Numerical Methods

Math 3338 – Spring 2022

Worksheet 6

Integration

1 Reading

CP	5.1, 5.6
NMEP	6.2

Table 1: Sections Covered

2 Integration

Evaluate the following,

$$\int_0^8 (x+1)\sqrt{1-\frac{1}{2}\sin^2(x)}\,dx$$

Remember, an integral is an area under a curve. Let's plot this graph and see how it looks. Figure 1 does

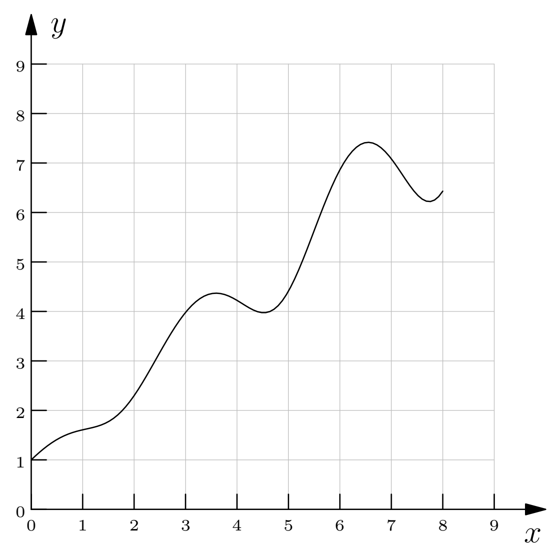

Figure 1: The graph of $(x+1)\sqrt{1-\frac{1}{2}\sin^2(x)}$

not make this function look easier to integrate.

We can't find the antiderivative of this function, I'm sorry if you tried. Luckily, we're in a class called Numerical Methods, so we'll evaluate this numerically. The basic idea is to approximate the trouble function by a nicer function, and integrate that instead.

2.1 Constant Function

Break the interval into regions, and approximate the function by a constant in each region. Figure 2 shows an example of this. you *should* recognize this.

It's a Riemann sum using midpoints. This is a nice example, the subintervals are the same width and we're using a consistent point in each region. Mathematically, neither of these are necessary, you can have different sized subintervals and choose whatever point your heart desires.

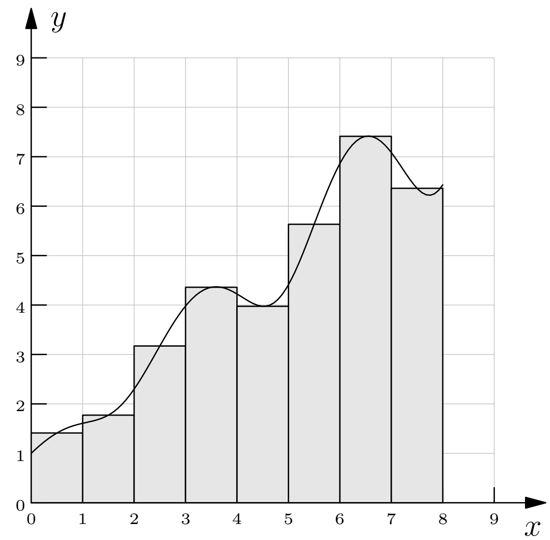

Figure 2: Constant Approximation

As a summation, this is

$$\sum_{i=0}^{n-1} f(x_i)\Delta x$$

where there are n subintervals, x_i is a point in the i^{th} subinterval, and Δx is the width of the subintervals (assuming a constant width). Notice, $f(x_i)\Delta x$ is just the area of a rectangle.

2.2 Linear

We want more accuracy! Instead of a constant function, let's approximate the function by a line. Figure 3 shows this.

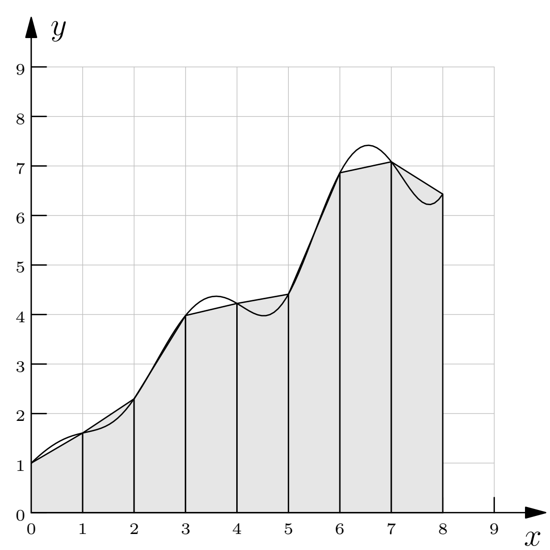

Figure 3: Linear Approximation

This is the trapezoid method. The area of a trapezoid is $\frac{(\text{height}_1+\text{height}_2)}{2}\cdot\text{base}$ (this actually makes perfect sense if you think about it). Therefore, if there are n subintervals of $[a, b]$, the trapezoid method is,

$$\sum_{i=0}^{n-1}\left(\frac{f(a+i\Delta x)+f(a+(i+1)\Delta x)}{2}\right)\Delta x=\frac{1}{2}f(a)\Delta x+\sum_{i=1}^{n-1}f(a+i\Delta x)\Delta x+\frac{1}{2}f(b)\Delta x$$

2.3 Quadratic

This is Simpson's method. Figure 4 shows a representation. The idea is to represent the function using

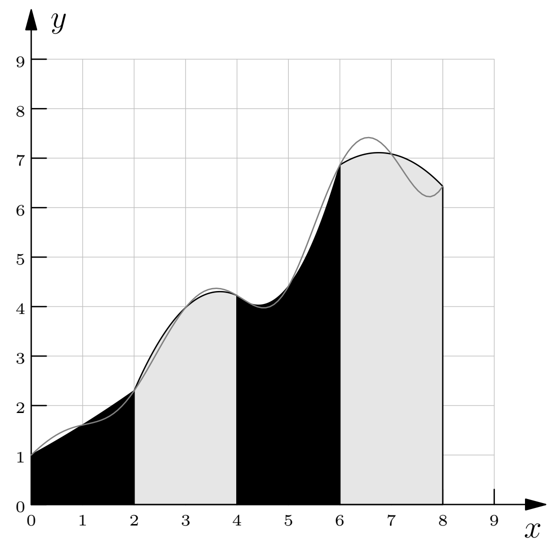

Figure 4: Simpson's Method

quadratics. However, you need 3 points to fit a quadratic, that's why the figure has only 4 distinct regions. This also implies you need an even number of subintervals, which isn't always feasible with real data. The formula is,

$$\frac{\Delta x}{3}\left(f(x_0)+4f(x_1)+2f(x_2)+4f(x_3)+2f(x_4)+\cdots+4f(x_{2n-1})+f(x_{2n})\right)$$

where $x_i = a + i\Delta x$.

2.4 Higher Order

You could proceed to approximate the function with higher order polynomials. This works, but is not the best way to do it. There is a method called *Gaussian quadrature*. This is a clever method that varies the width of each interval to more accurately approximate the integral. This is covered in detail in Section 5.6 in the text. I highly recommend reading through this. In general, we'll black box Gaussian quadrature (we'll use it, but won't know exactly how it works).

Numerical Methods

Math 3338 – Spring 2022

Homework 6 (Due: Thursday, February 3)

Problem 1 (1 pt) Write three functions, `riemann`, `trapezoidal`, and `simpsons`. The functions will have 4 inputs, a function f, a, b, and N. Your programs should be robust enough to handle the situations $a = b$ and $b < a$. Each of these functions should use vectorization and numpy.

There is a Pickle file on Canvas. It's a list `[function,a,b,N,riemann,trap,simp]`. There is also a file called "my_functions.py". You'll need to import this into the namespace with the pickle file `from my_functions import *`.

Problem 2 (1 pt) Let $f(x) = \int_0^x (t+1)\sqrt{1 - \frac{1}{2}\sin^2(t)}\, dt$. Make a graph of $f(x)$ for $-5 \leq x \leq 5$.

Problem 3 (1 pt) The goal of this problem is to explore the speed of numpy vs non-numpy. Write a new trapezoidal function `trapezoidal_new` that doesn't use numpy (the opposite of your original). Evaluate $\int_{-2}^{2} e^{-x^2}\, dx$ using both functions and many small and large values of N, timing each trial. Make a table comparing the computation times. Discuss the results.

Note: My numpy trapezoid rule computes the integral in 0.045578 seconds for $N = 1,000,000$. Yours should be close to this (or faster).

Problem 4 (1 pt) The goal of this problem is to explore the speed of our generic `trapezoidal` method. Evaluate $\int_{-2}^{2} e^{-x^2}\, dx$ two ways, first using our `trapezoidal` function and second using an ad-hoc function, on where the function is within the definition of the trapezoid method. Evaluate this integral for many values of N, timing each trial. Make a table comparing the computation times. Discuss the results.