

APPM4600 Homework #11

Owen O'Connor

1a) See code on GitHub

b) Error formula for composite trap:

$$|E_T| = \frac{-(b-a)^3}{12n_T^2} \max |f'''|_{[a,b]}$$

$$f(s) = \frac{1}{1+s^2}$$

$$f'(s) = \frac{-2s}{(1+s^2)^2}$$

$$f'''(s) = \frac{-2}{(1+s^2)^2} + \frac{8s^2}{(1+s^2)^3}$$

$$= \frac{-2(1+s^2) + 8s^2}{(1+s^2)^3}$$

$$f''(s) = \frac{6s^2 - 2}{(1+s^2)^3}$$

$$\max |f''(s)|_{[-s, s]} = 2 \text{ when } s=0$$

$$\Rightarrow |E_T| = \frac{(s+s)^3}{12m_T^2}(2) < 10^{-4}$$

$$n_T^2 > \frac{2(10)^3}{12(10^{-4})}$$

$$n_T > 1290.9$$

$$\underline{\underline{n_T = 1291}}$$

for Composite Simpson's:

$$|E_S| = \frac{-(b-a)^5}{180n^4} \max |f^{(4)}|_{[a, b]}$$

$$f''(s) = \frac{6s^2 - 2}{(1+s^2)^3}$$

$$f'''(s) = \frac{8s}{(1+s^2)^3} - \frac{6s(s^2-2)}{(1+s^2)^4}$$

$$f'''(s) = \frac{8s(1+s^2) - 6s(s^2-2)}{(1+s^2)^4}$$

$$f'''(s) = \frac{2s^3 - 4s}{(1+s^2)^4}$$

$$f^{(4)}(s) = \frac{6s^2 - 4}{(1+s^2)^4} - \frac{8s(2s^3 - 4s)}{(1+s^2)^5}$$

$$f^{(4)}(s) = \frac{(6s^2 - 4)(1+s^2) - 16s^4 - 32s^2}{(1+s^2)^5}$$

$$= \frac{6s^2 - 6s^4 - 4 - 4s^2 - 16s^4 - 32s^2}{(1+s^2)^5}$$

$$f^{(4)}(s) = \frac{-20s^4 - 30s^2 - 4}{(1+s^2)^5}$$

$\max = 4$ at $s = 0$:

$$|E_s| = \frac{10^s}{180n_s^4} (4) < 10^{-4}$$

$$n_s^4 > \frac{4(10)^s}{180(10^{-4})}$$

$$n_s > 63.65$$

$$\underline{\underline{n_s = 70}} \text{ since } n \text{ must be even.}$$

Using n_T and n_s in code, I get the following results:

$$I_{top} = 2.744673731677375$$

$$I_{\text{Simp}} = 2.7130049798793787$$

$$I_{\text{quad}} = 2.7468015338900327$$

$$I_{\text{quad}} (\text{tol} = 10^{-6}) = 2.746801533909586$$

$$2) \int_1^\infty \frac{\cos(x)}{x^3} dx$$

$$t = \frac{1}{x} \Rightarrow t^3 = \frac{1}{x^3}$$

$$x = \frac{1}{t}$$

$$dx = -\frac{1}{t^2} dt$$

when $x = 1, t = 1$

when $x = \infty, t \rightarrow 0$

$$\Rightarrow \int_1^0 t^3 \cos\left(\frac{1}{t}\right) \left(-\frac{1}{t^2} dt\right)$$

$$= \int_0^1 t \cos\left(\frac{1}{t}\right) dt$$

$t=0$ will be a problem so change lower limit to

1×10^{-8} :

$$= \int_{1 \times 10^{-8}}^1 t \cos\left(\frac{1}{t}\right) dt$$

evaluating using code from Q1:

$$= \underline{\underline{0.011748}}$$



3)

$$\text{Assume } I - I_n = \frac{c_1}{n\sqrt{n}} + \frac{c_2}{n^2} + \frac{c_3}{n^2\sqrt{n}} + \frac{c_4}{n^3} + \dots$$

$$I_n = I - \frac{c_1}{n\sqrt{n}} - \frac{c_2}{n^2} - \frac{c_3}{n^2\sqrt{n}} - \dots$$

$$I_{n/2} = I - \frac{c_1}{\frac{n}{2}\sqrt{\frac{n}{2}}} - \frac{c_2}{\left(\frac{n}{2}\right)^2} - \frac{c_3}{\left(\frac{n}{2}\right)^2\sqrt{\frac{n}{2}}} - \dots$$

$$I_{n/4} = I - \frac{c_1}{\frac{n}{4}\sqrt{\frac{n}{4}}} - \frac{c_2}{\left(\frac{n}{4}\right)^2} - \frac{c_3}{\left(\frac{n}{4}\right)^2\sqrt{\frac{n}{4}}}$$

$$I' = aI_n + bI_{n/2} + cI_{n/4}$$

$$= a\left(I - \frac{c_1}{n\sqrt{n}} - \frac{c_2}{n^2} - \dots\right) + b\left(I - \frac{c_1}{\frac{n}{2}\sqrt{\frac{n}{2}}} - \dots\right) \\ + c\left(I - \frac{c_1}{\frac{n}{4}\sqrt{\frac{n}{4}}} - \dots\right)$$

$$= I(a+b+c) - c_1\left(\frac{a}{n\sqrt{n}} + \frac{b}{\frac{n}{2}\sqrt{\frac{n}{2}}} + \frac{c}{\frac{n}{4}\sqrt{\frac{n}{4}}}\right) \\ - c_2\left(\frac{a}{n^2} + \left(\frac{b}{\frac{n}{2}}\right)^2 + \left(\frac{c}{\frac{n}{4}}\right)^2\right) + \dots$$

we want $c_1 = c_2 = 0$

$$\frac{a}{\sqrt{n}} + \frac{b}{\sqrt[2]{n}} + \frac{c}{\sqrt[4]{n}} = 0$$

$$a + 2\sqrt{2}b + 3c = 0 \quad ①$$

$$\frac{a}{n^2} + \frac{b}{(\frac{n}{2})^2} + \frac{c}{(\frac{n}{4})^2} = 0$$

$$a + 4b + 16c = 0 \quad ②$$

$$a + b + c = 1 \quad ③$$

$$\begin{bmatrix} 1 & 2\sqrt{2} & 3 \\ 1 & 4 & 16 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Solving with python:

$$a = 2.06255755, \quad b = -1.2448636, \quad c = 0.18230605$$

$$I' = 2.06I_n - 1.24I_{n/2} + 0.182I_{n/4}$$

1 Code

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad
from scipy.linalg import solve

f = lambda x: 1 / (1 + x**2)

def composite_trap(a, b, n, f):
    h = (b-a) / n
    x = np.linspace(a, b, n)
    I = h * (0.5 * (f(x[0]) + f(x[-1])) + np.sum(f(x[1:-1])))
    return I

def composite_simpsons(a, b, n, f):
    h = (b-a) / n
    x = np.linspace(a, b, n)
    print(x.size)
    I = (h / 3) * (f(x[0]) + f(x[-1]) + 4*np.sum(f(x[1::2]))
                   + 2*np.sum(f(x[2:-1:2])))
    return I

a, b = -5, 5

n_trap = 1291 # calculated from error formulas
n_simps = 70

I_trap = composite_trap(a, b, n_trap, f)
I_simps = composite_simpsons(a, b, n_simps, f)
I_scipy, err = quad(f, a, b)
I_scipy_lowtol, err = quad(f, a, b, epsabs=1e-4)

print(f'The approximate integral using the composite
      Trapezoidal method is {I_trap}')
print(f'The approximate integral using the composite
      Simpsons method is {I_simps}')
print(f'The approximate integral using the built in scipy
      quad is {I_scipy}')
print(f'The approximate integral using the built in scipy
      with tol={1e-4} is {I_scipy_lowtol}')

f2 = lambda x: x*np.cos(1/x)
a2 = 0
b2 = 1
n2 = 5

I2_simps = composite_simpsons(a2, b2, n2, f2)
print(f'The approximate integral with composite Simpsons
      method is {I2_simps}')
```

```
'''question 3'''

A = np.array([[1, 2*np.sqrt(2), 8],
              [1, 4, 16],
              [1, 1, 1]])
B = np.array([0,
              0,
              1])

X = solve(A, B)
print(X)
```