CS770: Assignment 4

Ronghao Yang
ID: 20511820

November 12, 2017

1 Question 1

1.1 Question 1a

For cubic spline, let the function $S(x)$ define the spline function
where $a = x_0 < x_1 < x_2 < x_3 < ...... < x_n = b$

$$S(X) = \begin{cases} S_0(x), & x_0 < x < x_1 \\ S_1(x), & x_1 < x < x_2 \\ ...... \\ S_i(x), & x_i < x < x_{i+1} \\ ...... \\ S_{n-1}(x), & x_{n-1} < x < x_n \end{cases} \tag{1}$$

Where each $S_i(x)$ has degree 3 in this case.
In cubic spline, $S(x)$ satisfies

$$\begin{cases} S_i(x_i) = S_{i+1}(x_i) \\ S'_i(x_i) = S'_{i+1}(x_i) \\ S''_i(x_i) = S''_{i+1}(x_i) \\ S_i(x_i) = y_i \end{cases}$$

By the definition of natural cubic spline, we have two additional constraints,

$$\begin{cases} S''_0(x_0) = 0 \\ S''_{n-1}(x_{n-1}) = 0 \end{cases}$$

1.2 Question 1b

```
function [coeffs] = nSpline(X,y)
%This function returns the coefficients of the natural cubic spline
%X and y are the input points where f(X(i)) = y(i)
%Each spline function on each interval has degree 3
```

```
%Si = a+bx+cx^2+dx^3
%We have n such Si's, where is n = length(X)-1
%coeffs should be [a1;b1;c1;d1;a2;b2;.....;an;bn;cn;dn]
%coeffs is a 4n by 1 vector

    numP = length(X);
    %numP is the number of points
    n =numP - 1;
    %n is the number of spline functions

    K = [];
    %initialize the matrix to be empty
    A = [];
    %A contains all the known values
    %K*coeffs = A

    %we construct the matrix using for loop
    for i = 1:numP
        a = (i-1)*4+1;
        b = a+1;
        c = b+1;
        d = c+1;
        %a,b,c,d are indices for the convinience of calculation
        %a,b,c,d indicate the next polynomial
        %to access the previous polynomial
        %use a-4, b-4, c-4, d-4

        if i==1
            tempK = zeros(1,4*n);
            tempK(1,c) = 2;
            tempK(1,d) = 6*X(i);
            K = [K;tempK];
            A = [A;0];
            tempK = zeros(1,4*n);
            tempK(1,a) = 1;
            tempK(1,b) = X(i);
            tempK(1,c) = X(i)^2;
            tempK(1,d) = X(i)^3;
            K = [K;tempK];
            A = [A;y(i)];
        end
```

```
if i==numP
    tempK = zeros(1,4*n);
    tempK(1,c-4) = 2;
    tempK(1,d-4) = 6*X(i);
    K = [K;tempK];
    A = [A;0];
    tempK = zeros(1,4*n);
    tempK(1,a-4) = 1;
    tempK(1,b-4) = X(i);
    tempK(1,c-4) = X(i)^2;
    tempK(1,d-4) = X(i)^3;
    K = [K;tempK];
    A = [A;y(i)];
end
%these the special end points contraints for natural cubic constraint

if i>1 && i<numP
    tempK = zeros(1,4*n);
    tempK(1,a-4) = 1;
    tempK(1,b-4) = X(i);
    tempK(1,c-4) = X(i)^2;
    tempK(1,d-4) = X(i)^3;
    K = [K;tempK];
    A = [A;y(i)];
    tempK = zeros(1,4*n);
    tempK(1,a) = 1;
    tempK(1,b) = X(i);
    tempK(1,c) = X(i)^2;
    tempK(1,d) = X(i)^3;
    K = [K;tempK];
    A = [A;y(i)];
end
%this is the constraint for S(xi) = yi

if i>1 && i<numP
    tempK = zeros(1,4*n);
    tempK(1,b-4) = 1;
    tempK(1,c-4) = 2*X(i);
    tempK(1,d-4) = 3*X(i)^2;
    tempK(1,b) = -1;
    tempK(1,c) = -2*X(i);
    tempK(1,d) = -3*X(i)^2;
```

```
            K = [K;tempK];
            A = [A;0];
            %this is the constraint for Si'(xi) = Si+1'(xi)

            tempK = zeros(1,4*n);
            tempK(1,c-4) = 2;
            tempK(1,d-4) = 6*X(i);
            tempK(1,c) = -2;
            tempK(1,d) = -6*X(i);
            K = [K;tempK];
            A = [A;0];
            %this is the constraint for Si''(xi) = Si+1''(xi)
        end
    end

    coeffs = K\A;

end
```

This code has been proved to run correctly by running the comparison between this function and the matlab built in *spline* function. Both functions generated the same results.

2 Question 2

For an interpolating polynomial P, it can be expressed as

$$P(x) = a_0p_0(x) + a_1p_1(x) + a_2p_2(x) + ... + a_{n-1}p_{n-1}(x)$$

In this question, we have four give points $(-1, -5)$, $(0, 1)$, $(1, 1)$, $(2, 1)$

- Monomial Basis

For monomial basis, we have

$$p_i(x) = x^{i-1}, \text{ for } i = 1, 2, ..., n$$

Since we have four points here, the interpolating polynomial should have form of

$$P(x) = a_0 + a_1x + a_2x^2 + a_3x^3$$

By applying matrix calculation, we get

$$a_0 = 1 \quad a_1 = 2 \quad a_2 = -3 \quad a_3 = 1$$

Therefore, $P(x) = 1 + 2x - 3x^2 + x^3$

- Lagrange Basis

For Lagrange basis, we have

$$p_i(x) = L_i(x) = \prod_{j=1,j\neq i}^{n} \frac{x-x_j}{x_i-x_j}$$

Since we have four points here, the interpolating polynomial should have form of

$$P(x) = a_0p_0(x) + a_1p_1(x) + a_2p_2(x) + a_3p_3(x)$$

Since for Lagrange basis,

$$L_i(x) = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$$

Then we have

$$P(x) = \sum_i^n L_i(x)f_i$$

In this case, we have

$$\begin{aligned} P(x) = & -5\frac{x(x-1)(x-2)}{-6} \\ & +1\frac{(x+1)(x-1)(x-2)}{2} \\ & +1\frac{(x+1)x(x-2)}{-2} \\ & +1\frac{(x+1)x(x-1)}{6} \end{aligned} \tag{2}$$

- Newton Basis

For Newton basis, we have

$$p_i(x) = N_i(x) = \prod_{j=1,j\neq i}^n (x - x_j)$$

Our interpolating function should have the form

$$\begin{aligned} P(x) = & a_0 \\ & + a_1(x-x_0) \\ & + a_2(x-x_0)(x-x_1) \\ & + a_3(x-x_0)(x-x_1)(x-x_2) \end{aligned} \tag{3}$$

After calculation we have:

$$a_0 = -5 \;\; a_1 = 6 \;\; a_2 = -3 \;\; a_3 = 1$$

Therefore,

$$\begin{aligned} P(x) = & -5 \\ & + 6(x+1) \\ & - 3(x+1)x \\ & + (x+1)x(x-1) \end{aligned} \tag{4}$$

3 Question 3

3.1 Question 3a

For $f(x) = x^3$, we have

$$f(0) = 0, \; f(1) = 1$$

Therefore, $P_n(x) = x$.

Define $\phi(t) = f(t) - P(t) - \frac{f(x)-p(x)}{\omega(x)}\omega(t)$, where $\omega(x) = \prod(x - x_i)$.

In this case, we have $\phi(t) = t^3 - t - \frac{x^3-x}{(x-0)(x-1))}(t^2 - t)$

$\phi'(t) = 3t^2 - 1 - \frac{x^3-x}{(x-0)(x-1))}(2t - 1)$

$\phi''(t) = 6t - 2\frac{x^3-x}{(x-0)(x-1))}$

$\xi = \frac{x^3-x}{3(x^2-x)}$

3.2 Question 3b

For $f(x) = (2x - 1)^4$, we have

$$f(0) = 1,\ f(1) = 1$$

Therefore, $P_n(x) = 1$.

Define $\phi(t) = f(t) - P(t) - \frac{f(x)-p(x)}{\omega(x)}\omega(t)$, where $\omega(x) = \prod(x - x_i)$

In this case, we have $\phi(t) = (2t - 1)^4 - 1 - \frac{(2x-1)^4-1}{(x-0)(x-1))}(t^2 - t)$

$\phi'(t) = 8(2t - 1)^3 - \frac{(2x-1)^4-1}{(x-0)(x-1))}(2t - 1)$

$\phi''(t) = 48(2t - 1)^2 - 2\frac{(2x-1)^4-1}{(x-0)(x-1))}$

Since ξ is in [0,1], then $\xi = \frac{\sqrt{\frac{(2x-1)^4-1}{24x(x-1)}}+1}{2}$

4 Question 4

4.1 • $f(x) = sin(\pi x)$

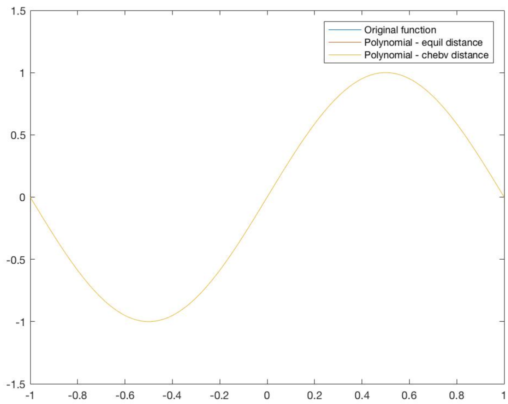

Figure 1: Polynomial Interpolation

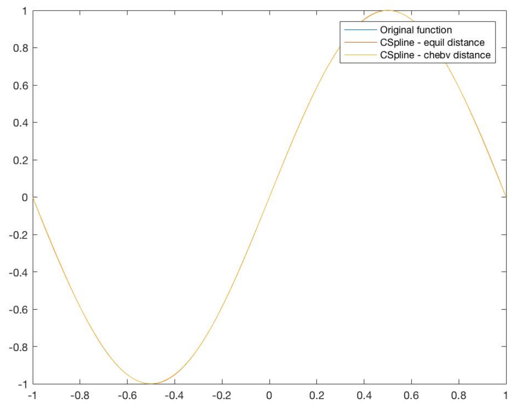

Figure 2: Spline Interpolation

4.2 • $f(x) = \frac{1}{1+25x^2}$

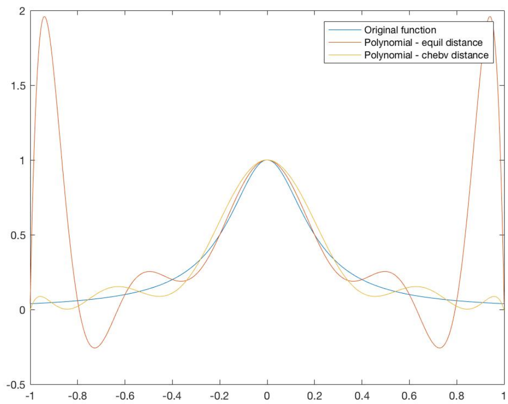

Figure 3: Polynomial Interpolation

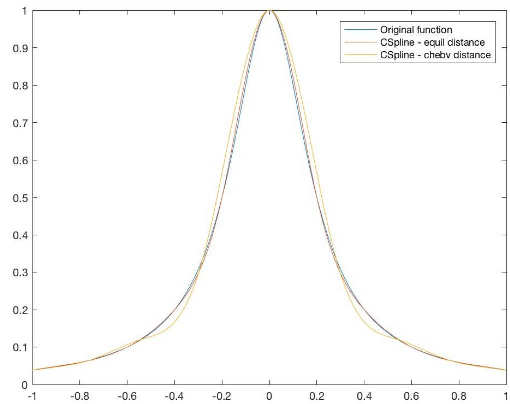

Figure 4: Spline Interpolation

4.3 • $f(x) = \mid x \mid$

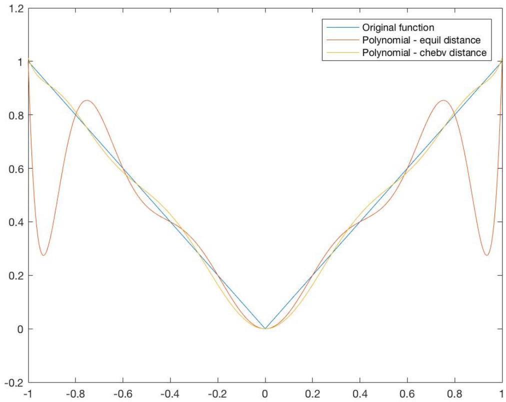

Figure 5: Polynomial Interpolation

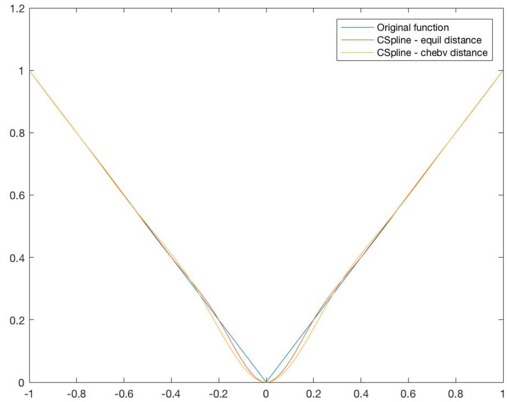

Figure 6: Spline Interpolation

By comparing between polynomial interpolation and cubic spline, we can see that cubic spline generates smaller errors. For polynomial interpolation, cheby points give much better results than equil distant points. For cubic spline interpolation, the two kinds of points don't have much difference from each other.

5 Question 5

5.1 Question 5a

Let $x = tan(\theta)$, then

$$\frac{dx}{d\theta} = sec^2\theta$$

Then

$$dx = sec^2\theta d\theta$$

Then

$$\int \frac{4}{1+x^2}dx = \int \frac{4}{1+tan^2(\theta)}sec^2\theta d\theta$$

Since

$$1 + tan^2(\theta) = sec^2(\theta)$$

Then

$$\int \frac{4}{1+x^2}dx = \int 4\ d\theta = 4\ \theta + C, \text{ where } C \text{ is a constant}$$

Since we have set $x = tan(\theta)$, then $\theta = arctan(x)$
Then we have

$$\int \frac{4}{1+x^2}dx = 4arctan(x) + \text{C}$$

Then $\int_0^1 \frac{4}{1+x^2}dx = \pi$

5.2 Question 5b

• Gauss-Legendre Quadrature

When using Gauss-Legendre Quadrature, $x_i's$ are

$$x_1 = 0.0338, x_2 = 0.1694, x_3 = 0.3807, x_4 = 0.6193, x_5 = 0.8306, x_6 = 0.9662$$

And the wrights are

$$w_1 = 0.0857, w_2 = 0.1804, w_3 = 0.2340, w_4 = 0.2340, w_5 = 0.1804, w_6 = 0.0857$$

The value of the quadrature is approximated to be **3.141592611187587**.

• Composite Trapezoidal Rule

For Gauss-Legendre Quadrature, the number of function evaluations is 6. To have the same number of function evaluations for composite trapezoidal rule, we set the number of subinterval to be 5. Then the value of the quadrature is approximated to be **3.134926113810990**.

Comparing between Gauss-Legendre Quadrature and the Composite Trapezoidal Rule, we can see that the integral approximated by Gauss-Legendre Quadrature is closer to the actual integral of the function. As we can see here, with the same number of function evaluations, Gauss-Legendre has higher accuracy than Composite Trapezoidal Rule for integral problems.

6 Question 6

$$\frac{Q(n)-Q(2n)}{Q(2n)-Q(4n)} = \frac{(\int_a^b f(x)-Q(2n))-(\int_a^b f(x)-Q(n))}{(\int_a^b f(x)-Q(4n))-(\int_a^b f(x)-Q(2n))}$$

For composite trapezoid rule

$$E(f) = -\frac{(b-a)h^2}{12}f''(\xi), \text{ where } h = \frac{b-a}{n}, \text{ n is the number of subintervals}$$

Then

$$\frac{Q(n)-Q(2n)}{Q(2n)-Q(4n)} = \frac{-h_2^2+h_1^2}{-h_4^2+h_2^2}, \text{ where } h_1 = \frac{b-a}{n}, h_2 = \frac{b-a}{2n}, h_4 = \frac{b-a}{4n}$$

Therefore,

$$\frac{Q(n)-Q(2n)}{Q(2n)-Q(4n)} \to 4, \text{ when } n \to \infty$$

7 Question 7

For Simpson's rule,

$$\int_a^b f(x) \approx \frac{b-a}{6}(f(a) + 4f(\frac{a+b}{2}) + f(b))$$

Then for composite Simpson's rule, we have:

$$\int_a^b f(x) \approx \frac{h}{3}(f_0 + 4f_1 + 2f_2 + 4f_3 + ... + 2f_{n-2} + 4f_{n-1} + f_n)$$

$$\int_a^b f(x) \approx \frac{h}{3}(f(x_0) + 2\sum_{j=1}^{\frac{n}{2}-1} f(x_{2j}) + 4\sum_{j=1}^{\frac{n}{2}} f(x_{2j-1}) + f(x_n))$$

For the error analysis, let $n = 2M$, then we have

$$E(f) = \int_a^b (f(x) - s(x))dx$$

$$E(f) = \int_a^b \frac{f^{n+1}(\xi)}{(n+1)!} \prod(x - x_i)dx$$

Then

$$E(f) = \frac{f^{n+1}(\xi)}{(n+1)!} \int_a^b \prod(x - x_i)dx, \text{ for some } a < \xi < b$$

For Simpson's Rule, we have:

$$\prod(x - x_i) = (x - a)(x - \frac{a+b}{2})^2(x - b)$$

Therefore, on one subinterval, we have error $O(h^5)$

When having composite Simpson's rule on the entire interval, we have the error being

$$O(h^4)$$

Since we loss one order of accuracy from local to global.