Michael Goforth
CAAM 550
HW 9
11/05/2021

Problem 1
See Jupyter notebook for code and results.

Problem 2

part a
Let $h_i = x_{i+1} - x_i$ and

$$\hat{x_i} = \frac{x - x_i}{h_i}$$

Then the Hermite polynomials are

$$\begin{aligned}
H_0(\hat{x}) &= (1-\hat{x})^2(1+2\hat{x}) \\
H_1(\hat{x}) &= \hat{x}^2(3-2\hat{x}) \\
h_0(\hat{x}) &= \hat{x}(1-\hat{x})^2 \\
h_1(\hat{x}) &= \hat{x}^2(\hat{x}-1) \\
H_0'(\hat{x}) &= \frac{1}{h_i}(6\hat{x}^2 - 6\hat{x})dx \\
H_1'(\hat{x}) &= \frac{1}{h_i}(-6\hat{x}^2 - 6\hat{x})dx \\
h_0'(\hat{x}) &= \frac{1}{h_i}(-3\hat{x}^2 - 4\hat{x} + 1)dx \\
h_1'(\hat{x}) &= \frac{1}{h_i}(3\hat{x}^2 - 2\hat{x})dx
\end{aligned}$$

The interpolating polynomials can then be constructed as

$$P_i(\hat{x}) = c_1 H_0(\hat{x}) + c_2 H_1(\hat{x}) + c_3 h_0(\hat{x}) + c_4 h_1(\hat{x}),\ x \in [x_i, x_i + 1]$$

In order for continuity of both the function and its derivative

$$\begin{aligned}
P_i(x_i) &= f_i \\
P_i(x_i + 1) &= f_{i+1} \\
P_i'(x_i) &= f_i' \\
P_i'(x_i + 1) &= f_{i+1}'
\end{aligned}$$

Then

$$
\begin{aligned}
P_i(x_i) &= c_1H_0(0) + c_2H_1(0) + c_3h_0(0) + c_4h_1(0) = c_1 = f_i \\
P_i(x_{i+1}) &= c_1H_0(1) + c_2H_1(1) + c_3h_0(1) + c_4h_1(1) = c_2 = f_{i+1} \\
P_i'(x_i) &= c_1H_0'(0) + c_2H_1'(0) + c_3h_0'(0) + c_4h_1'(0) = \frac{1}{h_i}c_3 = f_i' \\
P_i'(x_{i+1}) &= c_1H_0'(1) + c_2H_1'(1) + c_3h_0'(1) + c_4h_1'(1) = \frac{1}{h_i}c_4 = f_{i+1}'
\end{aligned}
$$

Combining the above equations gives the transformation

$$
\begin{aligned}
P_i(x) = f_iH_0(\frac{x-x_i}{x_{i+1}-x_i}) &+ f_{i+1}H_1(\frac{x-x_i}{x_{i+1}-x_i}) \\
&+ (x_{i+1}-x_i)f_i'h_0(\frac{x-x_i}{x_{i+1}-x_i}) + (x_{i+1}-x_i)f_{i+1}'h_1(\frac{x-x_i}{x_{i+1}-x_i})
\end{aligned}
$$

part b
See Jupyter notebook for code and results.

Problem 3

part a

$$B_n^i(x) = \binom{n}{i}(1-x)^{n-i}x^i$$

where

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

for $0 \leq i \leq n$. On the interval [0, 1], $1-x$, $n-i$, and x are all greater than or equal to zero, so the Bernstein polynomials also are greater than or equal to zero.
From the binomial theorem:

$$(x+y)^n = \sum_{i=0}^{n}\binom{n}{i}x^iy^{n-i}$$

if $y = 1-x$ then it is easy to see from the binomial theorem that the sum of the Bernstein Polynomials is

$$\sum_{i=0}^{n} B_i^n(x) = \sum_{i=0}^{n}\binom{n}{i}x^i(x-1)^{n-i} = (x+1-x)^n = 1$$

part b

$$B_i^n(x) = \binom{n}{i}(1-x)^{n-i}x^i$$
$$xB_{i-1}^{n-1}(x) = \binom{n-1}{i-1}(1-x)^{n-i}x^{i-1}$$
$$(1-x)B_i^{n-1}(x) = \binom{n-1}{i}(1-x)^{n-i}x^i$$
$$\binom{n-1}{i-1} = \frac{(n-1)!}{(i-1)!(n-i)!} = \frac{i}{n}\frac{n!}{i!(n-i)!}$$
$$\binom{n-1}{i} = \frac{(n-1)!}{i!(n-i-1)!} = \frac{n-i}{n}\frac{n!}{i!(n-i)!}$$
$$\binom{n-1}{i} + \binom{n-1}{i} = \frac{n!}{i!(n-i)!} = \binom{n}{i}$$

So then

$$xB_{n-1}^{i-1}(x) + (1-x)B_{n-1}^{i}(x) = B_n^i(x)$$

part c

Let

$$B(x) = \sum_{i=0}^{n} b_i B_i^n$$

be a polynomial of degree n written in the Bernstein basis with coefficients $b_0, b_1, \ldots, b_n$. Let $T \in \mathbb{R}^{(n+1)\times(n+1)}$ be the transformation matrix that converts the coefficients of the Bernstein basis into coefficients of the monomial basis. Then

$$T\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix}$$

where $c_0, c_1, \ldots, c_n$ are the coefficients in the monomial basis such that

$$B(x) = \sum_{i=0}^{n} b_i B_i^n = \sum_{i=0}^{n} c_i x^i$$

Consider the j-th term in the Bernstein polynomial sum above.

$$c_j B_j^n = \binom{n}{j}(1-x)^{n-j}x^j$$

Using the binomial theorem to expand $(1-x)^{n-j}$ gives

$$c_j B_j^n = c_j \binom{n}{j} x^j \sum_{l=0}^{n-j} \binom{n-j}{l} (-x)^{n-j-l}$$
$$c_j B_j^n = c_j \binom{n}{j} \sum_{l=0}^{n-j} \binom{n-j}{l} (-1)^{n-j-l} x^{n-l}$$

which converts B_j^n into a sum of the monomial basis vectors where the l-th term in the sum corresponds to the $(n-l)$-th monomial. Then

$$c_i = \sum_{j=0}^{n} c_j \binom{n}{j} \binom{n-j}{n-i} (-1)^{i-j}$$

From this sum it can be seen that the elements in the transformation matrix T can be expressed as

$$T_{i,j} = \binom{n}{j} \binom{n-j}{n-i} (-1)^{i-j} \text{ if } j \leq i$$
$$T_{i,j} = 0 \text{ if } j > i$$

where indices i, j range from 0 to n.
Further

$$\binom{n}{j} \binom{n-j}{n-1} = \frac{n!}{j!(n-j)!} \frac{(n-j)!}{(n-i)!(n-j-(n-i))!}$$
$$\binom{n}{j} \binom{n-j}{n-1} = \frac{n!}{j!(n-i)!(i-j)!}$$

For example, for $n = 3$:

$$T = \begin{bmatrix} T_{0,0} & T_{0,1} & T_{0,2} & T_{0,3} \\ T_{1,0} & T_{1,1} & T_{1,2} & T_{1,3} \\ T_{2,0} & T_{2,1} & T_{2,2} & T_{2,3} \\ T_{3,0} & T_{3,1} & T_{3,2} & T_{3,3} \end{bmatrix}$$

$$T = \begin{bmatrix} \frac{n!}{0!(n-0)!(0-0)!}(-1)^{0-0} & 0 & 0 & 0 \\ \frac{3!}{0!(3-1)!(1-0)!}(-1)^{1-0} & \frac{3!}{1!(3-1)!(1-1)!}(-1)^{1-1} & 0 & 0 \\ \frac{3!}{0!(3-2)!(2-0)!}(-1)^{2-0} & \frac{3!}{1!(3-2)!(2-1)!}(-1)^{2-1} & \frac{3!}{2!(3-2)!(2-2)!}(-1)^{2-2} & 0 \\ \frac{3!}{0!(3-3)!(3-0)!}(-1)^{3-0} & \frac{3!}{1!(3-3)!(3-1)!}(-1)^{3-1} & \frac{3!}{2!(3-3)!(3-2)!}(-1)^{3-2} & \frac{3!}{3!(3-3)!(3-3)!}(-1)^{3-3} \end{bmatrix}$$

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}$$

Problem 4
part i
See Jupyter notebook for code and results.

part ii
See Jupyter notebook for code and results.

part iv (iii?)
See Jupyter notebook for code and results.

Problem 5

part i
This method leads to a system of equations that can be represented as:

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ x_0 & x_1 & x_2 & \dots & x_n \\ x_0^2 & x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_0^n & x_1^n & x_2^n & \dots & x_n^n \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} b-a \\ 1/2(b^2-a^2) \\ 1/3(b^3-a^3) \\ \vdots \\ \frac{1}{n+1}(b^{n+1}-a^{n+1} \end{bmatrix}$$

The first equation is the transpose of the Vandermonde matrix, and similar to the Vandermonde matrix, this matrix is ill-conditioned so computing quadrature weights with this method could lead to large errors.

part ii
See Jupyter notebook for code and results.
For $n = 10$ and $n = 15$ negative weights occur. This is not surprising due to the poor conditioning of the problem. $n = 10$ and $n = 15$ also show large error (see part iv results for closer approximations).

part iii
Based on our discussion of polynomial interpolation, Chebyshev Points will give better quadrature than equidistant points.

$$x_i = \frac{1}{2}(a+b) + \frac{1}{2}(b-a)\cos(\frac{(2i-1)\pi}{2n})$$

See Jupyter notebook for code and results.
There are no longer negative weights used for approximations when $n = 10$ and $n = 15$. Results are also more consistent.

part iv
See Jupyter notebook for code and results.