科学计算作业 练习 3

任云玮

2016 级 ACM 班

516030910586

1 教材练习

13、求次数小于等于 3 的多项式 $P(x)$……

解 设

$$P(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) + A(x - x_0)(x - x_1) + B(x - x_0)^2(x - x_1).$$

成立 $P(x_0) = f(x_0)$，$P(x_1) = f(x_1)$.

$$\begin{cases} P'(x_0) = \dfrac{f(x_1) - f(x_0)}{x_1 - x_0} + A(x_0 - x_1) = f'(x_0) \\ P''(x_0) = 2A + 2B(x_0 - x_1) = f''(x_0) \end{cases}$$

$$\Rightarrow \begin{cases} A = \dfrac{f(x_1) - f(x_0) - f'(x_0)(x_1 - x_0)}{(x_1 - x_0)^2} \\ B = \dfrac{f''(x_0)(x_0 - x_1)^2 - 2(f(x_1) - f(x_0) + f'(x_0)(x_0 - x_1))}{2(x_0 - x_1)^3} \end{cases}$$

所以，$P(x)$ 为

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

15、证明两点三次 Hermite 插值余项……

证明 设插值多项式为 $P(x)$，记余项

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

同时，对于固定的 x，设

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

则 $\varphi(t)$ 在 $t = x_k, x_{k+1}, x$ 处为零. 则 $\varphi(t)$ 在 $t = \xi_1, \xi_2, x_k, x_{k+1}$ 处为零，其中 $\xi_1 \in (x_k, x)$，$\xi_2 \in (x, x_{k+1})$. 反复应用 Rolle 定理，可得存在 $\xi \in (x_k, x_{k+1})$，成立 $\varphi^{(4)}(\xi) = 0$. 即

$$k(x) = \frac{f^{(4)}}{4!}(\xi).$$

所以对于余项，成立

$$R(x) = \frac{f^{(4)}(\xi)}{4!}(x-x_k)^2(x-x_{k+1})^2. \tag{1}$$

对于分段三次 Hermite 插值，每一段上的余项满足式 (1).

对于每一段，误差满足

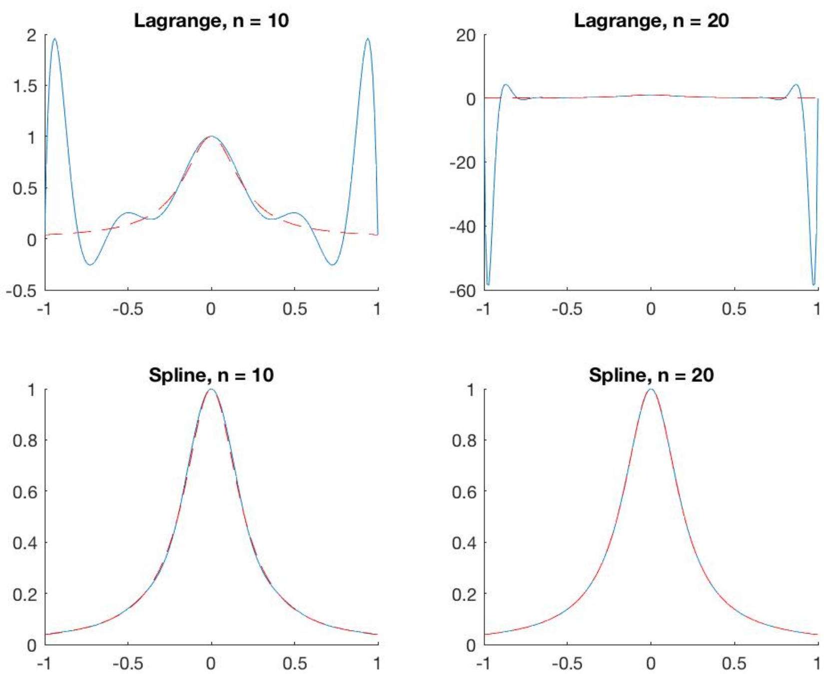
$$|R(x)| \le \frac{f^{(4)}(\xi)}{4!}\left(\frac{(x_{k+1}-x_k)^2}{4}\right)^2 = \frac{f^{(4)}(\xi)}{384}h_k^4.$$

设 M_4 为 $f^{(4)}$ 的上确界，$h = \max h_k$，则有

$$|f(x) - S(x)| \le \frac{M_4}{384}h^4. \quad \blacksquare$$

2、在区间 $[-1,1]$ 上分别取……

解 其中红线为函数图像，蓝线为插值多项式图像，源代码见后。



```matlab
f = @(x) 1 ./ (1 + 25 * x.^2);

x = {linspace(-1, 1, 11), linspace(-1, 1, 21)};
y = {f(x{1}), f(x{2})};

drawX = -1 : 1/100 : 1;
ansY = f(drawX);

subplot(2, 2, 1);
plot(drawX, lagrangeInterp(x{1}, y{1}, drawX), '-', drawX, ansY, 'r--');
title('Lagrange, n = 10');
box off

subplot(2, 2, 2);
plot(drawX, lagrangeInterp(x{2}, y{2}, drawX), '-', drawX, ansY, 'r--');
title('Lagrange, n = 20');
box off

subplot(2, 2, 3);
plot(drawX, spline(x{1}, y{1}, drawX), '-', drawX, ansY, 'r--');
title('Spline, n = 10');
box off

subplot(2, 2, 4);
plot(drawX, spline(x{2}, y{2}, drawX), '-', drawX, ansY, 'r--');
title('Spline, n = 20');
box off

function ansY = lagrangeInterp(interpX, interpY, queryX)
    n = length(interpX);
    function yi = li(i, x)
        yi = 1;
        for j = 1 : n
            if (j ~= i)
                yi = yi .* ...
                    (x - interpX(j)) ./ (interpX(i) - interpX(j));
            end
        end
    end

    ansY = 0;
    for i = 1 : n
        ansY = ansY + interpY(i) * li(i, queryX);
    end
end
```

2 补充练习

1、基于 spline 求解其他两类边界条件对应的插值多项式。

```
function ansY = cubicSpline(px, py, qx, type)
    if string(type) ~= 'period'
        fp0 = py(1);
        fpn = py(end);
        py = py(2:end-1);
    end
    h = px(2:end) - px(1:end-1);
    % divided difference
    dd = (py(2:end) - py(1:end-1)) ./ (px(2:end) - px(1:end-1));

    lambda = [1, h(2:end) ./ (h(1:end-1) + h(2:end)), 1];
    mu = [1, h(1:end-1) ./ (h(1:end - 1) + h(2:end)), 1];
    d = [0; 6 * ((dd(2:end) - dd(1:end-1)) ./ (h(2:end) + h(1:end-1)))'; 0];

    if string(type) == 'endslope'
        d(1) = 6 / h(1) * (dd(1) - fp0);
        d(end) = 6 / h(end) * (fpn - dd(end));
        A = 2 * eye(length(d)) + diag(mu(2:end), -1) + diag(lambda(1:end-1), 1);
        M = linsolve(A, d);
    elseif string(type) == 'moment'
        lambda(1) = 0;
        mu(end) = 0;
        d(1) = 2 * fp0;
        d(end) = 2 * fpn;
        A = 2 * eye(length(d)) + diag(mu(2:end), -1) + diag(lambda(1:end-1), 1);
        M = linsolve(A, d);
    elseif string(type) == 'period'
        lambda(end) = h(1) ./ (h(end) + h(1));
        mu(end) = 1 - lambda(end);
        d(end) = 6 * (dd(1) - dd(end)) ./ (h(1) + h(end));
        A = 2 * eye(length(d)) + diag(mu(2:end), -1) + diag(lambda(1:end-1), 1);
        A = A(2:end, 2:end);
        d = d(2:end);
        A(1, end) = mu(1);
        A(end, 1) = lambda(end);
        M = linsolve(A, d);
        M = [M(end); M];
    else
        error("Undefined type");
    end

    M = M';
    pos = discretize(qx, px);
    ansY = M(pos) .* (px(pos + 1) - qx).^3 ./ 6 ./ h(pos) + ...
           M(pos + 1) .* (qx - px(pos)).^3 ./ 6 ./ h(pos) + ...
           (py(pos) - M(pos) .* h(pos).^2 / 6) .* (px(pos + 1) - qx) ./ h(pos) + ...
           (py(pos + 1) - M(pos + 1) .* h(pos).^2 / 6) .* (qx - px(pos)) ./ h(pos);
end
```