Calculus tutorial

Excerpt from the **No bullshit guide to math and physics** by Ivan Savov

***Abstract*—This tutorial summarizes the key ideas of calculus using math formulas, visual intuition, and Python code examples.**

Click this link to view the tutorial online: `bit.ly/calctut3`.

CONTENTS

I. Introduction

Calculus is the study of functions that change over time. We use calculus concepts to describe various quantities in physics, chemistry, biology, engineering, business, and other fields where quantitative math models are used. If we know that some quantity of interest is described by the function $f(t)$ at time t, then the techniques of calculus allow us to do all kinds of useful calculations based on the function $f(t)$. The two main techniques of calculus involve calculating how functions *change* over time (derivatives), and how to compute the total *accumulation* of functions over time (integrals). Derivatives and integrals might sound like fancy math jargon, but actually they are common-sense concepts that you're already familiar with, as you'll see in the following example.

A. Example 1: file download

Suppose you're downloading a 720[MB] file from the internet to your computer. At $t = 0$ you click "save as" in your browser and the download starts. Consider the function $f(t)$ that describes the amount of disk space taken by the partially-downloaded file at time t. At time t, your browser reports the download progress as a percentage that corresponds to the fraction $\frac{f(t)}{720[\text{MB}]}$.

Download rate: The *derivative function* $f'(t)$, pronounced "f prime," describes how the function $f(t)$ changes over time. In our example $f'(t)$ is the download speed. If your downloading speed is $f'(t) = 2[\text{MB/s}]$, then the file size $f(t)$ will increase by 2[MB] each second. If you maintain this download speed, the file size will grow at a constant rate: $f(0) = 0[\text{MB}]$, $f(1) = 2[\text{MB}]$, $f(2) = 4[\text{MB}]$, …, $f(100) = 200[\text{MB}]$, and so on until $t = 360[\text{s}]$ when we expect the download to complete.

Let's look at how your browser calculates the "estimated time remaining" for the download at time t. To calculate the time until the download completes, we divide the amount of data that remains to be downloaded by the current download speed:

$$\text{time remaining at } t = \frac{720 - f(t)}{f'(t)} \quad [\text{s}].$$

The bigger the derivative $f'(t)$, the faster the download will finish. If your internet connection were 10 times faster, the download would finish 10 times more quickly.

Inverse problem: Let's now consider the download scenario from the point of view of the modem that connects your computer to the internet. Any data you download comes through the modem, so the modem knows the download rate $f'(t)[\text{MB/s}]$ at all times during the download.

The modem is separate from your computer, so it doesn't know the file size $f(t)$ as the download progresses. Nevertheless, the modem can infer the file size at time t from the transmission rate $f'(t)$. Think about it—if the modem sees data flowing through at the rate of $f'(t) = 2[\text{MB/s}]$, then it knows that the data accumulated on your computer is growing at the rate of 2[MB] each second. In calculus, we describe the total file size accumulated until time $t = \tau$ (the Greek letter *tau*) as the *integral* of the download rate $f'(x)$ between $t = 0$ and $t = \tau$:

$$f(\tau) \;=\; \int_{t=0}^{t=\tau} f'(t)\, dt.$$

The symbol $\int$ is an elongated S that stands for *sum*. Indeed, the "integral of $f'(t)$ between 0 and τ" is in some sense the sum of $f'(t)$ during each time instant dt between $t = 0$ and $t = \tau$. To calculate the total accumulated file size, we split the time interval between $t = 0$ and $t = \tau$ into many short time intervals dt of length 1[s]. During each second, the file size grows by $f'(t)\, dt$, where $f'(t)$ is the the download rate measured in [MB/s], and dt is 1 [s]. Note the units check out, the data downloaded during one second is $f'(t)dt[\text{MB}]$. The file size on your computer at $t = \tau$ is the sum of these 1-second contributions $f'(t)\, dt$ as t varies from $t = 0$ to $t = \tau$.

The situation described in the above example shows that calculus concepts are not some theoretical constructs reserved for math specialists, but something you encounter everyday. The derivative $q'(t)$ describe the rate of change of the quantity $q(t)$. The integral $\int_a^b q(t)\, dt$ measures the total accumulation of the quantity $q(t)$ during the time period from $t = a$ to $t = b$.

B. Infinity

The math symbol ∞ describes the concept of *infinity*. Infinity is the key building block for everything we do in calculus, so it's important that you develop the right way to think about infinity.

Infinity is not a number but a process. Consider the set of natural numbers $\mathbb{N} \stackrel{\text{def}}{=} \{0, 1, 2, 3, 4, 5, 6, \ldots\}$. The natural numbers describe the process of counting starting at 0. The natural number n is obtained by starting at 0 and performing the $+1$ operation n times. Geometrically, you can think of the $+1$ operation as taking one step to the right on the number line shown in Figure 1 (page 5). In this context, you can think of infinity ∞ as performing the $+1$ operation forever. Infinity is greater than any natural number n. Indeed, getting to n takes a finite number of steps, but ∞ describes taking an infinite number of steps so ∞ must be to the right of n.

Infinity is the main new concept in calculus. Everything else we'll talk about (numbers, variables, expressions, algebra, equations, functions, etc.) are standard topics from high school math, which I assume you're familiar with. Indeed, calculus can be described as the "infinity upgrade" to the high school math calculations you're familiar with that gives you a language for describing and solving a new class of problems.

Let's look at another example.

C. Example 2: Euler's number

Suppose you take out a loan with 100% nominal interest rate. This is a very bad loan that nobody would agree to the real world, but we'll use it for this example to make the math come out simpler. An interest of 100% calculated yearly means at the end of one year, you'll owe $(1 + 100\%) = (1 + 1) = 2$ times the amount you borrowed initially.

However, most banks don't calculate the interest owed only once per year, but more often. If the bank calculates the interest twice per year, during the first six months you'll have accrued $\frac{100\%}{2} = 50\%$ of interest, so you'll owe them $(1+50\%) = (1+\frac{1}{2}) = 1.5$ times the initial amount. Then during the second six months, the amount owed will grow by an additional $(1+50\%) = (1+\frac{1}{2}) = 1.5$, so at the end of the year, you'll owe $(1+\frac{1}{2})(1+\frac{1}{2}) = 2.25$.

If the bank computes the interest three times per year, the amounted owed after one year is $(1+\frac{1}{3})(1+\frac{1}{3})(1+\frac{1}{3}) = 2.370$. If they compute the interest four times per year (quarterly), then you'll owe $(1+\frac{1}{4})(1+\frac{1}{4})(1+\frac{1}{4})(1+\frac{1}{4}) = 2.441$. Note the amount owed after one year keeps changing, as the compounding is performed more frequently. In general, when the compounding is performed n times per year, the amount owed at the end of the year will be

$$\underbrace{\left(1+\tfrac{1}{n}\right)\left(1+\tfrac{1}{n}\right)\cdots\left(1+\tfrac{1}{n}\right)}_{n \text{ times}} = \left(1+\tfrac{1}{n}\right)^n.$$

With monthly compounding ($n = 12$), the amount owed will be $(1+\frac{1}{12})^{12} = 2.613$ at the end of one year. With daily compounding, the amount would be $(1+\frac{1}{365})^{365} = 2.715$. If computing the interest $n = 1000$ times per year, the amount ill be $(1+\frac{1}{1000})^{1000} = 2.717$. The amount owed keeps increasing, but it seems to "stabilize" around the value 2.71.

What happens if we perform the compounding even more frequently? Specifically, we want to know what happens if the compound interest interest is calculated infinitely often. The infinitely-often calculation corresponds to computing the *limit* of expression $(1+\frac{1}{n})^n$, as n goes to infinity, which is written as follows using math notation:

$$\lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n \quad = \quad e \quad = \quad 2.718281828\ldots.$$

This limit expression *converges* to the value $e = 2.71828\ldots$, which is known as *Euler's number*. If we borrow \$1000, we'll owe $\$1000e = \$2718.28$ at the end of one year.

The definition of the number e as a limit is a fascinating new concept that goes beyond the "regular" math operations that we learn in high school math. We're not talking about any particular large number n when calculating the expression $(1+\frac{1}{n})^n$, but the *process* of plugging in large and larger ns. This is what the limit notation $\lim\limits_{n\to\infty}(1+\dfrac{1}{n})^n$ means: it describes the behaviour of the expression $(1+\frac{1}{n})^n$ as n goes to infinity. We'll learn more about limits in Section III.

Euler's number e can also be obtained from another limit expression:

$$e = 1 + 1 + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \cdots = \lim_{n\to\infty}\sum_{k=0}^{n}\frac{1}{k!} = 2.718281828\ldots.$$

This alternative expression tells us we can compute e as the sum ($\sum$) with an infinite number of terms. Each term comes from a common "pattern" $\frac{1}{k!}$, where $k! = k\cdot(k-1)\cdots 3\cdot 2\cdot 1$ is the factorial function. The notation $\sum_{k=0}^{n}$ describes the summation starting at $k = 0$ and going all the way to $k = n$. The limit $\lim_{n\to\infty}$ tells us the summation has infinitely many terms. This kind of infinite sum expression are called a *series*, and provides a powerful way to compute quantities by summing together a bunch of terms. We'll learn more about sequences and series in Section VI.

D. Applications

Many laws of nature are expressed in terms of derivatives and integrals. It is therefore essential that you learn the language of calculus if you want to understand physics, chemistry, biology, ecology, and other sciences. Calculus is also heavily used in engineering, business, economics, any many other subjects based on quantitative analysis. We also use calculus in probability, statistics, and machine learning.

In all this areas, there are quantities described by functions, and we use derivatives and integrals to do useful calculations. For example, optimization, solving differential equations, computing probabilities involving continuous random variables, etc.

The goal of this tutorial is to show you the basics of derivatives and integrals, so that you you can think more clearly about these types of problems. This is the power of math: we learn techniques to analyze functions in general, which means our technique apply to any domain.

E. Doing calculus

In the previous section I made a lot of promises about the usefulness of calculus, as motivation talk to motivate you to read the rest of this tutorial so that you'll be interested in learning all the complicated-looking topics concepts, symbols, etc. Befogging getting to this, it's worth describing more specifically what doing calculus looks like.

Symbolic calculations using pen and paper: The key ideas of calculus and were developed by Isaac Newton and Wilhelm Leibnitz in the 17th century using mostly pen and paper calculations. The pen-and-paper approach continues to be the best way to learn about limits, derivatives, and integrals even to this day.

I encourage you to keep a notebook or use printer paper to reproduce the calculations presented in this tutorial on your own. The goal is for you to get used to manipulating functions, variables, and get used to the new calculus notation.

Symbolic calculations using SymPy: We're no longer in the 17th century, so we don't *have* to use pen and paper for symbolic math calculations. Using a computer algebra system like SymPy allows us to do symbolic math calculations very similar to what we could do on paper. SymPy is a Python module for When using SymPy, we can define a symbols `x` that works like the math variable x. We can then write arbitrary math expressions that involve `x` and ask SymPy to `factor`, `expand`, or `simplify` them, which are the standard algebra operations we normally perform using pen and paper.

```
>>> import sympy as sp
>>> x = sp.symbols("x")
>>> expr = 4 - x**2
>>> expr
```

```
4 - x**2
>>> sp.factor(expr)
-(x - 2)*(x + 2)
>>> sp.expand( (x-4)*(x+2) )
x**2 - 2*x - 8
>>> sp.simplify(2*x + 3*x - sp.sin(x) + 42)
5*x - sin(x) + 42
```

You can also subsitue particular values for `x` into the expression and evaluate the expression to obtain an exact symbolic value or a numerical approximation as a floating point number.

```
>>> expr.subs({x:1})
3
```

We can also also ask SymPy to *solve* the equation $\hat{e}$xpr = 0, which means to find the values of x that satisfy the equation $4 - x^2 = 0$.

```
>>> sp.solve(expr, x)
[-2, 2]
```

We'll use SymPy code examples to illustrate some concepts.

Numerical computing using NumPy: Calculus also has an engineering lineage. From the first mechanical calculators to modern CPUs and GPUs, there has been many computational developments in industry too. An engineer doesn't care about exact analytical results like knowing that $\sqrt{2}$ (the length of the diagonal of a square with side length 1) is an irrational number (requires infinitely many digits after the decimal to describe exactly). For most engineering concerts, if we can represent $\sqrt{2}$ approximately as 1.4......15 then they're good. In fact probably 1.4143521 would be enough for most use cases.

What engineers give up in mathematical exactitude, they gain manyfold in the form of computational power. Defining the specific data format for representing numbers (float32, float64, etc.) allows computer engineers to build high-performance hardware for doing math calculations.

```
>>> import numpy as np
>>> np.array([1.0, 1.1, 1.2, 1.3, 1.4, 1.5])
array([1. , 1.1, 1.2, 1.3, 1.4, 1.5])
>>> np.linspace(1.0, 1.5, 6)
array([1. , 1.1, 1.2, 1.3, 1.4, 1.5])
```

Scientific computing using SciPy: The Python module SciPy is a toolbox of scientific computing helper functions that greatly simplify our life. For example, computing integral of the function $h(x) = 4 - x^2$ between $x = 0$ and $x = 2$ requires only two lines of code:

```
>>> def h(x):
        return 4 - x**2
>>> from scipy.integrate import quad
>>> quad(h, a=0, b=2)[0]
5.333333333333333
```

The above code example shows the complete level of WIN humankind has achieved over practical math calculations. Calculus ideas started with Archimedes, then levelled up by Newton and Leibniz, and formalized as analysis (pure math) and numerical analysis (applied math). In parallel, compute hardware has improved its raw performance exponentially for many years. This means today you can perform the integrals like the ones the ancients only dreamed of in less than a second.

II. Math prerequisites

Before we dig into the new calculus topics, let's do a quick review of some key concepts from high school math. These are the basic building blocks I assume you've seen before, or at least heard about them.

A. Notation for sets and intervals

Sets are collections of math objects. Many math ideas are expressed in the language of sets, so it's worth knowing the notation conventions for sets.

- { definition }: the curly brackets surround the definition of a set, and the expression inside the curly brackets describes what the set contains.
- $s \in S$: this statement is read "s is an element of S" or "s is in S"
- $\mathbb{N}$: the set of natural numbers $\mathbb{N} \stackrel{\text{def}}{=} \{0, 1, 2, 3, 4, 5, \ldots\}$
- $\mathbb{N}_+$: the set of positive natural numbers $\mathbb{N}_+ \stackrel{\text{def}}{=} \{1, 2, 3, 4, 5, \ldots\}$.
- $\mathbb{R}$: the set of real numbers.
- $\mathbb{R}_+$: the set of nonnegative real numbers.

We often use the *set-builder* notation $\{\cdot \mid \cdot\}$ to define sets. Inside the curly brackets, we first describe the general kind of mathematical objects we are talking about, followed by the symbol "|" (which stands for "such that"), followed by the conditions that identifies the elements of the set. For example, the nonnegative real numbers $\mathbb{R}_+$ are defined as "all real numbers x such that $x \geqslant 0$," which can be expressed more compactly as $\mathbb{R}_+ \stackrel{\text{def}}{=} \{x \in \mathbb{R} \mid x \geqslant 0\}$ using the set-builder notation.

The number line: The *number line* is a visual representation of the set of real numbers $\mathbb{R}$, as shown in Figure 1. The real numbers correspond to all the points on the number line, from $-\infty$ to ∞.

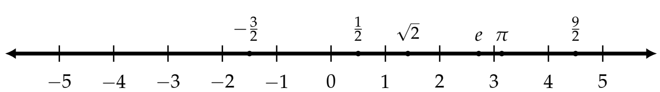


Figure 1. The real numbers $\mathbb{R}$ cover the entire number line.

The set of real numbers includes the natural numbers $\{0, 1, 2, 3, \ldots\}$, the integers $\{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$, the rational numbers like $\frac{5}{3}$, $\frac{22}{7}$, 1.5, as well as irrational numbers like $\sqrt{2}$, e, and π. This means any number you run into when solving a math problem can be visualized as a point on the number line.

The number line extends forever to the left and to the right. We use the notation $-\infty$ (negative infinity) to describe larger and larger negative numbers, and $+\infty$ to describe larger and larger positive numbers. Remember what we said in the introduction, ∞ is not a number but a process.

Number intervals: The number line can be used to represent subsets of the real numbers, which we call *intervals*. Figure 2 shows an illustration of the interval $[2, 4] \stackrel{\text{def}}{=} \{x \in \mathbb{R} \mid 2 \leqslant x \leqslant 4\}$, which is the subset of the real numbers between 2 and 4.

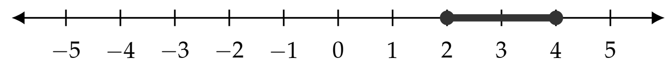


Figure 2. The interval $[2, 4] \stackrel{\text{def}}{=} \{x \in \mathbb{R} \mid 2 \leqslant x \leqslant 4\}$ is a subset of $\mathbb{R}$.

B. Functions

A *function* is a mathematical object that takes numbers as inputs and produces numbers as outputs. The output of the function f for the input x is denoted $f(x)$. For example, the function $f(x) \stackrel{\text{def}}{=} \frac{1}{2}x^2$ takes any number x as input, squares it and divides the result by two to produce the output. For example, $f(3) = \frac{1}{2}3^2 = \frac{9}{2} = 4.5$. Here is the Python code that defines the function `f` and evaluates it for the input $x = 3$.

```
>>> def f(x):
        return 0.5 * x**2
>>> f(3)
4.5
```

Not the Python syntax for evaluating the function `f` on the input `3` is the same as the math syntax $f(3)$.

Function graphs: The *graph* of a function is a line that passes through all input-output pairs of a function. Each input-output pair of the function f corresponds to the point $(x, f(x))$ in a coordinate system. We obtain the graph of the function by varying the input coordinate x and plotting all the points $(x, f(x))$, as illustrated in Figure **??**. The graph of the function f allows us to see at a glance the behaviour of the function for all possible inputs, and forms an essential visualization tool. Calculus calculations can be understood geometrically as operations based on the graph of the function.

We can use the Python modules `numpy` and `seaborn` to plot the graph of any function. For example, consider the function $f(x) \stackrel{\text{def}}{=} \frac{1}{2}x^2$ that we defined earlier. We start by importing the module `numpy` under the alias `np`, and evaluating the function for all inputs x in the interval $[-3, 3]$.

```
>>> import numpy as np
>>> xs = np.linspace(-3, 3, 1000)
>>> fxs = f(xs)
```

We used the function `np.linspace` to create an array (a list of numbers) `xs`, which contains 1000 input values that range from $x = -3$ until $x = 3$. Next we applied the function f to the array of inputs `xs` and stored the outputs of the function in the array `fxs`. At this point, the arrays `xs` and `fxs` contain 1000 input-output pairs of the form $(x, f(x))$, which is exactly what we need to plot the graph of the function.

```
>>> import seaborn as sns
>>> sns.lineplot(x=xs, y=fxs)
See Figure 3 for the output.
```

We imported the `seaborn` module under the alias `sns` then called the function `sns.lineplot` to produce the graph of $f(x)$ shown in Figure 3.

Inverse functions: The inverse function f^{-1} performs the *inverse operation* of the function f. If you start from some x, apply f, then apply f^{-1}, you'll arrive—full circle—back to the original input x:

$$f^{-1}(\ f(x)\ ) = x.$$

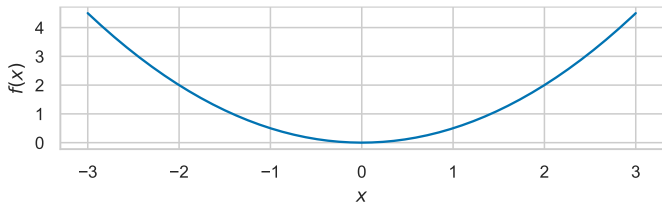


Figure 3. Graph of the function $f(x) = \frac{1}{2}x^2$ from $x = -3$ until $x = +3$. The graph of the function f consists of all the coordinate pairs $(x, f(x))$ over some interval of x values.

In Figure 4, the function f is represented as a forward arrow, and the inverse function f^{-1} is represented as a backward arrow that puts the value $f(x)$ back to the x it came from.

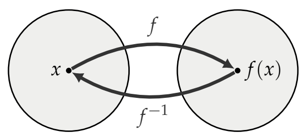


Figure 4. The inverse f^{-1} undoes the operation of the function f.

For example, when $x \geqslant 0$, the inverse of the function $f(x) = \frac{1}{2}x^2$ is the function $f^{-1}(x) = \sqrt{2x}$. Earlier we computed $f(3) = 4.5$. If we apply the inverse function to 4.5, we get $f^{-1}(4.5) = \sqrt{2 \cdot 4.5} = \sqrt{9} = 3$.

```
>>> from math import exp, log
>>> log(exp(5))
5.0
```

Function properties: We often think about the possible inputs and outputs of functions. We use the notation $f\colon A \to B$ to denote a function from the input set A to the output set B. The set of allowed inputs is called the *domain* of the function, while the set of possible outputs is called the *image* or *range* of the function. For example, the domain of the function $f(x) = \frac{1}{2}x^2$ is $\mathbb{R}$ (any real number) and it's image is $\mathbb{R}_+$ (nonnegative real numbers), so we write it as $f\colon \mathbb{R} \to \mathbb{R}_+$.

C. Function inventory

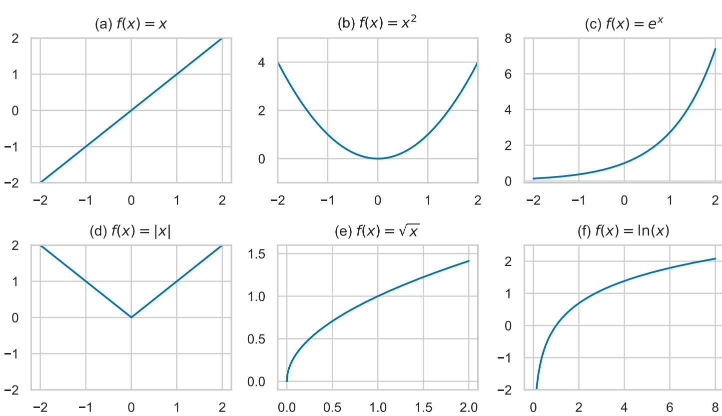


Figure 5. Graph of six math functions you should know about.

D. Functions with discrete inputs

Later in this tutorial, we'll study functions with discrete inputs, $a_k : \mathbb{N} \to \mathbb{R}$, which are called sequences. We often express sequences by writing explicitly the first possible value $[a_0, a_1, a_2, a_3, \ldots]$, which correspond to evaluating a_k for $k = 0$, $k = 1$, $k = 2$, $k = 3$, etc.

E. Geometry

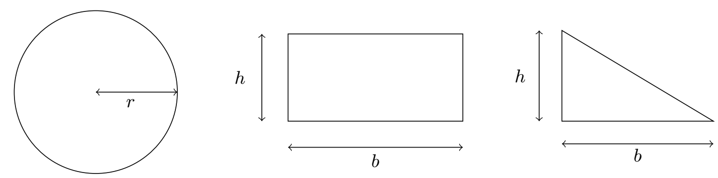
We'll now briefly review some geometry formulas.

Circle: The area enclosed by a circle of radius r is given by $A = \pi r^2$, where $\pi = 3.14159\ldots$. A circle of radius $r = 1$ has area π. The circumference of a circle of radius r is $C = 2\pi r$. A circle of radius $r = 1$ has circumference 2π.

Rectangle: The area of a rectangle of base b and height h is $A = bh$.

Triangle: The area of a triangle is equal to $\frac{1}{2}$ times the length of its base b times its height h $A = \frac{1}{2}bh$.

The three area formulas are illustrated in Figure 6.



(a) $A = \pi r^2$ (b) $A = bh$ (c) $A = \frac{1}{2}bh$

Figure 6. Area formulas for a circle, a rectangle, and a triangle.

F. Trigonometry

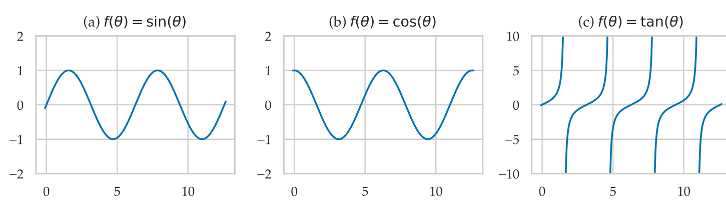


Figure 7. Graph of trigonometric functions $\sin\theta$, $\cos\theta$, and $\tan\theta$.

III. Limits

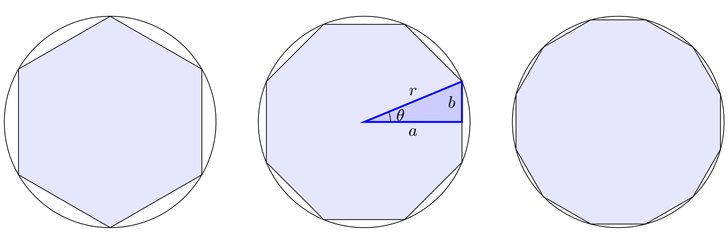
Limit expressions are a precise mathematical language for talking about infinitely large numbers, infinitely small lengths, and mathematical procedures with an infinite number of steps. The shorthand "lim" is common to all limit expressions, with the specifics of the limiting behaviour described below. Here are some examples:

- $\lim_{x\to\infty} f(x)$: limit expression that describes what happens to $f(x)$ when the input to the function x tends to infinity (gets larger and larger). In words, this limit expression is read as "limit of $f(x)$ as x goes to infinity."
- $\lim_{\delta\to 0} f(\delta)$: limit expression that describes the value $f(\delta)$ as the input δ tends to zero. The number δ (the Greek letter delta) usually describes a small distance, and the limit as delta goes to zero ($\delta \to 0$) describes the behaviour of the function $f(\delta)$ for infinitely short distance δ.
- $\lim_{n\to\infty} \text{proc}(n)$: limit expression that describes the value of $\text{proc}(n)$ as the integer n tends to infinity. The integer n usually describes the number of steps in a given procedure, and $\text{proc}(n)$ describes the output of this procedure when n steps are used.

Let's look at an example of an example of a math procedure with infinite number of steps that was invented by Archimedes, one of the OGs of calculus.

A. Example: area of a circle

Suppose we want to prove that area of a circle of radius r is described by the formula $A = \pi r^2$. We can approximate the circle as a regular polygon with n sides inscribed in the circle. Figure 8 shows the hexagonal (6-sides), octagonal (8-sides), and dodecagonal (12-sides) approximations to the circle.



(a) $n = 6$. (b) $n = 8$. (c) $n = 12$.

Figure 8. Approximations to the area of circle using a hexagon, an octagon, and a dodecagon inscribed inside a circle with radius r.

We can compute the area of the n-sided regular polygons by selling them up into triangular slices, and commuting the area of these slices using the formula for the area of a triangle $\frac{1}{2}bh$ and trigonometric functions sin and cos. Figure 8 (b) highlights one of the 16 identical triangular slices in the case when $n = 8$. The hypotenuse of this triangle has length r, the angle θ is $\frac{360\circ}{16} = \frac{2\pi}{16} = \frac{\pi}{8}$[rad], and its sides have length $a = r\cos(\frac{\pi}{8})$ and $b = r\sin(\frac{\pi}{8})$.

$$2n \times \tfrac{1}{2}ab = n \times r\cos(\tfrac{\pi}{n})\, r\sin(\tfrac{\pi}{n}) = n \times r^2 \cos(\tfrac{\pi}{n})\sin(\tfrac{\pi}{n}).$$

In the limit as $n \to \infty$, the n-sided-polygon approximation to the area of the circle becomes more and more accurate.

Here is the code that computes the approximations to the area of a circle of radius $r = 1$ with polygons with higher and higher number of sides.

```
>>> import math
>>> def calc_area(n, r=1):
        theta = 2 * math.pi / (2 * n)
        a = r * math.cos(theta)
        b = r * math.sin(theta)
        area = 2 * n * a * b / 2
        return area
>>> for n in [6, 8, 10, 50, 100, 1000, 10000]:
        area_n = calc_area(n)
        error = area_n - math.pi
        print(f"{n=}, {area_n=}, {error=}")
n=6, area_n=2.598076211353316, error=-0.5435
n=8, area_n=2.8284271247461903, error=-0.3132
n=10, area_n=2.938926261462365, error=-0.2027
n=50, area_n=3.133330839107606, error=-0.00826
n=100, area_n=3.1395259764656687, error=-0.002067
n=1000, area_n=3.141571982779476, error=-2.067e-05
n=10000, area_n=3.1415924468812855, error=-2.067e-07
```

As n goes to infinity we get πr^2, which is the formula for the area of a circle.

```
>>> n, r = sp.symbols("n r")
>>> A_n = n * r**2 * sp.cos(sp.pi/n) * sp.sin(sp.pi/n)
>>> sp.limit(A_n, n, sp.oo)
pi * r**2
```

This example shows practically why considering the limiting behaviour can lead us to computing quantities.

B. Limits at infinity

We're often interested in describing what happens to a certain function when its input variable tends to infinity. Does $f(x)$ approach a finite number, or does it keep on growing to ∞? For example, consider the limit of the function $f(x) = \frac{1}{x}$ as x goes to infinity:

$$\lim_{x\to\infty} f(x) = \lim_{x\to\infty} \tfrac{1}{x} = 0.$$

The function $\frac{1}{x}$ never *actually* reaches zero, so it would be wrong to write $f(x) = 0$. However, the the expression $\frac{1}{x}$ closer and closer to 0 as x goes to infinity. Limits are a useful concept because we can write $\lim_{x\to\infty} f(x) = 0$, even though $f(x) \neq 0$ for any number x.

The function $f(x)$ is said to *converge* to the number L if the function approaches the value L for large values of x:

$$\lim_{x\to\infty} f(x) = L.$$

We say "The limit of $f(x)$ as x goes to infinity is the number L." See Figure 9 for an illustration.

Example: Calculate $\lim_{x\to\infty} \frac{2x+1}{x}$. You are given the function $f(x) = \frac{2x+1}{x}$ and must determine what the function looks like for very large values of x.

$$\lim_{x\to\infty} \frac{2x+1}{x} = \lim_{x\to\infty} \left(2 + \frac{1}{x}\right) = 2 + \lim_{x\to\infty} \left(\frac{1}{x}\right) = 2 + 0 = 2.$$

As the denominator x becomes larger and larger, the fraction $\frac{1}{x}$ becomes smaller and smaller, so $\lim_{x\to\infty} \frac{1}{x} = 0$.

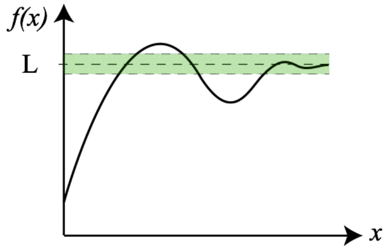


Figure 9. A function $f(x)$ that oscillates up and down initially, but after a while "settles down" close to the value L.

C. Limits to a number

The limit of $f(x)$ approaching $x = a$ *from the right* is defined as

$$\lim_{x\to a^+} f(x) = \lim_{\delta\to 0} f(a+\delta).$$

To find the limit from the right at a, we let x take on values like $a + 0.1$, $a + 0.01$, $a + 0.001$, $a + 0.0001$, etc.

The limit of $f(x)$ when x approaches *from the left* is defined analogously,

$$\lim_{x\to a^-} f(x) = \lim_{\delta\to 0} f(a-\delta).$$

If both limits from the left and from the right of some number exist and are equal to each other, we can talk about the limit as $x \to a$ without specifying the direction of approach:

$$\lim_{x\to a} f(x) = \lim_{x\to a^+} f(x) = \lim_{x\to a^-} f(x).$$

For the two-sided limit of a function to exist at a point, both the limit from the left and the limit from the right must converge to the same number. If the function $f(x)$ obeys, $f(a) = L$ and $\lim\limits_{x\to a} f(x) = L$, we say the function $f(x)$ is continuous at $x = a$.

D. Continuity

A function is said to be *continuous* if its graph looks like a smooth curve that doesn't make any sudden jumps and contains no gaps. If you can draw the graph of the function on a piece of paper without lifting your pen, the function is continuous.

A more mathematically precise way to define continuity is to say the function is equal to its limit for all x. We say a function $f(x)$ is *continuous* at a if the limit of f as $x \to a$ converges to $f(a)$:

$$\lim_{x\to a} f(x) = f(a).$$

Remember, the two-sided limit $\lim_{x\to a}$ requires both the left and the right limit to exist and to be equal. Thus, the definition of continuity implies the equality $\lim_{x\to a^-} f(x) = f(a)$ and $f(a) = \lim_{x\to a^+} f(x)$, which correspond to the idea of "not lifting the pen" when drawing the graph at $x = a$.

E. Limit formulas

The calculation of the limit of the sum, difference, product, and quotient of two functions is computed as follows, respectively:

$$\begin{aligned}
\lim_{x\to a}(f(x)+g(x)) &= \lim_{x\to a} f(x) + \lim_{x\to a} g(x),\\
\lim_{x\to a}(f(x)-g(x)) &= \lim_{x\to a} f(x) - \lim_{x\to a} g(x),\\
\lim_{x\to a} f(x)g(x) &= \lim_{x\to a} f(x)\cdot \lim_{x\to a} g(x),\\
\lim_{x\to a}\frac{f(x)}{g(x)} &= \frac{\lim_{x\to a} f(x)}{\lim_{x\to a} g(x)}.
\end{aligned}$$

The above formulas indicate we are allowed to *take the limit inside* of the basic arithmetic operations.

Euler's number e is defined as the limit $e \stackrel{\text{def}}{=} \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n$, which is the compound interest calculation for an annual interest rate of 100% with compounding is performed infinitely often.

F. Computing limits using SymPy

In SymPy, we use the symbol `oo` (two lowercase `o`s) to denote ∞. Infinity is not a number but a process: the process of counting forever. Thus, $\infty + 1 = \infty$, ∞ is greater than any finite number, and $1/\infty = 0$.

```
>>> from sympy import oo
>>> oo+1
oo
>>> 5000 < oo
True
>>> 1/oo
0
```

The SymPy function `limit` allows us to compute limit expressions. For example, here is the code for computing the limit $\lim_{x\to\infty}\frac{1}{x}$:

```
>>> import sympy as sp
>>> x = sp.symbols("x")
>>> sp.limit(1/x, x, oo)
0
```

The first line imports the `sympy` module under the alias `sp`. The second line defines the symbol `x`, which we can use to write math expressions. We provide the expression `1/x` as the first first argument to the function `limit`, then specify the variable `x` and destination `oo` as the second and third arguments.

Here is another example, that computes the limit of the fraction $\frac{2x+1}{x}$ as x goes to infinity:

```
>>> sp.limit( (2*x+1)/x , x, oo)
2
```

Recall the definition of Euler's number $e \stackrel{\text{def}}{=} \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n$ that we showed in the introduction. We can check this definition by making SymPy compute the limit.

```
>>> n = sp.symbols("n")
>>> limit((1+1/n)**n, n, oo)
E
```

Note SymPy produced the exact value `E` $= 2.718281828\ldots$.

G. Applications of limits

Limits are important in calculus because they are used in the formal definitions of derivatives, integrals, and series.

Limits for derivatives: The formal definition of a function's derivative is expressed in terms of the rise-over-run formula for an infinitely short run:

$$f'(x) \;=\; \lim_{\text{run}\to 0} \frac{\text{rise}}{\text{run}} \;=\; \lim_{\delta \to 0} \frac{f(x+\delta) \;-\; f(x)}{x+\delta \;-\; x}.$$

We'll continue the discussion of this formula in Section IV.

Limit for integrals: One way to approximate the area under the curve $f(x)$ between $x = a$ and $x = b$ is to split the area into n little rectangles of width $\Delta x = \frac{b-a}{n}$ and height $f(x)$, and then calculate the sum of the areas of the rectangles:

$$A_f(a,b) \approx \underbrace{\Delta x f(a) + \Delta x f(a+\Delta x) + \Delta x f(b - \Delta x)}_{n \text{ terms}}.$$

We obtain the exact value of the area in the limit where we split the area into an infinite number of rectangles with infinitely small width:

$$\int_a^b f(x)\,dx = A_f(a,b) = \lim_{n\to\infty} \Delta x \left[f(a) + f(a+\Delta x) + \cdots + f(b-\Delta x) \right].$$

Computing the area under a function by splitting the area into infinitely many rectangles is an approach known as a *Riemann sum*, which we'll discuss in Section V.

Limits for series: We use limits to describe the convergence properties of series. For example, the partial sum of the first n terms of the geometric series $a_k = r^k$ corresponds to the following expression:

$$S_n = \sum_{k=0}^{n} r^k = 1 + r + r^2 + r^3 + \cdots + r^n.$$

The *series* $\sum a_k$ is defined as the limit $n \to \infty$ of the above expression. For values of r that obey $|r| < 1$, the series *converges* to the a finite value:

$$S_\infty = \lim_{k\to\infty} S_k = \sum_{k=0}^{\infty} r^k = 1 + r + r^2 + r^3 + \cdots = \frac{1}{1-r}.$$

We'll learn more about series in Section VI.

In the remainder of this tutorial we'll use limits to evaluate derivatives, integrals, and series expressions.

IV. Derivatives

The *derivative* function, denoted $f'(x)$, $\frac{d}{dx}f(x)$, or $\frac{df}{dx}$, describes the *rate of change* of the function $f(x)$. For example, the constant function $f(x) = c$ has derivative $f'(x) = 0$ since the function $f(x)$ does not change at all. The derivative function describes the *slope* of the graph of the function $f(x)$. The derivative of a line $f(x) = mx + b$ is $f'(x) = m$, since the slope of this line is equal to m for all values of x. In general, the slope of a function is different at different values of x, so mathematicians invented a new notation $f'(x)$ for describing "the slope of the function f at x."

The derivative function $f'(x)$ is defined as the rate of change of the function f at x, and it is computed using the formula:

$$f'(x) \stackrel{\text{def}}{=} \lim_{\delta \to 0} \frac{f(x+\delta) \;-\; f(x)}{\delta}.$$

In words, this formula describes the general rise-over-run calculation for computing the slope of a line that connects the points $(x, f(x))$ and $(x+\delta, f(x+\delta))$, with the step-length δ becoming infinitely small.

Geometrically, the derivative function computes the slope of the graph of the function $f(x)$ for all values of x. In general, the slope of a function is different for different values of x. Figure 10 shows the slope calculation for the function $f(x) = \frac{1}{2}x^2$ for two different values of x: $x = -0.5$ and $x = 1$.

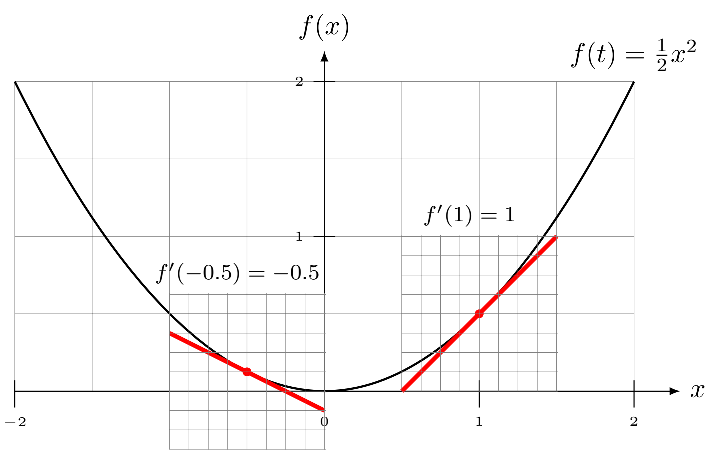


Figure 10. The derivative of the function at $x = a$ is denoted $f'(a)$ and describes the slope function at that point.

The derivative function $f'(x)$ is a property of the function $f(x)$. Indeed, this is where the name *derivative* comes from: $f'(x)$ is not an independent function—it is *derived* from the original function $f(x)$.

The *derivative operation*, denoted $\frac{d}{dx}$[<f>], takes as input a function $f(x)$ and produces as output the derivative function $f'(x)$. Applying the derivative operation to a function is also called "taking the derivative" of a function. For example, the derivative of the function $f(x) = \frac{1}{2}x^2$ is the function $f'(x) = x$. We can describe this relationship as $(\frac{1}{2}x^2)' = x$ or as $\frac{d}{dx}(\frac{1}{2}x^2) = x$. Look at Figure **??** and use the graph to prove to yourself that the slope of $f(x) = \frac{1}{2}x^2$ is described by $f'(x) = x$ everywhere on the graph.

A. Numerical derivative calculations

Here is a simple computer program for computing a numerical approximations to the derivative the function `f` at the point `x`:

```
>>> def differentiate(f, x, delta=1e-9):
        df = f(x+delta) - f(x)
        dx = delta
        return df / dx
```

The code performs the same calculation as in the definition of the derivative, but using a finite step `delta` $= 10^{-9}$ instead of the infinitely small step δ described by the limit calculation.

Consider the Python function `f` that corresponds to the math function $f = \frac{1}{2}x^2$. We can use `differentiate` to evaluate the derivative the function when $x = 1$:

```
>>> def f(x):
        return 0.5 * x**2
>>> differentiate(f, 1)
1.000000082740371
```

We obtain the approximation $f'(1) = 1.000000082740371$, which is not perfect but pretty close to the true value $f'(1) = 1$. For most practical applications, this approximation is good enough.

B. Derivative formulas

You don't need to apply the complicated derivative formula $f'(x) \stackrel{\text{def}}{=} \lim_{\delta\to 0} \frac{f(x+\delta)-f(x)}{\delta}$ every time you need to find the derivative of a function. For each function $f(x)$, it's enough to use the complicated formula once and record the formula you obtain for $f'(x)$, then you can reuse that formula whenever you need to compute $f'(x)$ in a later calculation.

Table I shows the derivative formulas for a number of commonly used functions. I invite you to mentally bookmark this page so you can come back to it when derivatives come up.

TABLE I
Derivative formulas for commonly used functions

$f(x)$	$-$ derivative $\to$	$f'(x)$
a	$-\frac{d}{dx}\to$	0
$\alpha f(x) + \beta g(x)$	$-\frac{d}{dx}\to$	$\alpha f'(x) + \beta g'(x)$
x	$-\frac{d}{dx}\to$	1
$mx + b$	$-\frac{d}{dx}\to$	m
x^n	$-\frac{d}{dx}\to$	nx^{n-1}
$\frac{1}{x} = x^{-1}$	$-\frac{d}{dx}\to$	$\frac{-1}{x^2} = -x^{-2}$
$\sqrt{x} = x^{\frac{1}{2}}$	$-\frac{d}{dx}\to$	$\frac{1}{2\sqrt{x}} = \frac{1}{2}x^{-\frac{1}{2}}$
e^x	$-\frac{d}{dx}\to$	e^x
$\ln(x)$	$-\frac{d}{dx}\to$	$\frac{1}{x}$
$\sin(x)$	$-\frac{d}{dx}\to$	$\cos(x)$
$\cos(x)$	$-\frac{d}{dx}\to$	$-\sin(x)$

C. Derivative rules

In addition to the table of derivative formulas show above, there are some important derivatives rules that allow you to find derivatives of *composite* functions.

Linearity: The derivative is a *linear* operation, which means:

$$\frac{d}{dx}\left[\alpha f(x) + \beta g(x)\right] = \alpha \frac{d}{dx} f(x) + \beta \frac{d}{dx} g(x).$$

In other words, the derivative of a linear combination of functions $\alpha f(x) + \beta g(x)$ is equal to the same linear combination of the derivatives $\alpha f'(x) + \beta g'(x)$.

Product rule: The derivative of a product of two functions is obtained as follows:

$$[f(x)g(x)]' = f'(x)g(x) + f(x)g'(x).$$

In each term, the derivative of one of the functions is multiplied by the other function.

Quotient rule: The *quotient rule* tells us how to obtain the derivative of a fraction of two functions:

$$\left[\frac{f(x)}{g(x)}\right]' = \frac{f'(x)g(x) - f(x)g'(x)}{g(x)^2}.$$

Chain rule: If you encounter a situation that includes an inner function and an outer function, like $f(g(x))$, you can obtain the derivative by a two-step process:

$$[f(g(x))]' = f'(g(x))g'(x).$$

In the first step, leave the inner function $g(x)$ alone, and focus on taking the derivative of the outer function $f(x)$. This step gives us $f'(g(x))$, the value of f' evaluated at $g(x)$. As the second step, we multiply the resulting expression by the derivative of the *inner* function $g'(x)$.

D. Higher derivatives

The second derivative of $f(x)$ is denoted $f''(x)$ or $\frac{d^2 f}{dx^2}$. It is obtained by applying the derivative operation to $f(x)$ *twice*: $\frac{d}{dx}\left[\frac{d}{dx}[\texttt{<f>}]\right]$. The second derivative $f''(x)$ encodes the information about the *curvature* of $f(x)$. Positive curvature means the function opens upward, and looks like the bottom of a valley. The function $f(x) = \frac{1}{2}x^2$ shown in Figure 10 has derivative $f'(x) = x$ and second derivative $f''(x) = 1$, which means it has positive curvature. Negative curvature means the function opens downward, and looks like a mountain peak. For example, the function $g(x) = -x^2$ has negative curvature.

E. Examples

Armed with these derivative formulas and rules, you can the derivative of any function, no matter how complicated. Let's look at some examples.

Example 1: To calculate the derivative of $f(x) = e^{x^2}$, we use the chain rule: $f'(x) = e^{x^2}[x^2]' = e^{x^2}2x$.

Example 2: To find the derivative of $f(x) = \sin(x)e^{x^2}$, we use the product rule and the chain rule: $f'(x) = \cos(x)e^{x^2} + \sin(x)2xe^{x^2}$.

Example 3: The derivative of $\sin(x^2)$ requires using the chain rule: $\left[\sin(x^2)\right]' = \cos(x^2)\left[x^2\right]' = \cos(x^2)2x$.

F. Computing derivatives analytically using SymPy

The `SymPy` function `diff` computes the derivative of any expression. For example, here is how we can compute the derivative of the function $f(x) = mx + b$:

```
>>> m, x, b = sp.symbols("m x b")
>>> sp.diff(m*x + b, x)
m
```

Let's also verify the derivative formula $\frac{d}{dx}[x^n] = nx^{n-1}$:

```
>>> x, n = sp.symbols("x n")
>>> sp.diff(x**n, x)
n * x**(n - 1)
```

The exponential function $f(x) = e^x$ is special because it is equal to its derivative:

```
>>> from sympy import exp
>>> sp.diff(exp(x), x)
exp(x)
```

Let's check the derivative calculations from the above examples:

```
>>> sp.diff(sp.exp(x**2), x)
2*x*exp(x**2)
>>> sp.diff(sp.sin(x)*sp.exp(x**2), x)
2*x*exp(x**2)*sin(x) + exp(x**2)*cos(x)
>>> sp.diff(sp.sin(x**2), x)
2*x*cos(x**2)
```

G. Applications of derivatives

We use derivatives to solve problems in physics, chemistry, computing, biology, business, and many other areas of science. The derivative operator comes up whenever we study the rate of change of a quantity. We use derivatives to obtain local liner approximations to functions (tangent lines).

Tangent lines: The *tangent line* to the function $f(x)$ at $x = x_0$ is the line that passes through the point $(x_0, f(x_0))$ and has the same slope as the function at that point. The tangent line to the function $f(x)$ at the point $x = x_0$ is described by the equation

$$T_1(x) = f(x_0) \;+\; f'(x_0)(x - x_0).$$

For example, the tangent line to $f(x) = \frac{1}{2}x^2$ at $x_0 = 1$ is $T_1(x) = f(1) + f'(1)(x-1) = \frac{1}{2} + 1 \cdot (x-1) = x - \frac{1}{2}$. Look back at Figure 10 for an illustration.

The tangent line T_1 is also called a *first order approximation* to the function f, since it has the same value and the same derivative as the function f, $T_1(1) = f(1)$ and $T_1'(1) = f'(1)$. In Section VI-G, we'll learn how to build order-n approximations $T_n(x)$.

H. Optimization

Derivatives are very useful for solving optimization problems, which consist of finding the maximum or minimum value of some function $f(x)$. For example, look the graph of the function $f(x) = \frac{1}{2}x^2$ in Figure 10. The minimum of the function occurs when $x = 0$ where the slope of the function is zero $f'(0) = 0$. Note also the second derivative of is positive at that point $f''(0) > 0$, which tells us the function locally looks like bottom of a bowl.

Analytical optimization: The values of x^* where the derivative is zero are called the *critical points* of the function. Optimum values (maximum or minimum) occurs at a critical point of the function. We identify a critical point x^* that corresponds to minimum if the second derivative is positive at that point $f''(x^*) > 0$ (positive curvature). In contrast, a critical point x^* here $f''(x^*) < 0$ (negative curvature) is a maximum. This observation suggests a general procedure for finding minima and maxima:

(1) Solve $f'(x) = 0$ to find the critical points x^*.
(2) For each critical point x^*, check to see if it is a maximum or a minimum by evaluating $f''(x^*)$:
   - If $f''(x^*) < 0$ then x^* is a max (mountain top)
   - If $f''(x^*) > 0$ then x^* is a min (bottom of a valley).

We can also perform the check in step (2) visually by looking at the graph of the function, or by evaluating the slope of the function near the critical point. If $f'(x^* - 0.1)$ is negative and $f'(x^* + 0.1)$ is positive, the point x^* is a minimum (as in Figure 10). If $f'(x^* - 0.1)$ is positive and $f'(x^* + 0.1)$ is negative, then the point x^* is a maximum. If $f'(x^* - 0.1)$ and $f'(x^* + 0.1)$ have the same sign, the value x^* is a *saddle point*, which is neither a minimum or a maximum.

Example: Let's use the two-step procedure to find the minimum and the maximum of the function $f(x) = x^3 - 2x^2 + x$. First we calculate its derivative $f'(x) = 3x^2 - 4x + 1 = 3(x-1)(x-\frac{1}{3})$. Next we find the critical points by solving the equation $f'(x) = 0$, which gives us two critical points $x_1^* = \frac{1}{3}$ and $x_2^* = 1$. The second derivative of the function is $f''(x) = 6x - 4$. For the critical value $x_1^* = \frac{1}{3}$, we find $f''(\frac{1}{3}) = -2 < 0$, which tells us x_1^* is a maximum. For $x_2^* = 1$, we find $f''(1) = 2$, which tells us x_1^* is a minimum.

Numerical optimization: Consider the shape of the function near a minimum value. The function is decreasing just before it reaches its minimum, and the function increases just after its minimum. This means we can start at any point $x = x_0$ and take "downhill" steps following the descending direction of the function, we'll end up at the minimum value. This simple procedure that repeatedly takes steps in the direction where the function is decreasing turns out to be a very powerful tool that can find the minimum of any function. This procedure is called the *gradient descent algorithm*, where the name *gradient* refers to the derivative operation for multivariable functions.

```
>>> def gradient_descent(f,x0=0,alpha=0.05,tol=1e-10):
        current_x = x0
        change = 1
        while change > tol:
            df_at_x = differentiate(f, current_x)
            next_x = current_x - alpha * df_at_x
            change = abs(next_x - current_x)
            current_x = next_x
        return current_x
```

The `gradient_descent` procedure takes two arguments as inputs: the function to minimize, and a initial value x_0 where to start the minimization process.

We can use use this procedure to find the minimum of any function. For example, let's see what happens when we call `gradient_descent` on the function $q(x) = (x-5)^2$ using the initial initial value `x0=10`.

```
>>> def q(x):
        return (x - 5)**2
>>> gradient_descent(q, x0=0)
5.000000000396651
```

The procedure repeats the "move downhill" operation several times, until it reaches the point $x^* = 5.00$, which is the minimum of the function $q(x)$.

Numerical optimization using SciPy: The module `scipy.optimize` provides a high-performance numerical optimization procedure called `minimize`. Here is a quick example that shows how to use this procedure on the $q(x) = (x-5)^2$.

```
>>> from scipy.optimize import minimize
>>> res = minimize(q, x0=0)
>>> res["x"][0]   # = argmin q(x)
4.9999999737
```

The answer is roughly the same as the procedure we defined, but the optimization was done much faster.

V. Integrals

Integration is the process of computing the "total" of some function $f(x)$ accumulated over a range of its input values. The symbol $\int$ we use to denote integrals is an elongated letter S, which is short for *summa*. This should give you a hint that integrals perform some kind of summation. There are actually two different integral operations:

- The *integral* of $f(x)$ from $x = a$ to $x = b$ is denoted $\int_{x=a}^{x=b} f(x)\,dx$ and corresponds to the area under the graph of $f(x)$ between a and b, which we also denote $A_f(a, b)$.
- The *integral function* $F_0(b) \overset{\text{def}}{=} A_f(0, b) = \int_{x=0}^{x=b} f(x)\,dx$ corresponds to the area-under-the-graph-of-$f(x)$ calculation as a function of the upper limit of integration b.

The integral $\int_{x=a}^{x=b} f(x)\,dx$ when both a and b are fixed is a number $A_f(a, b) \in \mathbb{R}$. In contrast, the integral $\int_{x=0}^{x=b} f(x)\,dx$ with a variable b is a function $F_0 : \mathbb{R} \to \mathbb{R}$. The integral function $F_0(b) \overset{\text{def}}{=} \int_{x=0}^{x=b} f(x)\,dx$ computes the area under the graph of $f(x)$ as a function of the upper limit of integration b. Both integral operations are important, and we'll discuss each of them in turn.

A. *Act 1: Integrals as area calculations*

Figure 11 shows a shaded region enclosed between the graph of $f(x)$ from above, the x-axis from below, and vertical lines at $x = a$ and $x = b$. The calculation of the *area* of this region is described by the following integral calculation:

$$A_f(a, b) = \int_{x=a}^{x=b} f(x)\,dx.$$

The numbers a and b and called the *limits of integration*. We refer to this type of integral as a *definite integral* since both limits of integration are defined.

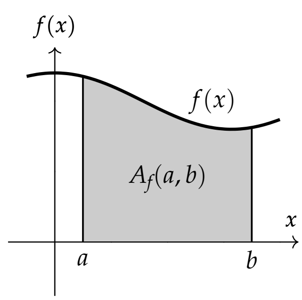


Figure 11. The integral of the function $f(x)$ between $x = a$ and $x = b$ corresponds to the area of the shaded region $A_f(a, b) = \int_a^b f(x)\,dx$.

We often use the simplified notation $\int_a^b f(x)\,dx$ as shorthand for $\int_{x=a}^{x=b} f(x)\,dx$ and read this expression as "the integral of $f(x)$ between a and b." If this is the first time you're seeing the notation for integrals, it might seem very intimidating and complicated to you, but don't freak out and bear with me for two more pages. You'll see this fancy-looking math notation is nothing to worry about! It's just the calculus way to denote a particular calculation that involves the function $f(x)$. You can think of $\int_a^b$ `<f>` dx as a "template" that you fill in by replacing `<f>` with the function $f(x)$ you're interested in whenever you need to compute the area $A_f(a, b)$.

B. *Properties of integrals*

We'll now state some properties of integrals that follow from their interpretation as area calculations.

- The sum of the integral from a to b and the integral from b to c is equal to the integral starting from a going all the way to c: $\int_a^b f(x)\,dx + \int_b^c f(x)\,dx = \int_a^c f(x)\,dx$.
- The integral of k times the function $f(x)$ is equal to k times the integral of $f(x)$: $\int_a^b kf(x)\,dx = k\int_a^b f(x)\,dx$.
- The integral of the sum of two functions is the sum of their integrals: $\int_a^b [f(x) + g(x)]\,dx = \int_a^b f(x)\,dx + \int_a^b g(x)\,dx$.
- Integrals over intervals with zero length have zero value: $\int_a^a f(x)\,dx = 0$. Geometrically, this integral defines a region with height $f(x)$ and width 0, so it has zero area.

Let's look at some examples.

Example 1. Integral of a constant function: Consider the constant function $f(x) = 3$. We can easily find the area under the graph of this function between any two points, since the region under the graph has a rectangular shape. The area under $f(x)$ between $x = 0$ and $x = 5$ is described by the following integral calculation:

$$A_f(0, 5) = \int_0^5 f(x)\,dx = 3 \cdot 5 = 15.$$

The area under the graph of $f(x)$ is a rectangle with height 3 and width 5, so its area is $3 \cdot 5 = 15$, as shown in Figure 12.

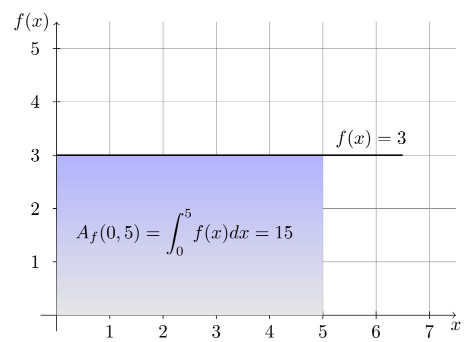


Figure 12. The area of a rectangle of height 3 and width 5 equals 15.

Example 2. Integral of a linear function: Consider now the area under the graph of the line $g(x) = x$ between $x = 0$ and $x = 5$, as shown in Figure 13. This area is described by the following integral calculation:

$$A_g(0, 5) = \int_0^5 g(x)\,dx = \tfrac{1}{2}5 \cdot 5 = \tfrac{1}{2}5^2 = \tfrac{25}{2} = 12.5.$$

The region under the graph of $g(x)$ has a triangular shape, so we can compute its area using the formula for the area of a triangle: base times height divided by 2.

I hope these two examples are starting to convince you that the scary-looking integral notation is not that complicated after all. It's just a fancy way to describe the "area under the graph of the function" calculation.

Example 3. Integral of a polynomial: Consider now the function $h(x) = 4 - x^2$. We want to know the area under the graph of

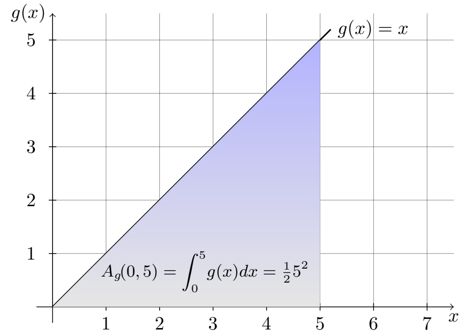


Figure 13. The area of a triangle with base 5 and height 5 is $\frac{1}{2}5^2 = \frac{25}{2} = 12.5$.

$h(x)$ between $x = 0$ and $x = 2$, as illustrated in Figure 14. We need to calculate the following integral:

$$A_h(0,2) = \int_0^2 h(x)\,dx \;= ???.$$

The area under the graph of $h(x)$ is a curved region and not a simple recognizable geometric shape with a known area formula. How could we compute the area in this case?

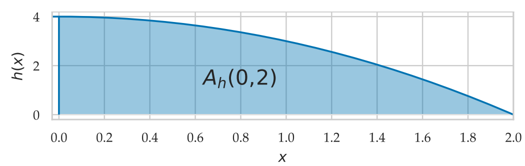


Figure 14. The area under the graph of $h(x)$ between $x = 0$ and $x = 2$.

One way to approximate the area under $h(x)$ is to compute the *Riemann sum* approximation, which splits the region into bunch of vertical rectangular strips of some fixed width, which we'll denote Δx. The height of each rectangular strip will vary depending on $h(x)$. Look ahead to figures 15 and 16 to see where we're going with this. Splitting up the area $A_h(0,2)$ into $n = 10$ strips, calculating the area of the individual strips, and summing them together produces the approximation $A_h(0,2) \approx 4.92$. If we split the area $A_h(0,2)$ into $n = 20$ strips, we obtain the more accurate approximation $A_h(0,2) \approx 5.13$. The approximation with $n = 1000$ rectangular strips gives us $A_h(0,2) \approx 5.329332$, and with $n = 1\,000\,000$ rectangles is $A_h(0,2) \approx 5.333329333332$. The more finely we chop up the region into rectangles, the closer we get to the *exact* value, which is $\int_0^2 h(x)\,dx = 5\frac{1}{3} = 5.\overline{3} = 5.3333333333333333\ldots$.

In the next section, we'll learn more about split-into-rectangles are area calculations (a.k.a. *integration*). Don't worry, I won't make you calculate sums with $n = 10$ or $n = 20$ terms by hand, let alone the sum with $n = 1\,000\,000$ terms! Instead, we'll write a computer program that performs the integration procedure for us. Modern computer are really good at this stuff. Indeed early computers were often called "numerical integrator" since they were built primarily to evaluate integrals.

C. Computing integrals numerically

Computing the integral $\int_a^b f(x)dx$ *numerically* means using a computer to compute the Riemann sum approximation to $A_f(a,b)$ by splitting the region of integration into many (think thousands or millions) of strips, computing the areas of each strip, then adding up the areas to obtain the total area under the graph of $f(x)$. The key step is to come up with a general mathematical expression that describes the approximate area calculation with n rectangular strips, then evaluate this expression for very large values of n.

Let's start by looking at the math required to calculate the approximation to $\int_0^2 h(x)dx$ using $n = 10$ rectangles, which is illustrated in Figure 15 (a). The width of each rectangle is $\Delta x = \frac{b-a}{n} = \frac{2-0}{10} = 0.2$. The x-coordinates of the right endpoints of the 10 rectangles are $[0.2, 0.4, 0.6, \ldots, 1.8, 2.0]$. To find the area of the rectangles, we need to know the height of the function h at these x-coordinates: $[h(0.2), h(0.4), h(0.6), \ldots, h(1.8), h(2.0)]$. The area of each rectangle is given by the height-times-width formula, and we sum together all of them to compute the total area:

$$A_h(0.2) \approx h(0.2)\cdot 0.2 + h(0.4)\cdot 0.2 + \cdots + h(2.0)\cdot 0.2 = 4.92.$$

Looking at figure Figure 15 (a), we can clearly see that the approximation computed in this way is an underestimate to the true area under the curve. Let's trust the process, and continue with the calculations, knowing that the "quality" of the approximations will improve when we split up the region into finer and finer strips.

The procedure we used for $n = 10$ works more generally for any n. In the general case, the rectangles will have width $\Delta x = \frac{b-a}{n} = \frac{2}{n}$, which will get smaller and smaller as n grows. The x-coordinates of the right endpoints of the n rectangles are $[\Delta x, 2\Delta x, 3\Delta x, \ldots, (n-1)\Delta x, n\Delta x]$. The height of the rectangles will be $[h(\Delta x), h(2\Delta x), h(3\Delta x), \ldots, h((n-1)\Delta x), h(n\Delta x)]$. To find the area under the graph of $h(x)$, we can sum together the individual height-times-width contributions of the n rectangular strips:

$$A_h(0,2) \approx h(\Delta x)\Delta x + h(2\Delta x)\Delta x + h(3\Delta x)\Delta x + \cdots + h(n\Delta x)\Delta x.$$

Observe that all the terms in this summation follow the same pattern: the k^{th} term in this summation is $h(k\Delta x)\,\Delta x$, as k varies from 1 to n. When working with long summations as in the above expressions, mathematicians use the symbol $\sum$ (the capital Greek letter *sigma*), which stands for sum. The approximation to the area under $h(x)$ between $x = a$ and $x = b$ using n rectangular strips corresponds to the following sum: $A_h(0,2) \approx \sum_{k=1}^{k=n} h(k\Delta x)\,\Delta x$. The labels above and below the summation symbol $\sum$ play the same role as the superscript and subscript in integral notation. The label $k = 1$ tells us where to start the summation, and label $k = n$ tells us where to stop the summation.

We can take what we learned from the particular example above to write a general formula for approximating the area under the graph of any function $f(x)$ between $x = a$ and $x = b$ using n rectangular strips:

$$A_f(a,b) \approx \sum_{k=1}^{k=n} f(a + k\Delta x)\,\Delta x, \quad \text{where} \quad \Delta x = \tfrac{b-a}{n}.$$

This is the famous *Riemann sum* formula for computing areas.

We'll now convert this math formula into a Python procedure that performs the n-rectangle area approximation calculation.

```
>>> def integrate(f, a, b, n):
        dx = (b - a) / n
        xs = [a + k*dx for k in range(1,n+1)]
        fxs = [f(x) for x in xs]
        area = sum([fx*dx for fx in fxs])
        return area
```

The code implements the same operations as described by the summation $A_f(a,b) \approx \sum_{k=1}^{k=n} f(a+k\Delta x)\,\Delta x$. We first compute the width of the rectangles `dx` $= \Delta x = \frac{b-a}{n}$. We then create the list `xs` that contains the x-coordinates of the right endpoints of the rectangles, `xs` $= [a+\Delta x, a+2k\Delta x, a+3k\Delta x, \ldots, n\Delta x]$, and evaluate the function `f` at these x-values to obtain `fxs` $= [f(a+\Delta x), f(a+2k\Delta x), f(a+3k\Delta x), \ldots, f(n\Delta x)]$. We calculate the areas of the rectangles by multiplying the heights `fxs` by the width `dx`, and summing everything together to obtain the total `area`, which we "return" as the output of the procedure.

Example 3 continued: Let's use the `integrate` procedure to compute the integral of the function $h(x) = 4 - x^2$. Recall we previously defined the Python function `h` that implements the same operation as the math function h:

```
>>> def h(x):
        return 4 - x**2
```

To calculate the $n = 10$ approximation for area under the graph of $h(x)$ between $x = 0$ and $x = 2$, we call the `integrate` procedure with the desired arguments.

```
integrate(h, a=0, b=2, n=10)
4.92
```

Then we can compute the approximation with $n = 20$ rectangles just as easily:

```
>>> integrate(h, a=0, b=2, n=20)
5.13
```

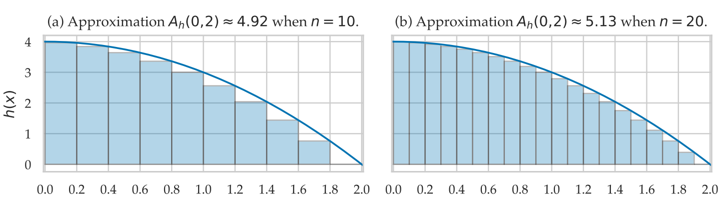


Figure 15. Approximations to the area under the graph of the function $h(x) = 4 - x^2$ computed using $n = 10$ and $n = 20$ rectangles.

Let's keep going to see what happens with $n = 50$ and $n = 100$:

```
>>> integrate(h, a=0, b=2, n=50)
5.2528
>>> integrate(h, a=0, b=2, n=100)
5.2932
```

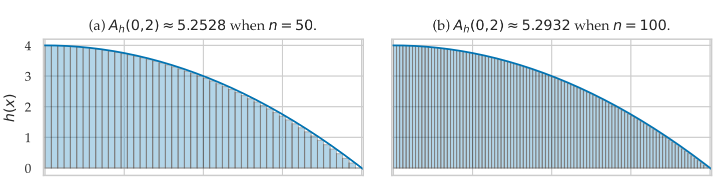


Figure 16. Approximations to the area under the graph of $h(x) = 4 - x^2$ computed using $n = 50$ and $n = 100$ rectangles.

The approximations get better and better as we increase the number of rectangles n.

```
>>> integrate(h, a=0, b=2, n=1000)
5.329332
>>> integrate(h, a=0, b=2, n=10000)
5.33293332
>>> integrate(h, a=0, b=2, n=1_000_000)
5.333329333332
```

The approximation computed with $n = 1M$ rectangles is accurate to 4 decimals. The exact value of the area $A_h(0,2)$ is $\frac{16}{3} = 5\frac{1}{3} = 3.\overline{3} = 5.3333333333\ldots$. To obtain the exact value, we have to **split up the region into infinitely many rectangular strips**, as we'll learn next.

D. Formal definition of the integral

In the limit as the number of rectangles n approaches ∞, the approximation to the area under the curve becomes *arbitrarily close* to the true area.

The integral between $x = a$ and $x = b$ is *defined* as the limit as n goes to infinity:

$$\int_a^b f(x)\,dx \stackrel{\text{def}}{=} \lim_{n\to\infty} \sum_{k=1}^{n} f(a+k\Delta x)\Delta x.$$

In words, the integral is defined as the limit of a Riemann sum that consists of infinitely thin rectangular strips. We previously defined the $\int_a^b f(x)\,dx$ geometrically as the area under the graph of $f(x)$, but now you know the formal math definition for integral that mathematicians use.

Note the structural similarity between the summation formula on the right and the integral notation on the left: in both cases we evaluate f at different values x values, multiply by a width, and add all these contributions together to get the total. Perhaps now the weird notation we use for integrals will start to make more sense to you. In the limit as $n \to \infty$, the summation sign $\sum$ becomes an integral sign $\int$, and the step size Δx becomes an infinitely small step dx.

The integral $\int_a^b f(x)\,dx$ is defined as a *procedure* with infinitely many steps ($\lim_{n\to\infty}$) that we perform on the function f. Recall that the formal definition of the derivative is also a procedure, specifically $f'(x) \stackrel{\text{def}}{=} \lim_{\delta\to 0} \frac{f(x+\delta)\, -\, f(x)}{\delta}$, which corresponds to rise-over-run calculation for computing the slope of f at x, with an infinitely short step-length δ. These two procedures are the foundations of calculus. The limits $\lim_{n\to\infty}$ and $\lim_{\delta\to 0}$ allow us to perform math operations on functions.

E. Act 2: Integrals as functions

The *integral function* $F_0(b)$ corresponds to the area calculation with a variable upper limit of integration $A_f(0,b)$:

$$F_0(b) \stackrel{\text{def}}{=} A_f(0,b) = \int_{x=0}^{x=b} f(x)\,dx\,.$$

As a matter of convention, we denote the integral function using the capital letter of the same letter as the original function.

Choosing $x = 0$ for the starting point of the integral function is an arbitrary choice. We can obtain another integral function if we use $x = a$ as the starting point, $F_a(b) = \int_a^b f(x)\,dx$. The integral functions F_a and F_0 differ only by a constant term: $F_0(b) = F_a(b) + C$, where $C = \int_{x=0}^{x=a} f(x)\,dx$.

The integral function $F_0(b)$ contains the "precomputed" information about the area under the graph of $f(x)$. Knowing F_0 allows us to compute the area using simple algebra calculations: the area under $f(x)$ between $x = a$ and $x = b$ is equal to the *change* in the integral function:

$$A_f(a,b) = \int_a^b f(x)\,dx = F(b) - F(a).$$

Intuitively, this formula computes the area $A_f(a,b)$ as the difference between the areas of two regions: the area until $x = b$ minus the area until $x = a$, as illustrated in Figure 17.

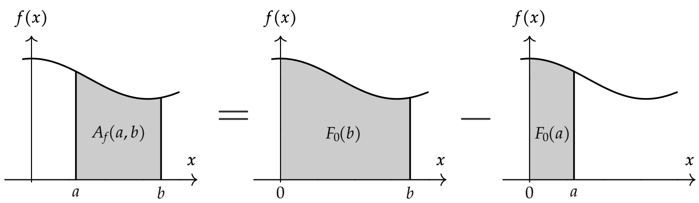


Figure 17. The area under $f(x)$ between $x = a$ and $x = b$ is computed using the formula $A_f(a,b) = F_0(b) - F_0(a)$, which describes the change in the output of $F_0(x)$ between $x = a$ and $x = b$.

Example 1 revisited: We can easily find the integral function for the constant function $f(x) = 3$, because the region under the curve is rectangular. Choosing $x = 0$ as the starting point, we obtain the integral function $F_0(b)$ that corresponds to the area under $f(x)$ between $x = 0$ and $x = b$ as follows:

$$F_0(b) = A_f(0,b) = \int_0^b f(x)\,dx = 3b.$$

The region is a rectangle of height 3 and with width b, as illustrated in Figure 18.

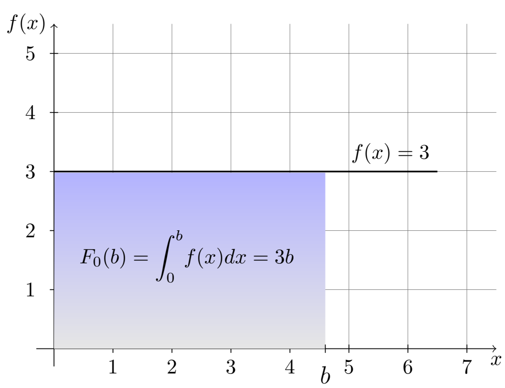


Figure 18. The integral function of the function $f(x) = 3$ is $F_0(b) = 3b$.

Knowing the function $F_0(b)$ allows us to compute the area under the graph of $f(x)$ between $x = 0$ and $x = 5$ as the difference $A_f(0,5) = F_0(5) - F_0(0) = 3 \cdot 5 - 3 \cdot 0 = 15$.

Example 2 revisited: Consider now the area under the graph of the line $g(x) = x$, starting from $x = 0$. Since the region is triangular, we can compute its area using the formula for the area of a triangle: base times height divided by two. The integral function of $g(x)$ is:

$$G_0(b) = A_g(0,b) = \int_0^b g(x)\,dx = \tfrac{1}{2}(b \cdot b) = \tfrac{1}{2}b^2.$$

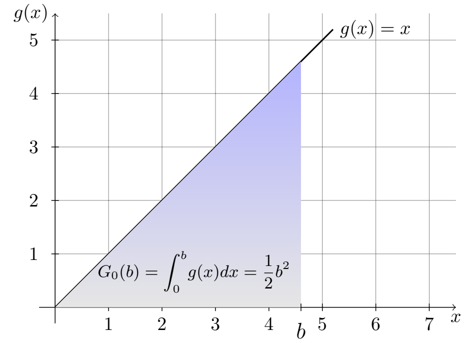


Figure 19. The integral function of the function $g(x) = x$ is $G_0(b) = \frac{1}{2}b^2$.

Knowing the function $G_0(b)$ allows us to compute the area under the graph of $g(x)$ between $x = 0$ and $x = 5$ as the difference $A_g(0,5) = G_0(5) - G_0(0) = \frac{1}{2}5^2 - \frac{1}{2}0^2 = 12.5$.

Example 3 revisited: The the area under $h(x) = 4 - x^2$ from $x = 0$ until $x = b$ is described by the following integral calculation:

$$H_0(b) = A_h(0,b) = \int_0^b h(x)\,dx = ???.$$

We were able to compute the integral functions $\int f(x)dx$ and $\int g(x)dx$ thanks to the simple geometries of the areas under the graphs, but $h(x)$ is a curved region so it requires some new integration methods. In the next few pages, we'll learn about symbolic integration techniques that we can use to find the integral function $H_0(b)$.

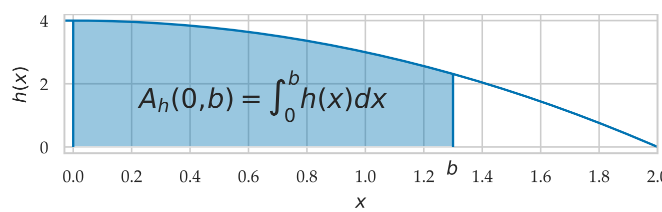


Figure 20. The integral of the function $h(x) = 4 - x^2$ from $x = 0$ to $x = b$.

F. Intermission

The integral in Example 3 was carefully chosen to motivate the need to learn more advanced methods for integration. How can we compute the integral function $H_0(b)$? Is there some formula that describes the area $A_h(0,b)$ of the region shown in Figure 20. We previously used numerical methods to compute the particular area $A_h(0,2)$ when $b = 2$, but now we're looking for a general math formula that computes the value $H_0(b) = A_h(0,b)$ for any b. In the next few pages we'll use *symbolic* calculations to fill in the ??? with an exact formula obtained by manipulating math equations. The secret weapon that will allow us to compute integrals using symbolic math is called the *fundamental theorem of calculus*, which we'll discuss next. The section is the "Intermission" in the calculus show. I want you, dear reader, to have a break before next pages where the level of "mathyness" will increase significantly As in a real-world intermission, this is also your chance to skip the rest of the "Integration show." Perhaps you have better things to do right now than being subjected to Ivan's teaching of advanced concepts you "might" want to know through model means of computation like Python and old-school math techniques like pen and paper. I won't get offended if you skip ahead—no worries! As a teacher, i'm happy that you now know that integrals are "area under the graph of" geometrically and can be approximated numerically using Riemann sums. This are the key takeaways so I feel I've done my job! I think you should stay though—you might enjoy the knowledge buzz moments that are coming your way in the next few pages. My college calculus professor in college described the realization you get after understanding the FTC as similar to the feeling you you get smoke some of that funny stuff. With your consent we continue with the integrals, else skip ahead to Section V-I for some tech talk about high performance numerical integrals, or straight away to Section VI to learn about sequences and series, which is the third foundational topics in calculus.

G. Act 3: Fundamental theorem of calculus

Note the pattern in the formulas for the integral functions $F_0(b)$, $G_0(b)$, and $H_0(b)$. The integral function of the constant function $f(x) = 3$ was a linear function $F_0(b) = 3b$. The integral of the linear function $g(x) = x$ is a quadratic function $G_0(x) = \frac{1}{2}b^2$. In each case, the integral function seems to increase the degree of the function. What is up with that? Is this a coincidence, or some fundamental math pattern we could use to "guess" the integral function of any polynomial?

The fundamental theorem of calculus (FTC) describes the inverse relation that exists between the integration operation $\int$`<f>`$\,dx$ and the differentiation operation $\frac{d}{dx}$[`<f>`]. A priori, there is no reason to suspect that integrals would be related to derivatives. The integral corresponds to the computation of an area, whereas the derivative operation computes the slope of a function. Yet behold:

Theorem (fundamental theorem of calculus): Let $f(x)$ be a continuous function, and let $a \in \mathbb{R}$ be a constant. Define the function $F_a(x)$ as follows:

$$F_a(x) \stackrel{\text{def}}{=} A_f(a, x) = \int_a^x f(u)\, du.$$

Then, the derivative of $F_a(x)$ with respect to x is equal to $f(x)$:

$$\frac{d}{dx}\big[F_a(x)\big] = f(x),$$

for any $x \in (a, b)$.

In words, the FTC tells us that ...

In order to understand the inverse relationship between integration and differentiation, we can draw an analogy with the inverse relationship between a function f and its inverse function f^{-1}, which *undoes* the effects of f. See Figure 4 on page 6. Given some initial value x, if we apply the function f to obtain the number $f(x)$, and apply the inverse function f^{-1} on the number $f(x)$, then the result will be the initial value x we started from:

$$f^{-1}(f(x)) = x.$$

Similarly, **the derivative operation is the inverse of the integral operation**. If you perform the integral operation $\int$`<f>`$\,dx$ followed by the derivative operation $\frac{d}{dx}$[`<f>`] on any function `<f>`, you'll get back to original function:

$$\frac{d}{dx}\int_c^x f(u)\, du = f(x).$$

We can verify this using SymPy by starting with some function `f` $= f(x) = x^2$, computing its integral function `F`, then using `sp.integrate` then take the derivative of `F`:

```
>>> f = x**2
>>> F = integrate(f, x)
>>> F
x**3/3          # + C
>>> diff(F, x)
x**2
```

For ordinary math functions, we know that if the function f^{-1} is the undo action for the function f, then f is also the undo action for f^{-1}: $f\big(f^{-1}(y)\big) = y$. Similarly, the inverse relationship between integrals and derivative holds in the other direction too. **The integral operation is the inverse operator of the derivative operation**. If we start with some function $G(x)$, calculate the derivative function $G'(x)$, then compute the integral of the derivative function $G'(x)$, we arrive back at the original function $G(x)$ (up to an additive constant):

$$G(x) = \int_c^x G'(u)\, du = G(x) + C.$$

Let's use SymPy to verify the fundamental theorem of calculus. Recall that the function `sp.diff` computes derivative functions and `sp.integrate` computes integral functions. To verify $\int_c^x G'(u)\ du = G(x) + C$, we start with some expression `G` for the function $G(x) = x^3$.

```
>>> G = x**3
>>> dGdx = diff(F, x)
>>> dGdx
3*x**2
>>> integrate(dGdx, x)
x**3    # + C
```

Define the anti-derivative function $F(x)$, which is a function whose derivative equals $f(x)$: $F'(x) = f(x)$.

Note the anti-derivative function is not unique; it is only defined up to an additive constant $F(x) + C$.

For example...

Computing integrals using anti-derivatives: The fundamental theorem of calculus gives us a way for computing integrals functions using "reverse engineering" thinking and the table of derivative formulas (see page **??**). To find an integral function of $f(x)$, we can look for a function $F(x)$ whose derivative is the the function $f(x)$. If we can find a function $F(x)$ such that $F'(x) = f(x)$, then we know the integral function of $f(x)$ is $F_a(x)$

$$\int_a^b f(x)\, dx = F_a(b) + C.$$

Example 3 continued: Suppose you're given a function $h(x) = 4 - x^2$ and asked to find its integral function $H_0(b) = \int_0^b h(x)\, dx$. This fundamental theorem of calculus tells us this problem is equivalent to finding a function $H(x)$ whose derivative is $h(x)$. The function $h(x) = 4 - x^2$ has two terms. The first term is a constant 4. We can guess that the corresponding term in the anti-derivative function $H(x)$ will be $4x$, since $\frac{d}{dx}[4x] = 4$. Now for the quadratic term $-x^2$. Remembering the derivative formulas for polynomials, we can guess that anti-derivative $H(x)$ must contain a x^3 term, because taking the derivative of a cubic term results in a quadratic term. Therefore, the anti-derivative function we're looking for has the form $H(x) = 3x - cx^3$, for some constant c. Pick the constant c that makes this equation true: $H'(x) = 4 - 3cx^2 = 4 - x^2$. Solving $3c = 1$, we find $c = \frac{1}{3}$ and so the anti-derivative function we're looking for is $H(x) = 4x - \frac{1}{3}x^3 + C$.

Using derivative formulas in reverse: This procedure based on using the derivative formulas in reverse to guess the value of $F(x)$ is very useful. We can use it for all the function listed in the table of derivative formulas. For example, the table tells us that the derivative of the linear function $f(x) = mx + b$ is the constant function $f'(x) = m$, which means the integral of a constant function is a linear function $\int m\,dx = mx + C$. The integral function of an exponential is also an exponential $\int e^x\,dx = e^x + C$, since $\frac{d}{dx}[e^x] = e^x$. The derivative of $\log_e(x)$ is $\frac{1}{x}$, therefore the integral of $\frac{1}{x}$ is $\log(x)$. Similarly for the trigonometric functions $\int \cos(x)\,dx = \sin(x)$ and $\int -\sin(x)\,dx = \cos(x)$. For economy of space, we'll verify all these integral formulas by computing the integral of the function $f(x) = m + e^x + \frac{1}{x} + \cos(x) - \sin(x)$ that contain the mix of several functions on the right side of the table of the derivative formulas table.

```
>>> fx = m + sp.exp(x) + 1/x + sp.cos(x) - sp.sin(x)
>>> sp.integrate(fx, x)
m*x + exp(x) + log(x) + sin(x) + cos(x)
```

SymPy tells us the integral function F_0 is $F_0(x) = mx + e^x + \log(x) + \sin(x) + \cos(x)$, which are all the corresponding terms on the left side of the table of derivative fromulas.

Okay, but what do we do if the function we want to integrate doesn't appear on the right side of the table?

H. Act 4: Techniques of integration

Okay we're getting into the fourth act of the calculus show, and I want you to remind you that you can "tap out" at any time. The material we'll show in the in this act is some of the most boring topics that CALC II students are forced to endure. If you're taking a CALC II class (a.k.a. CALC 102, Integral calculus), then you need to know this stuff because it is going to be your final exam. Everyone else, feel free to skip ahead to the next section.

There are a bunch of tricks that extend the reach of analytical integration methods (anti-differentiation) to more complicated functions. We don't have space to discuss all these tricks in this tutorial, so we'll focus on the two most important ones.

Substitution trick: Suppose the function we want to integrate has the structure $f(u(x))u'(x)$, which consists of inner function wrapped in an outer function multiplied by the derivative of the inner function. We can use the *substitution trick* to rewrite this integral in terms of the function $f(u)$ using u as the variable of integration:

$$\int_{x=a}^{x=b} f(u(x))\,u'(x)dx \;\;=\;\; \int_{u=u_a}^{u=u_b} f(u)\,du.$$

The substitution trick is also called a *change of variable* operation: we're replacing the variable x with the variable u, similar to a search-and-replace operation when editing a text file. Because we're doing the substitution "inside" an integral operation, we must also change the region of interval of integration (form $[a, b]$ to $[u_a, u_b]$) and change of the "step" parameter (from dx to du).

Follow these three steps to apply the substitution trick:
1) Replace dx with $\frac{1}{u'(x)}du$.
2) Replace all occurrences of $u(x)$ with u.
3) Replace the x-limits of integration $x = a$ and $x = b$ with u-limits of integration: $u_a = u(a)$ and $u_b = u(b)$.

Example: Let's compute the integral $\int_a^b \frac{1}{x-\sqrt{x}}\,dx$. This looks like a scary integral, but we can use the substitution trick to compute this integral. We'll apply the substitution $u = \sqrt{x}$, which implies $u'(x) = \frac{1}{2\sqrt{x}}$, and the new limits of integration $u_a = \sqrt{a}$ and $u_b = \sqrt{b}$. Performing the three steps of the substitution trick gives us:

$$\begin{aligned}
\int_{x=a}^{x=b} \frac{1}{x-\sqrt{x}}\,dx &= \int_{x=a}^{x=b} \frac{1}{x-\sqrt{x}}\,\frac{1}{2\sqrt{x}}\,du \\
&= \int_{x=a}^{x=b} \frac{1}{u^2-u}\,2u\,du \\
&= \int_{u(a)}^{u(b)} \frac{1}{u^2-u}\,2u\,du = \int_{u(a)}^{u(b)} \frac{2u}{u^2-u}\,du \\
&= \int_{u(a)}^{u(b)} \frac{2}{u-1}\,du = 2\ln(u-1)\Big|_{u(a)}^{u(b)} \\
&= 2\ln(\sqrt{b}-1) - 2\ln(\sqrt{a}-1).
\end{aligned}$$

Read this sequence of equations slowly and try to identify the what changes in each step. I know it looks like alien scribbles, but if you read it carefully you'll see the scribbles make sense logically. We're simply doing the search-and-replace on $u = \sqrt{x}$, but to do this right we need to also replace dx with du, and use the new limits of integration.

In the fourth line, we recognized the general form of the function inside the integral, $f(u) = \frac{2}{u-1}$, to be similar to the function $f(u) = \frac{1}{u}$ whose integral function is $\ln(u)$. Accounting for the -1 horizontal shift and the factor of 2 in the numerator, we obtain the answer $2\ln(u-1)$. In the last step, we changed back from u-variables to x-variables to compute the final answer.

The substitution trick for integrals comes from the chain rule for derivatives $[f(u(x))]' = f'(u(x))u'(x)$. The substitution rule only works for computing integrals of function that have the special structure $f'(u(x))u'(x)$.

Integration by parts: Integration by parts is useful when the function you're integrating has the special structure $f(x)g'(x)$.

$$\int f(x)\, g'(x)\, dx \;=\; f(x)g(x) \;-\int f'(x)g(x)\, dx.$$

It is easier to remember the integration by parts formula in its shorthand notation, $\int u\, dv = uv - \int v\, du$. You can think of integration by parts as a form of "double substitution," where you simultaneously replace u and dv. For definite integrals, integration by parts requires evaluating the product of the functions at the limits:

$$\int_a^b u\, dv = (uv)\Big|_a^b \;-\; \int_a^b v\, du.$$

Example: Let's calculate $\int_0^5 xe^x\, dx$ using the integration by parts procedure. We apply the substitutions $u = x$ and $dv = e^x dx$, which means $du = dx$ and $v = e^x$. Applying the formula for integration by parts gives us

$$\begin{aligned}\int_0^5 x\, e^x dx &= (xe^x)\Big|_0^5 \;-\; \int_0^5 e^x\, dx \\ &= (xe^x)\Big|_0^5 \;-\; e^x\Big|_0^5 \\ &= \left[5e^5 - 0e^0\right] \;-\; \left[e^5 - e^0\right] \\ &= 5e^5 - e^5 + 1 \\ &= 4e^5 + 1.\end{aligned}$$

Other tricks: The two tricks we showed are only the tip of the iceberg. Mathematicians and physicist have come up with hundreds of formulas and tricks for calculating integrals, since for centuries people were forced to use pen and paper calculations. There are tricks for trigonometric functions, square roots, fractions that that involve $x^2 + a^2$, etc. There is an entire course called integral calculus (CALC II), which is dedicated to learning integration tricks. CALC II is a required course for physicists and engineers.

I. Computing integrals numerically using SciPy

The Python function `integrate` is a useful teaching tool, but it would be much too slow to use for practical scientific computing tasks. The Python function `quad` defined in the module `scipy.integrate` is a much more powerful tool for computing numerical integral. The name `quad` is short for "quadrature," which a historical math name for the find-the-area-under-the-graph-of-a-function calculation performed by splitting it into small sections and adding up the total area of the sections.

Let's briefly revisit the Examples 1N, 2N, and 3N using the function `quad`. To compute the integral $\int_0^5 f(x)dx$ we call the function `quad` with inputs `f` as the first argument, and the limits of integration $a = 0$ and $b = 5$ as the second and third arguments.

```
>>> from scipy.integrate import quad
>>> quad(f, 0, 5)
(15.0, 1.1102230246251565e-13)
```

The function `quad` returns two numbers as outputs, (`area`,ϵ). The first number is the value of the area we're interested in. The second number ϵ tells us the accuracy of the procedure used to calculate the area. In the above calculation, the output tells us the definite integral $\int_0^5 f(x)dx$ is equal to `15.0` up to a precision on the order of 10^{-13}. Since we're usually only interested in the `area` value and not the precision ϵ, we often select the first half of `quad`'s output. This is why you'll often see the expression `quad(...)[0]` in code examples.

```
>>> quad(f, 0, 5)[0]
15.0
```

We can also use `quad` to calculate the integrals $\int_0^5 g(x)\, dx$ and $\int_0^2 h(x)\, dx$ form the other two examples.

```
>>> quad(g, 0, 5)[0]
12.5
>>> quad(h, 0, 2)[0]
5.3333333333333333
```

The answers we obtain match the results we obtained earlier. The main takeaway message is that the `quad` function is your friend whenever you need to compute integrals. All the scary-looking math equations that contain the $\int$ symbol can be computed using one or two lines of Python code. Specifically, whenever you see $\int_a^b$ `<f>` dx in a math formula, you can replace that with `quad(f,a,b)[0]`.

J. Computing integrals functions using SymPy

We can use Python to do *symbolic* integration using variables (symbols) instead of numbers. The SymPy function `sp.integrate` allows us to obtain the formulas for integrals and integral functions. We'll now revisit the integral calculations from the three examples using symbolic math calculations. Before we can begin, we must define symbolic symbols variables `x`, `a`, and `b`, which we'll use to express the function f, g, and h and the limits of integration.

```
>>> import sympy as sp
>>> x, a, b = sp.symbols("x a b")
```

Example 1S: Constant function: Consider the constant function $f(x) = 3$, which we can define as follows:

```
>>> fx = 3
>>> fx
3
```

To compute the integral $\int_a^b f(x)dx$, we call the SymPy function `sp.integrate`, passing in the function as the first argument, and the triple `(x,a,b)` as the second argument, where x specifies the variable of integration, and a and b are the limits of integration:

```
>>> sp.integrate(fx, (x,a,b))  # = A_f(a,b)
3*(b-a)
```

Since a and b are arbitrary constants, the answer we obtain for $A_f(a,b) = \int_a^b f(x)dx$ is a general-purpose formula that works for all possible intervals of integration $[a, b]$. Geometrically, we recognize the result as the height-times-width formula for the area of a rectangle, which we have seen several times already.

To compute the definite integral between $a = 0$ and $b = 5$, we specify the numerical limits of integration instead of the symbols a and b.

```
>>> sp.integrate(fx, (x,0,5))
15
```

This result matches the value we obtained using geometrical calculation in Figure 12, and the approximation we obtained using numerical integration `quad(f,0,5)`.

We can also compute the integral function $F_0(b)$, which is defined as $F_0(b) \stackrel{\text{def}}{=} \int_0^b f(x)dx$, for the function $f(x) =$ `fx`.

```
>>> F0b = sp.integrate(fx, (x,0,b))  # = F_0(b)
>>> F0b
3*b
```

Recall that the integral function F_0 is the area-under-the-graph calculation with a variable upper limit of integration b. See Figure 18 for an illustration of the integral function $F_0(b)$.

Given $F_0(b)$, we can compute the definite integral between $a = 0$ and $b = 5$ using the formula $\int_0^5 f(x)dx = F_0(5) - F_0(0)$. We'll need to used the method `subs` (short for substitute) on the expression `F0b` to "plug in" the values $b = 5$ and $b = 0$.

```
>>> F0b.subs({b:5}) - F0b.subs({b:0})
15
```

The `subs` method expects as inputs a Python dictionary whose keys are symbols, and whose values represent the numbers we want to plug into the expression.

Example 2S: Linear function: Let's now compute the integral function of the linear function $g(x) = x$, which corresponds to the following SymPy expression:

```
>>> gx = 1*x
>>> gx
x
```

To compute the integral function $G_0(b) \stackrel{\text{def}}{=} \int_0^b g(x)dx$, we call `sp.integrate` using the symbol `b` for the upper limit of integration:

```
>>> G0b = sp.integrate(gx, (x,0,b))  # = G_0(b)
>>> G0b
b**2 / 2
```

The expression $G_0(b) = \frac{1}{2}b^2$ we obtain is identical to the formula we obtained from the geometric calculation in Figure 19.

Given $G_0(b) =$ `G0b`, we can compute the definite integral $\int_0^5 g(x)\,dx$ using the formula $\int_0^5 g(x)\,dx = G_0(5) - G_0(0)$. We plug in $b = 5$ and $b = 0$ using the `subs` method:

```
>>> G0b.subs({b:5}) - G0b.subs({b:0})
25/2
```

SymPy computed the exact answer for us as a fraction $\frac{25}{2}$. This answer matches the value we obtained earlier using numerical integration, `quad(g,0,5)[0]` = 12.5.

Example 3S: Polynomial function: Recall $h(x) = 4 - x^2$.

```
>>> hx = 4 - x**2
>>> hx
4 - x**2
```

The integral function $H_0(b) = \int_0^b h(x)\,dx$ is given by:

```
>>> H0 = sp.integrate(hx, (x,0,b))
>>> H0
4*b - b**3/3
```

The integral function $H_0(b) = 4b - \frac{1}{3}b^3$ corresponds to the area calculation under $h(x)$ starting at $x = 0$.

K. Applications of integration

Intuitively, we use integrals whenever we want to compute the "total" of some quantity that varies over time or space.

Kinematics: Calculus was originally invented to describing the equations of motion $x(t)$, $v(t)$, and $a(t)$ as a function of time t. We call these the *kinematics* equations, from the Greek word *kinema* for motion. The velocity function $v(t)$ is the derivative of the position function, and the acceleration $a(t)$ is the derivative of the velocity:

$$a(t) \xleftarrow{\frac{d}{dt}} v(t) \xleftarrow{\frac{d}{dt}} x(t).$$

The starting point of kinematics the Newton's second law, which tells us the acceleration of the object of mass m that has a net force $\vec{F}_{\text{net}}$ acting on it is: $\frac{1}{m}\vec{F}_{\text{net}}(t) = a(t)$. Given the knowledge of acceleration $a(t)$, we can predict the position of the object $x(t)$ at any time t by "undoing" the derivative operations using integration:

$$\frac{1}{m}\vec{F}_{\text{net}}(t) \;=\; \underbrace{a(t) \;\xrightarrow{v_i+\int dt}\; v(t) \;\xrightarrow{x_i+\int dt}\; x(t)}_{\text{kinematics}}.$$

We use our knowledge of $a(t)$ and the initial velocity v_i to obtain $v(t)$. We then use integration a second time to obtain $x(t)$ from $v(t)$, taking into account the initial position x_i as the integration constant.

The case of *uniform accelerated motion* (UAM) is of particular interest, since it describes the the trajectory of falling objects under the effect of gravity. An object of mass m that has a constant net $\vec{F}_{\text{net}}$ acting on it will experience a constant acceleration $a(t) = a$. We can use integration to find the velocity of this object at a later time $t = \tau$,

$$v(\tau) = v_i + \int_0^\tau a(t)\,dt = v_i + \int_0^\tau a\,dt = v_i + a\tau.$$

Now that we know the velocity as a function of time $v(t)$, we can use integration a second time to find the position

$$x(\tau) = x_i + \int_0^\tau v(t)\,dt = x_i + \int_0^\tau (v_i + at)\,dt = x_i + v_i\tau + \tfrac{1}{2}a\tau^2.$$

Two simple calculus steps allows us to obtain the famous kinematics equation $x(t) = x_i + v_it + \frac{1}{2}at^2$, which describes the motion of objects undergoing constant acceleration a. Students taking a physics class are normally presented with this equation and it seems to come out of nowhere, but if you know calculus you don't have memorize it, you can always derive from first principles $a(t) = a$, and initial conditions $x_i \stackrel{\text{def}}{=} x(0)$ and $v_i \stackrel{\text{def}}{=} v(0)$.

Solving differential equations: Many important laws in science and engineering are described by *differential equations* that specify unknown function $f(t)$ in terms of their derivatives $f'(t)$, $f''(t)$, etc. For example, in the starting point of the kinematics equations with constant acceleration is the differential equation $x''(t) = a$, and we use use integration (twice) to find the unknown function $x(t) = x_i + v_i t + \frac{1}{2}at^2$, which is a *solution* to the differential equation $x''(t) = a$. We can verify this by taking the second derivative of $x(t)$ to confirm it satisfies $x''(t) = a$.

Here are some other examples of differential equations and their solutions;

- In biology, unconstrained bacterial growth is described by the equation $b'(t) = kb(t)$, where $b(t)$ is the number of bacteria at time t. Intuitively, the number of new bacteria $b'(t)$ is proportional to the number of existing bacteria. The solution to this equation is $b(t) = Ce^{\lambda t}$, where C describes the number of bacteria at time $t = 0$.
- Radioactive decay is described by the differential equation $r'(t) = -\lambda r(t)$, where $r(t)$ describes the number of atoms of some radioactive element. The solution is $r(t) = Ce^{-\lambda t}$.
- Simple harmonic motion is described by the second order differential equation $x''(t) + \omega^2 x(t) = 0$, which has solution $x(t) = C_1 \sin(\omega t) + C_2 \cos(\omega t)$.

Taking a course on differential equations you'll learn all kind of tricks and techniques for solving differential equations. Integration plays key roles in all the techniques, since it allows us to "reverse engineer" the derivative operation.

Probability calculations: Integration is an essential tool for computing probabilities of continuous random variables. A continuous random variable X is described by its *probability density function* f_X and the probability of the event $\{a \leqslant X \leqslant b\}$ is defined as the following integral:

$$\Pr(\{a \leqslant X \leqslant b\}) \stackrel{\text{def}}{=} \int_a^b f_X(x)\, dx.$$

The probability density f_X varies for different values of x, so if we want to compute the total probability of X falling between $x = a$ and $x = b$, we must compute the integral of f_X. For example, the standard normal random variable Z is described by the probability $f_Z(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}$. To calculate the probability of the event $\{-1 \leqslant Z \leqslant 1\}$, we must evaluate the integral $\int_{-1}^{1} f_Z(z)\, dz$, which is easy to do using SciPy helper function `quad(fZ,a=-1,b=1)` $= 0.68269$.

VI. Sequences and series

A *sequence* a_k is a function that take whole numbers as inputs and produce real numbers as outputs: $a_k : \mathbb{N} \to \mathbb{R}$. The *series* $\sum a_k$ describes the sum of all the terms in the sequence a_k. Series are a powerful computational tool that allow us to describe procedures with infinite number of steps and use these procedures to approximate irrational numbers like e, and transcendental functions like the exponential function $f(x) = e^x$. Sequences and series are the third pillar of the basic calculus knowledge that I want you to have.

A. Sequences are functions with discrete inputs

We use the notation $f : \mathbb{R} \to \mathbb{R}$ to describe the functions that take real numbers $x \in \mathbb{R}$ as inputs and produce real numbers as outputs $f(x) \in \mathbb{R}$. When studying functions that take natural numbers $k \in \mathbb{N}$ as inputs, we use a different notation: $a_k : \mathbb{N} \to \mathbb{R}$, where a_k describes the k^{th} term in the sequence. The sequence's input variable is usually denoted k. and corresponds to the *index* within the sequence. Usually k is a natural number $k \in \mathbb{N} \stackrel{\text{def}}{=} \{0, 1, 2, 3, 4, \ldots\}$ but some sequences are only defined nor positive natural numbers $k \in \mathbb{N}_+ \stackrel{\text{def}}{=} \{1, 2, 3, 4, \ldots\}$. Note the chance in notation: we use a_k for sequences instead of the usual function notation $a(k)$.

We can define a sequence by specifying the formula for the k^{th} term in the sequence a_k. For example, the *harmonic sequence* is defined by the formula $a_k = \frac{1}{n}$. Another way to define a sequence is by listing the first few values in the sequence: $[a_0, a_1, a_2, a_3, \ldots]$, which correspond to evaluating formula a_k for $k = 0$, $k = 1$, $k = 2$, $k = 3$, etc. We'll now look at some common examples sequences specified both as functions of k and show the the first few values in each sequence.

The natural numbers: The simplest possible example of a sequence is the identity function, that returns the index input k as output:

$$n_k \stackrel{\text{def}}{=} k, \text{ for } k \in \mathbb{N} \quad \Leftrightarrow \quad [0, 1, 2, 3, 4, 5, 6, 7, \ldots].$$

This is the fundamental counting sequence that describes the process of taking "unit step" to the right on the number line, starting at the origin.

Squares of natural numbers: The sequence-equivalent of the quadratic function $f(x) = x^2$ is the sequence of squares of the integers:

$$q_k \stackrel{\text{def}}{=} k^2, \text{ for } k \in \mathbb{N} \quad \Leftrightarrow \quad [0, 1, 4, 9, 16, 25, 36, 49, \ldots].$$

Harmonic sequence: We obtain another useful sequence by computing the fraction $\frac{1}{k}$:

$$h_k \stackrel{\text{def}}{=} \tfrac{1}{k}, \text{ for } k \in \mathbb{N}_+ \quad \Leftrightarrow \quad \left[1, \tfrac{1}{2}, \tfrac{1}{3}, \tfrac{1}{4}, \tfrac{1}{5}, \tfrac{1}{6}, \tfrac{1}{7}, \ldots\right].$$

The *harmonic sequence* appears in music. For most music instruments, when we want to play the note that corresponds to the frequency f, we also hear notes with frequencies that are integer multiple of the "dominant" frequency: $2f$, $3f$, $4f$, etc., which are called the harmonics. The harmonic sequence describes the wavelengths of the harmonics frequencies. On a string instrument, the harmonic sequence tells you where to place your fingers if you want to play higher harmonics.

The alternating harmonic sequence: Consider now a harmonic sequence with alternating positive an negative outputs:

$$a_k \stackrel{\text{def}}{=} \tfrac{(-1)^{k+1}}{k}, \text{ for } k \in \mathbb{N}_+ \quad \Leftrightarrow \quad \left[1, -\tfrac{1}{2}, \tfrac{1}{3}, -\tfrac{1}{4}, \tfrac{1}{5}, -\tfrac{1}{6}, \tfrac{1}{7}, \ldots\right].$$

The factor $(-1)^{k+1}$ is positive for all odd inputs $k \in \{1, 3, 5, 7, \ldots\}$ since $(-1)^m = +1$ for any even number m. The factor $(-1)^{k+1}$ is negative for all even inputs $k \in \{2, 4, 6, 8, \ldots\}$, hence the values in the sequence oscillate between positive and negative.

Inverse factorial sequence: The factorial function is denoted $k!$ and describes the product of the first k positive natural numbers; $k! \stackrel{\text{def}}{=} k \cdot (k-1) \cdots 3 \cdot 2 \cdot 1$. We'll see factorials in several formulas in this section. In particular, the following sequence is of particular interest:

$$f_k \stackrel{\text{def}}{=} \tfrac{1}{k!}, \text{ for } k \in \mathbb{N}_+ \quad \Leftrightarrow \quad \left[1, \tfrac{1}{2!}, \tfrac{1}{3!}, \tfrac{1}{4!}, \tfrac{1}{5!}, \tfrac{1}{6!}, \tfrac{1}{7!}, \ldots\right].$$

The values in the inverse factorial sequence quickly become very small, because the factorial function grows very quickly: $2! = 2$, $3! = 6$, $4! = 24$, $5! = 120$, $6! = 720$, $7! = 5040$, …, $10! = 3628800$, …, $13! \approx 6.2 \times 10^9$, …, $70! \approx 1.2 \times 10^{100}$, etc.

Geometric sequence: The sequences-equivalent of the exponential function $f(x) = e^x$ is the *geometric sequence* where the k^{th} value in the sequence is the k^{th} powers of some number r:

$$g_k \stackrel{\text{def}}{=} r^k, \text{ for } k \in \mathbb{N} \quad \Leftrightarrow \quad \left[1, r, r^2, r^3, r^4, r^5, r^6, r^7, \ldots\right].$$

Each term in the sequence equals r times the previous term, which describes *geometric process* that repeatedly grows/shrinks by the amount r. When $r < 1$ the values in sequence g_k quickly go to zero, similar to how exponential function e^{-x} goes to zero for large value of x. When $r > 1$ the sequence g_k increases quickly, similar to how exponential function e^x increases for large value of x.

Powers of two: The special case of the geometric series with $r = 2$ is of particular interest, so we'll alias-define it as the sequence b_k:

$$b_k \stackrel{\text{def}}{=} 2^k, \text{ for } k \in \mathbb{N} \quad \Leftrightarrow \quad [1, 2, 4, 8, 16, 32, 64, 128, \ldots].$$

This sequence comes up all over the place in computer science because it is describes the number of different numbers we can store in k bits of memory.

B. Convergence of sequences

What happens to sequences as k goes to infinity? We can use the limit notation $\lim_{k\to\infty}$ to describe this process. There are to possible behaviours: either the values of the sequence blow up to infinity, or they approach some fixed number.

For example, the sequences $n_k \stackrel{\text{def}}{=} k$, $q_k \stackrel{\text{def}}{=} k^2$, and $b_k \stackrel{\text{def}}{=} 2^k$ keep getting larger and larger as k goes to infinity:

$$\lim_{k\to\infty} k = \infty, \qquad \lim_{k\to\infty} k^2 = \infty, \qquad \lim_{k\to\infty} 2^k = \infty.$$

We say these sequences are *divergent*. In contrast, the values in the sequences $h_k \stackrel{\text{def}}{=} \frac{1}{k}$, $a_k \stackrel{\text{def}}{=} \frac{(-1)^{k+1}}{k}$, and $f_k \stackrel{\text{def}}{=} \frac{1}{k!}$ *converge* to the value 0 in the limit as k goes to infinity.

$$\lim_{k\to\infty} \frac{1}{k} = 0, \qquad \lim_{k\to\infty} \frac{(-1)^{k+1}}{k} = 0, \qquad \lim_{k\to\infty} \frac{1}{k!} = 0.$$

The geometric series $g_k \stackrel{\text{def}}{=} r^k$ converges only if absolute value of r is less than one: $\lim_{k\to\infty} r^k = 0$, $|r| < 1$.

The limit of a sequence as k goes to infinity is directly analogous to the limit of function $f(x)$ as x goes to infinity.

C. Summation notation

We're often interested in computing sums of values in a sequence. To describe the sum of 3rd, 4th, and 5th elements of the sequence c_k, we turn to summation notation:

$$\sum_{k=3}^{5} c_k = c_3 + c_4 + c_5.$$

The capital Greek letter *sigma* stands in for the word *sum*, and the range of index values included in this sum is denoted below and above the summation sign.

The sum of the values in the sequence c_k from $k = 0$ until $k = n$ is denoted as

$$\sum_{k=0}^{n} c_k = c_0 + c_1 + c_2 + \cdots + c_{n-1} + c_n.$$

Since this is a calculus tutorial, you should expect that an infinity of some kind will show up soon, and indeed we'll learn about *infinite series*, that describe the sum of *all* the values in the sequence c_k: $\sum c_k \stackrel{\text{def}}{=} \lim_{n\to\infty} \sum_{k=0}^{n} c_k$. But before we get to infinite sums, let's start at some finite sums to get a feel of this thing.

D. Exact sums

We'll now show some useful formulas for calculating sum of the terms in certain sequences. For example, here is a formula for the sum of the first n terms in the geometric sequence:

$$G_n = \sum_{k=0}^{n} r^k = 1 + r + r^2 + \cdots + r^n = \frac{1 - r^{n+1}}{1 - r}.$$

We can use this formula to find the sum of the powers of 2:

$$\sum_{k=0}^{n} 2^k = 1 + 2 + 4 + 8 + \cdots + 2^n = \frac{1 - 2^{n+1}}{1 - 2} = 2^{n+1} - 1.$$

The sum of the first n positive integers and the sum of their squares are:

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2} \qquad \text{and} \qquad \sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}.$$

The *binomial coefficient* is denoted using the symbol $\binom{n}{k}$, which is read as "n choose k." The binomial coefficient counts the number of combinations of k items we can choose from a set of n items, and it is computed using the formula: $\binom{n}{k} \stackrel{\text{def}}{=} \frac{n!}{(n-k)!\,k!}$. For example, the number of combinations of size 2 selected from a list of 5 items is $\binom{5}{2} = \frac{5!}{(5-2)!\,2!} = \frac{5!}{3!\,2!} = \frac{5\cdot4\cdot3\cdot2\cdot1}{3\cdot2\cdot1\ \cdot\ 2\cdot1} = \frac{120}{6\cdot2} = 10$. The binomial coefficient appears in the expansion of the binomial expression $(a+b)^n$, which can written as the following summation:

$$\begin{aligned}(a+b)^n &= \tbinom{n}{0}a^n + \tbinom{n}{1}a^{n-1}b + \cdots + \tbinom{n}{n-1}ab^{n-1} + \tbinom{n}{n}b^n \\ &= \textstyle\sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k.\end{aligned}$$

This sum appears in several calculations in probability theory.

E. Series

Series are defined as the sums computed from the terms in the sequence c_k. The *finite series* $\sum_{k=1}^{n} c_k$ computes the first n terms of the sequence:

$$\sum_{k=1}^{n} c_k = c_1 + c_2 + c_3 + c_4 + c_5 + \cdots + c_{n-1} + c_n.$$

The *infinite series* $\sum c_k$ computes *all* the terms in the sequence:

$$\sum c_k \stackrel{\text{def}}{=} \lim_{n\to\infty} \sum_{k=1}^{n} c_k = c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + \cdots.$$

The infinite series $\sum c_k$ of the sequence $c_k : \mathbb{N} \to \mathbb{R}$ is analogous to the integral $\int_0^\infty f(x)\,dx$ of a function $f : \mathbb{R} \to \mathbb{R}$.

Some infinite series converge to a finite value. For example, when $|r| < 1$, the limit as $n \to \infty$ of the geometric series converges to the following value:

$$G_\infty = \lim_{n\to\infty} G_n = \sum_{k=0}^{\infty} r^k = 1 + r + r^2 + r^3 + \cdots = \frac{1}{1-r}.$$

This expression describes an infinite sum, which is not possible to compute in practice, but we can see the truth of this equation using our mind's eye. The formula for first n terms is the geometric series is $G_n = \frac{1-r^{n+1}}{1-r}$. The term r^{n+1} goes to zero as $n \to \infty$, so the only part of the formula that remains is $\frac{1}{1-r}$.

Example 1: sum of a geometric series: Let's use the formula to compute infinite series of the geometric sequence with $r = \frac{1}{2}$:

$$\sum_{k=0}^{\infty} \left(\tfrac{1}{2}\right)^k = 1 + \tfrac{1}{2} + \tfrac{1}{4} + \tfrac{1}{8} + \tfrac{1}{16} + \tfrac{1}{32} + \cdots = \frac{1}{1 - \frac{1}{2}} = 2.$$

Figure 21 shows a visualization for this infinite sum.

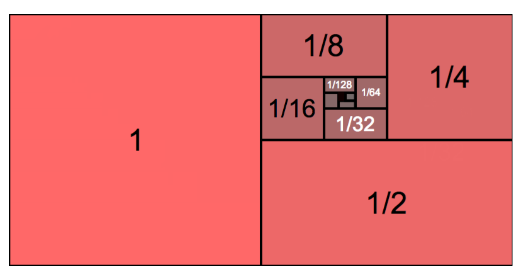


Figure 21. A graphical representation of the infinite sum of the geometric series with $r = \frac{1}{2}$. The area of each region corresponds to one of the terms in the series. The total area is equal to $\sum_{k=0}^{\infty}(\frac{1}{2})^k = \frac{1}{1-\frac{1}{2}} = 2$.

That's kind of cool, no? We're able to compute the value of a summation with infinitely meany terms, because we have the general pattern G_n for the sum with n terms then evaluate the limit as n goes to infinity.

Convergent and divergent series: We say the geometric series $G_\infty = \sum g_k = \sum_{k=0}^{\infty} r^k$ *converges* to the value $\frac{1}{1-r}$. We can also say that the infinite geometric series $\sum g_k$ *is convergent*, meaning it has a finite value and doesn't blow up. Another example of a converged infinite series is $F_\infty = \sum f_k$, which converges to the number e, as we'll see in Example 2 below.

In contrast, the regular harmonic series $\sum h_k$ *diverges*. When we sum together more and more terms of the sequences h_k, the total computed keeps growing and the series blows up to infinity $\sum h_k = \infty$, We say the harmonic series *is divergent*.

Using convergent series for practical calculations: We can use infinite series to compute the value of irrational numbers.

Example 2: Euler's number: The infinite sum of the sequence $f_k \stackrel{\text{def}}{=} \frac{1}{k!}$ converges to Euler's number $e = 2.71828\ldots$:

$$F_\infty = \lim_{n\to\infty} F_n = \sum_{k=0}^{\infty} \tfrac{1}{k!} = 1 + 1 + \tfrac{1}{2} + \tfrac{1}{6} + \tfrac{1}{24} + \tfrac{1}{720} + \cdots = e.$$

The calculation above is not just cool math fact, but a useful computational procedure that we can use to compute the value of $e = 2.71828\ldots$ using only basic arithmetic operations like repeated multiplication (factorial), division, and addition.

Let's look at some practical calculations where we compute the first $n = 10$, $n = 15$, and $n = 20$ terms in the infinite series.

```
>>> import math
>>> def f_k(n):
        return 1 / math.factorial(n)
>>> sum([f_k(k) for k in range(0,10)])
2.718281...
>>> sum([f_k(k) for k in range(0,15)])
2.71828182845...
>>> sum([f_k(k) for k in range(0,20)])
2.71828182845905...
```

Summing together the first 10 terms in the infinite series gives us an approximation to Euler's number e that is accurate to six decimals. With 15 terms we get an approximation that is accurate to 11 decimals. The more terms we include in the summation, the closer we get to the true value of $e = 2.71828182845905\ldots$.

If we want to compute the *exact* value of e, we would need to compute the infinite series $\sum_{k=0}^{\infty} \frac{1}{k!}$. We can do this using SymPy by calling the function `sp.summation` whose syntax is similar to the function `sp.integrate` we used to compute integrals. The first argument is an the expression for the k^{th} term in the sequence, then we specify the index variable, the starting point and the end point of the summation:

```
>>> import sympy as sp
>>> k = sp.symbols("k")
>>> sp.summation(1/sp.factorial(k), (k,0,sp.oo))
E
```

We use `sp.oo` to instruct SymPy to perform the infinite sum, which produced the exact symbolic answer `E` $= e$.

There other series we can use to compute values of interest.

Example 3: We can calculate the value $\ln(2)$ by computing the infinite sum of the alternating harmonic sequence $a_k \stackrel{\text{def}}{=} \frac{(-1)^{k+1}}{k}$:

$$A_\infty = \lim_{n\to\infty} \sum_{k=1}^{n} \frac{(-1)^{k+1}}{k} = 1 - \tfrac{1}{2} + \tfrac{1}{3} - \tfrac{1}{4} + \tfrac{1}{5} - \tfrac{1}{6} + \tfrac{1}{7} + \cdots = \ln(2).$$

To obtain the exact value $\ln(2)$, we need to sum together an infinite number of terms in the series $\sum a_k$, but we can obtain successively better approximations to $\ln(2)$ using finite sums.

```
>>> def a_k(k):
        return (-1)**(k+1) / k
>>> sum([a_k(k) for k in range(1,100+1)])
0.6...
>>> sum([a_k(k) for k in range(1,1000+1)])
0.69...
>>> sum([a_k(k) for k in range(1,1_000_000+1)])
0.69314...
```

The series approximation to $\ln(2)$ converges more slowly that the series approximation to e we saw in the previous example. We need to sum 1M terms in the series to obtain and approximation that is accurate to five decimals. Nevertheless if we keep calculating the series with more and more terms, we can obtain an approximation that arbitrarily close to the true value $\ln(2) = 0.6931471805599453\ldots$.

To get the exact value $\ln(2)$, we can make SymPy compute the infinite series:

```
>>> sp.summation((-1)**(k+1) / k, (k,1,sp.oo))
log(2)
```

In this manner, we can come up with all kinds of other infinite series expression to calculate other number of interest like π. Instead of showing you other series for approximating numbers, I'll show you something even more powerful: a technique for approximating functions as infinite series.

F. Power series

The term *power series* describes a series whose terms contain different powers of the variable x. The k^{th} term in a power series consists of some coefficient c_k and k^{th} power of the variable x:

$$P_n(x) = \sum_{k=0}^{n} c_k x^k = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + \cdots + c_n x^n.$$

The math expression we obtain in this way is a *polynomial* of degree n in x, which we'll denote $P_n(x)$. Depending on the choice of the coefficients $(c_0, c_1, c_2, c_3, \ldots, c_n)$ we can make the polynomial function $P_n(x)$ *approximate* some other function $f : \mathbb{R} \to \mathbb{R}$. To find such approximations, we need to figure out how to choose the coefficients c_k of the power series.

G. Taylor series

The *Taylor series approximation* to the function $f(x)$ is a power series whose coefficients c_k are computed by evaluating the k^{th} derivative of the function $f(x)$ at $x = 0$, which we'll denote $f^{(k)}(0)$. Specifically, the k^{th} coefficient in the Taylor series approximation for the function $f(x)$ is $c_k \stackrel{\text{def}}{=} \frac{f^{(k)}(0)}{k!}$. The finite series with n terms produces the following approximation:

$$f(x) \approx f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \cdots + \frac{f^{(n)}(0)}{n!}x^n = \sum_{k=0}^{n} \frac{f^{(k)}(0)}{k!}x^k.$$

In the limit as n goes to infinity, the approximation becomes exact:

$$f(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \frac{f^{(3)}(0)}{3!}x^3 + \cdots = \sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!}x^k.$$

Using this formula and your knowledge of derivative formulas, you can compute the Taylor series of any function $f(x)$. For example, let's find the Taylor series of the function $f(x) = e^x$ at $x = 0$. The first derivative of $f(x) = e^x$ is $f'(x) = e^x$. The second derivative of $f(x) = e^x$ is $f''(x) = e^x$. In fact, all the derivatives of $f(x)$ will be e^x because the derivative of e^x is equal to e^x. The k^{th} coefficient in the power series of $f(x) = e^x$ at the point $x = 0$ is equal to the value of the k^{th} derivative of $f(x)$ evaluated at $x = 0$. In the case of $f(x) = e^x$ we have $f^{(k)}(0) = e^0 = 1$, so the coefficient of the k^{th} term is $c_k = \frac{f^{(k)}(0)}{k!} = \frac{1}{k!}$, so the Taylor series of $f(x) = e^x$ is

$$e^x = \sum_{k=0}^{\infty} \frac{1}{k!}x^k = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \frac{x^7}{7!} + \cdots$$

Series are a powerful computational tool for approximating numbers and functions. As we compute more terms from the above series, our the polynomial approximation to the function $f(x) = e^x$ becomes more accurate. .

Table II shows the Taylor series obtained using the formula $f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!}x^k$ for several important functions.

TABLE II
TAYLOR SERIES EXPANSIONS FOR COMMONLY USED FUNCTIONS

$$\frac{1}{1-x} = \sum_{k=0}^{\infty} x^k = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + \cdots$$

$$\frac{1}{1+x} = \sum_{k=0}^{\infty} (-x)^k = 1 - x + x^2 - x^3 + x^4 - x^5 + x^6 + \cdots$$

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \cdots$$

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!}x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!}x^{2n} = 1 - \frac{x^2}{2} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots$$

$$\ln(x+1) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n}x^n = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \frac{x^5}{5} + \cdots$$

Reader who are familiar with the concept of a basis from linear algebra can think of the Taylor series shown in Table II as *representations* of the corresponding functions with respect to the basis of polynomial functions: $(1, x, x^2, x^3, x^4, x^5, \ldots)$. The Taylor series coefficients $c_k = \frac{f^{(k)}(0)}{k!}$ are the coordinates of the function $f(x)$ in the polynomial basis.

H. Obtaining Taylor series using SymPy

The `SymPy` function `series` is a convenient way to obtain the Taylor series of any function. Calling `series(fun,var,x0,n)` will show you the series expansion of any function `fun` near `var=x0` up to powers of `n`. We can quickly fact-check the Taylor series given in Table II using SymPy.

```
>>> import sympy as sp
>>> x = sp.symbols("x")
>>> sp.series(1/(1-x), x, x0=0, n=7)
1 + x + x**2 + x**3 + x**4 + x**5 + x**6 + O(x**7)
>>> sp.series(1/(1+x), x, x0=0, n=7)
1 - x + x**2 - x**3 + x**4 - x**5 + x**6 + O(x**7)
>>> sp.series(sp.E**x, x, x0=0, n=6)
1 + x + x**2/2 + x**3/6 + x**4/24 + x**5/120 + O(x**6)
>>> sp.series(sp.sin(x), x, x0=0, n=8)
x - x**3/6 + x**5/120 - x**7/5040 + O(x**8)
>>> sp.series(sp.cos(x), x, x0=0, n=8)
1 - x**2/2 + x**4/24 - x**6/720 + O(x**8)
>>> sp.series(sp.ln(x+1), x, x0=0, n=6)
x - x**2/2 + x**3/3 - x**4/4 + x**5/5 + O(x**6)
```

The big-O notation `O(x**n)` appears in all the above outputs as a reminder that the exact Taylor series contains additional terms, and the Taylor series approximations shows are only accurate up to an error "on the order of" x^n.

I. Series applications

We've already seen we can use compute e using the infinite series $\sum f_k$. Now that we know the Taylor series of e^x, we can also compute e^5 as ...

Another application of Taylor series representation for the function $f(x)$, is the relative easy with which we can compute its derivative function $f'(x)$ and its integral function $F_0(x) \stackrel{\text{def}}{=} \int_0^x f(u)\,du$. Since the Taylor series approximation consists only of polynomial terms of the form $c_n x^n$, we just need to compute the derivative function $f'(x)$ we simply compute the derivative of each term: $nc_n x^{n-1}$. Similarly, to compute the integral function $F_0(x) \stackrel{\text{def}}{=} \int_0^x f(u)\,du$, we compute the integrals of the individual terms, which gives us $\frac{c_n}{n+1}x^{n+1}$.

EXAMPLE

This example is of historical importance, since it was issued as challenge by Isaac Newton to Wilhelm Leibnitz. No formulas for the integral of ln x was not know at the time, and was considered a difficult computation problem. Newton had discover the Taylor series approximation formula for ln x but had not published his result. A bit like closed source software. Leibnitz had independently discovered calculus and series so he was able to compute the answer. When Newton saw that Leibnitz was able to solve the challenge, he was motivated to publish his results, which is why we still credit calculus to him.

TODO ODEs

VII. Multivariable calculus

Multivariable calculus is the extension of the ideas of differential and integral calculus to functions like take multiple variables as inputs. If you were able to follow the single-variable calculus concepts, then you won't have much new math to learn in multivariable calculus: it's essentially the same concepts but with more variables.

A. Multivariable functions

A single variable function $f : \mathbb{R} \to \mathbb{R}$ takes a real number $x \in \mathbb{R}$ as input and produces the real number $f(x) \in \mathbb{R}$ as output. A *multivariable function* takes multiple real numbers as inputs and produces real number as output. For example, a bivariate function $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ takes two real numbers as inputs $(x, y) \in \mathbb{R} \times \mathbb{R}$ and produces the real number $f(x, y) \in \mathbb{R}$ as output. We'll use the function $f(x, y) = 4 - x^2 - \frac{1}{4}y^2$ for all plots and calculations in the remainder of this section.

We can plot the function $f(x, y) = 4 - x^2 - \frac{1}{4}y^2$ as a *surface* in a three dimensional space, as shown in Figure 22. The height of the surface above the point (x, y) is function output $f(x, y)$.

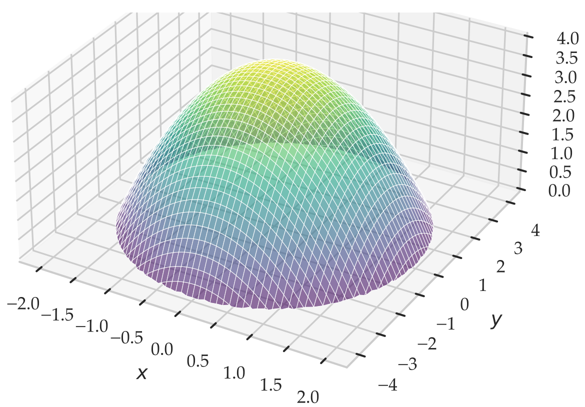


Figure 22. The 3D surface plot of the the function $f(x, y)$.

Three dimensional surface plots are very good for visualizing multivariable functions, but they can be difficult to draw by hand. Another approach for representing the functoin $f(x, y)$ is to use a two-dimensional plot that shows the "view from above" of the surface $f(x, y)$. We can trace *level curves* in the surface, to produce a "topographic map" of the surface where each level curve show the points that are at a certain height.

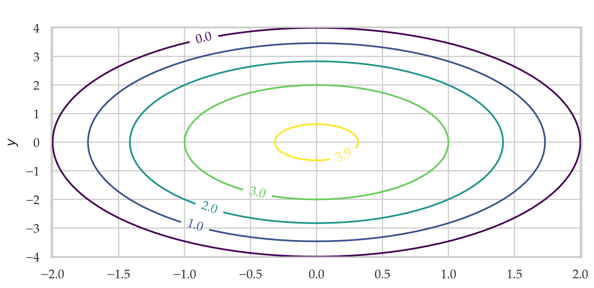


Figure 23. Topographic map that shows the function $f(x, y)$ as level curves.

The curve labeled 0.0 you see in Figure 23 represents the solution to the equation $f(x, y) = 0$, which is where the function $f(x, y)$ intersects the xy-plane.

B. Partial derivatives

For a function of two variables $f(x, y)$, there is an "x-derivative" operator $\frac{\partial}{\partial x}$ and a "y-derivative" operator $\frac{\partial}{\partial y}$. The operation $\frac{\partial}{\partial x} f(x, y)$ describes taking the derivative of $f(x, y)$ with respect to the input variable x, while keeping the input variable y constant. Taking the derivative of a multivariable function with respect to one of its input variables is called a *partial derivative* and denoted with the symbol ∂.

The partial derivative of $f(x, y)$ with respect to x is

$$\frac{\partial}{\partial x} f(x, y) = \frac{\partial f}{\partial x} \stackrel{\text{def}}{=} \lim_{\delta \to 0} \frac{f(x + \delta, y) - f(x, y)}{\delta}.$$

Similarly the partial derivative of with respect to y is

$$\frac{\partial}{\partial y} f(x, y) = \frac{\partial f}{\partial y} \stackrel{\text{def}}{=} \lim_{\delta \to 0} \frac{f(x, y + \delta) - f(x, y)}{\delta}.$$

Note that both $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ are function of x and y. Indeed, we can ask the questions "what is the slope in the x-direction" and "what is the slope in the y-direction" at any point (x, y) on the surface of the function. That's precisely the information returned by the functions $\frac{\partial f}{\partial x}(x, y)$ and $\frac{\partial f}{\partial y}(x, y)$.

TODO: example

C. Gradient

The operator ∇ is a combination of both the x and y derivatives:

$$\nabla f(x, y) \stackrel{\text{def}}{=} \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right).$$

Note that ∇ acts on a function $f(x, y)$ to produce a vector output. This vector is called the *gradient* vector and it tells you the combined x- and y-slopes of the surface. More specifically, the gradient vector tells you the direction of the function's maximum increase—the "uphill" direction at the surface of graph of $f(x, y)$ at the point (x, y). The gradient vector is always perpendicular to the *level curve* at that point.

D. Partial integration

We can perform integration with respect to one of the input variables:

$$f(y) = \int_{x \in \mathbb{R}} f(x, y)\, dx \quad \text{and} \quad f(x) = \int_{y \in \mathbb{R}} f(x, y)\, dy.$$

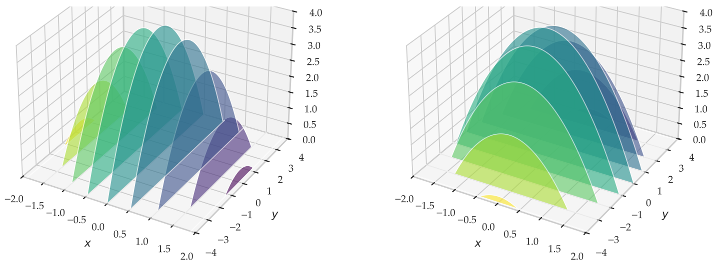
The result of this partial integration is a function of the variable that we didn't integrate.

E. Double integrals

The multivariable generalization of the integral $\int_{x \in I} f(x)\, dx$ that computes the "total" amount of $f(x)$ on some interval $I = [a, b]$ is the multivariable integral of the form:

$$\int\!\!\int_{(x,y) \in R} f(x, y)\, dxdy,$$

where R is called the *region of integration* and corresponds to some subset of the cartesian plane $\mathbb{R} \times \mathbb{R}$. The idea behind



(a) Slices at different values of x. (b) Slices at different values of y.

Figure 24. Visualization of the partial integration procedures.

multivariable integrals is the same as for single variable integrals—to compute the total amount of some function for some range of input values. For single-variable integrals, we split the region into thing rectangular strips of width dx. For double integrals we split the two-dimensional region of integration into small squares of area $dxdy$, and compute the total volume of a many vertical columns whose base area is $dxdy$ and whose height is given by the function $f(x,y)$.

TODO: insert graphic of 3D integral split into vertical columns

TODO: explain "sweep along x then sweep along y" idea + hint at change-of-variables techniques

F. Applications of multivariable calculus

Optimization: The notion of an uphill or downhill direction for the surface $f(x,y)$ turns out to be very useful for optimization. To find the lowest point on the surface (minimum value of $f(x,y)$), you can start at some point and keep moving downhill, that is in the opposite direction to the gradient $-\nabla f(x,y)$. Intuitively, this is the path that a water stream would take as it descends down the slope of the mountain until it reaches the minimum at the bottom of a valley. This intuitive notion of "keep moving downhill until you get to a local minimum" is the general idea behind the *gradient descent* optimization algorithm which is very important for machine learning applications.

VIII. Vector calculus

Vector calculus is the study of vector fields $\mathbf{F}$, which are functions of the form $\mathbf{F} : \mathbb{R}^3 \to \mathbb{R}^3$, which defines 3-dimensional output vector at each point (x, y, z) in space. For example, the electric field $\mathbf{E}(x, y, z)$ describes the strength and the direction of the electric force that a changed particle would experience if placed at (x, y, z).

Vector calculus is *waaaaay* out of scope for an introductory calculus tutorial, so I will just show you some simple definitions of the building blocks.

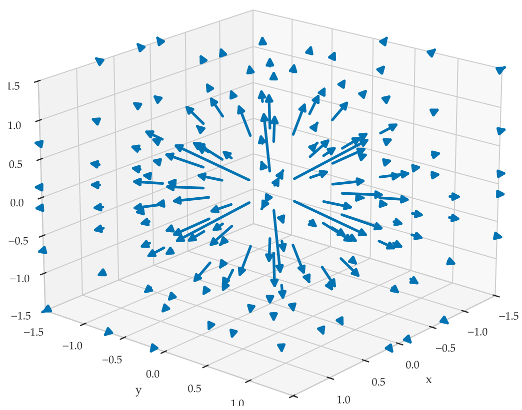


Figure 25. Visualization of the vector field $\vec{E}(x, y, z)$ around a positive charge.

For a point charge (q) located at the origin, the electric field at position $\vec{r} = (x, y, z)$ is

$$\vec{E}(x, y, z) = \frac{kq}{(x^2 + y^2 + z^2)^{3/2}}(x, y, z)$$

Then the field can be expressed compactly as:

$$\vec{E}(x, y, z) = \frac{kq}{r^2}\hat{r} = \frac{kq}{r^3}(x, y, z).$$

where $\vec{r} = (x, y, z)$, $r = |\vec{r}| = \sqrt{x^2 + y^2 + z^2}$, and $\hat{r} = \frac{\vec{r}}{r}$.

$$\vec{E}(x, y, z) =$$

Let me know if you want the same expression in spherical coordinates or the field of a dipole or multipole.

A. Definitions

- $\nabla \stackrel{\text{def}}{=} \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$: the vector derivative operator (*nabla*).
- $\nabla \cdot \vec{F}(x, y, z)$: the *divergence* of the field $\vec{F}$ tells us if field $\vec{F}$ is acting as a "source" or a "sink" at the point (x, y, z).
- $\nabla \times \vec{F}(x, y, z)$: the *curl* of the field $\vec{F}$ tells us the "rotational tendency" of the vector field $\vec{F}$ at (x, y, z).

B. Path integrals

path integrals of vectors fields,

Scalar path integral.

$$\int_C f(\mathbf{r})\, dr \stackrel{\text{def}}{=} \int_{t_i}^{t_f} f(\mathbf{r})\, \|\mathbf{r}'(t)\|\, dt.$$

Here the curve $C \in \mathbb{R}^3$ is described by the parametrization $\mathbf{r} : \mathbb{R} \to \mathbb{R}^3$, which assigns a coordinate $\mathbf{r} = (x, y, z)$ for each value of the parameter (denoted t in the above). Note $dr \stackrel{\text{def}}{=} \|\mathbf{r}'(t)\|\, dt$, which involves computing the derivative of $\mathbf{r}(t)$ then computing the length.

Vector path integral.

$$\int_C \mathbf{F}(\mathbf{r}) \cdot d\mathbf{r} \stackrel{\text{def}}{=} \int_{t_i}^{t_f} \mathbf{F}(\mathbf{r}) \cdot \mathbf{r}'(t)\, dt.$$

This integral computes the total of the vector field $\mathbf{F}$ in the direction of the tangent line to the curve C describe by $\mathbf{r}(t)$. To obtain the component of $\mathbf{F}$ in the direction of the tangent line, we take the dot product with $d\mathbf{r} \stackrel{\text{def}}{=} \mathbf{r}'(t)dt$ during each step.

C. Surface integrals

flux integrals of vectors fields through surfaces,

Scalar surface integral.

$$\iint_S f(\mathbf{r})\, dS \stackrel{\text{def}}{=} \int_{v_i}^{v_f} \int_{u_i}^{u_f} f(\mathbf{r})\, \|\mathbf{r}'_u \times \mathbf{r}'_v\|\, dudv.$$

Here the surface $S \in \mathbb{R}^3$ is described by the parametrization $\mathbf{r} : \mathbb{R}^2 \to \mathbb{R}^3$, which assigns a coordinate $\mathbf{r} = (x, y, z)$ for each pair of the parameter u and v. Note $dS = \|\mathbf{r}'_u \times \mathbf{r}'_v\|\, dudv$, which involves computing the partial derivatives of $\mathbf{r}(u, v)$ with respect to the two parameters, taking the cross product, then computing the length.

Vector surface integral.

$$\iint_S \mathbf{F}(\mathbf{r}) \cdot d\mathbf{S} \stackrel{\text{def}}{=} \int_{v_i}^{v_f} \int_{u_i}^{u_f} \mathbf{F}(\mathbf{r}) \cdot (\mathbf{r}'_u \times \mathbf{r}'_v)\, dudv.$$

This integral computes the total *flux* of the vector field $\mathbf{F}$ flowing perpendicularly through the surface S. To obtain the component of $\mathbf{F}$ in the direction of the surface normal, we take the dot product with $d\mathbf{S} \stackrel{\text{def}}{=} \hat{\mathbf{n}}dS \stackrel{\text{def}}{=} (\mathbf{r}'_u \times \mathbf{r}'_v)dudv$, for each piece of the surface.

The main thing we'll have to learn is how to parametrize regions of space. In fact, we could even say that the main purpose of this course is to get you comfortable with parameterizations of curves, surfaces, and volumes. Once you have a parametrization for a region you can perform any integral calculation over this region.

D. Vector calculus theorems

The main results in vector calculus are two theorems: *Gauss' divergence theorem* and *Stokes theorem*. Both theorems can be understood as extensions of the fundamental theorem of calculus (FTC), which relates the integral of the differential of some quantity over a region R to the value of this quantity on the boundary of a region, denoted ∂R. In the case of the fundamental theorem of calculus, the region is the interval $I = [a, b] \subseteq \mathbb{R}$ whose boundary ∂I consists of the two points a and b. The fundamental theorem of calculus is

$$\int_a^b f'(x)\, dx = \int_I f'(x)\, dx = f_{\partial I} = f(b) - f(a),$$

Gauss' Divergence Theorem relates the volume integral of the quantity $\nabla \cdot \vec{F}$, which is called the divergence of $\vec{F}$, to the total flux of the vector field through the surface ∂V, which is the boundary of the volume V. Gauss' divergence theorem is:

$$\iiint_{\mathrm{V}} \nabla \cdot \vec{F} \; d\mathrm{V} = \iint_{\partial \mathrm{V}} \vec{F} \cdot d\vec{S}$$

Intuitively, the *divergence* of a vector field describes how much of the vector field emanates from a given point in space. The *flux* of a vector field over a surface S accounts for the strength of the vector field flowing through the surface. In the above example, we saw Gauss' divergence theorem applied to the electric field, but the vector field $\vec{F}$ could also represent thermal flows, or fluid flows.

Stokes' Theorem uses the "other" vector derivative $\nabla \times \vec{F}$, which is called the *curl* of $\vec{F}$. The curl of a vector field, denoted $(\nabla \times \vec{F})(x, y, z)$ describes the local rotational tendency of the vector field $\vec{F}$ at the point (x, y, z). Given any surface S in space, we can cut up the surface into tiny little rectangles and calculate the total surface area as a double integral $S = \int dS$. Stokes' theorem is the application of this "splitting up into little squares" idea and the fundamental theorem of calculus, which leads us to the following equation.

$$\iint_{\Sigma} \nabla \times \vec{F} \cdot d\vec{S} = \int_{\partial \Sigma} \vec{F} \cdot d\vec{r},$$

The surface integral of the curl $\nabla \times \vec{F}$ over any surface Σ is equal to the circulation of $\vec{F}$ along the boundary of the surface $\partial\Sigma$. Both the left and right sides of this equation correspond to scalar numbers. The left side is the vector surface integral of a vector quantity (the curl of $\vec{F}$), the right side corresponds to a vector path integral of a vector quantity over an oriented curve $\partial\Sigma$.

E. Applications of vector calculus

Vector calculus is the math machinery used for electricity and magnetism, which is the study electric field $\mathbf{E}(x, y, z)$, the magnetic field $\mathbf{B}(x, y, z)$, and the interactions between them.

Practice problems

This means learning calculus is all about getting practical experience calculating limits, derivatives, and integrals of functions, which is best achieved by solving lots of problems.

TODO: add exercises

TODO: link to notebook for solutions

Links

I hope this tutorial helped you see as practical and useful math that allows you to do calculations—just look at the name of the thing!

[*Essence of calculus* series by 3Blue1Brown]
`https://tinyurl.com/CALCess`

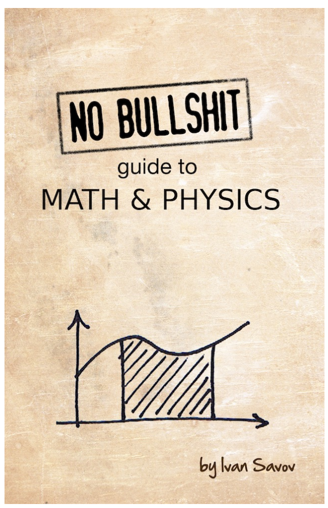
[*Calculus made simple* by Silvanus P. Thompson]
`https://gutenberg.org/ebooks/33283`

If you want to learn more about calculus, I invite you to check out my book, the **No bullshit guide to math and physics**.

This book contains short lessons on mechanics, differential and integral calculus written in a style that is jargon-free and to the point. This textbook covers both subjects in an integrated manner and aims to highlight the connections between them.

Contents:

- HIGH SCHOOL MATH
- VECTORS
- MECHANICS
- DIFFERENTIAL CALCULUS
- INTEGRAL CALCULUS
- SEQUENCES AND SERIES



5½[in] × 8½[in] × 528[pages]

For more information, see the book's website `minireference.com` or you can get in touch with me by email here `ivan@minireference.com`.