Calculus tutorial

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

Abstract—**This tutorial introduces the key ideas of calculus. We'll learn about limits, derivatives, integrals, sequences, and series. This material is normally taught in two separated university courses CALC1 and CALC2, but we choose to present all topics together to highlight the connections between them. The explanations in this tutorial combine words, math formulas, visualizations and Python code examples. Calculus techniques are used in businesses, engineering, and science. Visit the URL `bit.ly/calctut3` to follow along the calculations in this tutorial in an interactive computational environment.**

Contents

I. Introduction

Calculus is the study of functions and their properties. The two calculus techniques we'll learn in this tutorial are *derivatives*, which tell how functions *change* over time, and *integrals*, which compute the total *accumulation* of functions over time. 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" to start the download. Consider the function $f(t)$ that describes the amount of disk space taken by the partially-downloaded file. 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 download 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 will be done.

Let's look at how to calculate 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:

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

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 a time interval of duration 1[s].

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. Doing calculus: then and now

The key ideas of calculus were developed by Newton and Leibniz in the 17th century using pen and paper calculations. We're not in the 17th century anymore, and today we have computers at our disposal that are extremely good at certain calculus operations. Let's briefly outline the different ways of doing calculus to learn their use cases and benefits.

Symbolic calculations: The pen-and-paper approach continues to be the best way to learn calculus to this day. Writing math symbols on paper allows us to focus calculations at a high level of abstraction and arrive at exact symbolic answers. Manipulating math symbols gives you hands-on experience with calculus procedures and builds your intuitive understanding.

Symbolic calculations using SymPy: The Python library SymPy allows us to do symbolic math calculations on a computer. Using a computer algebra system like SymPy extends the reach of symbolic calculus operations to broader set of problems by doing some of the tedious steps for you and jumping straight to the answers. In this tutorial, we'll show how to use SymPy to check the answers we obtained using pen and paper calculations.

Numerical computing using NumPy and SciPy: Most practical applications of calculus don't use exact symbolic calculations, but instead work with numerical approximations. Engineers don't care about the exact value of square root of two $\sqrt{2}$, and instead represent $\sqrt{2}$ approximately the floating point number 1.4142135623730951 on a computer. This approximation is good enough for most engineering and scientific use cases. What we give up in mathematical exactitude, we gain manyfold in the form of computational power: modern computers can perform trillions (10^{12}) of floating point operations per second!

The Python libraries NumPy and SciPy make it easy to access all this computational power for doing calculus operations. The calculus answer we obtain using numerical computing are not exact like the answer we obtain using pen-and-paper or SymPy, but engineers are willing to accept that trade-off for the ability to perform any derivative, integral, or summations almost instantly.

C. Applications of calculus

We use calculus concepts to describe various quantities in physics, chemistry, biology, engineering, business, economics and other domain where quantitative analysis is used. Many laws of nature are expressed in terms of derivatives and integrals, so it's essential that you learn the language of calculus if you want to study science. In all this areas, the quantities of interest are described by functions and we use derivatives and integrals to do various useful calculations based on these functions. For example, derivatives are used to for optimization and to find function approximations. Integrals are used to solve differential equations and to compute probabilities in statistics and machine learning. The goal of this tutorial is to introduce you to the language and key ideas of calculus. This is the power of math: you learn some techniques for analyzing functions in general, then you're able to solve real-world problems in any domain.

II. Math prerequisites

Before we get started with the new calculus topics, let's do a quick review of key concepts from high school math. These are the basic building blocks I assume you've seen before.

A. Set notation

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

- $\{$ definition $\}$: we use curly brackets to define sets. The definition in the curly brackets is either a math description of the set's contents, or a list of elements in the set.
- $\mathbb{N}$: the set of natural numbers $\mathbb{N} \overset{\text{def}}{=} \{0, 1, 2, 3, 4, 5, \ldots\}$
- $\mathbb{N}_+$: the positive natural numbers $\mathbb{N}_+ \overset{\text{def}}{=} \{1, 2, 3, 4, 5, \ldots\}$.
- $\mathbb{R}$: the set of real numbers.
- $\mathbb{R}_+$: the set of nonnegative real numbers.
- $x \in S$: this statement is read "x is an element of S." We use this notation to indicate the "type" of the variable x. For example, writing "$x \in \mathbb{R}$" tells us x is a real number.

We use the math symbol $\overset{\text{def}}{=}$ to define new concepts. 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 is expressed more compactly as $\mathbb{R}_+ \overset{\text{def}}{=} \{x \in \mathbb{R} \mid x \geqslant 0\}$ 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 ∞.

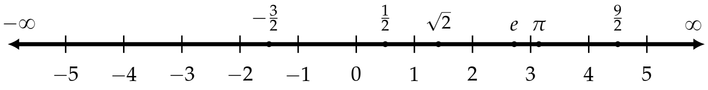


Figure 1. The real numbers $\mathbb{R}$ cover the entire number line.

The set of real numbers includes the natural numbers $\{0, 1, 2, 3, \ldots\}$, the integers $\{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$, rational numbers like $-\frac{3}{2}$, 0.5, and $\frac{9}{2}$, as well as irrational numbers like $\sqrt{2}$, e, and π. All the number you will run into when doing math can be visualized as a point on the number line.

Infinity: The math symbol ∞ describes the concept of *infinity*. We use the symbol ∞ to represent an infinitely large quantity, that is greater than any number you can think of. Geometrically speaking, we can imagine the number line extends to the right forever towards infinity, as illustrated in Figure 1. The number line also extends forever to the left, which we denote as negative infinity $-\infty$.

Infinity is not a number but a *process*. When we use the symbol $+\infty$, we're describing the process of moving to the right on the number line *forever*. We go past larger and larger positive numbers and never stop.

Infinity is a key concepts in calculus, so it's important that you develop the right way to think about infinitely large numbers, infinitely small distances, and procedures with infinite number of steps. We'll continue the discussion of infinity-related topics in Section III.

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 $f(x) \overset{\text{def}}{=} x^2$ takes any number x as input, squares it and divides the result by two to produce the output. For example, $f(3) = 3^2 = 9$.

In this tutorial, we'll often show code examples that mirror the math calculations. For example, here is the Python code that defines the function `f` and evaluates it for the input $x = 3$.

```
>>> def f(x):
        return x**2
>>> f(3)
9.0
```

Note 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 curve that passes through all input-output pairs of a function. Each input-output pair corresponds to a point $(x, f(x))$ in a Cartesian 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 2. The graph of the function f allows us to see at a glance the behaviour of the function for many inputs. Function graphs are an essential visualization tool for calculus calculations.

Let's see how we can use the Python modules `numpy` and `seaborn` to plot the graph of the function $f(x) \overset{\text{def}}{=} x^2$ that we defined in the previous code block. We start by importing the module `numpy` under the alias `np`. Next, we use the function `np.linspace` to create an array (a list of numbers) `xs` that contains 1000 input x-values that range between $x = -3$ and $x = 3$. We then evaluate the function for all inputs `xs` and store the outputs of the function in an array called `fxs`.

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

At this point, the arrays `xs` and `fxs` contain 1000 input-output coordinate pairs of the form $(x, f(x))$. To generate the graph of $f(x)$, we just need to trace a line passing through these coordinate pairs. We can do this by importing the `seaborn` module (alias `sns`) and calling the function `sns.lineplot`.

```
>>> import seaborn as sns
>>> sns.lineplot(x=xs, y=fxs)
See Figure 2 for the output.
```

We can use this combination of `np.linspace`, function evaluation, and `sns.lineplot` whenever we need to plot the graph of any function.

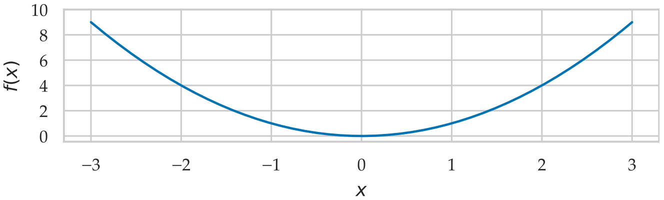


Figure 2. Graph of the function $f(x) = x^2$ from $x = -3$ until $x = +3$. The graph of the function f passes through the coordinates pairs $(x, f(x))$ for all x-values between $x = -3$ and $x = 3$.

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:

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

In Figure 3, 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.

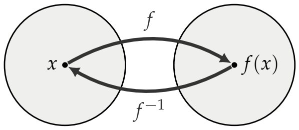


Figure 3. The inverse f^{-1} undoes the operation of the function f.

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

```
>>> from math import sqrt
>>> sqrt(9)
3
```

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* of the function. For example, the domain of the function $f(x) = x^2$ 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

Your function "vocabulary" determines which math expressions you'll be able to read and understand in the same way your English vocabulary determines determines which English sentences you'll be able to read and understand. Figure 4 shows the graphs of six important functions that are used in many areas of mathematical modelling.

Constant function: The constant function $f(x) \overset{\text{def}}{=} c$ produces the same output c for all inputs x.

Linear function: The linear function $f(x) \overset{\text{def}}{=} mx$ describes an input-output relationship where the output value $f(x)$ are *proportional* to the input value x, and the constant of proportionality is m. Geometrically, m is the slope in the graph of $f(x)$. Figure 4 show the graph of $f(x) \overset{\text{def}}{=} x$ which is the linear function with $m = 1$ for which the outputs $f(x)$ is equal to the input x. More generally, we can define the line $f(x) \overset{\text{def}}{=} mx + b$, where m describes the slope of the line, and b describes the value of the function when $x = 0$.

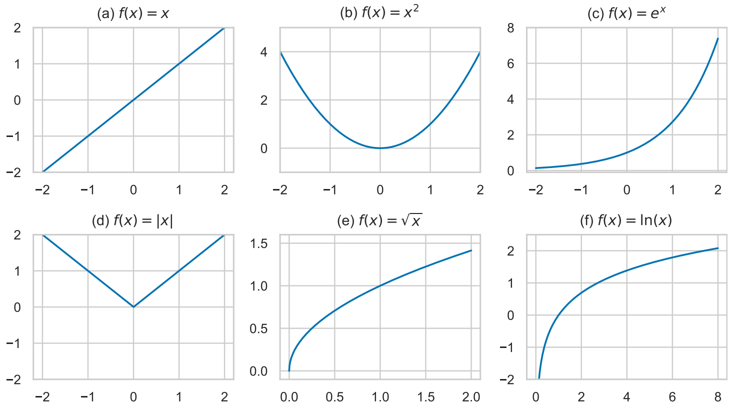


Figure 4. Graph of six math functions that you should know about.

Quadratic function: The quadratic function $f(x) \overset{\text{def}}{=} x^2$ calculates the square of the input x. The name "quadratic" comes from the Latin *quadratus* for square. Geometrically, x^2 is the area of a square with side length x. See Figure 4 (b) for the graph. The outputs the quadratic function are always nonnegative numbers since $x^2 \geqslant 0$, for all real numbers x.

Polynomial functions: We can combine the constant, linear, and quadratic functions to obtain the polynomial function $f(x) \overset{\text{def}}{=} a_2x^2 + a_1x + a_0$, where a_2, a_1, a_0 are arbitrary constants, which are called the *coefficients* of the polynomial. This is called a second degree polynomial, since the highest power of x it contains is x^2. The general equation for a polynomial function of degree n is

$$P_n(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots + a_nx^n.$$

Polynomials are a very useful family of functions.

Exponential function: The exponential function base $e = 2.7182818\ldots$ is defined as $f(x) \overset{\text{def}}{=} e^x = \exp(x)$. Figure 4 (c) shows the graph of the exponential function $f(x) = e^x$, which passes through the following points: $(-2, \frac{1}{e^2})$, $(-1, \frac{1}{e})$, $(0, 1)$, $(1, e)$, and $(2, e^2)$.

Absolute value function: The *absolute value* function tells us the size of numbers without paying attention to whether the number is positive or negative. We compute the absolute value of the number x by *forgetting* the sign of x. Geometrically, $|x|$ corresponds to its distance between x and the origin of the number line. We see the absolute values whenever we apply the combination of squaring followed by square-root on some number, $\sqrt{x^2} = |x|$, since squaring destroys the sign.

Square root function: The square root function is denoted $f(x) \overset{\text{def}}{=} \sqrt{x}$. The square root $\sqrt{x}$ is the inverse function of the square function x^2, when $x \geqslant 0$. The symbol $\sqrt{c}$ refers to the *positive* solution to the equation $x^2 = c$. Note that $-\sqrt{c}$ is also a solution of $x^2 = c$. Another notation for the square root function is $f(x) \overset{\text{def}}{=} x^{\frac{1}{2}}$, where the fractional exponent $\frac{1}{2}$ makes sense since if we square $x^{\frac{1}{2}}$, we get back to x: $\left(x^{\frac{1}{2}}\right)^2 = x^{\frac{2}{2}} = x^1 = x$. In addition to *square* root, there is also *cube* root $f(x) = \sqrt[3]{x} = x^{\frac{1}{3}}$, which is the inverse function for the cubic function $f(x) = x^3$. We have $\sqrt[3]{8} = 2$ since $2 \times 2 \times 2 = 8$.

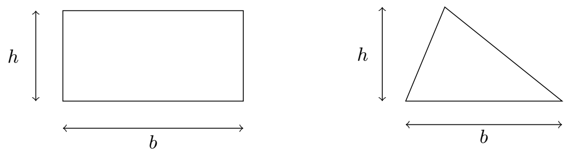
Logarithmic function: The natural logarithm function is denoted $f(x) \overset{\text{def}}{=} \ln(x) = \log_e(x)$. The function $\ln(x)$ is the inverse function of the exponential e^x. The graph of the function $\ln(x)$ passes through the following coordinates: $(\frac{1}{e^2}, -2)$, $(\frac{1}{e}, -1)$, $(1, 0)$, $(e, 1)$, $(e^2, 2)$, $(e^3, 3)$, $(e^4, 4)$, etc.

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 few values the sequence $[a_0, a_1, a_2, a_3, \ldots]$, which correspond to evaluating a_k for $k = 0$, $k = 1$, $k = 2$, $k = 3$, etc.

E. Geometry of rectangles and triangles

The area of a rectangle of base b and height h is $A = bh$, as illustrated in Figure 5 (a). The area of a triangle is equal to $\frac{1}{2}$ times the length of its base b times its height h: $A = \frac{1}{2}bh$, as shown in Figure 5 (b).



(a) $A_{\square} = bh$ (b) $A_{\triangle} = \frac{1}{2}bh$

Figure 5. Formulas for calculating the area of a rectangle and a triangle.

F. Trigonometric functions

The *unit circle* is a circle of radius one centred at the origin, as illustrated in Figure 6. The unit circle consists of all points (x, y) that satisfy the equation $x^2 + y^2 = 1$. A point on the unit circle has coordinates $(\cos\theta, \sin\theta)$, where θ is the angle the point makes with the x-axis.

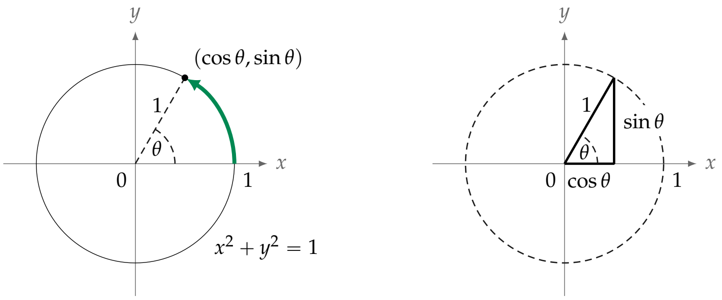


Figure 6. The coordinates of the point on the unit circle are $\cos\theta$ and $\sin\theta$.

In math, we use *radians* to measure angles instead of degrees °. One full circle is 360° or 2π radians. Some common angle measures are $30° = \frac{\pi}{6}$, $45° = \frac{\pi}{4}$, $60° = \frac{\pi}{3}$, and $90° = \frac{\pi}{2}$. The trigonometric functions sin, cos, and tan expect inputs in radians, so we often convert angles from degrees to radians.

Sine function: The graph of the sine function $f(\theta) \overset{\text{def}}{=} \sin(\theta)$ *oscillates* up and down and crosses the x-axis multiple times, as shown in Figure 7 (a). This graph corresponds to the vertical position of the point turning around on the unit circle, as illustrated in Figure 6 (a). We also use the sine function to find the y-component of unit length, as shown in Figure 6 (b).

Cosine function: The cosine function is the same as the sine function *shifted* by $\frac{\pi}{2}$ to the left: $f(\theta) = \cos(\theta) = \sin(\theta + \frac{\pi}{2})$, as shown in Figure 7 (b). The cosine function represents the horizontal position of the unit circle, and the x-component of unit length, as illustrated in Figure 6.

Tangent function: The tangent function is defined as the ratio of the sine and cosine functions: $f(\theta) = \tan(\theta) \overset{\text{def}}{=} \frac{\sin(\theta)}{\cos(\theta)}$. The graph of the tangent functionis shown in Figure 7 (c).

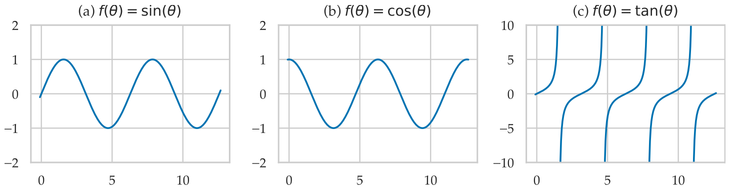


Figure 7. Graphs of the trigonometric functions $\sin(\theta)$, $\cos(\theta)$, and $\tan(\theta)$.

We can use the trigonometric functions $\sin(\theta)$, $\cos(\theta)$, and $\tan(\theta)$ to calculate the *components* of vectors. Sines and cosines are also describe waves and periodic motion in physics.

Overview and a look ahead

A look ahead

Figure

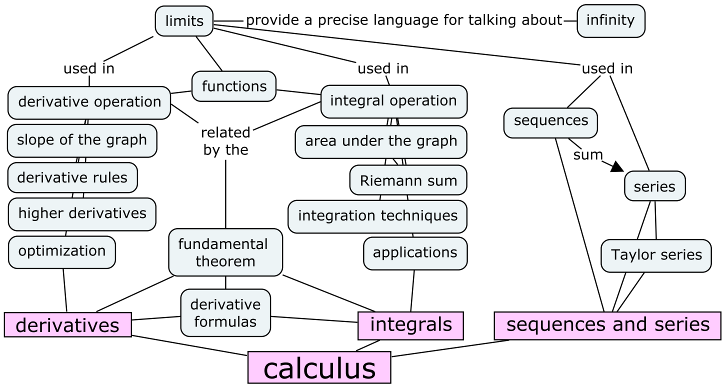


Figure 8. The calculus concepts and topics you'll learn in this tutorial.

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 used to denote limit expressions. Here are some examples:

- $\lim\limits_{x\to\infty} f(x)$: 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."
- $\lim\limits_{\delta\to 0} g(a+\delta)$: limit expression that describes the value of the function g as the input approaches a from the right. The number δ (the Greek letter *delta*) usually describes a small distance, and the limit as delta goes to zero, $\lim_{\delta\to 0} g(a+\delta)$ describes the behaviour of the function g for infinitely close to the value a.
- $\lim\limits_{n\to\infty} \mathrm{proc}(n)$: limit expression that describes the outcome of some computational procedure with n steps, as the number of steps n goes to infinity.

Let's look at an example of a math procedure with infinite number of steps.

A. Example: 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.

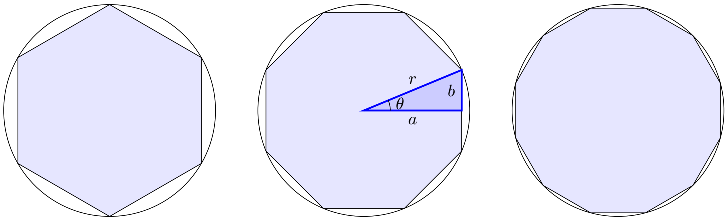
The definition of the number e as a limit is a fascinating new concept that goes beyond the "regular" math operations that we learn in high school math. We're not talking about any particular large number n when calculating the expression $(1+\frac{1}{n})^n$, but the *process* of plugging in large and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

Example: 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.

B. Example: the area of a circle

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

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

We can compute the area of the n-sided regular polygons by splitting them up into triangular slices, and computing the area of these slices using the formula for the area of a triangle $\frac{1}{2}bh$. Figure 9 (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 $b = r\sin(\frac{\pi}{8})$. The total area of the polygon with n sides is:

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

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 with polygons with more and more 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=2.5980762113533156, 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
```

To calculate the limit as n goes to infinity, we can perform the symbolic math calculation using SymPy:

```
>>> 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
```

The result is πr^2, which is the formula for the area of a circle.

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

C. 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 growing to ∞?

The function $f(x)$ is said to *converge* to 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 L." See Figure 10 for an illustration.

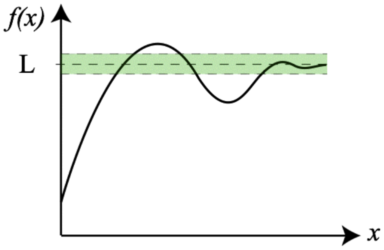


Figure 10. A function $f(x)$ that oscillates up and down initially, but it "settles down" close to the value L for large values of x.

Example 1: Consider the limit of the function $f(x) = \frac{1}{x}$ as x goes to infinity, as illustrated in Figure 11 (a):

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

The function $\frac{1}{x}$ never *actually* reaches zero, so it would be wrong to write $f(x) = 0$ for any $x \in \mathbb{R}$. However, the expression $\frac{1}{x}$ gets closer and closer to 0 as x goes to infinity. Limits are useful because they allow us describe this tendency as $\lim\limits_{x\to\infty} f(x) = 0$.

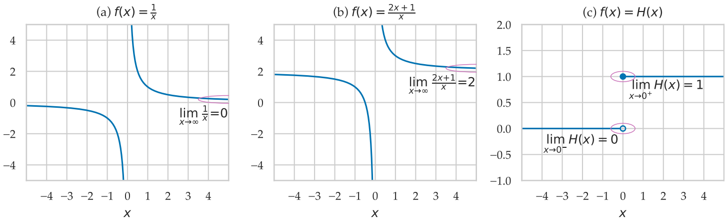

(a) $f(x) = \frac{1}{x}$ (b) $f(x) = \frac{2x+1}{x}$ (c) $f(x) = H(x)$

Figure 11. Visual representation of the limit calculations for three functions.

D. Limit formulas

The limit of the sum, difference, product, and quotient of two functions are can be computed as follows:

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

In words, these formulas tell us we can bring the limit calculations "inside" basic arithmetic operations.

Example 2: Calculate $\lim\limits_{x\to\infty} \frac{2x+1}{x}$. We're given the function $f(x) = \frac{2x+1}{x}$ and must determine what the function looks like for very large values of x. We can rewrite the function as $\frac{2x+1}{x} = 2 + \frac{1}{x}$ then apply the sum formula for limits:

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

As the denominator x becomes larger and larger, the fraction $\frac{1}{x}$ becomes smaller and smaller, so $\lim\limits_{x\to\infty} \frac{1}{x} = 0$, so the second term goes zero, leaving only the 2. See Figure 11 (b) for an illustration.

E. 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.

F. Continuity

If the function $f(x)$ obeys, $f(a) = L$ and $\lim_{x\to a} f(x) = L$, we say the function $f(x)$ is *continuous* at $x = a$. Geometrically, the graph of the function at $x = a$ is "smooth" and doesn't make a sudden jump at $x = a$. Intuitively, when a function is continuous, we can draw its graph using a single stroke without lifting the pen. In contrast, functions that blow up to infinity or make sudden jumps are not continuous.

Example 3: Consider the *Heaviside step function*, which is defined as follows:

$$H(x) \overset{\text{def}}{=} \begin{cases} 1, & \text{if } x \geqslant 0 \\ 0, & \text{if } x < 0. \end{cases}$$

The Heaviside function is zero for negative values of x, then suddenly jumps to one at $x = 0$, as shown in Figure 11 (c). The limit as x approaches $x = 0$ from the left is $\lim_{x\to 0^-} H(x) = 0$. The limit as x approaches $x = 0$ from the right is $\lim_{x\to 0^+} H(x) = 1$. The two limits are different, $\lim_{x\to 0^-} H(x) = 0 \neq 1 = \lim_{x\to 0^+} H(x)$, so the function is discontinuous at $x = 0$.

G. Computing limits using SymPy

The SymPy function `sp.limit(expr,var,value)` allows us to compute the limit of the expression `expr`, as the variable `var` approaches `value`. Let's use SymPy to revisit the limit calculations we saw above.

Recall the limit expression for Euler's number $e \overset{\text{def}}{=} \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n$. To compute this limit using SymPy, we start by importing the `sympy` module under the alias `sp`, defining the symbolic variable `n` $= n$, then calling `sp.limit`.

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

SymPy produced the exact value `E` $= e = 2.718281828\ldots$. Note we use the symbol `sp.oo` (two lowercase os) to denote infinity, which is a clever name since it resembles the infinity symbol ∞.

Let's now look at the code for computing the limit $\lim_{x\to\infty}\frac{1}{x}$. We define the symbol `x` then call the function `sp.limit` to evaluate the limit of the expression `1/x` $= \frac{1}{x}$ as `x` $= x$ goes to infinity `sp.oo` $= \infty$.

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

Here is another example, that computes the limit of the fraction $\frac{2x+1}{x}$ as x goes to infinity, which is illustrated in Figure 11 (b).

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

To calculate the limit form the left and the right of a number, we must provide a fourth argument `"-"` or `"+"`. The following SymPy calculations confirm the limits of the Heaviside step function when approaching $x = 0$ from the left and from the right.

```
>>> from sympy import Heaviside
>>> sp.limit(Heaviside(x,1), x, 0, "-")
0
>>> sp.limit(Heaviside(x,1), x, 0, "+")
1
```

See Figure 11 (c) for an illustration.

H. 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 derivatives in Section IV.

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

$$A_f(a,b) \approx \underbrace{\Delta x f(a+\Delta x) + \Delta x f(a+2\Delta x) + \Delta x f(b)}_{n \text{ terms}}.$$

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

$$\int_a^b f(x)\,dx = A_f(a,b) = \lim_{n\to\infty}\Delta x\left[f(a+\Delta x) + f(a+2\Delta x) + \cdots + f(b)\right].$$

We'll have a lot more to say about integrals in Section V.

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

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

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

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

We'll learn more about sequences and series in Section VI.

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 constant function doesn't change. Geometrically, the derivative function describes the *slope* of the graph of the function $f(x)$. The derivative of the 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 the notation $f'(x)$ for describing "the slope of the function f at x."

Consider the rise-over-run formula for calculating the slope of a function applied to the points $(x, f(x))$ and $(x + \delta, f(x + \delta))$ on the graph of the function. Figure REF shows the slopes calculated when $\delta = 0.1$, $\delta = 0.01$, and $\delta = 0.001$.

The average slope between ... and ... is

TODO: figure

but what about the instantaneous slope at one instant?

The derivative function $f'(x)$ is defined as the following limit expression:

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

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 tells us the slope of the graph of the function $f(x)$ for all values of x. Figure 12 shows the slope calculation for the function $f(x) = x^2$ at two different locations: at $x = -0.5$ and at $x = 1$.

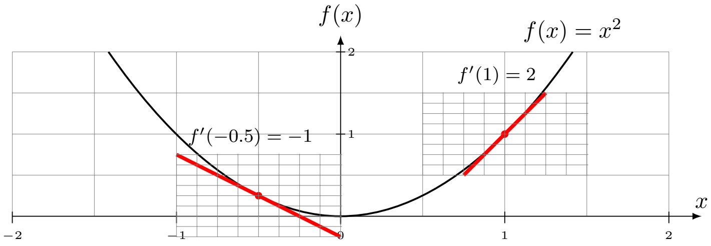


Figure 12. The derivative of the function at $x = a$ is denoted $f'(a)$ and describes the slope function at that point.

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 slope property of the original function $f(x)$.

More generally, we can define the *derivative operation*, denoted $\frac{d}{dx}$[<f>], which takes as input a function $f(x)$ and produces as output the derivative function $f'(x)$. Applying the derivative operation to the function is also called "taking the derivative" of a function. For example, the derivative of the function $f(x) = x^2$ is the function $f'(x) = 2x$. We can also describe this relationship as $(x^2)' = 2x$ or as $\frac{d}{dx}(x^2) = 2x$. Look at the graph in Figure 12 to convince yourself that the slope of $f(x) = x^2$ is indeed described by $f'(x) = 2x$ for any x. For example, when $x = 0$, we see the graph has zero slope and the derivative gives us the same thing: $f'(0) = 0$.

A. Numerical derivative calculations

Here is the Python code 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 function `differentiate` calculates the derivative using a finite step `delta` $= 10^{-9}$ instead of the infinitely small step δ in the math definition of the derivative. This means, the value returned by `differentiate` will be an approximation to the true derivative.

Let's now define a Python function `f` that corresponds to the math function $f(x) = x^2$ and use `differentiate` to find the slope of f when $x = 1$:

```
>>> def f(x):
        return x**2
>>> differentiate(f, 1)
2.000000165480742
```

Using the numerical method, we obtain the approximation $f'(1) = 2.000000165480742$, which is not perfect, but pretty close to the true value $f'(1) = 2$. For most practical applications, this numerical approximation is good enough.

B. Derivative formulas

You don't need to apply the complicated derivative formula $f'(x) \overset{\text{def}}{=} \lim_{\delta \to 0} \frac{f(x+\delta) - f(x)}{\delta}$ 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 later calculations.

Table I shows the derivatives several functions. I invite you to mentally bookmark this page so you can come back to it when you need to know the derivatives of some function.

Table I presents the results in terms of the derivative operator $\frac{d}{dx}$[<f>], which takes as input some function $f(x)$ and produces as output its derivative function $f'(x)$.

C. Derivative rules

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

Constant multiple rule: The derivative of k times the function $f(x)$ is equal to k times the derivative of $f(x)$:

$$\big[k f(x)\big]' = k f'(x).$$

Sum rule: The derivative of the sum of two functions is the sum of their derivatives:

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

TABLE I
DERIVATIVE FORMULAS FOR COMMONLY USED FUNCTIONS

$f(x)$	$-$ derivative $\rightarrow$	$f'(x)$
a	$-\frac{d}{dx}\rightarrow$	0
x	$-\frac{d}{dx}\rightarrow$	1
$mx+b$	$-\frac{d}{dx}\rightarrow$	m
x^n, for $n \neq 0$	$-\frac{d}{dx}\rightarrow$	nx^{n-1}
$\frac{1}{x} = x^{-1}$	$-\frac{d}{dx}\rightarrow$	$\frac{-1}{x^2} = -x^{-2}$
$\sqrt{x} = x^{\frac{1}{2}}$	$-\frac{d}{dx}\rightarrow$	$\frac{1}{2\sqrt{x}} = \frac{1}{2}x^{-\frac{1}{2}}$
e^x	$-\frac{d}{dx}\rightarrow$	e^x
$\ln(x)$	$-\frac{d}{dx}\rightarrow$	$\frac{1}{x}$
$\sin(x)$	$-\frac{d}{dx}\rightarrow$	$\cos(x)$
$\cos(x)$	$-\frac{d}{dx}\rightarrow$	$-\sin(x)$

Product rule: The derivative of a product of two functions is the sum of two contributions:

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

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

Quotient rule: This formula 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, we leave the inner function $g(x)$ alone and focus on taking the derivative of the outer function $f(x)$. This step gives us $f'(g(x))$, which is the value of f' evaluated at $g(x)$. In the second step, we multiply this expression by the derivative of the *inner* function $g'(x)$.

D. Higher derivatives

The second derivative of $f(x)$ is denoted $f''(x)$ or $\frac{d^2f}{dx^2}$. It is obtained by applying the derivative operation to $f(x)$ *twice*: $\frac{d}{dx}\left[\frac{d}{dx}[\texttt{<f>}]\right]$. Geometrically, the second derivative $f''(x)$ tells us the *curvature* of $f(x)$. Positive curvature means the function opens upward and looks like the bottom of a valley. The function $f(x) = x^2$ shown in Figure 12 has derivative $f'(x) = 2x$ and second derivative $f''(x) = 2$, 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 the derivative formulas from Table I and the derivative rules from the previous section, 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 using SymPy

The SymPy function `sp.diff` computes the derivative of any expression. For example, here is how to 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 the only function that is equal to its derivative:

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

Let's use SymPy to check the derivative calculations from the 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.

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 $f(x) = x^2$ at $x_0 = 1$ is $T_1(x) = f(1) + f'(1)(x-1) = 1 + 2(x-1) = 2x - 1$. Look at the right side of Figure 12 for an illustration of this tangent line.

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 a fancier approximation $T_n(x)$ that matches the second, third, and higher derivatives of $f(x)$.

H. Optimization

Derivatives are used to solve optimization problems, which consist of finding the maximum or minimum value of some function $f(x)$. For example, look the graph of the function $f(x) = x^2$ in Figure 12. The minimum of the function occurs when $x = 0$. This is the bottom of a valley.

Observations: 1. The slope of the function is zero $f'(0) = 0$. 2. The second derivative of is positive at that point $f''(0) > 0$, which tells us the function looks locally like bottom of a valley. We'll use these observations to come up with a mathematical (analytical) optimization algorithm: a strategy for finding the minimum(s) and the maximum(s) of a function.

Analytical optimization: The values of x where the derivative is zero are called the *critical points* of the function and denoted x_1^*, x_2^*, etc. Optimum values (maximum or minimum) occurs at a critical point of the function. We identify a critical point x_j^*, that corresponds to minimum if the second derivative is positive at that point $f''(x_j^*) > 0$ (positive curvature). In contrast, a critical point x_k^* here $f''(x_k^*) < 0$ (negative curvature) is a maximum. These observations lead us to the following procedure for finding minima and maxima of the function $f(x)$:

(1) Solve $f'(x) = 0$ to find the critical points $[x_1^*, x_2^*, x_3^*, \ldots]$.
(2) For each critical point x_i^*, check to see if it is a maximum or a minimum by evaluating $f''(x_i^*)$:
- If $f''(x_i^*) < 0$ then x_i^* is a max (mountain top)
- If $f''(x_i^*) > 0$ then x_i^* 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 on the left and the right of the critical point. If $f'(x^* - 0.01)$ is negative and $f'(x^* + 0.01)$ is positive, the point x^* is a minimum (like near $x* = 0$ in Figure 12). If $f'(x^* - 0.01)$ is positive and $f'(x^* + 0.01)$ is negative, then the point x^* is a maximum. If $f'(x^* - 0.01)$ and $f'(x^* + 0.01)$ have the same sign, the value x^* is a *saddle point*, which is neither a minimum or a maximum.

Example: Let's use the analytical optimization 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 $x_1^* = \frac{1}{3}$ is a maximum. For $x_2^* = 1$, we find $f''(1) = 2$, which tells us $x_2^* = 1$ is a minimum.

Numerical optimization: Consider the shape of the function near a minimum value. The function is decreasing on the left of the minimum, and the increasing on the right of the minimum. This means we can start at any point $x = x_0$ and take "downhill" steps to get to the minimum value. This simple procedure that repeatedly takes steps in the direction where the function is decreasing is a powerful algorithm 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.

Let's use this procedure to find the minimum of the function $q(x) = (x-5)^2$ using the initial initial value `x0` $= 10$ as the starting point of the gradient descent algorithm.

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

The `while` loop inside the `gradient_descent` function ran several times, and in each iteration takes a small "downhill" step until we get close to the minimum (the bottom of the valley). The procedure reaches the point $x = 5.000000000396651$, which is close to true minimum of the function $q(x)$.

Numerical optimization using SciPy: The module `scipy.optimize` provides a high-performance numerical optimization procedure called `minimize` that runs much faster than the function `gradient_descent` that we defined above. Here is a quick example that shows how to use `minimize` to find the minimum of the function $q(x) = (x-5)^2$.

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

Once more, the value we obtain 4.9999999737 is very close to the true minimum of the function $x^* = 5$.

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 are some kind of summation.

A. Act 1: Integrals as area calculations

Figure 13 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.

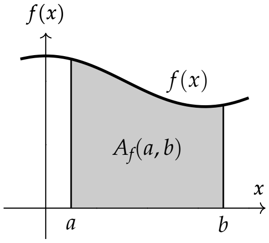


Figure 13. The integral of the function $f(x)$ between $x = a$ and $x = b$ corresponds to the area of the shaded region $A_f(a,b) = \int_a^b f(x)\,dx$.

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, 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 region with zero width 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 because the region 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 14.

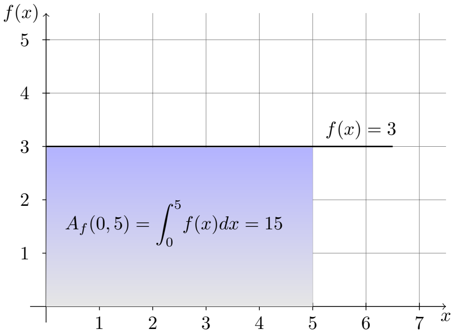


Figure 14. The area of a rectangle of height 3 and width 5 equals 15.

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 15. 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.

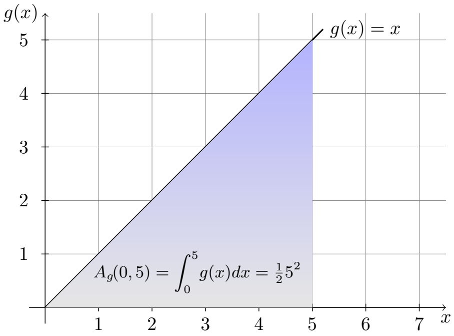


Figure 15. The area of a triangle with base 5 and height 5 is $\frac{1}{2}5^2 = \frac{25}{2} = 12.5$.

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 $h(x)$ between $x = 0$ and $x = 2$, as illustrated in Figure 16. We need to calculate the following integral:

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

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

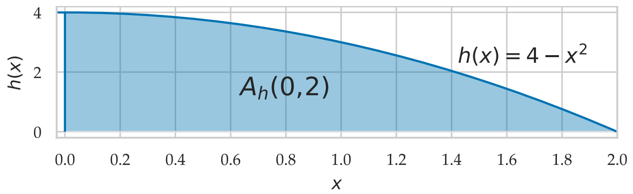


Figure 16. The area under the graph of $h(x)$ between $x = 0$ and $x = 2$.

One way to approximate the area under the graph of $h(x)$ is to compute the *Riemann sum* approximation, which splits the region into bunch of vertical rectangular strips of fixed width. The height of each rectangular strip varies depending on $h(x)$. Look ahead to figures 17 and 18 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 $A_h(0,2) \approx 5.329$, and $n = 1\,000\,000$ rectangles produces $A_h(0,2) \approx 5.333329$. 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 the split-area-into-rectangles calculation (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 into many (think 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 17 (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 approximation to 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 17 (a), we can clearly see that the area computed in this way is an underestimate to the true area under the graph. Let's trust the process, and continue with the calculations, knowing that the "quality" of the approximations will improve when we split the region into thinner and thinner strips.

The procedure we used for $n = 10$ works more generally for any n. In the general case, the rectangles have width $\Delta x = \frac{b-a}{n} = \frac{2}{n}$, which gets smaller and smaller as n grows. The x-coordinates of the right endpoints of the n rectangles are located at $[\Delta x, 2\Delta x, 3\Delta x, \ldots, (n-1)\Delta x, n\Delta x]$. The heights of the rectangles are $[h(\Delta x), h(2\Delta x), h(3\Delta x), \ldots, h((n-1)\Delta x), h(n\Delta x)]$. To find the area under the graph of $h(x)$, we 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 $h(k\Delta x)\Delta x$, and k goes from 1 to n. Mathematicians use the symbol $\sum$ (the capital Greek letter *sigma*) to describe long summations. 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 known as the *Riemann sum* formula for computing areas. We'll now 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` $= [a + \Delta x, a + 2\Delta x, a + 3\Delta x, \ldots, n\Delta x]$, and evaluate the function `f` at these x-values to obtain `fxs` $= [f(a + \Delta x), f(a + 2\Delta x), f(a + 3\Delta x), \ldots, f(n\Delta x)]$. 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:

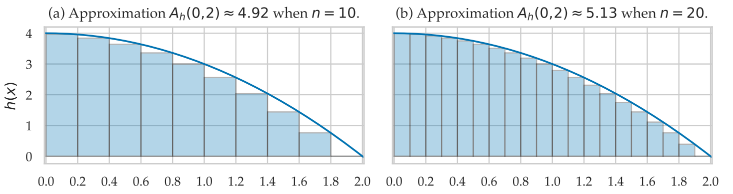
```
>>> def h(x):
        return 4 - x**2
```

To calculate the $n = 10$ approximation to the 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
```

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

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



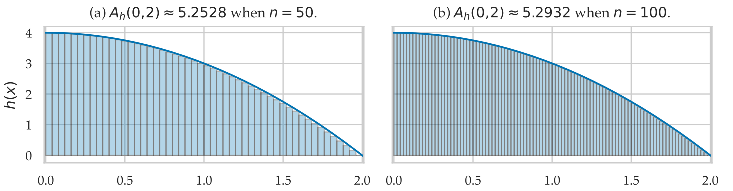
(a) Approximation $A_h(0,2) \approx 4.92$ when $n = 10$.

(b) Approximation $A_h(0,2) \approx 5.13$ when $n = 20$.

Figure 17. 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
```



(a) $A_h(0,2) \approx 5.2528$ when $n = 50$.

(b) $A_h(0,2) \approx 5.2932$ when $n = 100$.

Figure 18. 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 $\frac{16}{3} = 5\frac{1}{3} = 5.\overline{3} = 5.3333333333\ldots$. 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 graph of $f(x)$ 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 of the Riemann sum:

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

In words, the integral a Riemann sum that consists of infinitely thin rectangular strips. We previously defined the integral $\int_a^b f(x)\,dx$ geometrically as the area under the graph of $f(x)$, but now you know the formal math definition for the 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 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 with an infinitely short run δ. These two procedures are the foundations of calculus. The limits $\lim_{n\to\infty}$ and $\lim_{\delta\to 0}$ are essential tools that allow us to perform these calculus 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 of the letter used to denote 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, $F_a(b) \stackrel{\text{def}}{=} \int_a^b f(x)\,dx$. 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 under $f(x)$ between $x = a$ and $x = b$ as the *change* in the integral function:

$$A_f(a,b) = \int_a^b f(x)\,dx = F_0(b) - F_0(a).$$

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 19.

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,

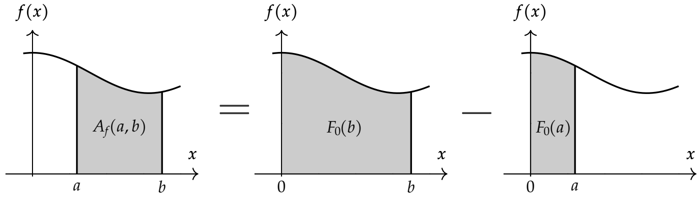


Figure 19. The area under $f(x)$ between $x = a$ and $x = b$ is computed using the formula $A_f(a,b) = F_0(b) - F_0(a)$, which describes the change in the output of $F_0(x)$ between $x = a$ and $x = b$.

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 integral function corresponds to the area of a rectangle of height 3 and with width b, as shown in Figure 20.

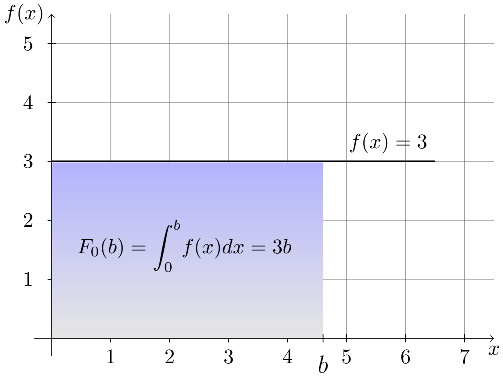


Figure 20. The integral function of the function $f(x) = 3$ is $F_0(b) = 3b$.

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:

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

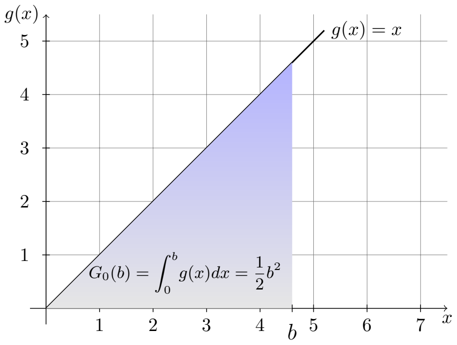


Figure 21. The integral function of the function $g(x) = x$ is $G_0(b) = \frac{1}{2}b^2$.

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 $F_0(b)$ and $G_0(b)$ thanks to the simple geometries of the areas under the graphs, but $h(x)$ is a curve so it requires some new integration methods. In the next few pages, we'll learn about symbolic integration techniques that will allow us to find the integral function $H_0(b)$.

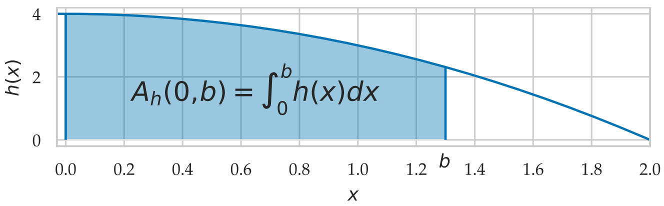


Figure 22. The integral of the function $h(x) = 4 - x^2$ from $x = 0$ to $x = b$.

F. Intermission

The integral in Example 3 was chosen to motivate the need for more advanced methods for integration. Is there a math formula that describes the area $A_h(0,b)$ of the region shown in Figure 22? We previously used numerical methods to compute the particular area $A_h(0,2)$ for $b = 2$, but now we're looking for a general math formula that computes the integral function $H_0(b) = A_h(0,b)$ for any b. In the next section, we'll learn about the *fundamental theorem of calculus*, which will allow us to find the exact formula for $H_0(b)$.

The section is like the intermission in the calculus show. As in a real-world intermission, this is also your chance to skip the rest of the show. Perhaps you have better things to do right now than learning about advanced calculus concepts. I won't get offended—no worries! Feel free to skip ahead to Section V-I for more technical details about high-performance numerical integration, or jump straight to Section VI to learn about sequences and series.

As a teacher, I'm happy that you know that integrals compute areas under the graph of functions, and can be approximated numerically using Riemann sums. These are the two key ideas related to integrals, 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 calculus teacher in college described the realization you get after understanding the fundamental theorem of calculus as similar to the feeling you get when smoking some of that "funny stuff."

Still with me? Okay, with your consent, let's continue with the calculus show.

G. Act 3: Fundamental theorem of calculus

Note the pattern in the formulas for the integral functions $F_0(b)$ and $G_0(b)$. 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 is there some fundamental math pattern we could follow to "guess" integral functions?

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:

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

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

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

To understand the inverse relationship between integration and differentiation, we can draw an analogy with the relationship between a function f and its inverse function f^{-1}, which *undoes* the effects of f. See Figure 3 on page 4. Given some initial value x, if we apply the function f to obtain the number $f(x)$, then apply the inverse function f^{-1} on the number $f(x)$, we get back to 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:

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

Let's use SymPy to verify that the fundamental theorem of calculus is true. We'll start with the function `f` $= f(x) = x^2$, compute its integral function `F` using `sp.integrate`, then take the derivative of `F` using `sp.diff`.

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

We see that sequence `diff(integrate(`$f(x)$`))` $= \frac{d}{dx}\int_0^x f(u)\,du$ bring us back to the original $f(x)$ we started from.

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 its 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):

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

Let's use SymPy to verify this formula. We'll start with the function `G` $= G(x) = x^3$, commute its derivative `dGdx` $= G'(x)$ using `sp.diff`, then use `sp.integrate` to compute the integral function of $G'(x)$.

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

We see the operations `integrate(diff(`$G(x)$`))` $= \int_0^x G'(u)\,du$ bring us back to the original $G(x)$ we started from.

Anti-derivative functions: 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... To find an integral function of $f(x)$, we can look for a function $F(x)$ whose derivative is the the function $f(x)$.

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 Table I on page 10). If we can find a function $F(x)$ such that $F'(x) = f(x)$, then we know $F(x)$ is an integral function of $f(x)$ and we can compute the integral using:

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

Computing integral functions using anti-derivatives: Recall that all integral functions differ by a constant $+C$, which is related to the starting point of the integral function:

$$F_a(b) = \int_a^b f(x)\,dx = F_0 + C.$$

This means $F(x) + C = F_0(b)$ for some $+C$.

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 powerful. We can use it for all the function listed in the table of derivative formulas (see page 10). For example, the table tells us that the derivative of the linear function $f(x) = mx + b$ is the constant function $f'(x) = m$. This 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 Table I.

```
>>> 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 formulas.

Okay, but what do we do if the function we want to integrate doesn't appear in Table I?

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 in this act is some of the most boring stuff. If you're taking a CALC II class, 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, but we'll show the two most important tricks.

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 change the limits integration (form a and b to u_a and u_b), and also change the "step" parameter (from dx to du).

Follow these three steps to apply the substitution trick:

1) Replace all occurrences of $u(x)$ with u.
2) Replace dx with $\frac{1}{u'(x)}du$.
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 formula, 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 $dx = 2\sqrt{x}\,du = 2u\,du$. The new limits of integration are $u_a = \sqrt{a}$ and $u_b = \sqrt{b}$.

Performing the three steps of the substitution trick gives us:

$$\int_{x=a}^{x=b} \frac{1}{x-\sqrt{x}}\,dx \;=\; \int_{u(a)}^{u(b)} \frac{1}{u^2-u}\,2u\,du.$$

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.

We can now simplify and solve the this integral 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.

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

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: The integration by parts trick can be used when the function we're integrating looks like $f(x)g'(x)$.

TODO: Explain

$$\int_a^b f(x)g'(x)\,dx \;=\; \big[f(b)g(b) - f(a)g(a)\big] \;-\; \int_a^b f'(x)g(x)\,dx.$$

Example: Let's calculate $\int_0^5 xe^x\,dx$ using the integration by parts procedure. The expression xe^x consists of two factors: x and e^x. We'll identify x with $f(x)$ and e^x as $g'(x)$. This means $f'(x) = 1$

and $g(x) = \int g'(x)\,dx = e^x$. We now know all the parts we need to apply the integration by parts formula:

$$\begin{aligned}\int_0^5 \underbrace{x}_{f(x)}\,\underbrace{e^x}_{g'(x)}\,dx &= \big[f(5)g(5) - f(0)g(0)\big] - \int_0^5 f'(x)g(x)\,dx \\ &= \big[5e^5 \qquad - 0e^0\big] \qquad - \int_0^5 1\cdot e^x\,dx \\ &= \big[5e^5 - 0e^0\big] - \big[e^5 - e^0\big] \\ &= 5e^5 - e^5 + 1 = 4e^5 + 1.\end{aligned}$$

Other tricks: Substitution and integration by parts are only small sample of the multitude of integration techniques. 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.

I. Computing integrals using SciPy

The Python function `integrate` that we showed in Section V-C is a useful teaching tool, but it would be much too slow to use for practical scientific computing tasks. The function `quad` from the `scipy.integrate` module is a much more powerful tool for computing numerical integrals. The name `quad` is short for "quadrature," which a historical name for the find-the-area-under-the-graph-of-a-function calculations.

Let's 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.

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

We can similarly use `quad` to calculate the integrals $\int_0^5 g(x)\,dx$ and $\int_0^2 h(x)\,dx$ from 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`:

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

We'll use these symbols to express the functions and the limits of integration.

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 to specify the variable of integration and the limits of integration a and b.

```
>>> 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 formula that works for all possible limits of integration a and b. Geometrically, we recognize the result `3*(b-a)` as the height-times-width formula for the area of a rectangle, which we have seen several times already.

To compute the definite integral $\int_0^5 f(x)\,dx$, 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 14, and the approximation we obtained using numerical integration `quad(f,0,5)[0]`.

We can also compute the integral function $F_0(b)$, which is defined as $F_0(b) \overset{\text{def}}{=} \int_0^b f(x)dx$, for the function `fx` $= f(x) = 3$.

```
>>> F0b = sp.integrate(fx, (x,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 20 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 use 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
```

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))
>>> 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 21.

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: We start by defining a SymPy expression that corresponds to the function $h(x) = 4 - x^2$.

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

We can now make SymPy compute the integral function $H_0(b) = \int_0^b h(x)\,dx$ by calling `sp.integrate`:

```
>>> H0b = sp.integrate(hx, (x,0,b))
>>> H0b
4*b - b**3/3
```

The integral function $H_0(b) = 4b - \frac{1}{3}b^3$ corresponds to the area calculation under $h(x) = 4 - x^2$ 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 describe the equations of motion $x(t)$, $v(t)$, and $a(t)$, which correspond to the object's *position*, *velocity*, and *acceleration* at 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, which we can summarize as follows:

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

The starting point of kinematics is Newton's second law, which tells us that the acceleration of an 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:

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

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. Consider an object that experiences a constant acceleration $a(t) = a$. We can use integration to find the velocity of this object at a later time $t = \tau$:

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

Knowing the velocity as a function of time $v(t)$, we can use integration a second time to find its position at time τ:

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

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 know where it comes from: the integration operation applied to the acceleration function $a(t) = a$ and the 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.

Here are some examples of differential equations and their solutions;

- The kinematics equations when the acceleration is constant come front he differential equation $x''(t) = a$. We use integration (twice) to find the unknown function $x(t) = x_i + v_it + \frac{1}{2}at^2$. We can verify that $x(t)$ is a solution to the differential equation $x''(t) = a$ by computing the the second derivative of $x(t)$.
- 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 bacterial growth rate $b'(t)$ is proportional to the number of existing bacteria. The solution to this equation is $b(t) = b_0e^{\lambda t}$, where b_0 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 $r(t) = r_0e^{-\lambda t}$.
- Simple harmonic motion is described by the differential equation $x''(t) + \omega^2 x(t) = 0$, which has solution $x(t) = c_1\sin(\omega t) + c_2\cos(\omega t)$, for some constants c_1 and c_2.

If you take a course on differential equations, you'll learn all kind of tricks and techniques for solving differential equations. Integration plays a key role in all these techniques, since it allows us to "undo" 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: $\Pr(\{a \leqslant X \leqslant b\}) \overset{\text{def}}{=} \int_a^b f_X(x)\,dx$. For example, the standard normal random variable Z is described by the probability density function $f_Z(z) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}z^2}$. To calculate the probability of the event $\{-1 \leqslant Z \leqslant 1\}$, we 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 natural 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. Sequences and series are the third pillar of the basic calculus knowledge that I want you to have because they are powerful computational tools that allow us to describe procedures with infinite number of steps.

A. Sequences are functions with discrete inputs

We use the notation $f : \mathbb{R} \to \mathbb{R}$ to describe 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 for positive natural numbers $k \in \mathbb{N}_+ \stackrel{\text{def}}{=} \{1, 2, 3, 4, \ldots\}$. Note the chance in notation: we use subscript to denote the input variable of a sequence a_k instead of the usual notation for functions $a(k)$.

We can define a sequence by specifying the formula for the k^{th} term in the sequence. For example, the *harmonic sequence* is defined by the formula $h_k \stackrel{\text{def}}{=} \frac{1}{n}$. Another way to define a sequence is by listing the first few values in the sequence: $[h_0, h_1, h_2, h_3, \ldots]$, which correspond to evaluating formula h_k for $k = 0$, $k = 1$, $k = 2$, $k = 3$, etc. We'll now look at some examples of sequences, specifying both their formulas and showing the the first few values of each sequence.

The natural numbers: The simplest possible example of a sequence is the identity function, which 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 a "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 natural numbers:

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

Harmonic sequence: We obtain another useful sequence by computing the fractions $\frac{1}{k}$ for each $k \in \{1, 2, 3, \ldots\} = \mathbb{N}_+$:

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

This is called the *harmonic sequence* because it describes the wavelengths of harmonic frequencies produced by musical instruments. When we 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 terms:

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

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 indices $k \in \{2, 4, 6, 8, \ldots\}$, 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 will be of 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 a *geometric* process that repeatedly grows/shrinks by the amount r. When $r < 1$, the values in the 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: We'll also use the label b_k for the special case of the geometric series with $r = 2$:

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

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 a sequence as k goes to infinity? We can use the limit notation $\lim_{k\to\infty}$ to describe this process. There are two behaviours we're interested in: sequences that blow up to infinity, and sequences that approach some fixed number as k goes to ∞.

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:

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

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

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, and indeed we'll soon 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, we'll start by looking at some finite sums to gain some experience with the summation notation.

D. *Exact formulas for finite summations*

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 described by the following formulas:

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

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:

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

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

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

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 23 shows a visualization for this infinite sum.

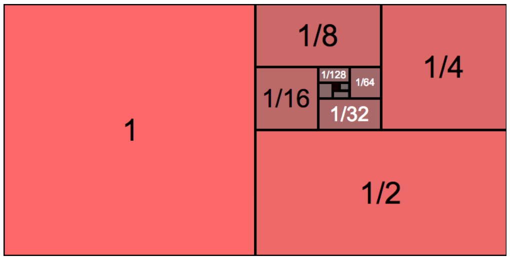


Figure 23. A graphical representation of the infinite sum of the geometric series with $r = \frac{1}{2}$. The area of each region corresponds to one of the terms in the series. The total area is equal to $\sum_{k=0}^{\infty}(\frac{1}{2})^k = \frac{1}{1-\frac{1}{2}} = 2$.

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 converging 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 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 infinite series blows up to infinity $\sum h_k = \infty$. We say that the harmonic series *is divergent*.

Using convergent series for practical calculations: We can use infinite series to compute 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 $e = 2.71828182845905\ldots$:

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

The calculation above is not just cool math fact, but a useful computational procedure that we can use to approximate 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 $n = 10$ and $n = 15$ terms in the series $\sum_{k=0}^{m} f_k$:

```
>>> 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...
```

Summing together the first 10 terms in the series gives us an approximation to 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, which is $2.71828182845905\ldots$.

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 used `sp.oo` to make SymPy compute 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 $a_k \overset{\text{def}}{=} \frac{(-1)^{k+1}}{k}$:

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

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 sums with more and more terms, we can obtain an approximation that is 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)
```

We can come up with all kinds of other infinite series expression for calculating other numbers. Instead of showing you other series for approximating numbers, I'll show you an even more powerful calculus technique: a way to approximate *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 the 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 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 some way to choose the coefficients c_k of the power series, so that the resulting polynomial approximates the function: $P_n(x) \approx f(x)$.

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 denote $f^{(k)}(0)$. Specifically, the k^{th} coefficient in the Taylor series approximation for the function $f(x)$ is $c_k \overset{\text{def}}{=} \frac{f^{(k)}(0)}{k!}$. 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 $x = 0$ divided by $k!$. 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 $c_k = \frac{f^{(k)}(0)}{k!} = \frac{1}{k!}$. The Taylor series of $f(x) = e^x$ is

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

Taylor series are a powerful computational tool for approximating 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

$$\begin{aligned}
\frac{1}{1-x} &= \sum_{k=0}^{\infty} x^k = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + \cdots \\
\frac{1}{1+x} &= \sum_{k=0}^{\infty} (-x)^k = 1 - x + x^2 - x^3 + x^4 - x^5 + x^6 + \cdots \\
e^x &= \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \cdots \\
\sin(x) &= \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots \\
\cos(x) &= \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} = 1 - \frac{x^2}{2} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \\
\ln(x+1) &= \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} x^n = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \frac{x^5}{5} + \cdots
\end{aligned}$$

Readers 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 `sp.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. Applications of series

The Taylor series representation for the function $f(x)$ provides a relatively easy way to compute its integral function $F_0(x) \overset{\text{def}}{=} \int_0^x f(u)\,du$. The Taylor series of $f(x)$ consists only of polynomial terms of the form $c_n x^n$. To compute the integral function $F_0(x) \overset{\text{def}}{=} \int_0^x f(u)\,du$, we can compute the integrals of the individual terms, which gives us $\frac{c_n}{n+1}x^{n+1}$. For example, if we want to compute the integral of $\cos(x)$, $\int \cos(x)\,dx =$??.

$$a$$

we can recognize this is the series for $\sin(x)$, so we conclude that $\int \cos(x)\,dx = \sin(x)$.

VII. Multivariable calculus

Multivariable calculus is the extension of the ideas of differential and integral calculus to functions that take multiple variables as inputs. If you were able to follow the single-variable calculus concepts, then you can easily learn 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 24. The height of the surface above the point (x, y) is function output $f(x, y)$.

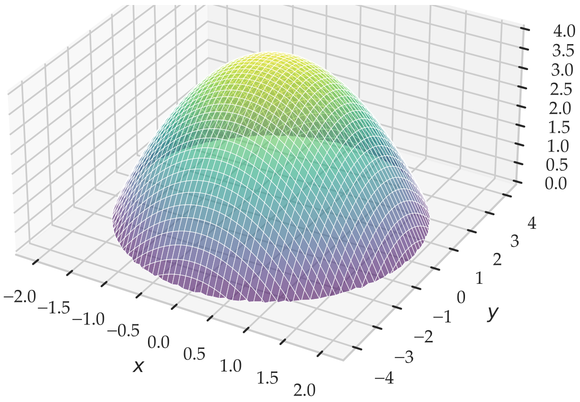


Figure 24. The 3D surface plot of the the function $f(x, y)$.

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 function $f(x, y)$ 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.

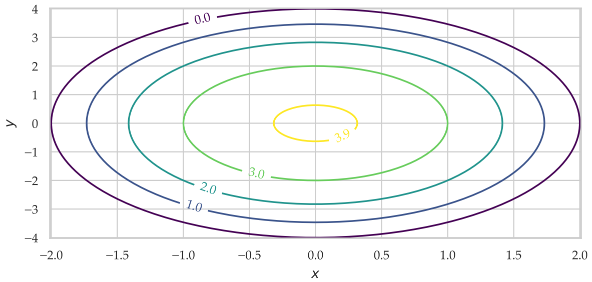


Figure 25. Topographic map that shows the function $f(x, y)$ as level curves.

The curve labeled 0.0 you see in Figure 25 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

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

Similarly the partial derivative of with respect to y is

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

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 $\frac{\partial f}{\partial y}(x, y)$.

Example: The partial derivatives of the function $f(x, y) = 4 - x^2 - \frac{1}{4}y^2$ are as follows:

$$\frac{\partial f}{\partial x} = -2x \qquad \frac{\partial f}{\partial y} = -\tfrac{1}{2}y.$$

Intuitively, we ignore the other variable.

C. *The gradient operator*

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

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

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.

Example: The gradient of the function $f(x, y) = 4 - x^2 - \frac{1}{4}y^2$ is

$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) = \left(-2x, -\tfrac{1}{2}y \right).$$

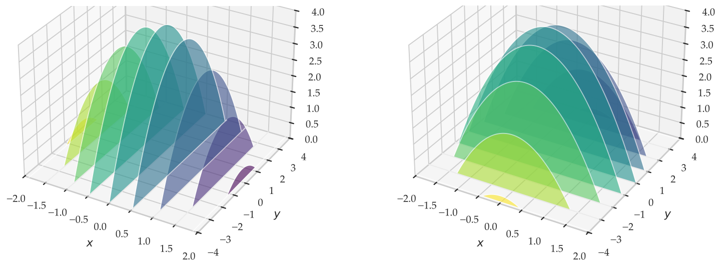
The other half of multivariable calculus involves applying the integration operation to multivariable functions.

D. *Partial integration*

We can integrate with respect to one of the input variables to produce a function that depends only on the other input variable:

$$f(y) = \int_{x=-\infty}^{x=\infty} f(x, y)\, dx \quad \text{and} \quad f(x) = \int_{y=-\infty}^{y=\infty} f(x, y)\, dy.$$

The functions $f(y)$ and $f(x)$ are called the *partial integrals* or *marginals* of the function $f(x,y)$. For example, the partial integrals of the function $f(x,y) = 4 - x^2 - \frac{1}{4}y^2$ are obtained by computing the areas of "slices" throughout he function $f(x,y)$, as illustrate in Figure 27.



(a) Slices at different values of x.

(b) Slices at different values of y.

Figure 26. Visualization of the partial integration procedures.

E. Double integrals

The multivariable generalization of the integral $\int_a^b f(x)\,dx$ that computes the "total" amount of $f(x)$ on between a and 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 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 functions, we integrate by splitting 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: Gradients are the key for the optimization algorithms used in modern machine learning techniques. Indeed, we "learn" model parameters of a machine learning model by minimizing some objective function, which involves repeated "downhill" steps in the direction of the gradient $-\nabla f(x,y)$.

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.

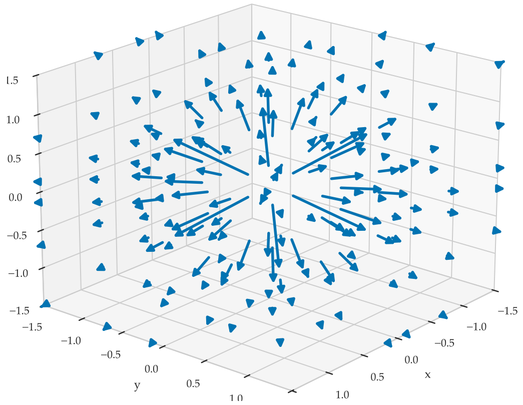


Figure 27. Visualization of the vector field $\vec{E}(x, y, z)$ around a positive charge.

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

- $\mathbf{r} = (x, y, z)$: position in space with coordinates (x, y, z).
- $\nabla \overset{\text{def}}{=} \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$: 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

Consider the curve C traced by $r(t)$ as t varies from t_i and t_f.

path integrals of vectors fields,

Scalar path integral.

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

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 $dr \overset{\text{def}}{=} \|\mathbf{r}'(t)\|\, dt$, which involves computing the derivative of $\mathbf{r}(t)$ then computing the length.

Vector path integral.

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

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 $d\mathbf{r} \overset{\text{def}}{=} \mathbf{r}'(t)dt$ during each step.

C. Surface integrals

flux integrals of vectors fields through surfaces,

Scalar surface integral.

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

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.

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

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 $d\mathbf{S} \overset{\text{def}}{=} \hat{\mathbf{n}}dS \overset{\text{def}}{=} (\mathbf{r}'_u \times \mathbf{r}'_v)dudv$, 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 $\partial\mathrm{V}$, which is the boundary of the volume V. Gauss' divergence theorem is:

$$\iiint\limits_{\mathrm{V}} \nabla \cdot \vec{F}\, d\mathrm{V} = \iint_{\partial \mathrm{V}} \vec{F} \cdot d\vec{S}$$

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.

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

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

F. Problem 3: integration using substitution

Compute $\int_0^1 \frac{4x}{(1+x^2)^3}\,dx$ using the substitution $u = 1 + x^2$.

Solution: When using the change of variable $u = 1 + x^2$, we must also change the differential $du = 2x\,dx$, which conveniently contains x that appears in the faction numerator, which allows us to write:

$$\int_{x=0}^{x=1} \frac{4x}{(1+x^2)^3}\,dx = \int_{x=0}^{x=1} \frac{2}{u^3}\,du = \int_{x=0}^{x=1} 2u^{-3}\,du.$$

Next we must change the x-limits of integration to u-limits of integration: The lower limit $x = 0$ became $u = 1 + 0^2 = 1$, and the upper limit $x = 1$ becomes $u = 1 + 1^2 = 2$, which the complete substation:

$$\int_{x=0}^{x=1} \frac{4x}{(1+x^2)^3}\,dx = 2\int_{u=1}^{u=2} u^{-3}\,du.$$

We can now proceed using the integral rule $\int x^n\,dx = \frac{1}{n+1}x^{n+1} + C$ to obtain

$$\begin{aligned} 2\int_{u=1}^{u=2} u^{-3}\,du &= 2\left[\frac{u^{-2}}{-2}\right]_1^2 = -\left[u^{-2}\right]_1^2 = -\left[\frac{1}{u^2}\right]_1^2 \\ &= -\left(\frac{1}{2^2} - \frac{1}{1^2}\right) = -\left(\frac{1}{4} - 1\right) = \frac{3}{4} \end{aligned}$$

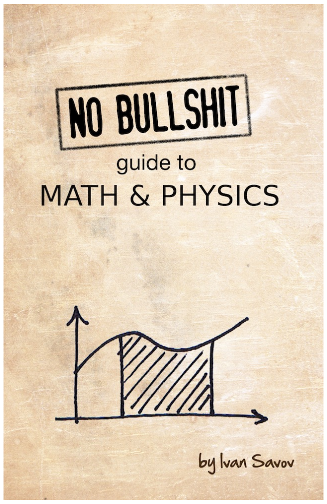
Links

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`.