young

2024 年 3 月 7 日

1 实验目的

对函数

$$f\left(x\right)=\frac{1}{1+25x^2},x\in\left[-1,1\right]$$

构造 Lagrange 插值多项式 $p_N\left(x\right)$，其中插值结点为：

(1) 等间距网格点：$x_i=1-\frac{2}{N}$, $i=0,1,\ldots,N$;

(2) Chebyshev 网格点：$x_i=-\cos\left(\frac{2i+1}{2N+2}\pi\right)$, $i=0,1,\ldots,N$

并计算如下误差：

$$error=\max_i\left\{\left|f\left(y_i\right)-p_N\left(y_i\right)\right|,y_i=\frac{i}{50}-1,i=0,1,\ldots,100\right\}.$$

分别对 $N=5,10,20,40$ 比较上述两组节点的结果，并在绘制 $N=20$ 时两种插值方案的计算结果。

2 实验方法

我们根据 Newton 插值的公式，在 $x_1,x_2,\ldots,x_N$ 点处，得到插值多项式的表达式为

$$p_N\left(x\right)=a_0+a_1\left(x-x_1\right)+a_2\left(x-x_1\right)\left(x-x_2\right)+\cdots+a_n\left(x-x_1\right)\left(x-x_2\right)\cdots\left(x-x_N\right), \tag{1}$$

其中

$$a_k=f\left[x_0,x_1,\ldots,x_k\right],k=0,1,\ldots,N,$$

为 k 阶差商。

分别对 $N=5,10,20,40$ 和要求的两组插值点进行计算，得到插值多项式后，在 $y_i=\frac{i}{50}-1,i=0,1,\ldots,100$ 处计算 p_N-f 的离散无穷模，作为插值精确度的度量。

具体而言，定义了函数

- **function** y = interpolate_n(xx, yy, x)

其中 xx 和 yy 是插值所用的节点对，计算并返回插值函数在 x 点处的函数值 y。相关代码见脚本 interpolate_n.m。

对于要求的不同的 N，分别产生插值节点对 xx 和 yy，并计算在 y_i 处的插值函数与 $f\left(x\right)$ 的差的绝对值；遍历所有 i，得到上述绝对值的最大值；绘制 $N=20$ 时两组插值节点处的插值函数的图像。相关代码见脚本 code_2.m。

3 实验结果

经过计算，两组插值节点得到的插值函数在 $y_i, i = 0, 1, \ldots, 100$ 处的离散无穷模如表 1所示。

插值节点数 N	网格 (1) 误差	网格 (2) 误差
5	0.432692	0.555911
10	1.915643	0.108929
20	58.278125	0.015325
40	78689.037485	0.000274

表 1: 两组插值节点处的插值函数精度比较

当 $N = 20$ 时，分别绘制两组插值点处的插值多项式和函数 $f(x)$ 的图像，如图 1所示。

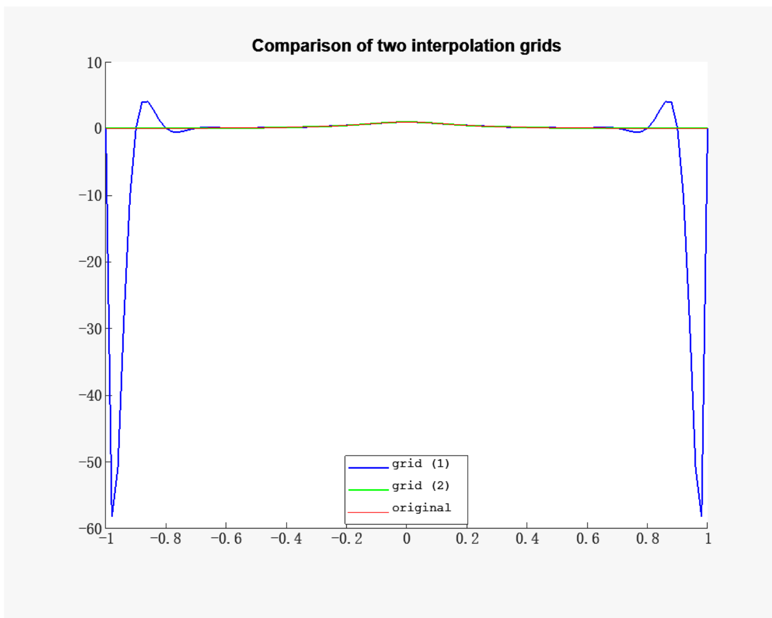

图 1: 两组插值节点处的插值函数与 $f(x)$ 的图像

4 后续讨论

从上述实验结果中可以看出，无论是从数值上来看，还是从图像上来看，在插值节点 (2) 上计算得到的插值多项式都比在插值节点 (1) 上计算得到的插值多项式更精确，即能够更好地逼近原来的函数 $f(x)$。这也与“Chebyshev 点是最佳结点”这一结论相吻合。

A 代码

本部分包含了主要用到的代码。

interpolate_n.m

```
function y = interpolate_n(xx, yy, x)
    % Newton interpolate, return the interpolated value at point x
    n = length(xx);
    for i = 2:n
        for j = n:-1:i
            yy(j) = (yy(j) - yy(j-1)) / (xx(j) - xx(j-i+1));
        end
    end
    y = yy(end);
    for i = n:-1:2
        y = yy(i-1) + (x - xx(i-1))*y;
    end
end
```

code_2.m

```
% code 2
NList = [5, 10, 20, 40];
error1 = zeros(4, 1);
error2 = zeros(4, 1);
for k = 1:length(NList)
    N = NList(k);
    x1 = 1 - (0:N) * 2 / N;
    x2 = -cos((2*(0:N) + 1)*pi/(2*N+2));
    y1 = 1./(1 + 25*x1.^2);
    y2 = 1./(1 + 25*x2.^2);

    yTest = (0:100)/50 - 1;
    p1 = zeros(length(yTest), 1);
    p2 = p1;
    for i = 1:length(yTest)
        p1(i) = interpolate_n(x1, y1, yTest(i));
        p2(i) = interpolate_n(x2, y2, yTest(i));
    end
    error1(k) = max(abs(p1' - (1./(1+25*yTest.^2))));
    error2(k) = max(abs(p2' - (1./(1+25*yTest.^2))));

    if N == 20
        plot(yTest, p1, 'b', 'LineWidth', 1.5);
        hold on;
        plot(yTest, p2, 'g', 'LineWidth', 1.5);
        hold on;
        xx = -1:0.01:1;
        yy = 1./(1+25*xx.^2);
        plot(xx, yy, 'r','LineWidth', 1);
        legend('grid (1)', 'grid (2)', 'original', 'Location', 'south');
        title("Comparison of two interpolation grids");
    end
end
```

```
for k = 1:length(NList)
    fprintf("N=%d\n", NList(k));
    fprintf("Max Error of grid (1): %.6f\n", error1(k));
    fprintf("Max Error of grid (2): %.6f\n", error2(k));
end
```