Homework 6

Panupong (Peach) Chaphuphuang

April 25, 2024

Github URL for homework 6:
`https://github.com/pach2648/APPM4600/tree/main/Homework/HW6`

Problem 1

Solution

a)

$$\int_0^1 f(x)\,dx = \frac{1}{2}f(x_0) + c_1 f(x_1)$$

Since there are three unknowns, we would expect the $f(x)$ to be at least $1, x, x^2$

$$\int_0^1 1\,dx = 1 = \frac{1}{2}\cdot 1 + c_1 \cdot 1 \tag{1}$$

$$\int_0^1 x\,dx = \frac{1}{2} = \frac{1}{2}\cdot x_0 + c_1 \cdot x_1 \tag{2}$$

$$\int_0^1 x^2\,dx = \frac{1}{3} = \frac{1}{2}\cdot x_0^2 + c_1 \cdot x_1^2 \tag{3}$$

From (1), we can solve that $c_1 = \frac{1}{2}$.

From (2) and (3), we can solve the system since there are 2 equations and 2 unknowns.

$$\frac{1}{2} = \frac{1}{2}\cdot x_0 + \frac{1}{2}\cdot x_1 \tag{4}$$

$$\frac{1}{3} = \frac{1}{2}\cdot x_0^2 + \frac{1}{2}\cdot x_1^2 \tag{5}$$

From equation 4, we can rearrange to $x_0 = 1 - x_1$ and plug this into equation 5. We will get:

$$\frac{2}{3} = (1 - x_1)^2 + x_1^2$$

Therefore, $x_1 = 1 - \frac{\sqrt{3}\pm 1}{2\sqrt{3}}$ and plug in back to equation 4, so $x_0 = \frac{3\pm\sqrt{3}}{6}$

Problem 2

Solution

```
Approximation of the integral using composite Trapezoidal rule: 2.746801287341883
Approximation of the integral using composite Simpson's rule: 2.7468015338893297

Number of subintervals for Trapezoidal rule (n): 1291
Error using composite Trapezoidal rule: 2.4654814900770816e-07

Number of subintervals for Simpson's rule (n): 108
Error using composite Simpson's rule: 7.02105040772949e-13

Using quad with default tolerance (1e-6):
Result: 2.7468015338900327
Estimated error: 1.4334139675000002e-08

Using quad with tolerance set to 1e-4:
Result: 2.746801533909586
Estimated error: 1.0279997850748401e-05
```

Figure 1: The results from part a-c

a) Github URL for part a: `https://github.com/pach2648/APPM4600/blob/main/Homework/HW6/Q2-1.py`

b) The error estimate for Trapezoidal rule

$$|E_T| \leq \frac{(b-a)^3}{12n^2} M_2 \tag{6}$$

The error estimate for Simpson's rule

$$|E_S| \leq \frac{(b-a)^5}{180n^4} M_4 \tag{7}$$

Where M_2 is the absolute maximum of $f''(x)$ in $[-5, 5]$ and M_4 is the absolute maximum of $f^{(4)}(x)$ in $[-5, 5]$.

Find M_2 and M_4

$$\begin{aligned}
f(x) &= \frac{1}{1+x^2} \\
f'(x) &= -\frac{2x}{(1+x^2)^2} \\
f''(x) &= -\frac{2(-3x^2+1)}{(1+x^2)^3} \\
f'''(x) &= \frac{24x(-x^2+1)}{(1+x^2)^4} \\
f''''(x) &= \frac{24(5x^4-10x^2+1)}{(1+x^2)^5}
\end{aligned}$$

Plot both $f''(x)$ and $f''''(x)$ in Desmos to find the absolute maximum value in $[-5, 5]$

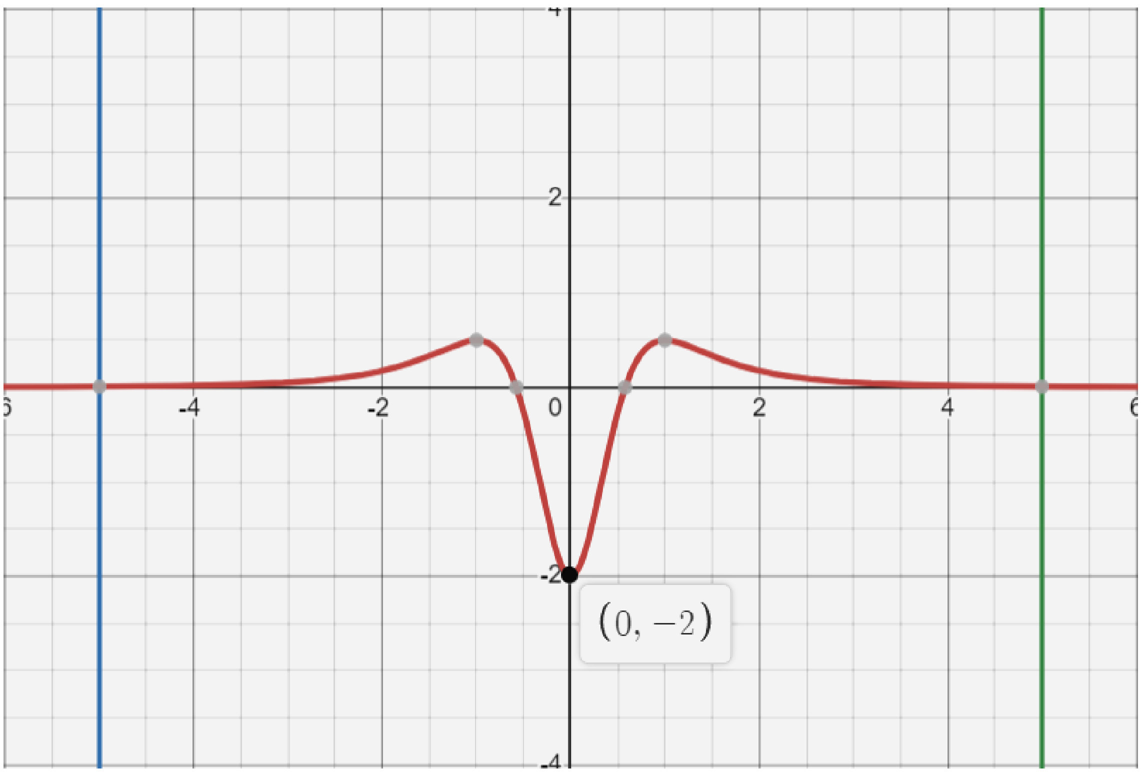


Figure 2: The plot of $f''(x) = -\frac{2(-3x^2+1)}{(1+x^2)^3}$

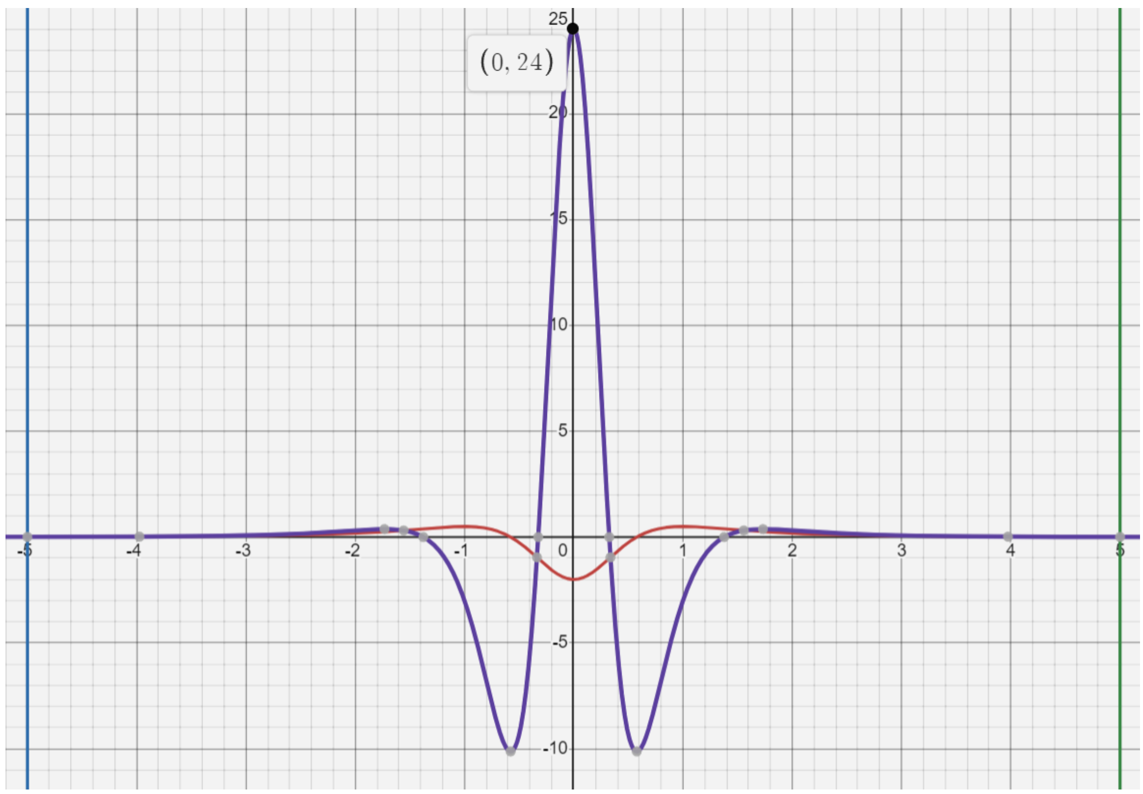


Figure 3: The plot of $f''''(x) = \frac{24(5x^4-10x^2+1)}{(1+x^2)^5}$

Therefore, the absolute maximum value of $f''(x) = -\frac{2(-3x^2+1)}{(1+x^2)^3}$ is 2, and the absolute maximum value of $f''''(x) = \frac{24(5x^4-10x^2+1)}{(1+x^2)^5}$ is 24. I use those values to plug in my code to find the n. (The exact value of the integration from -5 to 5 of $f(x) = \frac{1}{1+x^2}$ is $2\arctan 5$ to find the errors of each method).

c) Github URL for part c: `https://github.com/pach2648/APPM4600/blob/main/Homework/HW6/Q2-1.py`

Problem 3

Solution

a)

$$I - I_n = \frac{C_1}{n\sqrt{n}} + \frac{C_2}{n^2} + \frac{C_3}{n^2\sqrt{n}} + \frac{C_4}{n^3} + \cdots$$

If $n = n/2$

$$\begin{aligned} I - I_{n/2} &= \frac{C_1}{n/2\sqrt{n/2}} + \frac{C_2}{(n/2)^2} + \frac{C_3}{(n/2)^2\sqrt{n/2}} + \frac{C_4}{(n/2)^3} + \cdots \\ &= \frac{2\sqrt{2}C_1}{n\sqrt{n}} + \frac{4C_2}{n^2} + \frac{4\sqrt{2}C_3}{n^2\sqrt{n}} + \frac{8C_4}{n^3} + \cdots \end{aligned}$$

If $n = n/4$

$$\begin{aligned} I - I_{n/4} &= \frac{C_1}{n/4\sqrt{n/4}} + \frac{C_2}{(n/4)^2} + \frac{C_3}{(n/4)^2\sqrt{n/4}} + \frac{C_4}{(n/4)^3} + \cdots \\ &= \frac{8C_1}{n\sqrt{n}} + \frac{16C_2}{n^2} + \frac{32C_3}{n^2\sqrt{n}} + \frac{64C_4}{n^3} + \cdots \end{aligned}$$

We need to solve for I since we assume that three values $I_n, I_{n/2}$, and $I_{n/4}$ have been computed. Therefore, we will have 3 equations below:

$$I_n = I - \frac{C_1}{n\sqrt{n}} - \frac{C_2}{n^2} - \frac{C_3}{n^2\sqrt{n}} + \frac{C_4}{n^3} - \cdots \tag{8}$$

$$I_{n/2} = I - \frac{2\sqrt{2}C_1}{n\sqrt{n}} - \frac{4C_2}{n^2} - \frac{4\sqrt{2}C_3}{n^2\sqrt{n}} - \frac{8C_4}{n^3} - \cdots \tag{9}$$

$$I_{n/4} = I - \frac{8C_1}{n\sqrt{n}} - \frac{16C_2}{n^2} - \frac{32C_3}{n^2\sqrt{n}} - \frac{64C_4}{n^3} - \cdots \tag{10}$$

Multiply equation 8 by $2\sqrt{2}$, and subtract to equation 9. (This will remove C_1.)

$$2\sqrt{2}I_n - I_{n/2} = (2\sqrt{2}-1)I - (2\sqrt{2}-4)\frac{C_2}{n^2} - (2\sqrt{2}-4\sqrt{2})\frac{C_3}{n^2\sqrt{n}} - (2\sqrt{2}-8)\frac{C_4}{n^3} - \cdots \tag{11}$$

Multiply equation 9 by $\frac{8}{2\sqrt{2}}$, and subtract to equation 10. (This will remove C_1 as well.)

$$2\sqrt{2}I_{n/2} - I_{n/4} = (2\sqrt{2}-1)I - (8\sqrt{2}-16)\frac{C_2}{n^2} - (-16)\frac{C_3}{n^2\sqrt{n}} - (16\sqrt{2}-64)\frac{C_4}{n^3} - \cdots \tag{12}$$

Multiply equation 11 by 4, and subtract to equation 12. (This will remove C_2.)

$$8\sqrt{2}I_{n/2} - 4I_{n/4} - 2\sqrt{2}I_{n/2} + I_{n/4} = (6\sqrt{2}-3)I - (16-8\sqrt{2})\frac{C_3}{n^2\sqrt{n}} - (32-8\sqrt{2})\frac{C_4}{n^3} - \cdots \tag{13}$$

Therefore,

$$I = \frac{1}{6\sqrt{2}-3}(8\sqrt{2}I_{n/2} - 4I_{n/4} - 2\sqrt{2}I_{n/2} + I_{n/4} + (16-8\sqrt{2})\frac{C_3}{n^2\sqrt{n}} + (32-8\sqrt{2})\frac{C_4}{n^3} + \cdots)$$

Problem 4

Solution

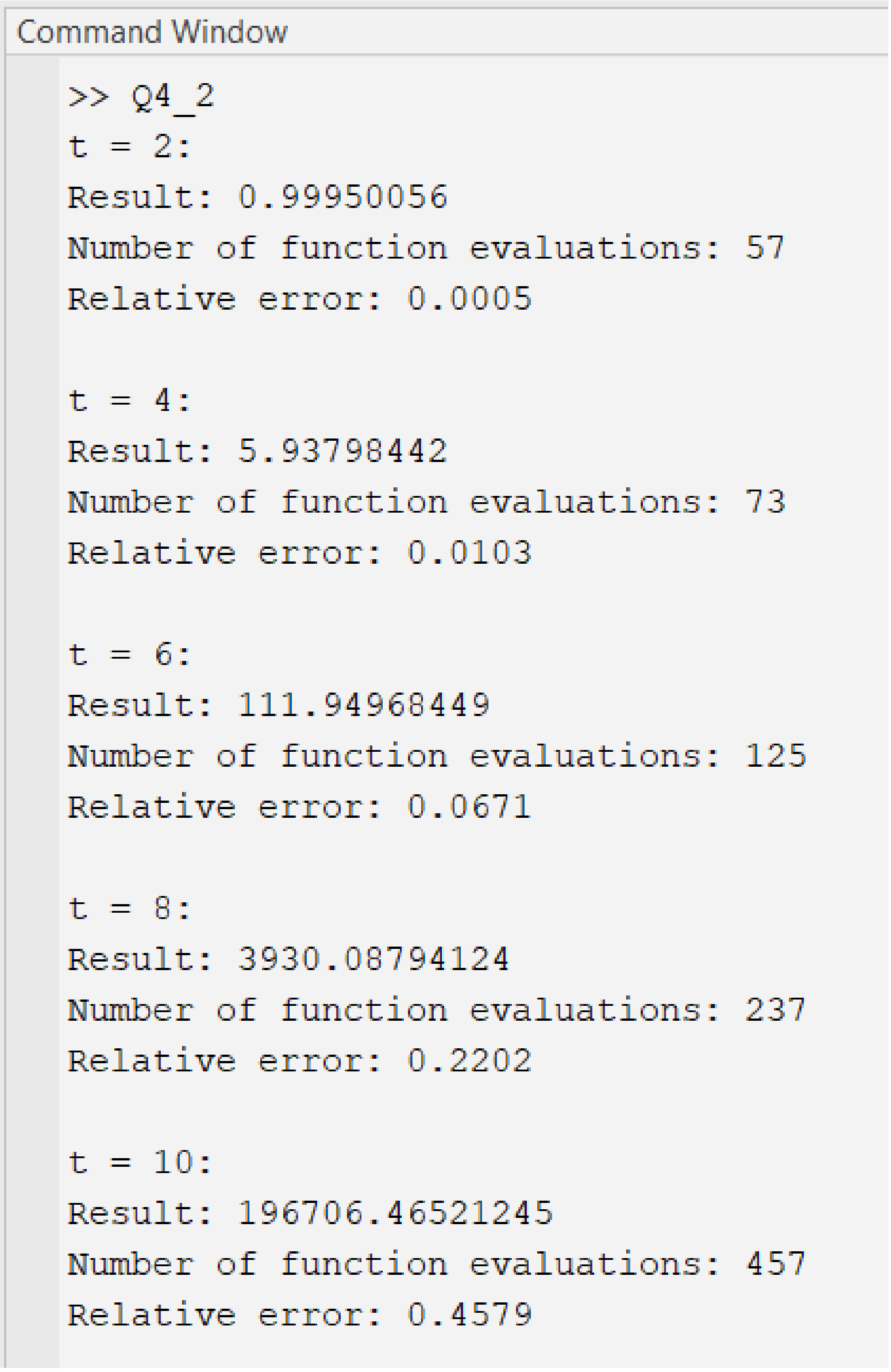
a) Github URL for part a: `https://github.com/pach2648/APPM4600/blob/main/Homework/HW6/Q4-1.py`

For part a, I start part b first to find the value of n for each t. Then, I use the same number of n of each t (You can see the values of n in figure 5) to compare the relative error of part a and b. We can see that if the t is small the function from MATLAB looks better in terms of the relative error. However, when increasing the value of t, the results are almost similar. For the lower and upper limits of integration, I try to have wider range as possible, but my laptop is not powerful enough to run from 0 to ∞, so I choose to run from 0 to 10 which gives us pretty reasonable results which is really close to the exact values with fast running time. Moreover, with this range, I can see the difference between each methods clearly.

```
t=2:
Trapezoidal Rule: 0.9994922640759575
Scipy's gamma function: 1.0
Relative Error: 0.0005077359240425183

t=4:
Trapezoidal Rule: 5.9379834311950335
Scipy's gamma function: 6.0
Relative Error: 0.010336094800827755

t=6:
Trapezoidal Rule: 111.94966553788129
Scipy's gamma function: 120.0
Relative Error: 0.06708612051765593

t=8:
Trapezoidal Rule: 3930.08680612887 9
Scipy's gamma function: 5040.0
Relative Error: 0.2202208717998 2563

t=10:
Trapezoidal Rule: 196706.42737915152
Scipy's gamma function: 362880.0
Relative Error: 0.4579298187302923
```

Figure 4: The results of each value of t

b) Github URL for part b: `https://github.com/pach2648/APPM4600/blob/main/Homework/HW6/Q4_2.m`



```
Command Window
>> Q4_2
t = 2:
Result: 0.99950056
Number of function evaluations: 57
Relative error: 0.0005

t = 4:
Result: 5.93798442
Number of function evaluations: 73
Relative error: 0.0103

t = 6:
Result: 111.94968449
Number of function evaluations: 125
Relative error: 0.0671

t = 8:
Result: 3930.08794124
Number of function evaluations: 237
Relative error: 0.2202

t = 10:
Result: 196706.46521245
Number of function evaluations: 457
Relative error: 0.4579
```

Figure 5: The results of each value of t in MATLAB

c) Github URL for part b: `https://github.com/pach2648/APPM4600/blob/main/Homework/HW6/Q4-3.py`

For this part, I use the number of $n = 57$ to see that even if I use the smallest size of

n from MATLAB, we still get pretty accurate results from Gauss-Laguerre quadrature (We can look at all relative errors which are really small compared to part a and b).

```
t=2:
Gauss-Laguerre Quadrature: 0.9999999999998825
Scipy's gamma function: 1.0
Relative Error: 1.1746159600534156e-13

t=4:
Gauss-Laguerre Quadrature: 5.999999999999421
Scipy's gamma function: 6.0
Relative Error: 9.651538827408028e-14

t=6:
Gauss-Laguerre Quadrature: 119.99999999999066
Scipy's gamma function: 120.0
Relative Error: 7.780442956573097e-14

t=8:
Gauss-Laguerre Quadrature: 5039.999999999627
Scipy's gamma function: 5040.0
Relative Error: 7.398667216803583e-14

t=10:
Gauss-Laguerre Quadrature: 362879.9999999725
Scipy's gamma function: 362880.0
Relative Error: 7.571102279309033e-14
```

Figure 6: The results of each value of t with Gauss-Laguerre quadrature with $n = 57$