

MATH 4640 Numerical Analysis - HW 3 Solutions

Austin Barton

March 27, 2025

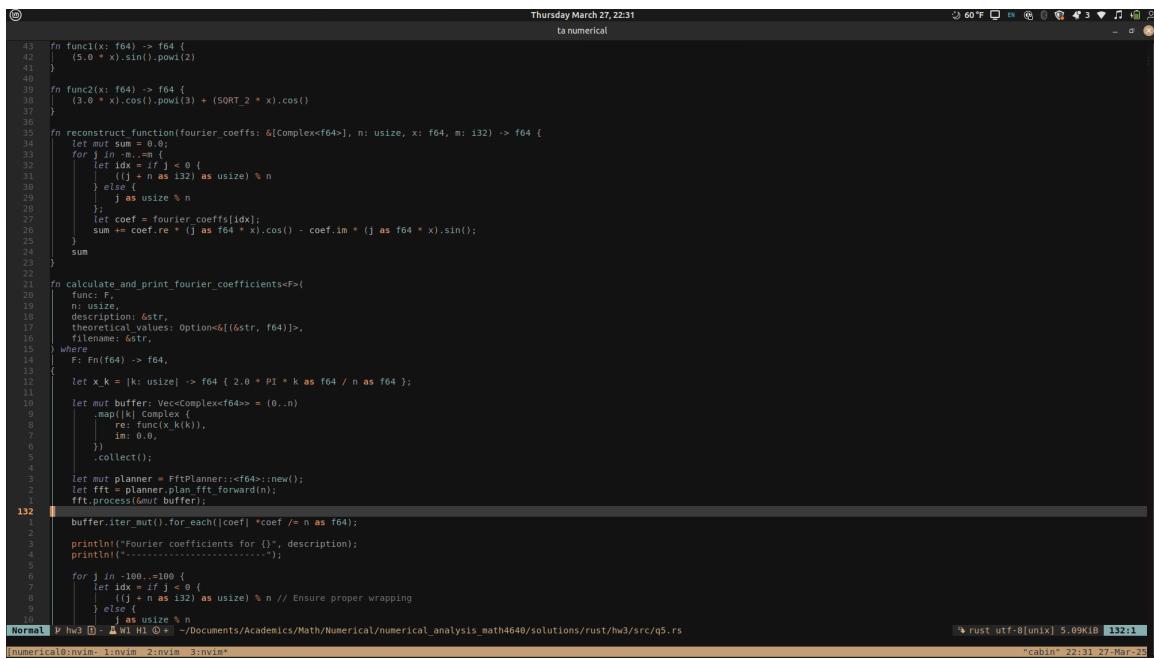
Problem 1: Please see attached PDF with all written solutions for questions 1-4 and 6-7.

Problem 2: Please see attached PDF with all written solutions for questions 1-4 and 6-7.

Problem 3: Please see attached PDF with all written solutions for questions 1-4 and 6-7.

Problem 4: Please see attached PDF with all written solutions for questions 1-4 and 6-7.

Problem 5:



The screenshot shows a terminal window titled "to numerical" with the following content:

```
Thursday March 27, 22:31
```

```
43 fn func1(x: f64) -> f64 {
44     (5.0 * x).sin().powi(2)
45 }
46
47 fn func2(x: f64) -> f64 {
48     (5.0 * x).cos().powi(3) + (SQRT_2 * x).cos()
49 }
50
51 fn reconstruct_function(fourier_coeffs: &[Complex<f64>], n: usize, x: f64, m: i32) -> f64 {
52     let mut sum = 0.0;
53     for i in 0..m {
54         let idx = if j < 0 {
55             ((j + n as i32) as usize) % n
56         } else {
57             j as usize % n
58         };
59         let coef = fourier_coeffs[idx];
60         sum += coef.re * (j as f64 * x).cos() - coef.im * (j as f64 * x).sin();
61     }
62     sum
63 }
64
65 fn calculate_and_print_fourier_coefficients<F>(
66     func: F,
67     n: usize,
68     description: &str,
69     numerical_values: Option<&[(&str, f64)]>,
70     filename: &str,
71 ) where
72     F: Fn(f64) -> f64,
73 {
74     let x_k = |k: usize| -> f64 { 2.0 * PI * k as f64 / n as f64 };
75
76     let mut buffer: Vec<Complex<f64>> = (0..n
77         .map(|k| Complex {
78             re: func(x_k(k)),
79             im: 0.0,
80         })
81         .collect();
82
83     let mut planner = FftPlanner::new();
84     let fft = planner.plan_fft(forward);
85     fft.process(&mut buffer);
86
87     buffer.iter_mut().for_each(|coef| *coef /= n as f64);
88
89     println!("Fourier coefficients for {}", description);
90     println!("-----");
91
92     for j in -100..=100 {
93         let idx = if j < 0 {
94             ((j + n as i32) as usize) % n // Ensure proper wrapping
95         } else {
96             j as usize % n
97         };
98         let coef = &mut buffer[idx];
99         print!("{} ", coef.re);
100    }
101 }
```

The terminal window has a title bar "Thursday March 27, 22:31" and "to numerical". The status bar at the bottom right shows "rust utf-8(unix) 5.09KIB 132:1" and "cabin" 22:31 27-Mar-23". The command prompt is "(numerical@ovim: ~hw3 [0] - □ W1 H1 ⊕ + -Documents/Academics/Math/Numerical/numerical_analysis_math4640/solutions/rust/hw3/src/q5.rs".

Figure 1: Screenshot of code for calculating fourier coefficients using FFT.

```
Thursday March 27, 22:32  
to numerical

19  
9 let mut planner = FftPlanner::new();  
8 let fft = planner.plan_fft_forward(n);  
7 fft.process(&mut buffer);  
6  
7 buffer.iter_mut().for_each(|coef| *coef /= n as f64);  
8  
9 println!("Fourier coefficients for {}:", description);  
10 println!("-----");
138 11 for j in -100..=100 {  
12     let idx = if j < 0 {  
13         |j + n as i32| as usize % n // Ensure proper wrapping  
14     } else {  
15         |j| as usize % n  
16     };  
17  
18     let threshold = 1e-12;  
19     if buffer[idx].re.abs() > threshold || buffer[idx].im.abs() > threshold {  
20         println!(  
21             "f hat({:3}): {:.12}+{:12}i",  
22             j, buffer[idx].re, buffer[idx].im  
23         );
24     }
25  
26     plot_functions(func, &buffer, filename); // use `let _ = ...` to ignore the resulting value: `let _ =`  
27  
28     if let Some(values) = theoretical_values {  
29         println!("Theoretical values: -");  
30         for (label, value, values) in  
31             values {  
32                 println!("{} = {:.12}, label, value, values");
33             }
34         println!("All other coefficients = 0");
35     }
36  
37     println!("-----");
38 }
39  
fn solve(q5) -> Result<(), Box> {
40     let n = 200;
41  
42     let func1_theory = [  
43         ("f_hat(0)", 0.5),  
44         ("f_hat(10)", -0.25),  
45         ("f_hat(-10)", -0.25),
46     ];
47  
48     calculate_and_print_fourier_coefficients(
49         func1,
50         n,
51         |fx| fx == sin(25x),
52         Some(&func1_theory),
53         "plots/first-func-plot.png",
54     );
55 }
```

Figure 2: Screenshot of code for calculating fourier coefficients using FFT.

```
Thursday March 27, 22:32  
to numerical  
18 }  
19  
20 pub fn plot_functions(func: &buffer, filename): ■■ use 'let _ = ...' to ignore the resulting value: 'let _ = '  
21 {  
22     if let Some(values) = theoretical_values {  
23         println!("n--Theoretical Values--");  
24         for (label, value) in values {  
25             println!("{} = {:.12}", label, value);  
26         }  
27         println!("All other coefficients = 0");  
28     }  
29     println!("-~-~-~-~-~-");  
30 }  
31  
32 fn solve_q5() -> Result<(), Box33     let n = 200;  
34  
35     let func1_theory = [  
36         ("f_hat(0)", 0.5),  
37         ("f_hat(10)", -0.25),  
38         ("f_hat(-10)", -0.25),  
39     ];  
40  
41     calculate_and_print_fourier_coefficients(  
42         func1,  
43         n,  
44         "f(x) = sin^2(5x)",  
45         Some(&func1_theory),  
46         "plots/first-func-plot.png",  
47     );  
48  
49     calculate_and_print_fourier_coefficients(  
50         func2,  
51         n,  
52         "f(x) = cos^2(3x) + cos(\sqrt{2}x)",  
53         None,  
54         "plots/second-func-plot.png",  
55     );  
56  
57     Ok(())
58 }
59
60 pub fn main() {
61     if let Err(e) = solve_q5() {
62         eprintln!("Error: {}", e);
63         std::process::exit(1);
64     }
65 }
```

Figure 3: Screenshot of code for calculating fourier coefficients using FFT.

```

warning: 'hw0' (bin "hw0") generated 4 warnings
  Finished 'dev' profile [unoptimized + debuginfo] target(s) in 1.73s
    Running target/debug/hw0 q3
---MATH 4640, Numerical Analysis, Homework 3---

-----
Running solution to question 5...
Fourier coefficients for f(x) = sin^2(5x)

f_hat(10): -0.250000000000 + 1.0e-000000000000i
f_hat( 0): 0.500000000000 + 1.0e-000000000000i
f_hat(-10): -0.250000000000 + 1.0e-000000000000i
Plot saved to plots/first-func-plot.png

--Theoretical Values--
f_hat(0) = 0.500000000000
f_hat(10) = -0.250000000000
f_hat(-10) = -0.250000000000
All other coefficients = 0

-----
Fourier coefficients for f(x) = cos^2(3x) + cos(\sqrt{2}x)

f_hat(-100): 0.00461792771 + 1.0e-000000000000i
f_hat(-99): 0.004617622732 + 1.0e-000073014015i
f_hat(-98): 0.0046176016008 + 1.0e-000146641380i
f_hat(-97): 0.004616966333 + 1.0e-000219186514i
f_hat(-96): 0.004616653889 + 1.0e-000295174820i
f_hat(-95): 0.004616552089 + 1.0e-000375153564i
f_hat(-94): 0.004616774888 + 1.0e-000439351577i
f_hat(-93): 0.004616682998 + 1.0e-000513128705i
f_hat(-92): 0.00461674537 + 1.0e-000587162568i
f_hat(-91): 0.004616729998 + 1.0e-000662130300i
f_hat(-90): 0.004616114208 + 1.0e-000736153655i
f_hat(-89): 0.004616160893 + 1.0e-000811189033i
f_hat(-88): 0.00461591752 + 1.0e-000886637624i
f_hat(-87): 0.00461577108 + 1.0e-000962050000i
f_hat(-86): 0.004615694527 + 1.0e-001038939545i
f_hat(-85): 0.004615385619 + 1.0e-001115877897i
f_hat(-84): 0.004615149248 + 1.0e-001193399701i
f_hat(-83): 0.004614905594 + 1.0e-001271350594i
f_hat(-82): 0.004614622913 + 1.0e-00135036703i
f_hat(-81): 0.004614329998 + 1.0e-0014299727154i
f_hat(-80): 0.004614018174 + 1.0e-001510251421i
f_hat(-79): 0.0046136304956 + 1.0e-001591481242i
f_hat(-78): 0.004613332278 + 1.0e-001673430169i
f_hat(-77): 0.004612956586 + 1.0e-001756393802i
f_hat(-76): 0.004612579711 + 1.0e-001840349954i
f_hat(-75): 0.004612135288 + 1.0e-001925358824i
f_hat(-74): 0.004611771902 + 1.0e-002010831808i
f_hat(-73): 0.0046114113902 + 1.0e-002098788602i
f_hat(-72): 0.004610712828 + 1.0e-002187343618i
f_hat(-71): 0.004610183100 + 1.0e-002277220024i
f_hat(-70): 0.004609623330 + 1.0e-002366710000i
f_hat(-69): 0.004609141376 + 1.0e-0024555541376i
f_hat(-68): 0.004608407439 + 1.0e-002555549472i
[numerical0:nvim- 1:nvim- 2:nvim- 3:[tmux]*]

```

Figure 4: Screenshot of result of code for calculating fourier coefficients using FFT.

```

Fourier coefficients for f(x) = cos^2(3x) + cos(\sqrt{2}x)

f_hat(-100): 0.00461792771 + 1.0e-000000000000i
f_hat(-99): 0.004617622732 + 1.0e-000073014015i
f_hat(-98): 0.0046176016008 + 1.0e-000146641380i
f_hat(-97): 0.004616966333 + 1.0e-000219186514i
f_hat(-96): 0.004616653889 + 1.0e-000295174820i
f_hat(-95): 0.004616552089 + 1.0e-000375153564i
f_hat(-94): 0.004616774888 + 1.0e-000439351577i
f_hat(-93): 0.004616682998 + 1.0e-000513128705i
f_hat(-92): 0.00461674537 + 1.0e-000587162568i
f_hat(-91): 0.004616729998 + 1.0e-000662130300i
f_hat(-90): 0.004616114208 + 1.0e-000736153655i
f_hat(-89): 0.004616160893 + 1.0e-000811189033i
f_hat(-88): 0.004614018174 + 1.0e-001510251421i
f_hat(-79): 0.0046136304956 + 1.0e-001591481242i
f_hat(-78): 0.004613332278 + 1.0e-001673430169i
f_hat(-77): 0.004612956586 + 1.0e-001756393802i
f_hat(-76): 0.004612579711 + 1.0e-001840349954i
f_hat(-75): 0.004612135288 + 1.0e-001925358824i
f_hat(-74): 0.004611771902 + 1.0e-002010831808i
f_hat(-73): 0.0046114113902 + 1.0e-002098788602i
f_hat(-72): 0.004610712828 + 1.0e-002187343618i
f_hat(-71): 0.004610183100 + 1.0e-002277220024i
f_hat(-70): 0.004609623330 + 1.0e-002366710000i
f_hat(-69): 0.004609141376 + 1.0e-0024555541376i
f_hat(-68): 0.004608407439 + 1.0e-002555549472i
[numerical0:nvim- 1:nvim- 2:nvim- 3:[tmux]*]

```

Figure 5: Screenshot of result of code for calculating fourier coefficients using FFT.

```
Thursday March 27, 22:45  
59°F 10 4 3 0 0  
22:08:22 [88/210]  
  
f(hat,-29): 0.004487921679 + 10.089999219029  
f(hat,-27): 0.008476772248 + 10.010318796372  
f(hat,-26): 0.004464302313 + 10.010768897821  
f(hat,-25): 0.00845629552 + 10.0112532460539  
f(hat,-24): 0.004410561983 + 10.012324871717  
f(hat,-23): 0.008410561983 + 10.012342807198  
f(hat,-22): 0.004396116996 + 10.012595209921  
f(hat,-21): 0.008437266932 + 10.013632626899  
f(hat,-20): 0.004396116996 + 10.014032624465  
f(hat,-19): 0.008431045494 + 10.015187234163  
f(hat,-18): 0.008427709394 + 10.016992061998  
f(hat,-17): 0.008423335840 + 10.0171024254598  
f(hat,-16): 0.008420086598 + 10.0178124257868  
f(hat,-15): 0.008417785773 + 10.019326855050  
f(hat,-14): 0.008404642758 + 10.021080695867  
f(hat,-13): 0.008394415988 + 10.022784884666  
f(hat,-12): 0.008383757307 + 10.024380654522  
f(hat,-11): 0.0083726551794 + 10.027665581293  
f(hat,-10): 0.00835457108622 + 10.029934323273  
f(hat,-9): 0.12817358990 + 10.03347157693  
f(hat,-8): 0.00823865322 + 10.038195737300  
f(hat,-7): 0.008177801322 + 10.043004124007  
f(hat,-6): 0.0081238065958 + 10.052044124148  
f(hat,-5): -0.0080387073921 + 10.061705515199  
f(hat,-4): -0.00803616146480 + 10.084481037733  
f(hat,-3): -0.00803416146480 + 10.101377041415  
f(hat,-2): -0.0531292035883 + 10.259596565292  
f(hat,-1): 0.120166517395 + 1.0.295768618077  
f(hat, 0): 0.062401278907 + 10.08090908080809  
f(hat, 1): 0.008222375307 + 1.0.0817024254597  
f(hat, 2): 0.0051792035883 + 1.0.226595635292  
f(hat, 3): 0.363131679683 + 1.0.126674563415  
f(hat, 4): -0.003616146480 + 1.0.084491037733  
f(hat, 5): 0.003616146480 + 1.0.084491037733  
f(hat, 6): 0.0031292035883 + 1.0.052044124148  
f(hat, 7): 0.002177919813 + 1.0.043870599887  
f(hat, 8): 0.00277201322 + 1.0.0379657373808  
f(hat, 9): 0.120166517395 + 1.0.033477041415  
f(hat, 10): 0.00836514578652 + 1.0.033477041415  
f(hat, 11): 0.0036651365677 + 1.0.027065952943  
f(hat, 12): 0.008322375307 + 1.0.024699864522  
f(hat, 13): 0.00834427580 + 1.0.022708064600  
f(hat, 14): 0.00834427580 + 1.0.022708064600  
f(hat, 15): 0.0084117857793 + 1.0.019526889650  
f(hat, 16): 0.0084181072469 + 1.0.018238669786  
f(hat, 17): 0.008423335840 + 1.0.017102425698  
f(hat, 18): 0.008420086598 + 1.0.0167802425698  
f(hat, 19): 0.008410561984 + 1.0.015187234103  
f(hat, 20): 0.008434565999 + 1.0.014371761045  
f(hat, 21): 0.008437266932 + 1.0.013632624895  
f(hat, 22): 0.008431045494 + 1.0.013032624895  
f(hat, 23): 0.008410561983 + 1.0.012342867198  
f(hat, 24): 0.008434489187 + 1.0.011776171717  
f(hat, 25): 0.00845629552 + 1.0.0112532460539  
f(hat, 26): 0.008420086598 + 1.0.010832624895  
f(hat, 27): 0.008476772248 + 1.0.010318796372  
numerical0@nvim- 1:nvim 2:nvim 3:[thme]
```

Figure 6: Screenshot of result of code for calculating fourier coefficients using FFT.

Figure 7: Screenshot of result of code for calculating fourier coefficients using FFT.

```
43 use plotters::prelude::*;
44 use rustfft::num_complex::Complex_FftPlanner;
45 use std::f64::consts::(PI, SQR_T_2);
46
47 fn plot_functions<Fn>(fns: &[Fn], x_range: Range<f64>, y_range: Range<f64>) {
48     let mut chart = ChartBuilder::on(&x_range)
49         .caption("Function Comparison", ("sans-serif", 30).into_font())
50         .margin(10)
51         .x_label_area_size(40)
52         .y_label_area_size(60)
53         .build_cartesian_2d(x_range.clone(), y_range.clone())?;
54
55     chart.configure_mesh()
56         .x_labels(10)
57         .y_labels(10)
58         .x_desc("x")
59         .y_desc("f(x) or g(x)")
60         .drawn()?;
61
62     let points = Vec::from_fn(x_range.start..(x_range.end - x_range.start) * (1 as f64 / sample_points as f64), |i| {
63         map![i] {
64             let x = x_range.start + (x_range.end - x_range.start) * (i as f64 / sample_points as f64);
65             fns[i]
66         }
67     });
68
69     chart
70         .series(
71             FunctionSeries::new(
72                 points,
73                 "f(x)" if fns[0] == fns[1] else "g(x)",
74                 "black"
75             )
76         )
77         .show();
78 }
```

Figure 8: Screenshot of code for plotting function and reconstruction of function.

```
Thursday March 27, 22:32  
to numerical  
  
19 |     .y_labels(10)  
20 |     .x_desc("x")  
21 |     .y_desc("f(x) or g(x)")  
22 |     .drawn()?  
23 |  
24 | let points: Vec<f64, f64> = (0..=sample_points)  
25 |     .map(|i| {  
26 |         let x =  
27 |             x_range.start + (x_range.end - x_range.start) * (i as f64 / sample_points as f64);  
28 |         (x, func(x))  
29 |     })  
30 |     .collect();  
31 |  
32 | let points2: Vec<f64, f64> = (0..=sample_points)  
33 |     .map(|i| {  
34 |         let x =  
35 |             x_range.start + (x_range.end - x_range.start) * (i as f64 / sample_points as f64);  
36 |         (x, reconstruct_function(&fourier_coeffs, 200, x, 5))  
37 |     })  
38 |     .collect();  
39 |  
40 | chart  
41 |     draw series(LineSeries::new(points1.iter().map(|&(x, y)| (x, y)), &RED))?  
42 |     .label("(x)")  
43 |     .legend([(x, y)] PathElement::new(vec![(x, y), (x + 20, y)], &RED));  
44 |  
45 | chart  
46 |     draw series(LineSeries::new(points2.iter().map(|&(x, y)| (x, y)), &BLUE))?  
47 |     .label("g(x)")  
48 |     .legend([(x, y)] PathElement::new(vec![(x, y), (x + 20, y)], &BLUE));  
49 |  
50 | chart  
51 |     .configure_series_labels()  
52 |     .background_style(&WHITE.mix(0.8))  
53 |     .border_style(&BLACK)  
54 |     .drawn()?  
55 |  
56 | root.present()?  
57 |  
58 | println!("Plot saved to {}", filename);  
59 | Ok(())  
60 |}  
61 |  
62 | fn func1(x: f64) -> f64 {  
63 |     (5.0 * x).sin().powi(2)  
64 | }  
65 |  
66 | fn func2(x: f64) -> f64 {  
67 |     (3.0 * x).cos().powi(3) + (SORT_2 * x).cos()  
68 | }  
69 |  
70 | fn reconstruct_function(fourier_coeffs: &[Complex<f64>], n: usize, x: f64, m: i32) -> f64 {  
71 |     let mut sum = 0.0;  
72 |     for j in ..m.m {  
73 |         let mut sum = 0.0;
```

Figure 9: Screenshot of code for plotting function and reconstruction of function.

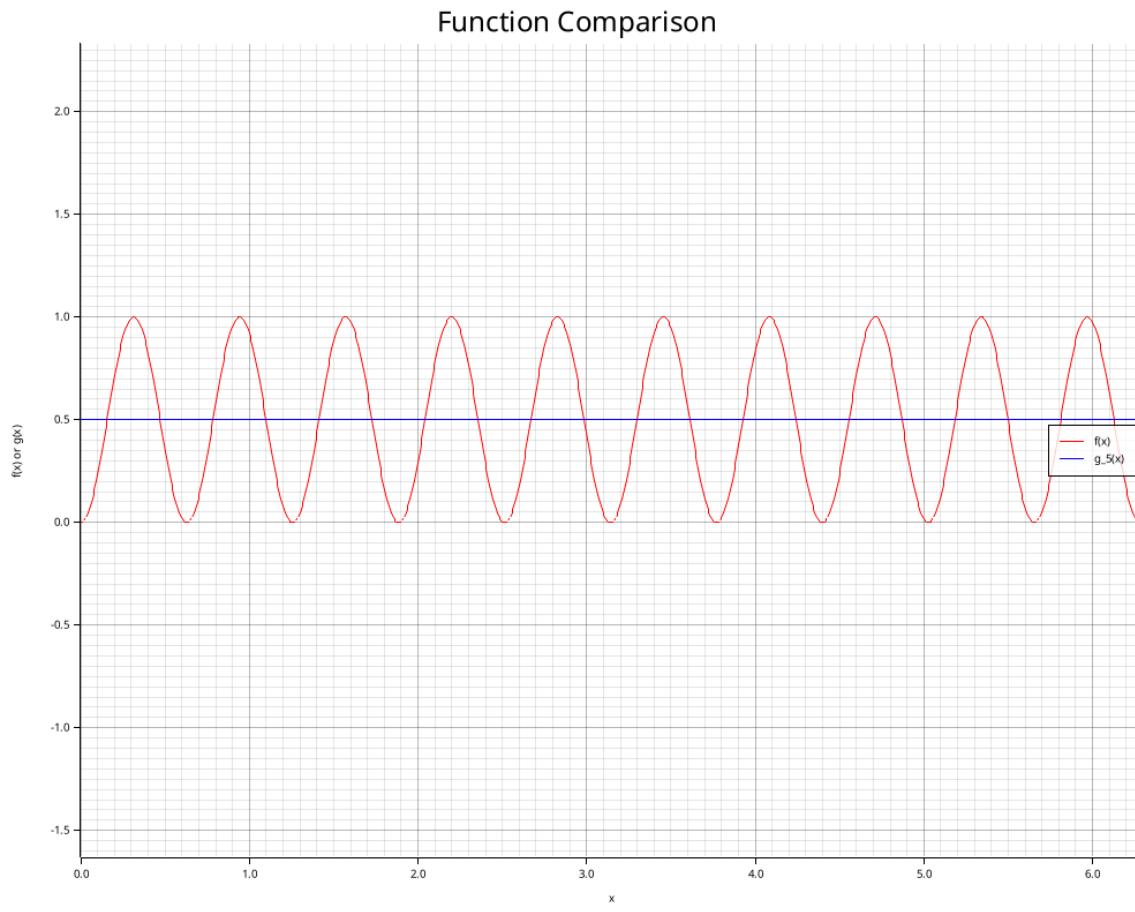


Figure 10: Screenshot of plot of first function and reconstruction for comparison (f vs g_5).

Function Comparison

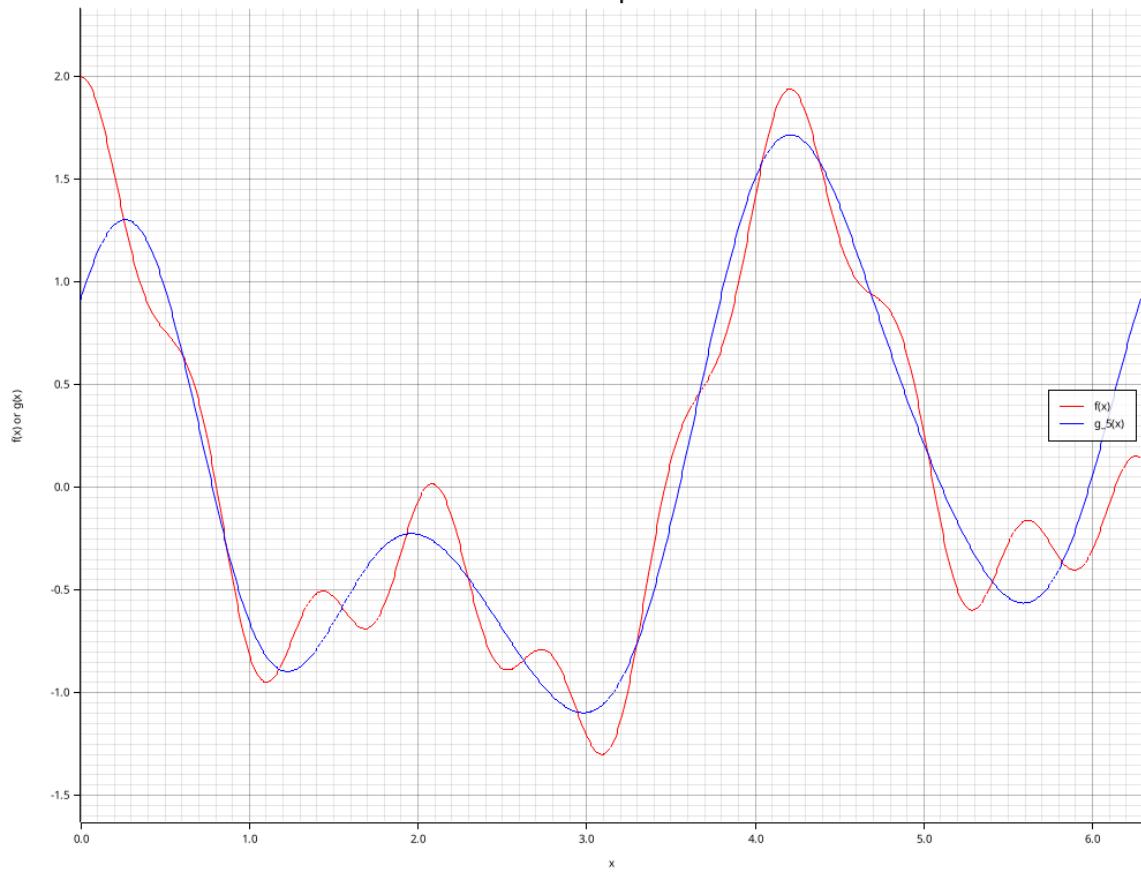


Figure 11: Screenshot of plot of first function and reconstruction for comparison (f vs g_5).

Problem 6: Please see attached PDF with all written solutions for questions 1-4 and 6-7.

Problem 7: Please see attached PDF with all written solutions for questions 1-4 and 6-7.

Problem 8:

1 Written Solutions

HW 3

Question 1 Let $\{P_i(x)\}_{i=0}^{\infty}$ be a sequence of orthogonal polynomials and let x_0, x_1, \dots, x_k be the $k+1$ distinct zeros of $P_{k+1}(x)$.

Prove that the Lagrange polynomials $l_i(x) = \prod_{t=0, t \neq i}^{k+1} (x - x_t) / (x_i - x_t)$ for these points are orthogonal $\int l_i l_j dx = 0$ to each other.

Hint: Show that for $i \neq j$, $\int l_i l_j dx = P_{k+1}(x) g(x)$, where $g(x)$ is some polynomial of degree $\leq k$.

Solution

$$l_i(x) = \prod_{\substack{t=0 \\ t \neq i}}^{k+1} \frac{x - x_t}{x_i - x_t} \quad \text{and} \quad l_j(x) = \prod_{\substack{s=0 \\ s \neq j}}^{k+1} \frac{x - x_s}{x_j - x_s}$$

$$l_i(x_t) = \begin{cases} 1 & t = i \\ 0 & t \neq i \end{cases} \quad l_j(x_s) = \begin{cases} 1 & s = j \\ 0 & s \neq j \end{cases}$$

$$\deg(l_i(x)) = \deg(l_j(x)) = k, \quad \deg(l_i(x) l_j(x)) \leq 2k$$

$$l_i(x) l_j(x) = \prod_{\substack{t=0 \\ t \neq i}}^{k+1} \frac{x - x_t}{x_i - x_t} \prod_{\substack{s=0 \\ s \neq j}}^{k+1} \frac{x - x_s}{x_j - x_s}$$

$$l_i(x_t) l_j(x_s) = \begin{cases} (1)(0) & t = i \\ (0)(1) & t = j \\ (0)(0) & t \neq i \text{ and } t \neq j \end{cases}$$

$$\text{Thus, } l_i(x_t) l_j(x_s) = 0 \quad x_0, \dots, x_k$$

So $l_i(x) l_j(x)$ is a degree $\leq 2k$ polynomial with $k+1$ distinct roots at x_0, \dots, x_k .

This means, for some $k+1$ degree or less polynomial $g(x)$

$$l_i(x) l_j(x) = P_{k+1}(x) g(x)$$

But $P_{k+1}(x)$ is orthogonal and $k+1$ degree, so $\langle P_{k+1}, g \rangle = 0$

$$\text{Hence } \langle l_i, l_j \rangle = 0$$

Question 2

Approximation by Orthogonal Projection

The best polynomial $p(x)$ is given by projecting onto the basis of:

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

$$\text{- Chebyshev polynomials for } \langle f, g \rangle = \int_{-1}^1 \frac{f(x)g(x)}{L(-x^2)^{1/2}} dx$$

$$\text{- Hermite polynomials for } \langle f, g \rangle = \int_{-\infty}^{\infty} f(x)g(x)e^{-x^2}dx$$

For up to degree 2 with our objective being to minimize

$$\int_{-1}^1 (\cos(\pi x) - p(x))^2 dx$$

we note this corresponds to $\langle f, g \rangle = \int_{-1}^1 f(x)g(x)dx$ inner product.

$$p_2(x) = \sum_{i=0}^2 \frac{\langle \cos(\pi x), L_i \rangle}{\langle L_i, L_i \rangle} L_i(x)$$

$$L_0 = 1, \quad L_1(x) = x, \quad L_2(x) = \frac{3x^2 - 1}{2} = \frac{3}{2}x^2 - \frac{1}{2}$$

$$\text{As } L_{n+1}(x) = \frac{(2n+1)xL_n(x) - nL_{n-1}(x)}{n+1} \text{ is Legendre polynomial general form.}$$

$$\text{Further, } \langle L_i, L_i \rangle = \frac{2}{2i+1}. \text{ Thus,}$$

$$p_2(x) = \frac{\langle \cos(\pi x), 1 \rangle}{2} + \frac{\langle \cos(\pi x), x \rangle}{3} x$$

$$+ \frac{\langle \cos(\pi x), \frac{3}{2}x^2 - \frac{1}{2} \rangle}{5} \left(\frac{3}{2}x^2 - \frac{1}{2} \right)$$

$$\cdot \langle \cos(\pi x), 1 \rangle = \int_{-1}^1 \cos(\pi x) dx = \frac{1}{\pi} \sin(\pi x) \Big|_{-1}^1$$

$$= \frac{1}{\pi} (\sin(\pi) - \sin(-\pi)) = \frac{1}{\pi} (\sin(\pi) + \sin(\pi)) = 0$$

$$\cdot \langle \cos(\pi x), x \rangle = \int_{-1}^1 \cos(\pi x) x dx = 0 \quad \text{(steps below)}$$

$$\int u dv = uv - \int v du \quad u = x \quad v = \frac{1}{\pi} \sin(\pi x)$$

$$du = dx \quad dv = \cos(\pi x) dx$$

$$\int x \cos(\pi x) dx = \frac{x}{\pi} \sin(\pi x) - \frac{1}{\pi} \int \sin(\pi x) dx$$

$$\int_{-1}^1 x \cos(\pi x) dx = \frac{x}{\pi} \sin(\pi x) \Big|_{-1}^1 + \frac{1}{\pi} \int_{-1}^1 \cos(\pi x) dx$$

$$= \frac{1}{\pi^2} (\cos(\pi) - \cos(-\pi)) = \frac{1}{\pi^2} (0) = 0$$

$$\cdot \langle \cos(\pi x), \frac{3}{2}x^2 - \frac{1}{2} \rangle = \frac{3}{2} \int_{-1}^1 x^2 \cos(\pi x) dx - \frac{1}{2} \int_{-1}^1 \cos(\pi x) dx$$

$$\cdot -\frac{1}{2} \int_{-1}^1 \cos(\pi x) dx = 0$$

$$\cdot \frac{3}{2} \int_{-1}^1 x^2 \cos(\pi x) dx = \frac{3}{2} \left(-\frac{4}{\pi^2} \right) = -\frac{12}{2\pi^2} = -\underline{\underline{6\pi^{-2}}}$$

$$u = x^2 \quad v = \frac{1}{\pi} \sin(\pi x)$$

$$du = 2x dx \quad dv = \cos(\pi x) dx$$

$$\int_{-1}^1 x^2 \cos(\pi x) dx = \frac{x^2}{\pi} \sin(\pi x) \Big|_{-1}^1 - \frac{1}{\pi} \int_{-1}^1 \sin(\pi x) 2x dx = -\frac{1}{\pi} \int_{-1}^1 \sin(\pi x) 2x dx$$

$$\Rightarrow -\frac{1}{\pi} \int_{-1}^1 \sin(\pi x) 2x dx = -\frac{2}{\pi} \left(\frac{\sin(\pi x) - \pi x \cos(\pi x)}{\pi^2} \Big|_{-1}^1 \right) = -\frac{2}{\pi} \left(\frac{2}{\pi} \right) = -\underline{\underline{4\pi^{-2}}}$$

$$\cdot \langle \cos(\pi x), 1 \rangle = 0$$

$$\cdot \langle \cos(\pi x), x \rangle = 0$$

$$\cdot \langle \cos(\pi x), \frac{3}{2}x^2 - \frac{1}{2} \rangle = -6\pi^{-2}$$

$$p_2(x) = \frac{-6\pi^{-2}}{\frac{2}{5}} \left(\frac{3}{2}x^2 - \frac{1}{2} \right) = \frac{-30}{2\pi^2} \left(\frac{3}{2}x^2 - \frac{1}{2} \right)$$

$$= -15\pi^{-2} \left(\frac{3}{2}x^2 - \frac{1}{2} \right) = \frac{-45\pi^{-2}}{2} x^2 + \frac{15\pi^{-2}}{2}$$

$$p_2(x) = -\frac{45}{2\pi^2} x^2 + \frac{15}{2\pi^2} . \quad \square$$

Question 3 If $f(x)$ is a 2π periodic function, prove that $g_\alpha(x) = f(x+\alpha)$ is also 2π -periodic.

Proof.

We want to show $g_\alpha(x+2\pi) = g_\alpha(x) \quad \forall x$.

$$g_\alpha(x+2\pi) = f((x+2\pi)+\alpha) = f(x+\alpha+2\pi)$$

Since f is 2π periodic, $f((x+\alpha)+2\pi) = f(x+\alpha)$

Hence, $g_\alpha(x+2\pi) = g_\alpha(x)$, as $f(x+\alpha) = g_\alpha(x)$.

Therefore, $g_\alpha(x)$ is also 2π -periodic. \square

$$f(j) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-ijx} dx \quad \hat{g}_\alpha(j) = \frac{1}{2\pi} \int_{-\pi}^{\pi} g_\alpha(x) e^{-ijx} dx$$

$$\hat{g}_\alpha(j) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x+\alpha) e^{-ijx} dx$$

$$u = x + \alpha \quad du = dx \quad \text{so} \quad 0 \mapsto \alpha, \quad 2\pi \mapsto 2\pi + \alpha$$

$$\hat{g}_\alpha(j) = \frac{1}{2\pi} \int_{\alpha}^{2\pi+\alpha} f(u) e^{-ij(u-\alpha)} du \quad x = u - \alpha$$

$$f(u) \text{ is } 2\pi\text{-periodic, so } \frac{1}{2\pi} \int f(u) e^{-ij(u-\alpha)} du = \frac{1}{2\pi} \int f(u) e^{-ij(u-\alpha)} du$$

$$\text{And thus, } \hat{g}_\alpha(j) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(u) e^{-ij(u-\alpha)} e^{ij\alpha} du$$

$$\text{which is: } \hat{g}_\alpha(j) = \frac{e^{ij\alpha}}{2\pi} \int_{-\pi}^{\pi} f(u) e^{iju} du$$

$$\text{Thus, } \hat{g}_\alpha(j) = e^{ij\alpha} f(j). \quad \square$$

Question 4 Verify that the 2π -periodic function $f(x)$ whose values on $[0, 2\pi]$ are given by

$$f(x) = \begin{cases} (\frac{x}{\pi})^2 - x/\pi & 0 \leq x < \pi \\ (x-\pi)\pi - \left(\frac{x-\pi}{\pi}\right)^2 & \pi \leq x < 2\pi \end{cases}$$

is continuous and has a continuous first derivative (as a 2π -periodic) but has jumps in the second derivative. Then construct the spectrum of $f(x)$ and show that it decays like j^{-3} , $j \rightarrow \infty$.

Solution

$$f'(x) = \begin{cases} \frac{2x}{\pi^2} - \frac{1}{\pi}, & x \in [0, \pi) \\ \frac{1}{\pi} - \frac{2(x-\pi)}{\pi^2}, & x \in [\pi, 2\pi) \end{cases}$$

$$f''(x) = \begin{cases} \frac{2}{\pi^2}, & x \in [0, \pi) \\ -\frac{2}{\pi^2}, & x \in [\pi, 2\pi) \end{cases}$$

- $\lim_{x \rightarrow \pi^-} f(x) = \left(\frac{\pi}{\pi}\right)^2 - \frac{\pi}{\pi} = 1 - 1 = 0$

- $\lim_{x \rightarrow \pi^+} f(x) = \frac{\pi-\pi}{\pi} - \left(\frac{\pi-\pi}{\pi}\right)^2 = 0$

- $\lim_{x \rightarrow \pi^-} f(x) = \lim_{x \rightarrow \pi^+} f(x)$, f is continuous over $[0, 2\pi]$

- $\lim_{x \rightarrow \pi^-} f'(x) = \frac{2\pi}{\pi^2} - \frac{1}{\pi} = \frac{1}{\pi}$

- $\lim_{x \rightarrow \pi^+} f'(x) = \frac{1}{\pi} - 0 = \frac{1}{\pi}$

- $\lim_{x \rightarrow \pi^-} f'(x) = \lim_{x \rightarrow \pi^+} f'(x)$, F' is continuous over $[0, 2\pi]$

- $\lim_{x \rightarrow \pi^-} f''(x) = \frac{2}{\pi^2} \neq \lim_{x \rightarrow \pi^+} f''(x) = -\frac{2}{\pi^2}$ f'' is not contn. over $[0, 2\pi]$.

$$\hat{f}(j) = \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-ijx} dx$$

$$= \frac{1}{2\pi} \left[\int_0^{\pi} \left(\frac{x}{\pi} \right)^2 e^{-ijx} - \frac{x}{\pi} e^{-ijx} dx + \int_{\pi}^{2\pi} \frac{x-\pi}{\pi} e^{-ijx} - \frac{(x-\pi)^2}{\pi^2} e^{-ijx} dx \right]$$

Or if using $f(x) = \sum_{j=1}^n a_j \sin jx + \sum_{j=0}^n b_j \cos jx$
form instead, where:

$$a_j = \langle f, \sin jx \rangle$$

$$b_j = \langle f, \cos jx \rangle$$

$$a_j = \frac{1}{2\pi} \int_0^{2\pi} f(x) \sin jx dx = \frac{1}{2\pi} \left[\int_0^{\pi} \left(\frac{x^2}{\pi^2} - \frac{x}{\pi} \right) \sin jx dx + \int_{\pi}^{2\pi} \left(\frac{x-\pi}{\pi} - \frac{(x-\pi)^2}{\pi^2} \right) \sin jx dx \right]$$

$$b_j = \frac{1}{2\pi} \int_0^{2\pi} f(x) \cos jx dx = \frac{1}{2\pi} \left[\int_0^{\pi} \left(\frac{x^2}{\pi^2} - \frac{x}{\pi} \right) \cos jx dx + \int_{\pi}^{2\pi} \left(\frac{x-\pi}{\pi} - \frac{(x-\pi)^2}{\pi^2} \right) \cos jx dx \right]$$

$f \in C^1[0, 2\pi]$ since $f^{(2)}(x)$ is not continuous over $[0, 2\pi]$ but $f, f^{(1)}$ are.

$$\text{Thus, } |\hat{f}(j)| = O(|j|^{-2}) \quad \square$$

Question 6

First need to determine the orthogonal polynomials associated with $w(x) = -x$ over $[-1, 0]$.

$$\text{Satisfy: } \int_{-1}^0 P_m(x) P_n(x) w(x) dx = 0 \quad \text{for } m \neq n.$$

These are the Legendre polynomials shifted by $x \mapsto -2x - 1$. Let $\tilde{P}_n(x)$ be non-shifted Legendre poly. $[-1, 0] \rightarrow [-1, 1]$

$$\text{So, } P_0(x) = 1, \quad P_1(x) = \tilde{P}_1(2x+1) = 2x+1$$

$$P_2(x) = \tilde{P}_2(2x+1) = \frac{1}{2}(3(2x+1)^2 - 1) = \frac{3}{2}(4x^2 + 4x + 1) - \frac{1}{2} \\ = 6x^2 + 6x + 1$$

(n=1) One-Point Quadrature

$P_1(x) = 2x+1$. The quadrature node is the root of $P_1(x)$, which is $x_0 = -\frac{1}{2}$.

$$\text{The weight is } A_0 = \int_{-1}^0 (-x) \lambda_0(x) dx = \int_{-1}^0 -x dx = \left[-\frac{x^2}{2} \right]_{-1}^0 = \frac{1}{2}$$

The one-point Gaussian quadrature rule is

$$I \approx \frac{1}{2} f(x_0) = \frac{1}{2} f(-\frac{1}{2})$$

(n=2) Two-Point Quadrature

$$P_2(x) = 6x^2 + 6x + 1. \text{ Solving } 6x^2 + 6x + 1 = 0 \text{ we find,}$$

$$x = \frac{-6 \pm \sqrt{36 - 24}}{12} = \frac{-6 \pm \sqrt{12}}{12} = \frac{-6 \pm 2\sqrt{3}}{12} = -\frac{1}{2} \pm \frac{1}{6}\sqrt{3}$$

$$x_0 = -\frac{1}{2} - \frac{1}{6}\sqrt{3}, \quad x_1 = -\frac{1}{2} + \frac{1}{6}\sqrt{3}$$

$$A_0 = \int_{-1}^0 -x \lambda_0(x) dx = \int_{-1}^0 -x \frac{x-x_1}{x_0-x_1} dx = \frac{1}{x_0-x_1} \int_{-1}^0 -x^2 + xx_1 dx = \frac{1}{x_0-x_1} \left[\frac{-x^3}{3} + \frac{x^2}{2} x_1 \right]_{-1}^0$$

$$A_0 = \frac{\frac{1}{3} + \frac{1}{2}x_1}{x_0-x_1}$$

$$A_1 = \int_{-1}^0 -x \lambda_1(x) dx = \int_{-1}^0 -x \frac{x-x_0}{x_1-x_0} dx = \frac{1}{x_1-x_0} \left[\frac{-x^3}{3} + \frac{x^2}{2} x_0 \right]_{-1}^0 = \frac{\frac{1}{3} + \frac{1}{2}x_0}{x_1-x_0}$$

The Two-Point Quadrature Rule is thus,

$$A_0 f(x_0) + A_1 f(x_1) = \frac{\frac{1}{3} + \frac{1}{2}x_1}{x_0 - x_1} f(x_0) + \frac{\frac{1}{3} + \frac{1}{2}x_0}{x_1 - x_0} f(x_1)$$

$$\therefore \frac{1}{3} + \frac{1}{2}x_1 = \frac{1}{3} - \frac{1}{4} + \frac{\sqrt{3}}{12} = \frac{1 + \sqrt{3}}{12}$$

$$\therefore \frac{1}{3} + \frac{1}{2}x_0 = \frac{1}{3} - \frac{1}{4} - \frac{\sqrt{3}}{12} = \frac{1 - \sqrt{3}}{12}$$

$$\therefore x_0 - x_1 = \left(-\frac{1}{2} - \frac{\sqrt{3}}{6}\right) - \left(-\frac{1}{2} + \frac{\sqrt{3}}{6}\right) = -\frac{\sqrt{3}}{3}$$

$$\therefore x_1 - x_0 = \left(-\frac{1}{2} + \frac{\sqrt{3}}{6}\right) - \left(-\frac{1}{2} - \frac{\sqrt{3}}{6}\right) = \frac{\sqrt{3}}{3}$$

Final Answer

$$\frac{1 + \frac{\sqrt{3}}{12}}{-\frac{\sqrt{3}}{3}} f\left(-\frac{1}{2} - \frac{\sqrt{3}}{6}\right) + \frac{1 - \frac{\sqrt{3}}{12}}{\frac{\sqrt{3}}{3}} f\left(-\frac{1}{2} + \frac{\sqrt{3}}{6}\right)$$

2

Question 7

The corrected trapezoidal rule is

$$I(f) = \int_a^b f(x) dx \approx \frac{1}{2} (b-a) (f(a)+f(b)) + \frac{(b-a)^2}{12} (f'(a)-f'(b))$$

For evenly spaced points we let $x_i = x_0 + ih$
for x_0, \dots, x_N and $h = \frac{b-a}{N}$.

So $x_{i+1} - x_i = h, \forall i = 0, \dots, N-1$. Now, we want to

$$I(f) = \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} f(x) dx \quad \text{compute}$$

But for the corrected trapezoidal rule,

$$\begin{aligned} \int_{x_i}^{x_{i+1}} f(x) dx &\approx \frac{1}{2} (x_{i+1} - x_i) (f(x_i) + f(x_{i+1})) + \frac{(x_{i+1} - x_i)^2}{12} (f'(x_i) - f'(x_{i+1})) \\ &= \frac{1}{2} h (f(x_i) + f(x_{i+1})) + \frac{h^2}{12} (f'(x_i) - f'(x_{i+1})) \end{aligned}$$

So,

$$\begin{aligned} I(f) &\approx \sum_{i=0}^{N-1} \left[\frac{h}{2} (f(x_i) + f(x_{i+1})) + \frac{h^2}{12} (f'(x_i) - f'(x_{i+1})) \right] \\ &= \frac{h}{2} \sum_{i=0}^{N-1} [f(x_i) + f(x_{i+1})] + \frac{h^2}{12} \sum_{i=0}^{N-1} [f'(x_i) - f'(x_{i+1})] \end{aligned}$$

$$\text{Ansver} \Rightarrow = \frac{h}{2} (f(a) + f(b)) + h \sum_{i=1}^{N-1} f(x_i) + \frac{h^2}{12} (f'(a) - f'(b)) \quad \star$$

$$\text{Since } \sum_{i=0}^{N-1} [f'(x_i) - f'(x_{i+1})]$$

$$\begin{aligned} &= f'(x_0) - f'(x_1) + f'(x_1) - f'(x_2) + \dots - f'(x_{N-1}) + f'(x_{N-1}) - f'(x_N) \\ &= f'(x_0) - f'(x_N) = f'(a) - f'(b) \quad \blacksquare \end{aligned}$$