# Zhang_Jeffrey_code5

April 10, 2022

## 1 AMSC 460 HW 5 Part 2

Given a function $f$ with a singularity at a point $x_0$, we may obtain a composite quadrature rule that uses more points near the singularity by performing an adequate change of variables.

(a) Let $f(x) = \cos(x)/\sqrt[3]{x}$. We want to estimate

$$I = \int_0^1 f(x)dx \approx 1.3212$$

Since $f$ blows up at $x = 0$, the composite trapezoid rule to estimate $I$ gives infinity as a result. However, it also holds that

$$\int_0^{10^{-8}} f(x)dx \approx 7 \cdot 10^{-6}.$$

So, potentially, we could apply a quadrature rule on the interval $\left[10^{-8}, 1\right]$ to obtain an approximation of $I$ correct up to 5 digits. Estimate $I$ using n = $\left\{10^3, 10^4, 10^5, 10^6\right\}$. How accurate are your results?

Let $f \in C^2[a,b], h = (b-a)/n$, and $x_j = a + jh$, for each $j = 0, 1, \ldots, n$. There exists a $\mu \in (a,b)$ for which the Composite Trapezoidal rule for $n$ subintervals can be written with its error term as

$$\int_a^b f(x)dx = \frac{h}{2}\left[f(a) + 2\sum_{j=1}^{n-1} f\left(x_j\right) + f(b)\right] - \frac{b-a}{12}h^2 f''(\mu)$$

or

$$\left|\int_a^b f(x)dx - h\left(\frac{f(a)}{2} + \sum_{j=1}^{n-1} f\left(x_j\right) + \frac{f(b)}{2}\right)\right| \leq \max_{x\in[a,b]} \left|f''(x)\right| \frac{(b-a)^3}{12}\frac{1}{n^2}$$

In the case of this problem, $a = 10^{-8}$, $b = 1$, and $x_j = 10^{-8} + j\frac{1-10^{-8}}{n}$.

To find the max error, we find the max of the 2nd derivative:

$$f'(x) = -\frac{3x\sin(x) + \cos(x)}{3x^{\frac{4}{3}}}$$
$$f''(x) = \frac{6x\sin(x) + \left(4 - 9x^2\right)\cos(x)}{9x^{\frac{7}{3}}}$$

Graphing the 2nd derivative shows that the maximum value occurs at $10^{-8}$ in the $[10^{-8}, 1]$

From the answers given to us above, we can approximate an answer for:

$$I = \int_{10^{-8}}^{1} f(x)dx \approx 1.3212 - 7 \cdot 10^{-6} = 1.321193$$

[3]:
```
import math
import numpy as np
from sympy import *
from scipy.integrate import quad
import scipy.special as special
from sympy.plotting import plot

def f(x):
    return math.cos(x)/math.pow(x,1/3)

def f2(x):
    return (6*x*math.sin(x) + (4-9*x**2)*math.cos(x))/(9*math.pow(x,7/3))

def Composite_Trapezoidal_Rule(n,a,b):
    h = (b-a)/n
    summation = 0
    for j in range(1,n):
        summation += f(a + j*h)
    return (h/2)*(f(a) + 2*summation + f(b))
```

[65]:
```
Composite_Trapezoidal_Rule(10**3,10**-8,1)
```

[65]: 1.5435687059970864

[67]:
```
1.5435687059970864 - 1.321193
```

[67]: 0.22237570599708634

[68]:
```
Composite_Trapezoidal_Rule(10**4,10**-8,1)
```

[68]: 1.342333717567031

[69]:
```
1.342333717567031 - 1.321193
```

[69]: 0.021140717567030842

[70]:
```
Composite_Trapezoidal_Rule(10**5,10**-8,1)
```

[70]: 1.3230915177381277

[71]:
```
1.3230915177381277 - 1.321193
```

[71]: 0.0018985177381276586

[72]:
```
Composite_Trapezoidal_Rule(10**6,10**-8,1)
```

[72]: 1.3213566203735232

[73]:
```
1.3213566203735232 - 1.321193
```

[73]: 0.0001636203735231323

We can see that with the given ns, as n increases, the error decreases using the composite trapezoid rule.

(b) Make the substitution $t = x^{2/3}$ in (1) to express $I$ as the integral of another function, which we shall call $g(t)$. (Note that $g$ is smooth, because the singularity of $f$ at $x_0 = 0$ is compensated by the differential of the change of variables.) Estimate $I$ with the same values of n as before and compare your results.

$$I = \int_0^1 f(x)dt = \int_0^1 \frac{3}{2}\cos(t^{3/2})dt$$

[4]:
```
def g(t):
    return (3/2)*math.cos(t**(3/2))

def Composite_Trapezoidal_Rule_t(n,a,b):
    h = (b-a)/n
    summation = 0
    for j in range(1,n):
        summation += g(a + j*h)
    return (h/2)*(g(a) + 2*summation + g(b))
```

[5]:
```
Composite_Trapezoidal_Rule_t(10**3,10**-8,1)
```

[5]: 1.3212229013700847

[9]:
```
1.3212229013700847 - 1.321193
```

[9]: 2.9901370084628454e-05

[6]:
```
Composite_Trapezoidal_Rule_t(10**4,10**-8,1)
```

[6]: 1.321223057568142

[10]:
```
1.321223057568142 - 1.321193
```

[10]: 3.0057568142005664e-05

[7]:
```
Composite_Trapezoidal_Rule_t(10**5,10**-8,1)
```

[7]: 1.3212230591301257

[11]:
```
1.3212230591301257 - 1.321193
```

[11]: 3.0059130125659195e-05

[8]:
```
Composite_Trapezoidal_Rule_t(10**6,10**-8,1)
```

[8]: 1.3212230591457068

[12]:
```
1.3212230591457068 - 1.321193
```

[12]: 3.0059145706751167e-05

The substitution made the errors lower.