Numerical Methods with Python

Numerical Integration

Problem Statement:

- Given some known function $f(x)$, we wish to calculate the definite integral from x_1 to x_2

Numerical Integration

Problem Statement:

- Given some known function $f(x)$, we wish to calculate the definite integral from x_1 to x_2
- Example: $p(t) = p_0 + \int_{t_i}^{t_f} F(t)dt$
  - Given the function $F(t)$ and the initial momentum p_0, we can calculate $p(t)$

Numerical Integration

Problem Statement:

- Given some known function $f(x)$, we wish to calculate the definite integral from x_1 to x_2
- Example: $p(t) = p_0 + \int_{t_i}^{t_f} F(t)dt$
  - Given the function $F(t)$ and the initial momentum p_0, we can calculate $p(t)$
- Example: $W = \int_{x_i}^{x_f} F_x(x)dx$

Numerical Integration

What *is* an integral?
Question: if $y(x)$ is a curve describing the boundary of some shape, what does $\int_{x_{min}}^{x_{max}} y(x)dx$ represent?

Numerical Integration

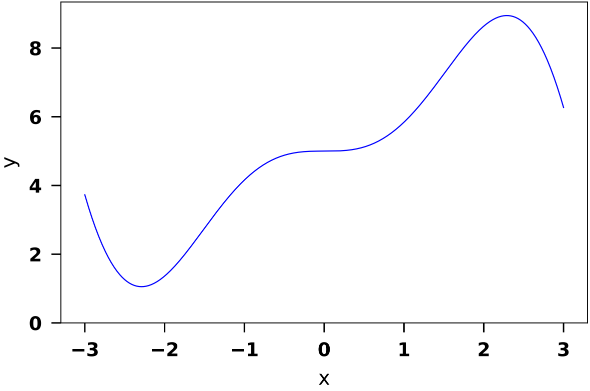
What *is* an integral?
Question: if $y(x)$ is a curve describing the boundary of some shape, what does $\int_{x_{min}}^{x_{max}} y(x)dx$ represent?

- It is just the **area under the curve**

Example: a triangle

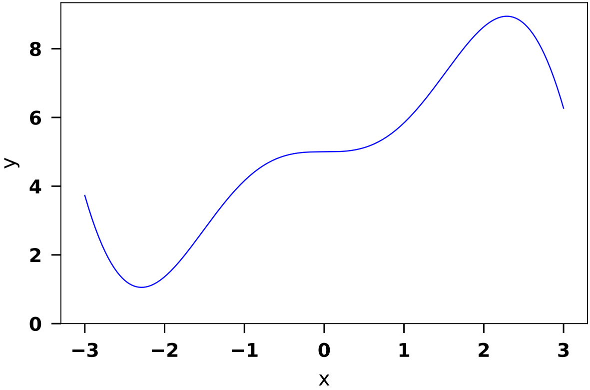
Numerical Integration

Not all functions are so easy...



Our Approach

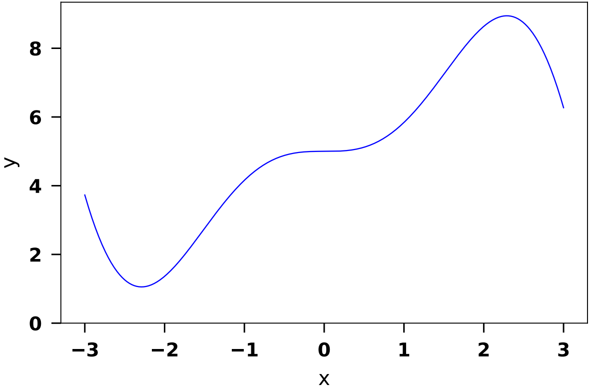
We don't know the area of this complicated function
We *do* know the area of a rectangle!



Our Approach

We don't know the area of this complicated function
We *do* know the area of a rectangle!

- We can divide the area under the curve into a bunch of tiny rectangles, then add the total area of all of the rectangles

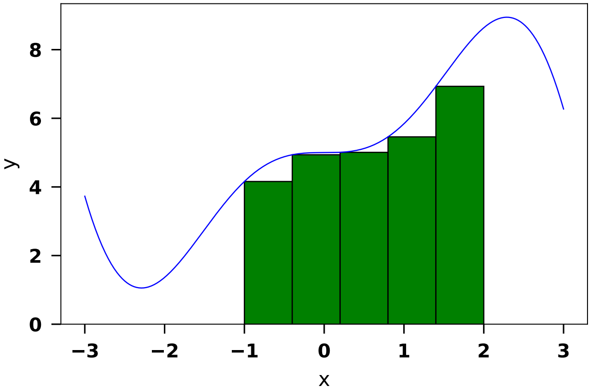


Our Approach

We don't know the area of this complicated function

We *do* know the area of a rectangle!

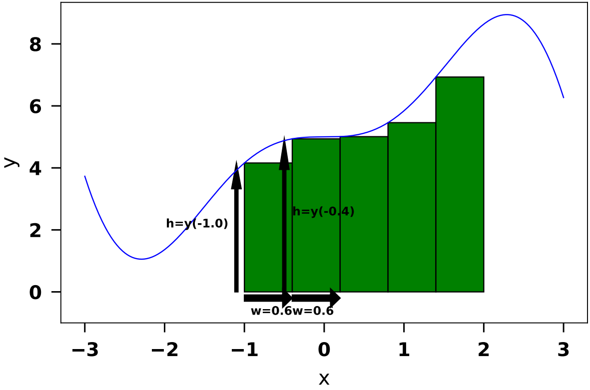
- We can divide the area under the curve into a bunch of tiny rectangles, then add the total area of all of the rectangles
- $\int_{-1}^{2} y(x)dx \approx$



Our Approach

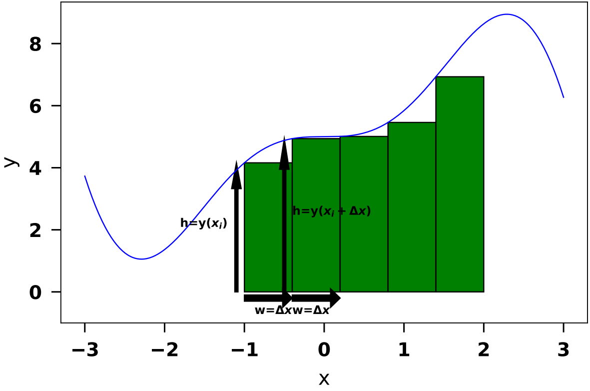
Each rectangle has a height $h = y(x)$ and a constant width $w = \Delta x$

The area of each rectangle is $h \cdot w$



Our Approach

More generally:



Our Approach

More generally:

$$\int_{x_i}^{x_f} y(x)dx \approxeq \sum_{k=0}^{n-1} y(x_i + k\Delta x) \cdot \Delta x$$

This is called a **Riemann sum**

Our Approach

We can easily code this!

1. Write a function to calculate $y(x)$
2. Given x_i, x_f, and n, find Δx
3. Loop from $k = 0$ to $k = n - 1$, and sum the quantity $y(x_i + k\Delta x) \cdot \Delta x$

https://upload.wikimedia.org/wikipedia/commons/1/19/Riemann_sum_%

Example

Integrate $y(x) = 2x$ from 0 to 3

Example

Integrate $y(x) = e^{-x^2}$ from -1 to 1

Example

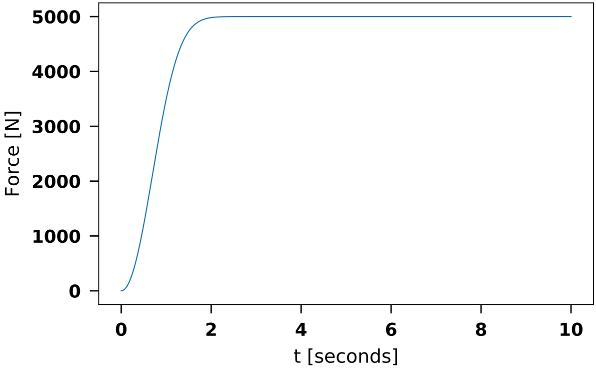
What if we don't know the functional form of $y(x)$ but we have a set of (x, y) points?

Example

Variable acceleration

A certain car has a transmission which enables it to supply a variable force to accelerate the car as a function of time:

$$F(t) = 5000\ \mathrm{N} \times (1 - e^{-0.2t^3 - t^2})$$



Starting from rest, what speed is the 1100 kg car able to travel at after 8 seconds?

Uncertainty of Riemann Estimate

This is only an approximation!
The integral

$$I = \int_{x_i}^{x_f} y(x)dx$$

has an *exact* answer.
If we knew what it was, we wouldn't need to approximate it!
We are *approximating* this result by:

$$A = \sum_{k=0}^{n-1} y(x_i + k\Delta x) \cdot \Delta x \approxeq \int_{x_i}^{x_f} y(x)dx$$

Uncertainty of Riemann Estimate

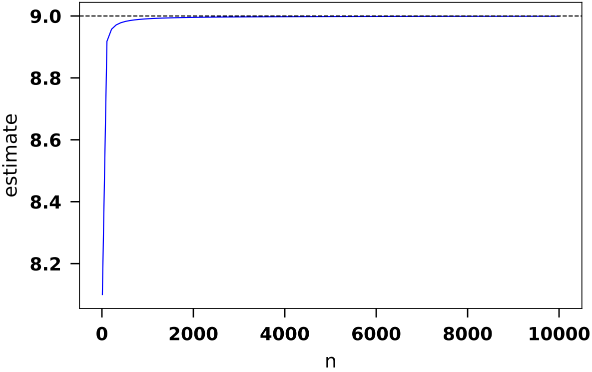
$$I = \int_{x_i}^{x_f} y(x)dx$$

$$A = \sum_{k=0}^{n-1} y(x_i + k\Delta x) \cdot \Delta x \approxeq \int_{x_i}^{x_f} y(x)dx$$

In general, A and I will not be the same.
We don't know the exact answer I, but we *can* estimate how close our estimate A guaranteed to be to it

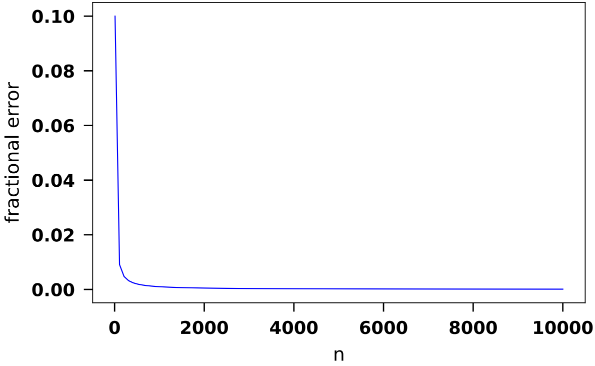
Uncertainty of Riemann Estimate

We have seen that using a larger number of rectangles n results in a more accurate estimate



Uncertainty of Riemann Estimate

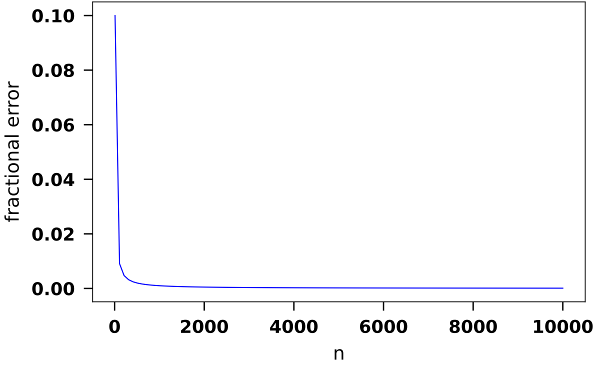
In fact, the error of our estimate seems to decrease like $1/n$



Uncertainty of Riemann Estimate

Let's see if we can quantify the error in our Riemann sum estimate without knowing the true value

This is very important!



Why Estimating Uncertainty is Important

As a scientist/engineer, you will be using computer programs to make predictions

Your prediction is completely useless unless you know how precise it is!

If you are asked to calculate how much weight a cable can bear and your program approximates the answer to be 3500 N.

- We know this answer is probably not the exactly correct value
- The correct way to report this result is not "the cable can hold 3500 N" but rather "The true strength of the cable lies somewhere in the range of 3400 N and 3600 N"

Why Estimating Uncertainty is Important

As a scientist/engineer, you will be using computer programs to make predictions

Your prediction is completely useless unless you know how precise it is!

If you are asked to calculate how much weight a cable can bear and your program approximates the answer to be 3500 N.

- We know this answer is probably not the exactly correct value
- The correct way to report this result is not "the cable can hold 3500 N" but rather "The true strength of the cable lies somewhere in the range of 3400 N and 3600 N"
  - "I estimate the strength of the cable to be 3500 N, with a precision of 100 N"

Why Estimating Uncertainty is Important

In this case, you probably wouldn't want to hang anything more than 3400 N from the cable

- Even though your estimate is 3500 N, the *true* value could be as low as 3400 N

Why Estimating Uncertainty is Important

In this case, you probably wouldn't want to hang anything more than 3400 N from the cable

- Even though your estimate is 3500 N, the *true* value could be as low as 3400 N
- If you didn't estimate the uncertainty of your measurement, you would never know this. You would suspend a 3500 N load, and the cable could snap.

Why Estimating Uncertainty is Important

In this case, you probably wouldn't want to hang anything more than 3400 N from the cable

- Even though your estimate is 3500 N, the *true* value could be as low as 3400 N
- If you didn't estimate the uncertainty of your measurement, you would never know this. You would suspend a 3500 N load, and the cable could snap.

Any measurement or numerical approximation consists of both a value and an associated uncertainty

Estimating the Uncertainty of a Riemann Sum

Estimating the Uncertainty of a Riemann Sum

The absolute error is bounded such that:

$$|E| \leq \frac{1}{2}\frac{M(x_f - x_i)^2}{n}$$

Where:

$$M = \max\left(|y'(x)|, x_i \leq x \leq x_f\right)$$

Estimating the Uncertainty of a Riemann Sum

$|E|$ is the **maximum possible** error on our estimate

- We *could* be closer to the actual value
- We are *guaranteed* to be within $|E|$ of it

Estimating the Uncertainty of a Riemann Sum

$|E|$ is the **maximum possible** error on our estimate

- We *could* be closer to the actual value
- We are *guaranteed* to be within $|E|$ of it

If A_{est} is our estimate, and A_{true} is the exact result, then:

$$A_{est} - |E| \leq A_{true} \leq A_{est} + |E|$$

Estimating the Uncertainty of a Riemann Sum

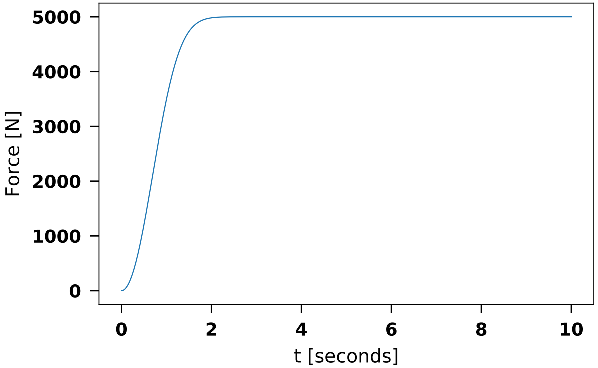
Let's modify our code to include the uncertainty along with our actual estimate

Example

Variable acceleration

A certain car has a transmission which enables it to supply a variable force to accelerate the car as a function of time:

$$F(t) = 5000\ \mathrm{N} \times (1 - e^{-0.2t^3 - t^2})$$



Starting from rest, what speed is the 1100 kg car able to travel at after 8 seconds?

Improving the approximation

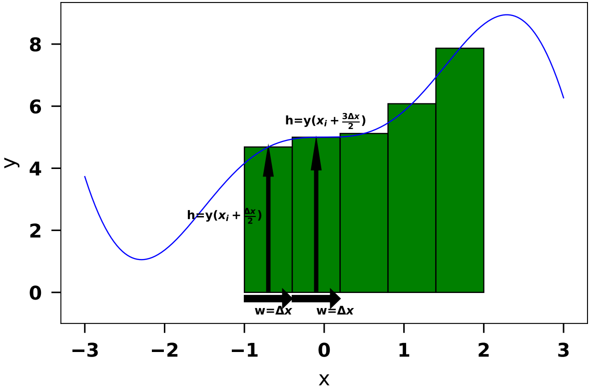
We have seen that by using more and more rectangles we can get a better estimate

Improving the approximation

We have seen that by using more and more rectangles we can get a better estimate

- Is there a more efficient way? (A way that would give a closer estimate with the same number of rectangles?)

The Midpoint Riemann Sum



Uncertainty of midpoint method

I won't derive this result like the last one, I'll just give you the formula:

$$|E| \leq \frac{1}{24}\frac{M_2(x_f - x_i)^3}{n^2}$$

Where M_2 is the maximum value of the *second* derivative

$$M_2 = \max\left(|y''(x)|, x_i \leq x \leq x_f\right)$$

Midpoint Riemann vs Left Riemann