104A: Homework 4

Daniel Naylor

Question 1

Observe that

$$\begin{aligned} c_0 &= \sum_{j=0}^{N-1} f_j e^{-2\pi i \frac{j0}{N}} \\ &= \sum_{j=0}^{N-1} f_j \end{aligned}$$

Since all f_j are real, c_0 is clearly real.

Now consider c_{N-k}

$$\begin{aligned} c_{N-k} &= \sum_{j=0}^{N-1} f_j e^{-2\pi i \frac{j(N-k)}{N}} \\ &= \sum_{j=0}^{N-1} f_j e^{-2\pi i j\left(1-\frac{k}{N}\right)} \\ &= \sum_{j=0}^{N-1} f_j e^{2\pi i j \frac{k}{N}} \cdot e^{-2\pi i j} \end{aligned}$$

Note that $e^{-2\pi i j} = 1$ since j is an integer.

$$\begin{aligned} &= \sum_{j=0}^{N-1} f_j e^{2\pi i j \frac{k}{N}} \\ &= \sum_{j=0}^{N-1} f_j \left(\cos\left(2\pi \frac{jk}{N}\right) + i \sin\left(2\pi \frac{jk}{N}\right)\right) \\ \bar{c}_{N-k} &= \sum_{j=0}^{N-1} f_j \left(\cos\left(2\pi \frac{jk}{N}\right) - i \sin\left(2\pi \frac{jk}{N}\right)\right) \\ &= \sum_{j=0}^{N-1} f_j e^{-2\pi i \frac{jk}{N}} \\ &= c_k \end{aligned}$$

Question 2

Let $\{c_k\}_{k=0}^{N-1}$ be the computed FFT coefficients. We know that

$$c_k = \sum_{j=0}^{N-1} f_j e^{-2\pi i \frac{jk}{N}} = \sum_{j=0}^{N-1} f_j \cos\left(2\pi \frac{jk}{N}\right) - i f_j \sin\left(2\pi \frac{jk}{N}\right)$$

From Question 1 we can deduce that

$$\begin{aligned}
c_k + c_{N-k} &= \sum_{j=0}^{N-1} f_j \cos\left(2\pi \frac{jk}{N}\right) - i f_j \sin\left(2\pi \frac{jk}{N}\right) + \sum_{j=0}^{N-1} f_j \cos\left(2\pi \frac{jk}{N}\right) + i f_j \sin\left(2\pi \frac{jk}{N}\right) \\
&= \sum_{j=0}^{N-1} f_j \cos\left(2\pi \frac{jk}{N}\right) - i f_j \sin\left(2\pi \frac{jk}{N}\right) + f_j \cos\left(2\pi \frac{jk}{N}\right) + i f_j \sin\left(2\pi \frac{jk}{N}\right) \\
&= \sum_{j=0}^{N-1} 2 f_j \cos\left(2\pi \frac{jk}{N}\right)
\end{aligned}$$

Recall that $x_j = j \frac{2\pi}{N}$

$$\begin{aligned}
c_k + c_{N-k} &= 2 \sum_{j=0}^{N-1} f_j \cos(k x_j) \\
&= 2 \sum_{j=1}^{N-1} f_j \cos(k x_j) + 2 f_0
\end{aligned}$$

Hence $a_k = \frac{c_k + c_{N-k} - 2f_0}{N}$. Also observe that $k > N/2$ is redundant by symmetry.

Now observe

$$\begin{aligned}
c_k - c_{N-k} &= \sum_{j=0}^{N-1} f_j \cos\left(2\pi \frac{jk}{N}\right) - i f_j \sin\left(2\pi \frac{jk}{N}\right) - \sum_{j=0}^{N-1} f_j \cos\left(2\pi \frac{jk}{N}\right) + i f_j \sin\left(2\pi \frac{jk}{N}\right) \\
&= \sum_{j=0}^{N-1} f_j \cos\left(2\pi \frac{jk}{N}\right) - i f_j \sin\left(2\pi \frac{jk}{N}\right) - f_j \cos\left(2\pi \frac{jk}{N}\right) - i f_j \sin\left(2\pi \frac{jk}{N}\right) \\
&= \sum_{j=0}^{N-1} -2 i f_j \sin\left(2\pi \frac{jk}{N}\right) \\
&= \sum_{j=0}^{N-1} -2 i f_j \sin(k x_j) \\
&= -2i \sum_{j=1}^{N-1} f_j \sin(k x_j)
\end{aligned}$$

So we see that

$$\begin{aligned}
c_{N-k} - c_k &= 2i \sum_{j=1}^{N-1} f_j \sin(k x_j) \\
\implies \frac{c_{N-k} - c_k}{iN} &= \frac{2}{N} \sum_{j=1}^{N-1} f_j \sin(k x_j)
\end{aligned}$$

Thus $b_k = \frac{c_{N-k} - c_k}{iN}$

Question 3

See attached PDF.

Question 4

(a)

Let $\{\psi_j\}_{j=0}^n$ be a collection of orthogonal jth degree polynomials forming a basis for $\mathcal{P}^n$. For any function f, we can construct a Least Squares Approximation polynomial P_n as such

$$P_n(x) = \sum_{j=0}^{n} \frac{\langle f, \psi_j \rangle}{\langle \psi_j, \psi_j \rangle} \psi_j(x),$$

Where $\langle \cdot, \cdot \rangle$ denotes an inner product. Consider

$$f(x) - P_n(x) = f(x) - \sum_{j=0}^{n} \frac{\langle f, \psi_j \rangle}{\langle \psi_j, \psi_j \rangle} \psi_j(x)$$

Now take any polynomial $q \in \mathcal{P}^n$. Note that

$$\langle f - P_n, q \rangle = \langle f, q \rangle - \langle P_n, q \rangle$$

Since q is a polynomial of at most degree n, we can find a linear combination of ψ_j such that

$$q(x) = \sum_{j=0}^{n} c_j \psi_j(x)$$

Hence

$$\begin{aligned}
\langle f, q \rangle - \langle P_n, q \rangle &= \langle f, c_1\psi_1 \rangle + \cdots + \langle f, c_n\psi_n \rangle - \langle P_n, c_1\psi_1 \rangle - \cdots - \langle P_n, c_n\psi_n \rangle \\
&= (\langle f, c_1\psi_1 \rangle - \langle P_n, c_1\psi_1 \rangle) + \cdots + (\langle f, c_n\psi_n \rangle - \langle P_n, c_n\psi_n \rangle) \\
&= c_1 (\langle f, \psi_1 \rangle - \langle P_n, \psi_1 \rangle) + \cdots + c_n (\langle f, \psi_n \rangle - \langle P_n, \psi_n \rangle)
\end{aligned}$$

So it suffices to show that $\langle f, \psi_j \rangle = \langle P_n, \psi_j \rangle$ for all $j = 0, \ldots, n$.

Indeed, since P_n is a linear combination of ψ_j, and that $\langle \psi_j, \psi_k \rangle = 0$ for $j \neq k$, we see that

$$\langle P_n, \psi_j \rangle = \frac{\langle f, \psi_j \rangle}{\langle \psi_j, \psi_j \rangle} \cdot \langle \psi_j, \psi_j \rangle = \langle f, \psi_j \rangle$$

(b)

If $\langle a - b, c \rangle = 0$ for vectors $a, b, c \in \mathbb{R}^k$ (under Euclidean dot product), then this is equivalent to saying that b is the projection of a onto c.

Similarly, $\langle f - P_n, q \rangle = 0$ is saying that P_n is the projection of f onto q; in other words, P_n is f's projection onto any polynomial of degree $\leq n$.

Question 5

(a)

Using the inner product

$$\langle f, g\rangle = \int_{-1}^{1} f(x)g(x)dx$$

We see that $\psi_0 = 1$ and $\psi_1 = x$ are orthogonal polynomials over $[-1, 1]$. To obtain the remaining two polynomials, we may apply the following recursive definition:

$$\psi_{k+1} = x\psi_k - \frac{\langle x\psi_k, \psi_k\rangle}{\langle \psi_k, \psi_k\rangle}\psi_k - \frac{\langle \psi_k, \psi_k\rangle}{\langle \psi_{k-1}, \psi_{k-1}\rangle}\psi_{k-1}$$

From this we see

$$\begin{aligned}
\psi_2 &= x\psi_1 - \frac{\langle x\psi_1, \psi_1\rangle}{\langle \psi_1, \psi_1\rangle}\psi_1 - \frac{\langle \psi_1, \psi_1\rangle}{\langle \psi_0, \psi_0\rangle}\psi_0 \\
&= x^2 - \frac{\langle x^2, x\rangle}{\langle x, x\rangle}x - \frac{\langle x, x\rangle}{\langle 1, 1\rangle} \\
&= x^2 - \frac{2/3}{2} \\
&= x^2 - \frac{1}{3}
\end{aligned}$$

Similarly,

$$\begin{aligned}
\psi_3 &= x\psi_2 - \frac{\langle x\psi_2, \psi_2\rangle}{\langle \psi_2, \psi_2\rangle}\psi_2 - \frac{\langle \psi_2, \psi_2\rangle}{\langle \psi_1, \psi_1\rangle}\psi_1 \\
&= x^3 - \frac{x}{3} - \frac{\langle x^3 - x/3, x^2 - 1/3\rangle}{\langle x^2 - 1/3, x^2 - 1/3\rangle}\left(x^2 - \frac{1}{3}\right) - \frac{\langle x^2 - 1/3, x^2 - 1/3\rangle}{\langle x, x\rangle}x \\
&= x^3 - \frac{x}{3} - \frac{8/45}{2/3}x \\
&= x^3 - \frac{x}{3} - \frac{4}{15}x \\
&= x^3 - \frac{3}{5}x
\end{aligned}$$

(b)

We know that

$$P_n(x) = \sum_{j=0}^{n} \frac{\langle e^x, \psi_j\rangle}{\langle \psi_j, \psi_j\rangle}\psi_j(x),$$

Hence, for $f(x) = e^x$

$$\begin{aligned}
P_1(x) &= \frac{\langle e^x, 1\rangle}{\langle 1, 1\rangle} + \frac{\langle e^x, x\rangle}{\langle x, x\rangle}x \\
&= \frac{e - e^{-1}}{2} + \frac{3}{e}x
\end{aligned}$$

For P_2,

$$\begin{aligned}
P_2(x) &= \frac{\langle e^x, 1\rangle}{\langle 1, 1\rangle} + \frac{\langle e^x, x\rangle}{\langle x, x\rangle}x + \frac{\langle e^x, x^2 - \frac{1}{3}\rangle}{\langle x^2 - \frac{1}{3}, x^2 - \frac{1}{3}\rangle}\left(x^2 - \frac{1}{3}\right) \\
&= \frac{e - e^{-1}}{2} + \frac{3}{e}x + \left(\frac{15e^2 - 105}{4e}\right)\left(x^2 - \frac{1}{3}\right)
\end{aligned}$$

And for P_3

$$\begin{aligned}
P_3(x) &= \frac{\langle e^x, 1\rangle}{\langle 1, 1\rangle} + \frac{\langle e^x, x\rangle}{\langle x, x\rangle}x + \frac{\langle e^x, x^2 - \frac{1}{3}\rangle}{\langle x^2 - \frac{1}{3}, x^2 - \frac{1}{3}\rangle}\left(x^2 - \frac{1}{3}\right) + \frac{\langle e^x, x^3 - \frac{3}{5}x\rangle}{\langle x^3 - \frac{3}{5}x, x^3 - \frac{3}{5}x\rangle}\left(x^3 - \frac{3}{5}x\right) \\
&= \frac{e - e^{-1}}{2} + \frac{3}{e}x + \left(\frac{15e^2 - 105}{4e}\right)\left(x^2 - \frac{1}{3}\right) + \left(\frac{2590 - 350e^2}{8e}\right)\left(x^3 - \frac{3}{5}x\right)
\end{aligned}$$

(c)

The polynomial would be x^3. This is because $\{\psi_j\}_{j=0}^4$ forms a basis for polynomials up to degree 4, hence the polynomial least squares approximation of at least degree 3 would simply be itself.

```python
'''
Daniel Naylor
5094024
11/23/2022
'''

# ------------------------------------------------------------------
#                            Question 3
# ------------------------------------------------------------------

import numpy as np
from numpy.fft import fft
from math import exp, sin, cos, pi, e
N=8 # number of data points


input_data = [
    exp(sin(j * 2 * pi / N))
    for j in range(N)
]

c = fft(input_data)


# note that f_0 = 1

# discrete 'a' coefficients
a = {
    0: 2/N * sum(input_data[1:])
}
for k in range(1, N//2 + 1):
    a[k] = np.real((c[N-k] + c[k] - 2)/(N))

# discrete 'b' coefficients
b = {
    0: 0
}
for k in range(1, N//2 + 1):
    b[k] = np.real((c[N-k] - c[k])/(1j * N))

def P_prime(x: float, /):
    '''
    The derivative of P_8
    '''
    fourier_sum = 0
    for k in range(1, N//2):
        fourier_sum += k * b[k] * cos(k * x) - k * a[k] * sin(k * x)

    return fourier_sum - 2 * a[4] * sin(4 * x)

def f_prime(x: float, /):
    '''
    The derivative of e^(sin(x))
    '''
    return cos(x) * e**(sin(x))

for j in range(8):
    x_j = j * 2 * pi / N
    print(f'Error for x_j = {j} * 2pi / N: {P_prime(x_j) - f_prime(x_j)}')

# Error for x_j = 0 * 2pi / N: -0.004317911098587035
# Error for x_j = 1 * 2pi / N: 1.2105558808948307
# Error for x_j = 2 * 2pi / N: -0.5000000000000002
# Error for x_j = 3 * 2pi / N: 0.20365768147826468
# Error for x_j = 4 * 2pi / N: 0.004317911098587146
# Error for x_j = 5 * 2pi / N: -0.2098362525369506
# Error for x_j = 6 * 2pi / N: 0.49999999999999994
```

```
# Error for x_j = 7 * 2pi / N: -1.2043773098361448
```