APPM 4600 Homework 11

15 November 2024

1. (a) The code implementing composite trapezoidal rule and composite simpson's rule is listed below.

```python
#! /usr/bin/env python3
import numpy as np

# APPM 4600, Homework 11, Problem 1a
# Edward Wawrzynek

def eval_composite_trap(M,a,b,f):
    h = (b-a)/M
    x = np.arange(0, M+1)*h + a
    # trapezoidal weights
    w = h*np.ones(M+1)
    w[0] = h*0.5
    w[-1] = h*0.5

    I = np.sum(f(x) * w)

    return I, x, w

def eval_composite_simpsons(M,a,b,f):
    # force M even
    M = int(M/2)*2

    h = (b-a)/(M)
    x = np.arange(0, (M)+1)*h + a
    w = np.ones((M)+1)
    for i in range(M):
        w[2*i+1] = 4
        if i < M-1:
            w[2*i+2] = 2

    w = w * h/3

    I = np.sum(f(x) * w)
    return I, x, w

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

    method = eval_composite_trap
    a = -5
    b = 5
    N = 1000
    I, _, _ = method(N, a, b, f)

    print(I)

driver()
```

(b) We are integrating the function

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

The function has derivatives

$$f^{(2)}(s) = \frac{8s^2}{(1+s^2)^3} - \frac{2}{(1+s^2)^2},$$
$$f^{(4)}(s) = \frac{-288s^2}{(s^2+1)^4} + \frac{24}{(s^2+1)^3} + \frac{384s^4}{(s^2+1)^5}.$$

Over $[-5,5]$, these are bound by $\left|f^{(2)}(s)\right| \leq 2$ and $\left|f^{(2)}(s)\right| \leq 24$.
The error estimate for composite Trapezoidal on f over the interval $[a,b]$ with spacing h is

$$|E_T(f,a,b,h)| = \frac{f^{(2)}(\eta)h^2(b-a)}{12},$$

for some $\eta \in [a,b]$. We have the endpoints $a=-5$, $b=5$, and interval $h = \frac{b-a}{N} = \frac{10}{N}$. We want $|E| < 10^{-4}$, which implies

$$10^{-4} > \frac{2h^2(10)}{12} = \frac{(2)(10^3)}{12N^2} \implies N > \sqrt{\frac{2(10^3)}{12(10^{-4})}} \approx 1290.$$

The error estimate for composite Simpson on f over $[a,b]$ with spacing h is

$$|E_S(f,a,b,h)| = \frac{f^{(4)}(\eta)h^4(b-a)}{180},$$

where $\eta \in [a,b]$. As before, we want $|E_T| < 10^{-4}$, which implies

$$10^{-4} > \frac{24(10^5)}{180N^4} \implies N > \left(\frac{24(10^5)}{180(10^{-4})}\right)^{\frac{1}{4}} \approx 108.$$

(c) The code used in this question is listed at the end of the question. Using scipy's `quad` gives an approximate answer $I \approx 2.7468015$. Using the number of terms found in the previous question, we get absolute error 1.5×10^{-7} and 5.2×10^{-9} for our implementation of trapezoidal and Simpsons', respectively.

Scipy's quadrature requires 63 nodes to achieve a tolerance of 10^{-4} and 147 nodes to achieve a tolerance of 10^{-6}.

```
#! /usr/bin/env python3
import numpy as np
import scipy
import scipy.integrate

# APPM 4600, Homework 11, Problem 1c
# Edward Wawrzynek

def eval_composite_trap(M,a,b,f):
    h = (b-a)/M
    x = np.arange(0, M+1)*h + a
    # trapezoidal weights
    w = h*np.ones(M+1)
    w[0] = h*0.5
    w[-1] = h*0.5

    I = np.sum(f(x) * w)

    return I, x, w

def eval_composite_simpsons(M,a,b,f):
    # force M even
```

```
    M = int(M/2)*2

    h = (b-a)/(M)
    x = np.arange(0, (M)+1)*h + a
    w = np.ones((M)+1)
    for i in range(int(M/2)):
        w[2*i+1] = 4
        if i < int(M/2)-1:
            w[2*i+2] = 2

    w = w * h/3

    I = np.sum(f(x) * w)
    return I, x, w

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

    a = -5
    b = 5
    I_exact, _ = scipy.integrate.quad(f, a, b, epsabs=1e-12)
    I_trap, _, _ = eval_composite_trap(1290, a, b, f)
    I_simp, _, _ = eval_composite_simpsons(108, a, b, f)

    print("Trapezoidal error:")
    print(np.abs(I_exact - I_trap))

    print("Simpsons' error:")
    print(np.abs(I_exact - I_simp))

    I1, _, info1 = scipy.integrate.quad(f, a, b, epsabs=1e-4, full_output=True)
    print("quadpack iters 10^-4:")
    print(info1['neval'])

    I2, _, info2 = scipy.integrate.quad(f, a, b, epsabs=1e-6, full_output=True)
    print("quadpack iters 10^-4:")
    print(info2['neval'])

driver()
```

2. The code used in this question is listed at end.

   We let $t = x^{-1}$ and have

$$I = \int_1^\infty \frac{\cos(x)}{x^3}\,\mathrm{d}x = \int_1^0 -t^3\cos\left(\frac{1}{t}\right)\frac{\mathrm{d}t}{t^2} = \int_0^1 t\cos\left(\frac{1}{t}\right)\mathrm{d}t.$$

   We apply composite Simpson's method with 5 nodes and get

$$I \approx 0.014685,$$

   which is a relative error of 0.19.

```python
#! /usr/bin/env python3
import numpy as np
import scipy
import scipy.integrate

# APPM 4600, Homework 11, Problem 2
# Edward Wawrzynek

def eval_composite_simpsons(M,a,b,f):
  # force M even
  M = int(M/2)*2

  h = (b-a)/(M)
  x = np.arange(0, (M)+1)*h + a
  w = np.ones((M)+1)
  for i in range(int(M/2)):
    w[2*i+1] = 4
    if i < int(M/2)-1:
      w[2*i+2] = 2

  print(w)
  w = w * h/3

  I = np.sum(f(x) * w)
  return I, x, w

def driver():
  f = lambda t: t*np.cos(1/t)

  a = 1e-12
  b = 1
  N = 4
  I, _, _ = eval_composite_simpsons(N, a, b, f)

  I_exact, _ = scipy.integrate.quad(f, a, b)


  print((I - I_exact) / I_exact)

  print(I)
  print(I_exact)

driver()
```

3. We have the asymptotic formulas

$$\begin{aligned}
I - I_n &= \frac{C_1}{n\sqrt{n}} + \frac{C_2}{n^2} + \frac{C_3}{n^2\sqrt{n}} + \frac{C_4}{n^3} + \dots, \\
I - I_{\frac{n}{2}} &= 2\sqrt{2}\frac{C_1}{n\sqrt{n}} + 4\frac{C_2}{n^2} + 4\sqrt{2}\frac{C_3}{n^2\sqrt{n}} + 8\frac{C_4}{n^3} + \dots, \\
I - I_{\frac{n}{4}} &= 4\sqrt{4}\frac{C_1}{n\sqrt{n}} + 16\frac{C_2}{n^2} + 16\sqrt{4}\frac{C_3}{n^2\sqrt{n}} + 64\frac{C_4}{n^3} + \dots.
\end{aligned}$$

We add the first two together to get a formula with error on order $\frac{1}{n^2}$,

$$\begin{aligned}
I - I_n - \frac{1}{2\sqrt{2}}I + \frac{1}{2\sqrt{2}}I_{\frac{n}{2}} &= \left(1 - \frac{4}{2\sqrt{2}}\right)\frac{C_2}{n^2} + \left(1 - \frac{4\sqrt{2}}{2\sqrt{2}}\right)\frac{C_3}{n^2\sqrt{n}} + \left(1 - \frac{8}{2\sqrt{2}}\right)\frac{C_4}{n^3} + \dots \\
&= \left(1 - \sqrt{2}\right)\frac{C_2}{n^2} - \frac{C_3}{n^2\sqrt{n}} + \left(1 - 2\sqrt{2}\right)\frac{C_4}{n^3} + \dots.
\end{aligned}$$

Similarly, we add the second two expansions to have

$$\begin{aligned}
I - I_n - \frac{1}{4\sqrt{4}}I + \frac{1}{4\sqrt{4}}I_{\frac{n}{4}} &= \left(1 - \frac{16}{4\sqrt{4}}\right)\frac{C_2}{n^2} + \left(1 - \frac{16\sqrt{4}}{4\sqrt{4}}\right)\frac{C_3}{n^2\sqrt{n}} + \left(1 - \frac{64}{4\sqrt{4}}\right)\frac{C_4}{n^3} + \dots \\
&= -\frac{C_2}{n^2} - 3\frac{C_3}{n^2\sqrt{n}} - 7\frac{C_4}{n^3} + \dots.
\end{aligned}$$

Thus, we have an expansion in order $\frac{1}{n^2\sqrt{n}}$,

$$\begin{aligned}
&I - I_n - \frac{1}{2\sqrt{2}}I + \frac{1}{2\sqrt{2}}I_{\frac{n}{2}} - \frac{1-\sqrt{2}}{1-\sqrt{4}}I + \frac{1-\sqrt{2}}{1-\sqrt{4}}I_n + \frac{1-\sqrt{2}}{1-\sqrt{4}}\frac{1}{4\sqrt{4}}I - \frac{1-\sqrt{2}}{1-\sqrt{4}}\frac{1}{4\sqrt{4}}I_{\frac{n}{4}} \\
&= \left(2 - \frac{\sqrt{2}}{4} - \sqrt{2} - \frac{1-\sqrt{2}}{8}\right)I + \left(-1 - 1 + \sqrt{2}\right)I_n + \frac{\sqrt{2}}{4}I_{\frac{n}{2}} + \frac{1-\sqrt{2}}{8}I_{\frac{n}{4}} \\
&= \left(\frac{15}{8} - \frac{9}{8}\sqrt{2}\right)I + \left(-2 + \sqrt{2}\right)I_n + \frac{\sqrt{2}}{4}I_{\frac{n}{2}} + \frac{1-\sqrt{2}}{8}I_{\frac{n}{4}} \\
&= \left(3\frac{1-\sqrt{2}}{1-\sqrt{4}} - 1\right)\frac{C_3}{n^2\sqrt{n}} + \left(1 - 2\sqrt{2} - \left(1 - 4\sqrt{4}\right)\frac{1-\sqrt{2}}{1-\sqrt{4}}\right)\frac{C_4}{n^3} + \dots.
\end{aligned}$$

Finally, this gives the extrapolation

$$I - \frac{16 + 16\sqrt{2}}{15 - 9\sqrt{2}}I_n + \frac{2\sqrt{2}}{15 - 9\sqrt{2}}I_{\frac{n}{2}} + \frac{1-\sqrt{2}}{15 - 9\sqrt{2}}I_{\frac{n}{4}} = O(\frac{1}{n^2\sqrt{n}}).$$