科学计算 Exercise 3

范舟

516030910574

致远学院 2016 级 ACM 班

1 教材练习 P49-50

13. 解：由于次数不超过三次的多项式 $P(x)$ 通过点 $(x_0, f(x_0)), (x_1, f(x_1))$，可设

$$P(x) = f(x_0) + f[x_0, x_1](x - x_0) + A(x - x_0)(x - x_1) + B(x - x_0)^2(x - x_1)$$

又有

$$\begin{aligned} P'(x_0) &= f[x_0, x_1] + A(x_0 - x_1) = f'(x_0) \\ P''(x_0) &= 2A + 2B(x_0 - x_1) = f''(x_0) \end{aligned}$$

解得

$$\begin{aligned} A &= \frac{f'(x_0) - f[x_0, x_1]}{x_0 - x_1} \\ &= \frac{f'(x_0)(x_0 - x_1) + f(x_1) - f(x_0)}{(x_0 - x_1)^2} \\ B &= \frac{f''(x_0)(x_0 - x_1) + 2f[x_0, x_1] - 2f'(x_0)}{2(x_0 - x_1)^2} \\ &= \frac{f''(x_0)(x_0 - x_1)^2 - 2f'(x_0)(x_0 - x_1) + 2f(x_0) - 2f(x_1)}{2(x_0 - x_1)^3} \end{aligned}$$

因此

$$\begin{aligned} P(x) = f(x_0) &+ \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) \\ &+ \frac{f'(x_0)(x_0 - x_1) + f(x_1) - f(x_0)}{(x_0 - x_1)^2}(x - x_0)(x - x_1) \\ &+ \frac{f''(x_0)(x_0 - x_1)^2 - 2f'(x_0)(x_0 - x_1) + 2f(x_0) - 2f(x_1)}{2(x_0 - x_1)^3}(x - x_0)^2(x - x_1) \end{aligned}$$

15. 证明：可设 $[x_k, x_{k+1}]$ 间的插值余项为

$$R_3(x) = f(x) - P(x) = k(x)(x-x_k)^2(x-x_{k+1})^2$$

其中 $k(x)$ 为待定函数，固定 $x \in [x_k, x_{k+1}]$，构造

$$\phi(t) = f(t) - P(t) - k(x)(t-x_k)^2(t-x_{k+1})^2$$

显然有 $\phi(x_k) = \phi(x_{k+1}) = \phi(x) = 0$，且 $\phi'(x_k) = \phi'(x_{k+1}) = 0$，因此在 $\phi(t)$ 在 $[x_k, x_{k+1}]$ 中至少有 5 个零点（二重根算两个）. 反复应用罗尔定理，可得 $\phi^{(4)}(t)$ 在 (x_k, x_{k+1}) 中至少有一个零点，不妨设其为 ξ，则有

$$\phi^{(4)}(\xi) = f^{(4)}(\xi) - 4!k(x) = 0$$

即得到

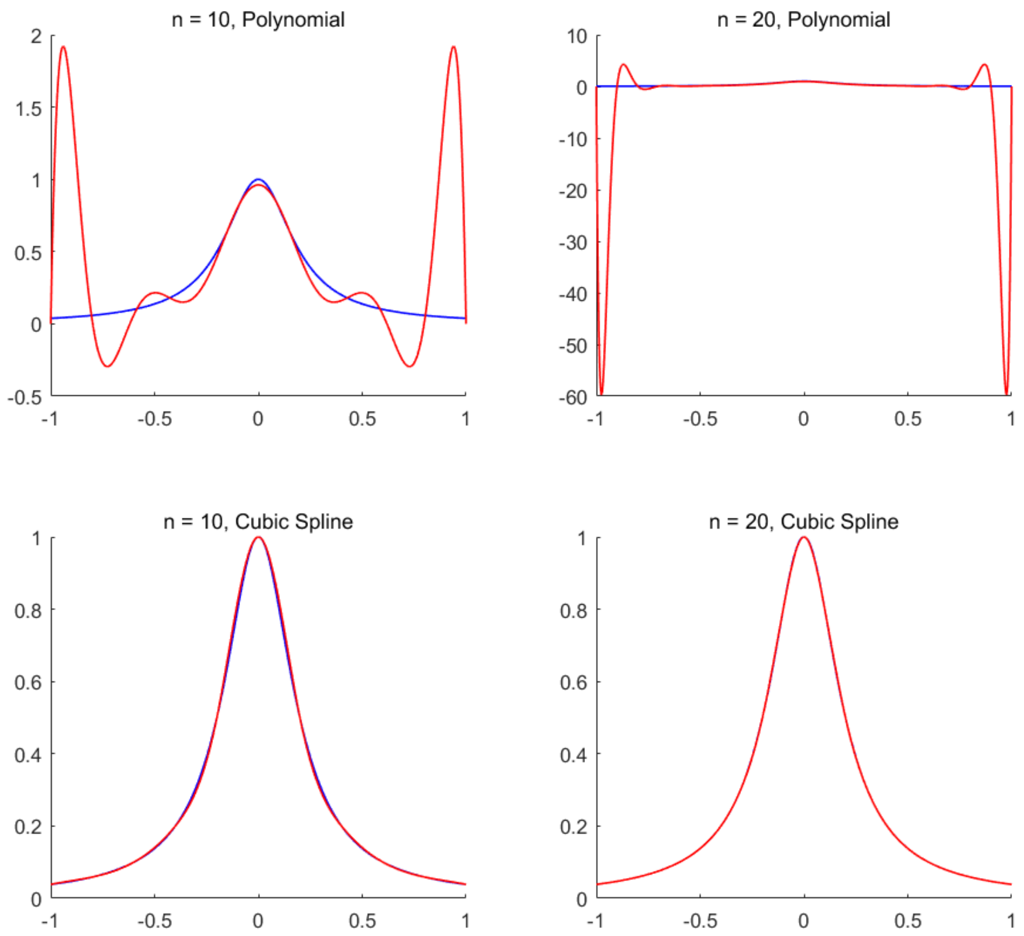
$$k(x) = \frac{f^{(4)}(\xi)}{4!},\ \xi \in (x_k, x_{k+1})$$
$$R_3(x) = \frac{f^{(4)}(\xi)}{4!}(x-x_k)^2(x-x_{k+1})^2$$

设 $h_k = x_{k+1} - x_k, h = \max h_k, M = \sup f^{(4)}$，则分段三次埃尔米特插值的误差

$$R_3(x) = \frac{f^{(4)}(\xi)}{4!}(x-x_k)^2(x-x_{k+1})^2 \le \frac{M}{4!}\frac{h^4}{16} = \frac{Mh^4}{384}$$

因此分段三次埃尔米特插值的误差限为 $\frac{Mh^4}{384}$.

2. 画出的插值图像如下图.



Matlab 代码：

```matlab
x_10 = -1 : (1 - -1) / 10 : 1;
x_20 = -1 : (1 - -1) / 20 : 1;
f_10 = f(x_10);
f_20 = f(x_20);

x_plot = -1 : (1 - -1) / 500 : 1;
f_plot = f(x_plot);

subplot(2, 2, 1);
plot(x_plot, f_plot, 'b', 'LineWidth', 1);
hold;
y1_plot = newton(x_10, f_10, x_plot);
plot(x_plot, y1_plot, 'r', 'LineWidth', 1);
title('n = 10, Polynomial');
box off;

subplot(2, 2, 2);
plot(x_plot, f_plot, 'b', 'LineWidth', 1);
hold;
y2_plot = newton(x_20, f_20, x_plot);
```

```
plot(x_plot, y2_plot, 'r', 'LineWidth', 1);
title('n = 20, Polynomial');
box off;

subplot(2, 2, 3);
plot(x_plot, f_plot, 'b', 'LineWidth', 1);
hold;
y1_plot = spline(x_10, f_10, x_plot);
plot(x_plot, y1_plot, 'r', 'LineWidth', 1);
title('n = 10, Cubic Spline');
box off;

subplot(2, 2, 4);
plot(x_plot, f_plot, 'b', 'LineWidth', 1);
hold;
y2_plot = spline(x_20, f_20, x_plot);
plot(x_plot, y2_plot, 'r', 'LineWidth', 1);
title('n = 20, Cubic Spline');
box off;

function ret = f(x)
    ret = 1 ./ (1 + 25 * x .^ 2);
end

function yq = newton(x, y, xq)
    n = length(x);
    d(:, 1) = y;
    for i = 2 : n
        dx(i : n) = x(i : n) - x(1 : n - i + 1);
        d(i : n, i) = (d(i : n, i - 1) - d(i - 1 : n - 1, i - 1)) ./ dx(i : n)';
        a(i) = d(i, i);
    end
    yq = calc(a, x, xq);
    function y = calc(a, x, xq)
        n = length(x);
        y = 0;
        for i = 1 : n
            t = a(i);
            for j = 1 : i - 1
                t = t .* (xq - x(j));
            end
            y = y + t;
        end
    end
end
```

2 补充练习

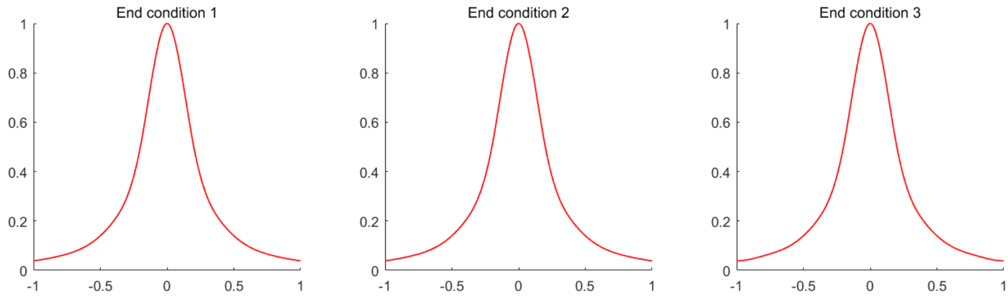
1. 下面的这段 Matlab 代码中，spline_zhou() 这个函数实现了三种边界条件的三次样条插值，函数的调用方式请见代码开头的注释：

```
% function spline_zhou(), code by Zhou Fan
% Cubic Spline Interpolation under three classes of end conditions
```

```matlab
% Usage:
%   condition (1): yq = spline_zhou(x, y, xq, 1, [a, b]);
%   condition (2): yq = spline_zhou(x, y, xq, 2, [a, b]);
%   condition (3): yq = spline_zhou(x, y, xq, 3, []);

function yq = spline_zhou(x, y, xq, cond_type, cond_v)
    n = length(x);
    h(1 : n - 1) = x(2 : n) - x(1 : n - 1);
    mu(2 : n - 1) = h(1 : n - 2) ./ (h(1 : n - 2) + h(2 : n - 1));
    lambda(2 : n - 1) = h(2 : n - 1) ./ (h(1 : n - 2) + h(2 : n - 1));
    d1_f(2 : n) = (y(2 : n) - y(1 : n - 1)) ./ h(1 : n - 1);
    d2_f(3 : n) = (d1_f(3 : n) - d1_f(2 : n - 1)) ./ (h(1 : n - 2) + h(2 : n - 1));
    d(2 : n - 1) = 6 * d2_f(3 : n);

    switch cond_type
        case 1
            lambda(1) = 1;
            d(1) = 6 / h(1) * (d1_f(2) - cond_v(1));
            mu(n) = 1;
            d(n) = 6 / h(n - 1) * (cond_v(2) - d1_f(n));
        case 2
            lambda(1) = 0;
            mu(n) = 0;
            d(1) = 2 * cond_v(1);
            d(n) = 2 * cond_v(2);
        case 3
            lambda(n) = h(1) / (h(n - 1) + h(1));
            mu(n) = 1 - lambda(n);
            d(n) = 6 * (d1_f(2) - d1_f(n)) / (h(1) + h(n - 1));
            d(1) = 0;
        otherwise
            throw(MException('Zhou:invalidArgument', 'Invalid argument for function spline_zhou'));
    end

    if cond_type <= 2
        A = 2 * eye(n);
        A = A + [zeros(n - 1, 1), lambda(1 : n - 1) .* eye(n - 1); zeros(1, n)];
        A = A + [zeros(1, n); mu(2 : n) .* eye(n - 1), zeros(n - 1, 1)];
    else
        B = 2 * eye(n - 1);
        B = B + [zeros(n - 2, 1), lambda(2 : n - 1) .* eye(n - 2); zeros(1, n - 1)];
        B = B + [zeros(1, n - 1); mu(3 : n) .* eye(n - 2), zeros(n - 2, 1)];
        B(1, n - 1) = mu(2);
        B(n - 1, 1) = lambda(n);
        A = [zeros(1, n); zeros(n - 1, 1), B];
        A(1, 1) = 1;
        A(1, n) = -1;
    end

    M = (A \ d')';
    j = discretize(xq, x);
    yq = M(j) .* (x(j + 1) - xq) .^ 3 ./ (6 * h(j)) + M(j + 1) .* (xq - x(j)) .^ 3 ./ (6 * h(j)) ...
        + (y(j) - M(j) .* h(j) .^ 2 ./ 6) .* (x(j + 1) - xq) ./ h(j) ...
        + (y(j + 1) - M(j + 1) .* h(j) .^ 2 ./ 6) .* (xq - x(j)) ./ h(j);
```

```
end
```

使用 spline_zhou() 对上一道题目中的函数分别进行三种条件下的三次样条插值（取 $n = 10$），得到的图像如下.



可以看到，在三种边界条件下，三次样条插值的效果都很好。