КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Ебаная хуйня, блядь

Побудова інтерполяційного сплайну

Виконав:

студент 4-го курсу

спеціальність математика

Сивак Назар

2016

Постановка задачі

Побудувати інтерполяційний сплайн $S(x,u)$ другого степеня дефекту 1, з крайовими умовами типу III, використовуючи метод $2M$ для пошуку системи лінійних рівнянь та метод релаксації для її розв'язку.

Теоретичні відомості

Інтерполяційний сплайн другого степеня дефекту 1 — це така функція $S(x,u) \in C^1([a,b])$, що для інтерполяційної сітки X та функції u виконується:

$$\forall x \in [X_i, X_{i+1}) : S(x,u) = a_2^i (x - X_i)^2 - a_1^i (x - X_i) + a_0^i ; \forall i : S(X_i, u) = u(X_i).$$

Але для сплайнів парного степеня використовують іншу сплайнову сітку $\{x_i\} \subset [a,b]$, точки якої зазвичай кладуть посередині відрізків інтерполяційної сітки:

$$x_i = (X_{i-1} + X_i)/2, i = \overline{2,n}; \quad x_1 = X_1, x_{n+1} = X_{n+1}.$$

Тоді

$$\forall x \in [x_i, x_{i+1}) : S(x,u) = a_2^i (x - X_i)^2 - a_1^i (x - X_i) + a_0^i ; \forall i : S(X_i, u) = u(X_i).$$

Для побудови системи рівнянь для коефіцієнтів використаємо другі похідні $2a_i = M_i = S''(X_i, u)$. З теорії відомі такі обмеження на M_i (навчальний посібник "Сплайн-функції та її застосування"):

$$h_{i-1}M_{i-1} + 3(h_{i-1} + h_i)M_i + h_i M_i = 8u(X_{i-1}; X_i; X_{i+1})(h_{i-1} + h_i);$$
$$h_{i-1}M_{i-1} + 3(h_{i-1} + h_i)M_i + h_i M_i = 8(u(X_i; X_{i+1}) - u(X_{i-1}; X_i)).$$

Де $h_i = X_{i+1} - X_i$. Тоді отримуємо систему рівнянь:

$$\begin{cases} M_1 = A; \\ h_{i-1}M_{i-1} + 3(h_{i-1} + h_i)M_i + h_i M_i = 8(u(X_i; X_{i+1}) - u(X_{i-1}; X_i)), i = \overline{2, n-1}; \\ M_n = B; \end{cases}$$

Для якої можна записати матрицю

$$\left(\begin{array}{ccccc|c} 1 & 0 & 0 & \dots & 0 & A \\ h_1 & 3(h_1 + h_2) & h_2 & \dots & 0 & 8(u(X_2; X_3) - u(X_1; X_2)) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & h_{n-2} & 3(h_{n-2} + h_{n-1}) & h_{n-1} & 8(u(X_{n-1}; X_n) - u(X_{n-2}; X_{n-1}) \\ 0 & \dots & 0 & 0 & 1 & B \end{array}\right)$$

В умові вимагається розв'язання системи лінійний рівнянь методом квадратного кореня, який в свою чергу вимагає ермітовості матриці, тому цю систему перетворюємо в симетричну відніманням від другого та передостаннього рядка відповідно першого та останнього, помножених на відповідні коефіцієнти.

Для обрахування саме коефіцієнтів сплайну використовуються наступні формули:

$$\begin{cases} a_i = \frac{M_i}{2}; \\ c_i = u_i; \\ b_1 = u(X_1, X_2) - \frac{1}{8}h_1(3M_1 + M_2); \\ b_i = u(X_{i-1}; X_i) + \frac{1}{8}h_1(M_{i-1} + 3M_i), i = \overline{2, n}. \end{cases}$$

Практична реалізація

Функція, що здійснює перевірку правильності побудови сплайну: побудову графіків та розрахунок сіткової норми.

perevirka.m

```
function [] = perevirka(func, X, T)
    if ~exist('func')
```

```
        func = @(t)(t * (1 - t));
    end;
    if ~exist('X')
        X = 0:0.1:1;
    end;
    if ~exist('T')
        T = 0 : 0.001 : 1;
    end;

    u = arrayfun(func, X);
    res = spl_23(X, u, T);

    plot(T, arrayfun(func, T), 'm', T, res, 'r', X, u, 'kx');
    legend('interpolyovana␣funciya', 'spline', 'X');
    title(sprintf('Rasnica␣norm␣%e', max(abs(arrayfun(func, T) - res))));
    print -dpdf ./result.pdf;
end;
```

Функція, що здійснює побудову сплайна.

spl_23.m

```
function res = spl_23(X, u, T)
    if isrow(X)
        X = X';
    end;
    if isrow(u)
        u = u';
    end;

    n = length(X) + 1;
    u = [u; u(2)];
    h = X(2 : end) - X(1 : end - 1);
    X1 = X(2 : end) - h / 2;
    h = [h; h(1)];
    d = (u(2 : end) - u(1 : end - 1)) ./ h(1 : end);
    m = [diag(h(1 : end - 1)), zeros(n - 2, 2)] + [zeros(n - 2, 2), diag(h(2 : end))] +
        3 * [zeros(n - 2, 1), diag(h(1: end - 1)) + diag(h(2 : end)), zeros(n - 2, 1)];
    m(:, [2, end - 1]) += m(:, [end, 1]);
    m = m(:, 2 : end - 1);
    b = [8 * (d(2 : end) - d(1 : end - 1))];
    m = [m, b];

    [r, c] = size(m);
    m = m / (norm(m, 1));
    B = diag(ones(r, 1)) - m(:, 1 : r);
    b = solve(diag(ones(r, 1)) - tril(B, -1), m(:, r + 1 : end));
    next = @(t)(solve(diag(ones(r, 1)) - tril(B, -1), triu(B) * t) + b);
    app = zeros(r, c - r);
    for i = 1 : 1000
        t = next(app);
```

```
            s = app + 1 * (t - app);
            if norm(t - app, 1) < 1e-5
                break
            end;
            app = s;
        end;

        n = n - 1;
        u = u(1 : end - 1);
        h = X(2 : end) - X(1 : end - 1);
        d = (u(2 : end) - u(1 : end - 1)) ./ h(1 : end);
        s = [s(end); s];

        spl = [s / 2, zeros(n, 1), u];
        spl(:, 2) = [d(1) - 0.125 * h(1) * (3 * s(1) + s(2)) ;...
         d(1 : end) + 0.125 * h(1 : end) .* (s(1 : end - 1) + 3 * s(2 : end))];
        splineFunction = @(t)(eval(X, X1, spl, t));
        res = arrayfun(splineFunction, T);
end;

function s = solve(m, b)
        [r, c] = size(b);
        s = zeros(r, c);
        for i = 1 : r
            s(i, :) = b(i, :) ./ m(i, i);
            b(i : end, :) -= m(i : end, i) * s(i, :);
        end;
end;

function y = eval(X, X1, spl, t)
        i = max([0; find((t - X1) >= 0)]) + 1;
        r = spl(i, :);
        t1 = t - X(i);
        p = t1 .^ (length(r) - 1 : -1 : 0);
        y = sum(r(1 : length(p)) .* p);
end;
```