Automatic differentiation

Contents

Types of differentiation

There are 3 methods of computing the derivative of a function evaluate at a specific point :

- Symbolic
- Numérical
- Automactic Differentiation

1 Symbolic differentiation

It's the same as we do when computing manually the derivative of a function to get its value at a specific point. We take the input expression and apply rules of differentiation such as :

- $$\frac{d}{dx}f \times \frac{d}{dx}g = \frac{d}{dx}f \times g + \frac{d}{dx}g \times f$$

- $$\frac{d\sin}{dx} = \cos$$

- $$\frac{d}{dx}\frac{f}{g} = \frac{g \times \frac{d}{dx}f - f \times \frac{d}{dx}g}{g^2}$$

Symbolic computing takes as input an expression and then only use its symbolic representation for each operation (in other words, the program keeps using ‘x‘ instead of the value of ‘x‘ for computing). It’s really helpful to get the exact form of a result since there cannot be any error of rounding or floating precision because no value is used at all.

For example, with Python symbolic differentiation works as following :

```
import sympy as sym

x = sym.Symbol('x') # declare x as a symbolic variable
t = sym.Symbol('t')

f = sym.sin(x*(t**2)) # definition of the expression
f_prime = f.diff(x) # compute the symbolic form of the derivative (first
    order, aka df/dx) with respect to x
```

Then

```
print(f)
>> sin(t**2*x)
print(f_prime)
>> t**2*cos(t**2*x)
```

To get the value of the derivative at one point (replacing the symbolic representations of the variables with their respective values) :

```
# Evaluate the derivative of f with x=2, t=3
f_prime.evalf(subs={x: 2, t:3})
>> 5.94285037419672
```

The main issue with symbolic computing is that it needs to expand each expression to get its reduced form. For example $(x + y + z)^3$ expands to $x^2 + 2xy + y^2 + 2xz + 2yz + z^2$. And that expansion/simplification has a prohibitive cost for longer expression.

And because of that overhead, time and resource consuming method, it’s not the usual way to compute.

2 Numerical differentiation

It’s the go to method for computing derivative in a simple and fast way. But it comes with a trade off : precision. Numerical differentiation approximates numerically the value of the derivative but does not compute it’s exact value.

Let’s dig into how does it works.

2.1 What is a derivative ?

First of all, we need to understand how derivative is defined to understand numerical differentiation.

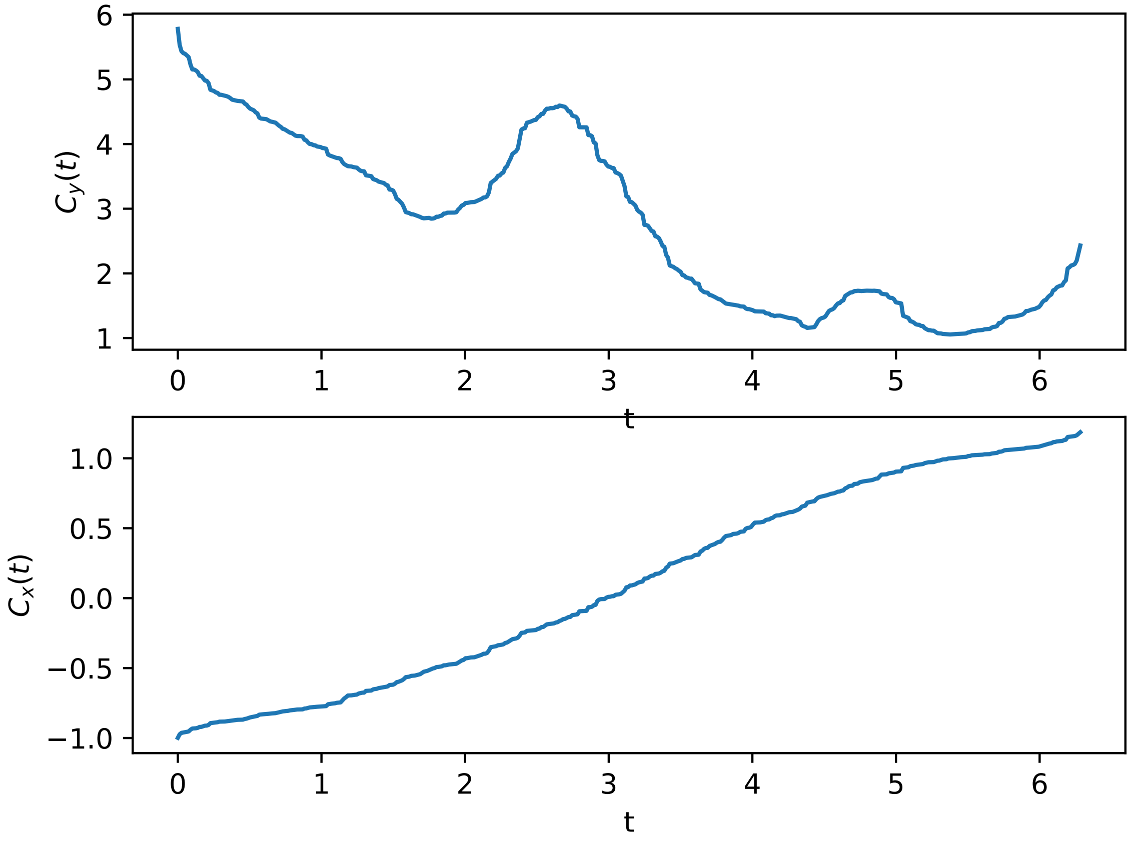
2.1.1 Definition of a function

A function describe a transformation/a relation from one set (in a mathematical sense) to another. For example, we could have a relation that give the coordinates of an object in the 2D plan with respect to time t. At any time t the object has for coordinate $(C_x(t), C_y(t))$ (usually coordinate are represented by $(x(t), y(t))$ but in order to mitigate any doubts from notations we made a distinction here). C_x and C_y are two functions :

$$C_x : \begin{cases} \mathbb{R} \to \mathbb{R} \\ t \longmapsto C_x(t) \end{cases}$$

$$C_y : \begin{cases} \mathbb{R} \to \mathbb{R} \\ t \longmapsto C_y(t) \end{cases}$$

The two functions could have for graphical representation the following graph :



Naturally we want to superimpose the two graph to see the evolution of the y coordinate with respect to the x coordinate (after all that the graph which really represent the motion of the object).

Let's define :

$$X := \{C_x(t) | t \in Ker(C_x) \cap Ker(C_y)\} = Im(C_x)$$

$$Y := \{C_y(t) | t \in Ker(C_x) \cap Ker(C_y)\} = Im(C_y)$$

Note that since C_x and C_y have the same Kernel (because they share the same time) :

$$Ker(C_x) \cap Ker(C_y) = Ker(C_x) = Ker(C_y)$$

Then, the corresponding composition function C is :

$$C : X \to Y$$

where for $a \in X$, $f(a) = b \in Y$ with

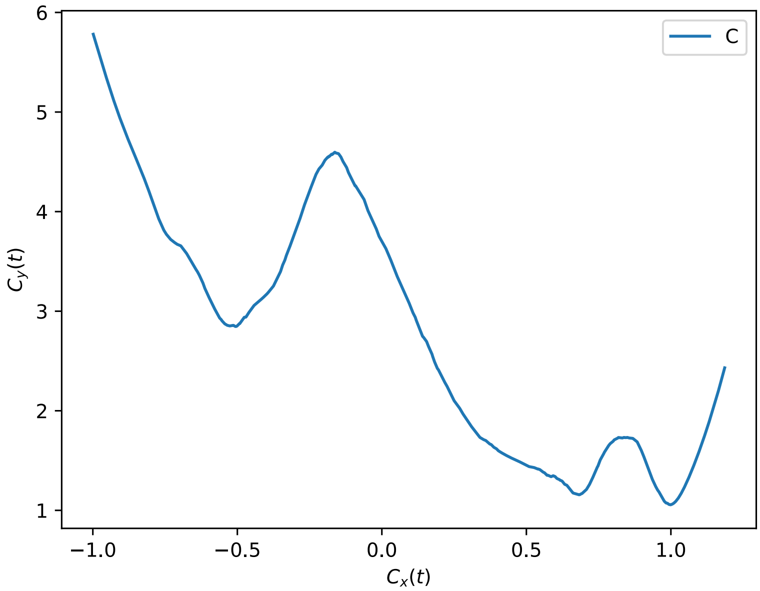
$$\begin{cases} t \in Ker(C_x) \cap Ker(C_y) \\ a = C_x(t) \\ b = C_y(t) \end{cases}$$

Note that C is the function that map for the same moment in time t the value of $C_x(t)$ to $C_y(t)$

C can be written as :

$$C : \begin{cases} X \to Y \\ a \mapsto C_y(C_x^{-1}(a)) \end{cases}$$

With C_x^{-1} the inverse image of C_x by X

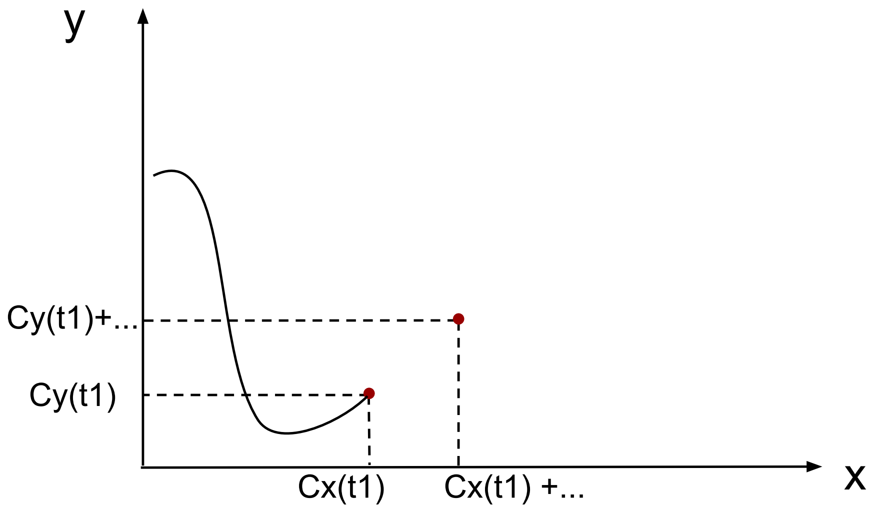


2.1.2 History of derivatives

The notion of derivatives is not so old and find its origin in the field of Physics. The question was : from a specific time t_1 where the object will be a few moment after ? In a graphical way, how to approximate the graph after the time t_1 (based on the knowledge of the previous points) ?

The main hypothesis which is natural is to consider that there is not so much changes if the two points are really really close, then we can approximate the graph between the two point as a straight line. And that is the foundation of differentiation.

Take a look at a potential graph of the function C. The goal is to find each quantity '...' (which could be different for each axis/dimension).



For simplicity let's define $x_i := C_x(t_i)$ and $y_i := C_y(t_i)$. Note that := means that the equality is by definition.

In fact it's not really about the proximity of the points chosen but the time elapsed to go from $P_1 := (C_x(t_1) = x_1, C(C_x(t_1)) = C_y(t_1) = y_1)$ and $P2 := (C_x(t_2) = x_2, C(C_x(t_2)) = C_y(t_2) = y_2) := (x_1 + ..., y_1 + ...)$, with $t_2 := t_1 + o$ where o is really really really small time quantity.

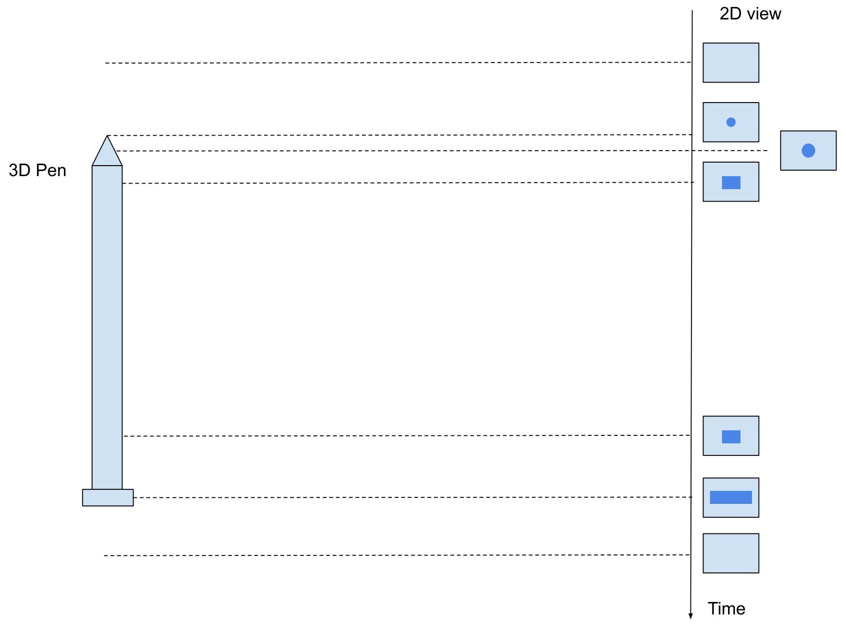
2.1.3 Notion of speed and velocity

The first derivatives were used to explain the physical phenomena which intrinsically implies the evolution along time. A lot of the origin of derivative is based things that are hard-wired to our daily experience of living.

Some words on Time Time is the named of the concept that we need to explain the motion of objects. In fact, the correlation between time and movement of objects has been named motion. We have an intrinsic understanding of time thanks to our body which has some clock protein which are produced and broken down in a cycle that lasts 24h (which is called the circadian cycle). The only way for us to explain, measure the motion of an object is time. In fact, the simple fact to speak about motion and time is already a tautology.

The definition of time is just something that we can related to describe some phenomena without what we could not understand them. In fact, the actual definition of time is given by the emission of radiation from atoms that have a regular rate. Then this rate is adjust to match what we know being 24h hours (which is approximately given by the clock protein that has probably matched the cycle of day/nigh in our planet).

if time was a physical dimension (4D) then we may not need this concept anymore because we could literally see object evolving along that "time" dimension. But in our vision of world there is something missing to explain motion. A living being in a 2D world would need the concept of time to describe the third dimension that we have and that he's missing. For them, an object is describe by the time needed to entirely see it and the shape of projection of our 3D object into their 2D plan view (living in a 2D world means having a view in a 2D plan. It's the normal plan for them, but for us it's 2D relative to our 3D world).



Time is quite the same thing as the axiom on top of what the euclidean geometry is built on. In particular, the 5th axiom, the parallel postulate tells us :

> If a line segment intersects two straight lines forming two interior angles on the same side that are less than two right angles, then the two lines, if extended indefinitely, meet on that side on which the angles sum to less than two right angles

And that postulate cannot be proven using the 4 others. Quite like time as been used has a "pseudo postulate". Moreover, non euclidean geometries like the Riemannian geometry have a curved space where two parallel line can intersect each other (for a curved space observed inside our 3D world these line are intersecting but for a living being of that curved space they are absolutely not).

Time is useful to resolve problem without really being able to define it. And maybe there is some other concept that describe motion better like other geometries better describe some problems.

Speed and motion Speed is a central concept in derivative and it's build on top of motion. The definition of speed is a pretty old concept. For example *The Physics* by Aristotle describes the notion of speed and velocity :

- [...] two things are of the same velocity if they occupy an equal time in accomplishing a certain equal amount of motion. [...] We may say, therefore, that things are of equal velocity in an equal time they traverse the same magnitude - *Book 7, part 4*

- Then, A the movement have moved B a distance G in a time D, then in the same time the same force A will move 1/2B twice the distance G, and in 1/2D it will move 1/2B the whole distance for G: thus the rules of proportion will be observed - *Book 7, part 4*

The speed is the amount of distance travelled with respect to the time elapsed. So why is it useful ? Because from a point t_a in time if we want to know the distance traveled to another moment t_b you just have to know the speed of the object at t_a(with the hypothesis that the speed at t_a is representative of the average speed in $[t_a, t_b]$). The distance travelled is then $(t_b - t_a) \times speed|_{t_a}$.

This definition does not come from nowhere. It's comes because we have observed that an object at each point of the time has what we call speed/velocity which comes from the fact that the object is put in motion by a force (external or internal). Speed can be measured experimentally at "each" (at least at enough moments in time that we can no longer make a difference) points of time by taking some really small intervals of time (Δ_t really small) and measuring during that interval how much an object has moved. Thus, the simple knowledge of that force (how much it's pushing the object and in what direction) give us a pretty rule of proportion (with the hypothesis that the force is constant : it's pushing with the same amount of strength, and in the same direction).

Since we assume that the initial force is representative of the average force between t_a and t_b, then we can approximate the distance at t_b only from the knowledge of the forces and position of the object (with its mass) at t_a. In other words, it's strictly the same as drawing a straight line between two point. We just have to know how the line is oriented and when to stop it.

Of course this is not true in general : forces are not always constants and are usually not representative of the average speed. But it's quite true if the time elapsed between t_a and t_b is small enough. And experimental testing tends to comfort that hypothesis. However, it was at least a starting point in history of speed an later derivative.

This short recap of history is only present to bring to the light the nature of speed which is not a tool simply invented but which is intrinsically related to the Physics of our world. We observe that in some cases to push an object along the same distance (let say from point A to B in a straight line), we need to apply a force F (let say a constant force) that will takes a specific time t to move. And we observe that we can move that same object, along the same distance half the time of the first try, by applying the double of the force. And to compare that two experiences, we need to compare how much the object has been moved at each moment. Some experiences could have different distance or different duration. So we need something that represent ratio to be able to compare them at each point of the time no matter the initial parameters (distance, duration, ...). That ratio is the today well known : $\frac{\Delta_d}{\Delta_t}$ where Δ_d is the total distance between A and B, and Δ_t is the time elapsed. That rate is then called speed.

Of course when the force is non constant, speed will change and the ratio $\frac{\Delta_d}{\Delta_t}$ can no longer be applied for the whole motion of the object because for certain moments/intervals of moments, this rate could be higher or lower than the previous or next moments. Then we understand why we need to repeat the same operation for lot of points of the curve with really small time intervals. But being in the scope of mathematics the "small" is in fact a limit : if we need to do the operation for a lot of points of the curve to gain accuracy, in fact we can go on for infinity (because the small interval can always be spitted in smaller interval for continuous function) and the only end is when we can have the speed of a single point (no more extremely small intervals that are representative of a specific point of the curve, just the speed for that single point at a single moment in time). This is what the limit represents.

By the way, what we have done for distance with respect to time (which give the speed) can also be done for its speed with respect to time which represents at each moment how much the speed changes (which is called acceleration).

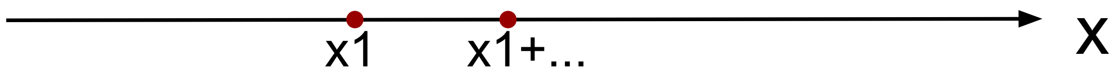
In conclusion :

> The speed is the instantaneous rate of change of the distance with respect to the time

2.1.4 Derivative definition

Going back to our initial problem, by knowing the speed of an object at a specific point, we then can determine form a specific point what will be its next position.

If we take a one dimensional example (our object is only moving along a line)



Then $(x_1 + ...) = x_1 + speed|_{t_1} \times o$ where $speed|_{t_1}$ is the speed of the object at the moment t_1 and o that really small quantity $(o \neq 0)$ in time. We take two moment in time really close and we suppose that all the forces and constraints remain the same during that really small change in time. Then we can apply a basic rule of proportion. Historically, the speed or velocity (the directional speed/the speed along a specific axis, along a specific degree of freedom of the object) is noted $\dot{x}$ (for the speed along the x axis).

The, we can write P_2 as $(x_1 + \dot{x_1} \times o, y_1 + \dot{y_1} \times o)$. Note that $\dot{x_1} = C_x(\dot{x}(t_1)) = speed|^x_{t_1}$ is the speed of the object along the x axis at the time t_1. This is the notation used by Newton (see 1,2) when he first introduces the derivative or its own version of derivative. The quantities along x, y was called ‘flowing‘ or ‘fluent‘ and the corresponding velocity was called ‘fluxion‘. The ‘moment‘ is the ‘fluxion‘ × ‘time‘.

If we calculate the slope α_{P_1} of the line made between P_1 and P_2 we will find :

$$\begin{aligned}\alpha_{P_1} &= \frac{C(x_2) - C(x_1)}{x_2 - x_1} \\ &= \frac{C(x_1 + \dot{x_1} \times o) - C(x_1)}{\big(x_1 + \dot{x_1} \times o\big) - x_1}\end{aligned}$$

Since by hypothesis $(y_2 = y_1 + ...)$, then by identification $y_2 = C(x_1 + \dot{x_1} \times o) = y_1 + \dot{y_1} \times o$. Note that, this brings to the light the following relation : $C(x_1 + \dot{x_1} \times o) = C(x_1) + \dot{C}(x_1) \times o$ which remind us the Taylor expansion of first order ! Of course at that step we still don't know what is a derivative but...

Then,

$$\alpha_{P_1} = \frac{\big(y_1 + \dot{y_1} \times o\big) - y_1}{\big(x_1 + \dot{x_1} \times o\big) - x_1}$$

Since $o \neq 0$:

$$\alpha_{P_1} = \frac{\dot{y_1}}{\dot{x_1}}$$

Since $\dot{y_1}$ is the velocity along the y axis at the time t_1, then $\dot{y_1} = \frac{\Delta_y}{\Delta_t}|_{t_1}$, then :

$$\alpha_{P_1} = \frac{\Delta_y}{\Delta_x}\Big|_{t_1}$$

When o is infinitesimally small α_{P_1} becomes the slope of the tangent at P_1. And we observe that P_2 is just a point along the tangent really close to P_1. In other words, we find a straight line that allow use to approximate where the next point will be on the graph of C.

Later, $\alpha_{P_1} = \frac{\dot{y_1}}{\dot{x_1}}$ was called a derivative.

In conclusion : "' the derivative of y is the instantaneous rate of change of y with respect to point x "'

Derivatives has then been generalized and applied to other fields than Physics where the notion of speed is no more defined in a Physics way but where it's described as the instantaneous rate of changes in a function. But the under laying concept comes from observation of our world.

2.1.5 Some notations

Usually, we note C_x as x (which must not be mingled with the x axis), C_y as y and C as f.

Nowadays, we would write :

- $$o \to dt$$
- $$\dot{y_1} \to \frac{dy}{dt}|_{t_1}$$
- $$\dot{x_1} \to \frac{dx}{dt}|_{t_1}$$
- $$\dot{x_1} \times o \to \frac{dx}{dt}|_{t_1} \times dt \to dx$$
- $$\dot{y_1} \times o \to \frac{dy}{dt}|_{t_1} \times dt \to dy$$
- $$\alpha_{P_1} \to \frac{dy}{dx}|_{t_1}$$

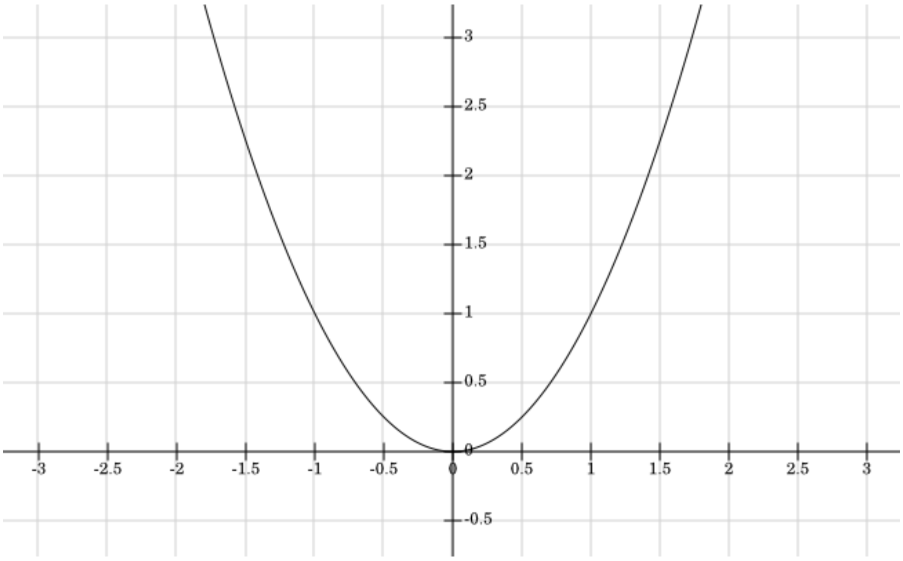
($\frac{dy}{dx}$ remains a function of t here, because $x : t \mapsto x(t)$ and $y : t \mapsto y(t)$ are functions of t. Formally we would have to write $\frac{dy(t)}{dx(t)}|_{t=t_1}$, or $\frac{dy}{dx}(t)|_{t=t_1}$)

dx, dy, dt are called the differential of the variable x, y, t. Which basically represent the infinitesimally step in the x, y, z direction.

2.1.6 Derivative calculation

But one question remains : how to derivative is formally calculated from explicit expression like $x + 2y$?

The Newton notation can be applied to arbitrary function with a defined expression. For example, let's take $y = x^2$ which give us a parabola curve :



The derivative of y with respect to x is obtained by replacing x by a small change along that its direction $x \to x + \dot{x} \times o$, same for y with $y \to y + \dot{y} \times o$ and then :

$$\begin{aligned} y = x^2 &\leftrightarrow (y + \dot{y} \times o) = (x + \dot{x} \times o)^2 \\ &\leftrightarrow (y + \dot{y} \times o) = x^2 + 2x\dot{x}o + o^2 \\ &\leftrightarrow (y - x^2) + \dot{y} \times o = 2x\dot{x}o + o^2 \end{aligned}$$

Since $y = x^2$, then :

$$(y - x^2) + \dot{y} \times o = 2x\dot{x}o + o^2 \leftrightarrow \dot{y} \times o = 2x\dot{x}o + o^2$$

And because $o \neq 0$:

$$\dot{y} = 2x\dot{x} + o$$

And because o is infinitesimally small we can ignore it. Then we end up with :

$$\begin{aligned} \dot{y} = 2x\dot{x} &\leftrightarrow \frac{\dot{y}}{\dot{x}} = 2x \\ &\leftrightarrow \frac{dy}{dx} = 2x \end{aligned}$$

Which is exactly what today we know being the derivative of x^2.

During the calculation of the derivative I used $\leftrightarrow$ instead of $\Leftrightarrow$. It's because at one point we say $o \neq 0$ to divide by o and then later because o is small we can ignore it (meaning $o = 0$). That exactly what Leibniz blame Newton's theory the use of infinitesimally quantity. For more context see Leibniz–Newton calculus controversy). And that's also why today we do not make calculation with infinitesimally quantities (dx, dy, dt) as a "number" but we use the notion of limits.

Nowadays we will write :

$$\begin{aligned} \frac{dy}{dx} &= \alpha_{P_1} \\ &= \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \\ &\underset{h \to 0}{=} \frac{f(x+h) - f(x)}{h} + o(h) \end{aligned}$$

Where $o(h)$ is the Landau notation (and not the Newton's o notation).

$$\begin{aligned}
\frac{dy}{dx} \underset{h\to 0}{=} & \frac{f(x+h)-f(x)}{h} + o(h) \\
& \Leftrightarrow f(x+h) \underset{h\to 0}{=} f(x) + \frac{dy}{dx} \times h - o(h) \\
& \Leftrightarrow f(x+h) \underset{h\to 0}{=} f(x) + \frac{dy}{dx} \times h + o(-h) \\
& \Leftrightarrow f(x+h) \underset{h\to 0}{=} f(x) + \frac{dy}{dx} \times h + o(h) \\
& \Leftrightarrow f(x+h) \underset{h\to 0}{=} f(x) + \frac{d}{dx} f \times h + o(h)
\end{aligned}$$

Which is the first order Taylor expansion of f.

2.1.7 Derivative computation

Now, how computer compute derivatives numerically ? As we would do with any given curve, or experimentally : take a really small step in the input set of the function and use :

$$\frac{f(x+h)-f(x)}{h} \approx f'(x) = \frac{df}{dx}$$

Because $o(h)$ is so small that it can be ignored when $h \to 0$. Usually $h = 10^{-6}$ in computer programs.

For example with $f : x \mapsto x^2$ the value of the derivative at the point $x = 3$ is with $h = 10^{-6}$

$$\begin{aligned}
\frac{f(3+10^{-6})-f(3)}{10^{-6}} &= \frac{(3+10^{-6})^2-9}{10^{-6}} \\
&= \frac{9,000006-9}{10^{-6}} \\
&= \frac{0,000006}{10^{-6}} \\
&= 6,000001 \\
&\approx 2 \times 3
\end{aligned}$$

The we need to do that for each desired points x

The main issue with this method is the precision of the result because some numbers cannot be represented exactly in computer, and there are some error which propagate in each operation made (rounding error, catastrophic cancellation, ...). Standards are defined to make computation with number replicable. Theses issues are avoided by symbolic differentiation because there is no value at all, thus the computation is exact.

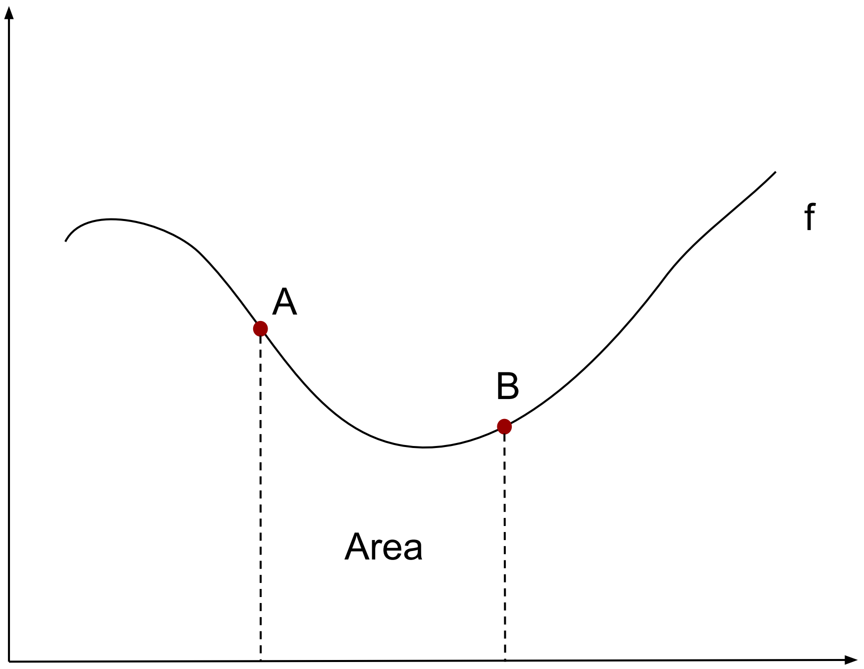
3 Automatic differentiation

There is two modes of operation for AD : forward and bacward mode.

3.1 Forward mode

3.1.1 Integration of a function

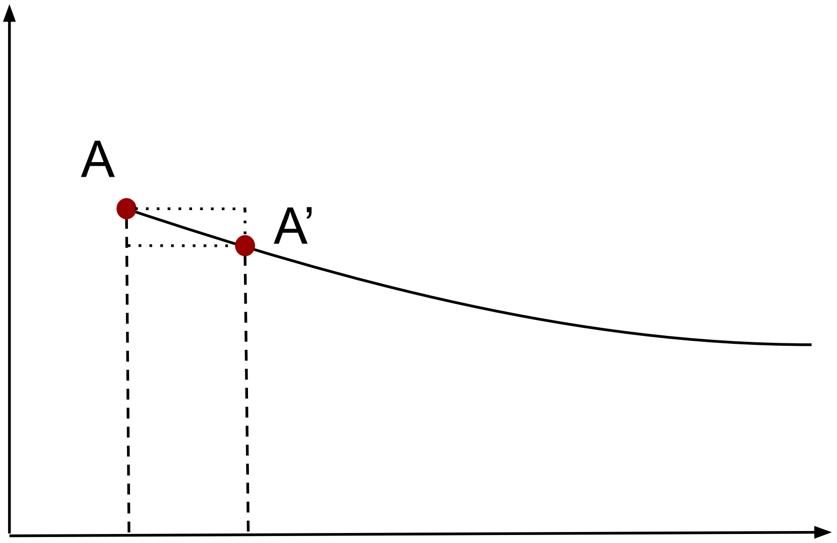
The integral of a function f from $x_a \in Ker(f)$ to $x_b \in Ker(f)$, noted $\int_{x_a}^{x_b} f(x)dx$, is by definition the area under that curve



Here, $A := (x_a, y_a := f(x_a))$ and $B := (x_b, y_b := f(x_b))$

There is lot of definition of integral but the simplest is the Riemann one. The basic idea behind Riemann integral is to split the interval $[A, B]$ into small rectangle. If we split this interval into small enough rectangle then the sum will have for limits the area under the curve.

Let's take a look at a zoom of the graph of f around the point A



We approximate the area under A, A' by the area of the rectangle with height $f(x_a)$. Then the area for that rectangle is : $f(x_a) \times (x_{a'} - x_a)$.

3.1.2 Integral of derivative

The fundamental theorem of calculus (FTOC) give us :

$$\int_a^b (\frac{d}{dx} f) dx = f(b) - f(a)$$

Remember that the two dx are quite not the same and we can not "simplify" by it. The dx in the integral represents along which direction (x axis) we integrate f.

But why is it true ?

Let's define $\mathcal{A}(x)_f$ the area of f from $c \in Ker(f)$ as is the first defined value for f with $c \geq 0$, to $x \in Ker(f)$. If f is defined at 0, then $c = 0$. In other words, $\mathcal{A}(x)_f = \int_c^x f(x)dx$

As A and A' and A' are really close, then, $A' := (x_a + h, f(x_a + h))$ with $h \neq 0$ as $(x + h) \in Ker(f)$.

The area from A to A' is then : $\mathcal{A}(x_a + h)_f - \mathcal{A}(x_a)_f$

$$f(x_a + h) \times ((x_a + h) - x_a) < \mathcal{A}(x_a + h)_f - \mathcal{A}(x_a)_f < f(x_a) \times (x_a - (x_a + h))$$

$$f(x_a + h) \times h < \mathcal{A}(x_a + h)_f - \mathcal{A}(x_a)_f < f(x_a) \times h$$

$h \neq 0$:

$$f(x_a + h) < \frac{\mathcal{A}(x_a + h)_f - \mathcal{A}(x_a)_f}{h} < f(x_a)$$

When $h \to 0$:

$$\begin{cases} f(x_a + h) \to f(x_a) \\ f(x_a) \to f(x_a) \\ \frac{\mathcal{A}(x_a + h)_f - \mathcal{A}(x_a)_f}{h} \to (\frac{d}{dx}\mathcal{A}(x)_f)|_{x=x_a} \end{cases}$$

Then,

$$f(x_a) \leq (\frac{d}{dx}\mathcal{A}(x)_f)|_{x=x_a} \leq f(x_a)$$

Thus,

$$(\frac{d}{dx}\mathcal{A}(x)_f)|_{x=x_a} = f(x_a)$$

Since x_a has been chosen arbitrarily, then :

$$\frac{d}{dx}\mathcal{A}(x)_f = f$$

With the usual notation we have for f continuous on $[a, x]$:

$$\begin{cases} F(x) = \mapsto \int_a^x f(x)dx \\ \frac{d}{dx} F = f \end{cases}$$

3.1.3 Taylor series

For $f : \mathbb{C} \to \mathbb{R}$ infinitely differentiable at $a \in \mathbb{C}$ we have the following Taylor power series :

$$f(x) = \sum_{0}^{\infty} \frac{f^n(a)}{n!}(x-a)^n$$

where $f^n(a)$ denotes the nth derivative of f evaluated at the point a and $f^0(a) = f(a)$.
An excellent proof is given here

3.1.4 Dual number

Dual numbers are the key to AD forward mode. A dual number is define as a real part (called the primal part) and a dual part (called the tangent part) :

$$a + b\epsilon$$

where :

$$\begin{cases} a, b \in \mathbb{R} \\ \epsilon \neq 0 \\ \epsilon^2 = 0 \end{cases}$$

Dual numbers are part of the family of hyper-numbers as complex numbers are. The definition of dual number can be quite out of context but as complex numbers it has an history. To better understand of such number are discovered take a look at the excellent video of *Veritasium.*

To better understand how calculus is done with dual number take a look here.

History of dual number can be linked to Newton calculus when in some calculation the term dx^2 was ignored (meaning $dx = 0$) while $dx \neq 0$.

So why dual number useful for differentiation ?

Because for any function that admin a Taylor expansion we have for $x = a + \epsilon$ where a is the differentiable point of f :

$$f(x) = \sum_{0}^{\infty} \frac{f^n(a)}{n!}(x-a)^n \Leftrightarrow$$

$$f(a+\epsilon) = \sum_{0}^{\infty} \frac{f^n(a)}{n!}(\epsilon)^n \Leftrightarrow$$

$$f(a+\epsilon) = f(a) + \frac{df}{dx}(a)\epsilon + \sum_{2}^{\infty} \frac{f^n(a)}{n!}\epsilon^n$$

Since for $n \geq 2$, $\epsilon^n = 0$, the :

$$f(a+\epsilon) = f(a) + \frac{df}{dx}(a)\epsilon$$

Thus, by computing tne value of f at $a + \epsilon$ we have the value of f at a and the exact value of the derivative of f at a. Note that, this is not an approximation but an exact value as if it was obtains with symbolic differentiation.

Denote that we can take a dual number as a complex number and use it as input for any function since all usual operations in the rings are defined (see 1, 2). We just need to use a norm (which has a continuity property) for comparison.

3.1.5 Gradient

Let's focus on a more general case : multi-variable scalar function, also called scalar field function. As its name stand for a multi-variable scalar function is a function $f : \mathbb{R}^n \to \mathbb{R}$ that takes as argument different variable x, y, z, ... and output a scalar in $\mathbb{R}$ or $\mathbb{C}$. For example the following function is a scalar field :

$$f : \begin{cases} \mathbb{R} \to \mathbb{R} \\ (x, y) \mapsto \sin(x) \times y \end{cases}$$

Then to talks about derivatives we need to know on which variable we want to focus : x, y, z. In other words the single case variable function derivatives extends to multi-variables by taking derivatives of the function for each variable. A derivative with respect to a variable is called a partial derivative and usually denoted as $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$, $\frac{\partial f}{\partial z}$, ...

Formally the directional derivative (if exists) is defined for f at a with respect to x :

$$\frac{\partial f}{x}(a_x, a_y, a_z, ...) = \lim_{h \to 0} \frac{f(a_x + h, a_y, a_z, ...) - f(a_x, a_y, a_z, ...)}{h}$$

Having all well defined partial derivatives at a does not means the function is differentiable at that point ! If f is continuous at a then its differentiable at a. But continuous then differentiable is given by another theorem and differentiation can be obtains without it (by the definition for example, which imply the knowledge of topology).

We denote as gradient the vector of partial derivatives : $\nabla f := (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}, ...)$. It could be a line or a column vector and we go from one form to another with the transpose :

$$\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \\ \vdots \end{bmatrix}^T = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}, ...]$$

We usually talks about gradient for vector spaces when dealing with common functions. A vector space has a dimension denoted n which describes the number of possibles directions along which we could move and that define the space at the most elementary level. For example, $\mathbb{R}$ is a vector space of dimension 3 : x, y, z. We could also imagine a 4th direction that make a diagonal in the x, y plan but then it would be a combination of the 3 previous directions. Since x, y, z cannot be expressed as a combination of other element inside that space it's forming the axis/direction of the space. Together theses axis are forming a family of the vector space and are usually denotes as $(e_1, e_2, e_3, ...)$.

Each element a of the vector space can be written as a combination of a family of the vector space as $a = \sum_{i=0}^{n} a_i \times e_i$ with a_i the coordinate of a along the e_i direction. For the common 2D and 3D vector space (e_1, e_2, e_3) are denotes as $(\vec{i}, \vec{j}, \vec{k})$.

Since the *gradient* is a vector we can express it through the axis of the output space as :

$$[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}, ...] \times \begin{bmatrix} \vec{i} \\ \vec{j} \\ \vec{k} \\ \vdots \end{bmatrix} = \frac{\partial f}{\partial x}\vec{i} + \frac{\partial f}{\partial y}\vec{j} + \frac{\partial f}{\partial z}\vec{k} + ...$$

And abusively we just write $\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} + \frac{\partial f}{\partial z}$. But the gradient is not a number !

3.1.6 Directional derivative

If the partial derivative of a function represents the rate of change of that function along the corresponding direction, then the directional derivative represents the rate of change of the function along an arbitrary chosen direction.

For $v = \in \begin{bmatrix} v_x \\ v_y \\ v_z \\ \vdots \end{bmatrix} \mathbb{R}^n$ the directional derivative of f for any $a \in \mathbb{R}^n$ is :

$$\nabla_v f(a) = \lim_{h \to 0} \frac{f(a + hv) - f(a)}{h}$$

Remind that h is a scalar

And, $\nabla_v f(a) + ...$

Here, the directional derivative is a scalar which is the sum of all partial derivative with respect to each axis defined by the directional vector v (it's a dot product - the multiplication extension to vector).

$$v \times \nabla f(a) = v_x \times \frac{\partial f}{\partial x}(a) + v_y \times \frac{\partial f}{\partial y}(a) + v_z \times \frac{\partial f}{\partial z}(a)$$

3.1.7 Jacobian

The gradient is in fact a particular case of the Jacobian. The Jacobian is the same thing as the gradient bu defined for function with multiple output value $f : \mathbb{R}^n \to \mathbb{R}^m$. Such a function is decomposed for each coordinate of the output as a new function :

$$f : \begin{cases} \mathbb{R}^n \to \mathbb{R}^m \\ \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \mapsto \begin{bmatrix} f_1(x_1, x_2, ..., x_n) \\ f_2(x_1, x_2, ..., x_n) \\ \vdots \\ f_m(x_1, x_2, ..., x_n) \end{bmatrix} \end{cases}$$

For example f can be :

$$f : (x, y) \mapsto (x \times y, \cos(x))$$

And if you think of f as the application of a single function sin for example to a vector, we get a vector that can also be described as different function :

$$f : \begin{bmatrix} x \\ y \end{bmatrix} \mapsto \sin\Big(\begin{bmatrix} x \\ y \end{bmatrix}\Big) = \begin{bmatrix} \sin(x) \\ \sin(y) \end{bmatrix} := \begin{bmatrix} f_1(x, y) \\ f_2(x, y) \end{bmatrix}$$

In fact, all functions than we can think of are built in the scenario of single variable function and then generalized for multi-variable function usually by applying that function for each component.

The Jacobian of a multi-variable function is then :

$$\begin{bmatrix} \frac{\partial f_1}{x_1}(x_1, x_2, ..., x_n) & \frac{\partial f_1}{x_2}(x_1, x_2, ..., x_n) & ... & \frac{\partial f_1}{x_n}(x_1, x_2, ..., x_n) \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_m}{x_1}(x_1, x_2, ..., x_n) & \frac{\partial f_m}{x_2}(x_1, x_2, ..., x_n) & ... & \frac{\partial f_m}{x_n}(x_1, x_2, ..., x_n) \end{bmatrix}$$

Now let's extend the Taylor series for scalar field function. Let's define $f : \mathbb{R}^n \to \mathbb{R}$ a scalar field function.

Remember that the goal is the get the Taylor expansion for scalar field function to extend the use of dual number to such a function. Then we just need to get the expansion to the 2nd order as seen previously. In fact there is a complete formula of Taylor series for multi-variable function but it comes without real context :

$$\sum_{n_1=0}^{\infty} \cdots \sum_{n_d=0}^{\infty} \frac{(x_1 - a_1)^{n_1} \cdots (x_d - a_d)^{n_d}}{n_1! \cdots n_d!} \left(\frac{\partial^{n_1+\cdots+n_d} f}{\partial x_1^{n_1} \cdots \partial x_d^{n_d}} \right) (a_1, \ldots, a_d)$$

The proof made previously with integral could possibly also be done like that, but we will need to use line integral] (use for integral of multi-variable function) which is an overhead.

In fact, there is more elegant and simple proof at least for our use case.

Let's consider f a C^2 scalar field function (f is continuous and its derivatives up to the 2nd orders are continuous). The proof could be made with less powerful hypothesis but will be harder to write and understand.

The directional derivative of f in the direction defined by $a \in \mathbb{R}^n$ at $r \in \mathbb{R}^n$ arbitrary chosen :

$$a \times \nabla f(r) = \lim_{h \to 0} \frac{f(r + ha) - f(r)}{h}$$

Now let's consider : $g : \begin{cases} \mathbb{R} \to \mathbb{R} \\ h \mapsto f(r + ha) \end{cases}$

g is a single variable function since r and a are fixed. The only changing par is h. And since f is a scalar field, then $f(r + ah) \in \mathbb{R}$.

Let's suppose that f is 2-differentiable (f admit at least derivatives of the 2nd order at r). Then f is continuous at 0, thus g is continuous at 0 and admit derivative of the 2nd order.

Then since g is a 1-dimensional function we can use the Taylor series expansion for any $h\mathbb{R}$:

$$g(h) = g(0) + g'(0)h + \frac{1}{2} g''(0)h^2 + ...$$

-

$$g(0) = f(r + a \times 0) = f(r)$$

- for $x \in \mathbb{R}$ arbitrary chosen (then, since f is continuous) :

$$\begin{aligned} g'(x) &= \lim_{l \to 0} \frac{g(x + l) - g(x)}{l} \\ &= \lim_{l \to 0} \frac{f(r + (x + l)a) - f(r + xa)}{l} \end{aligned}$$

Let's consider $X := r + xa$, then

- $$\begin{aligned} g'(x) &= \lim_{l \to 0} \frac{f(X + la) - f(X)}{l} \\ &= \nabla_a f(X) \\ &= \nabla_a f(r + xa) \end{aligned}$$

- $$g'(0) = \nabla_a f(r + 0 \times a) = \nabla_a f(r) = a\nabla f(r)$$

- $$\begin{aligned} g''(0) &= \lim_{l \to 0} \frac{g'(0 + l) - g'(0)}{l} \\ &= \lim_{l \to 0} \frac{\nabla_a f(r + la) - \nabla_a f(r)}{l} \end{aligned}$$

Let's consider $\phi : \begin{cases} \mathbb{R}^n \to \mathbb{R} \\ x \mapsto \nabla_a f(x) \end{cases}$

Since the directional derivative is a linear combination of the partial derivatives and since f is $\mathbb{C}^2$, then $\nabla_a f$ is differentiable and continuous. Thus,

$$\begin{aligned} g''(0) &= \lim_{l \to 0} \frac{\phi(r + la) - \phi(r)}{l} \\ &= \nabla_a \phi(r) \end{aligned}$$

$g''(0)$ is the directional derivative of the directional derivative of f at r along the direction of a. It's denoted as $\nabla_a^2 f(r)$ or $\nabla_a \nabla_a f(r)$.

Then,

Then,

$$\begin{aligned} g(h) &= f(r) + \nabla_a f(r)h + \nabla_a^2 f(r)h^2 + \wr(\langle^{\in}) \Leftrightarrow \\ f(r + ha) &= f(r) + \nabla_a f(r)h + \nabla_a^2 f(r)h^2 + \wr(\langle^{\in}) \end{aligned}$$

Since r, a, h were arbitrary choose, thus for $h = 0 + \epsilon$:

$$f(r + a\epsilon) = f(r) + \nabla_a f(r)\epsilon + 0$$

Then the dual number trick can also be applied to scalar field function. For $a = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, we get :

$$f(r + a\epsilon) = f(r) + \frac{\partial f}{\partial x_1}\epsilon$$

where x_1 is the 1th variable of the function f.

Finally, for multi-variables functions we need to compute each partial derivative through the use of dual number.

3.1.8 Hessian

For $f : \mathbb{R}^n \to \mathbb{R}$ the Taylor expansion at $a \in \mathbb{R}$ for $x \in \mathbb{R}^n$ is :

$$f(x) = f(a) + \nabla f(a)(x - a) + \frac{1}{2}(x - a)^T Hf(a)(x - a) + \wr((\S - \dashv)^{\in})$$

where Df is the gradient of f and Hf the hessian of f (which is the "derivative" of the 2nd order of f)

$$Hf = \begin{bmatrix} \frac{\partial^2 f}{\partial {x_1}^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial {x_2}^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_n} & \frac{\partial^2 f}{\partial x_n \partial x_n} & \cdots & \frac{\partial^2 f}{\partial {x_n}^2} \end{bmatrix}$$

For $f : \begin{cases} \mathbb{R}^n \to \mathbb{R}^m \\ x \mapsto (f_1(x), f_2(x), ..., f_m(x)) \end{cases}$ the hessian is : $Hf = (Hf_1, Hf_2, ..., Hf_m)$ a [tensor](https://en.wikipedia.org/wiki/Tensor). A tensor is a n-dimensional structure that holds data and adapts it to the transformation of the surrounding space where the tensor live. For example, with a tensor no matter how the usual 3D space coordinate system is oriented, rotated, translated, the value inside the tensor will represents the same thing by adapting the inner value seemless adapting to the transformation. So it's not just a matrix (see 1, 2).

Here is an example of such a hessian.

3.1.9 Matrix

For $f : \mathbb{R}^{n \times m} \to \mathbb{R}^{p \times q}$ function for matrices things get more complicated for the Taylor series since the multiplication of matrix is not commutative.

However, the Taylor series can be generalized to many other objects (see 1, 2).

3.1.10 Implementation

Finally, the forward mode of Automatic differentiation is the use of dual number to get one by one the partial derivative. This is can be achieved pragmatically by overloading operators and function in the programming language.

```
class DualNumber:
    def __init__(self, primal, tangent=1):
        self.primal = primal # real part
        self.tangent = tangent # dual part

    def __add__(self, v):
        # right addition : current_object + input object
        if type(v) is DualNumber:
            return DualNumber(self.primal+v.primal, self.tangent+v.tangent)
        else: # scalar addition
            return DualNumber(self.primal+v, self.tangent)

    ...
```

Then we need to define functions for dual numbers as following :

```
def sin(input_value):
    if type(input_value) is DualNumber:
        return DualNumber(math.sin(input_value.primal), input_value.tangent*
   math.cos(input_value.primal))

    return math.sin(input_value)
```

This mode of Automatic differentiation works well when the number of input is smaller than the number of output because we need to do the computation as many times as there is input (with the previous example once with $(a_1 + \epsilon, a_2)$ and twice with $(a_1, a_2 + \epsilon)$). When the number of output is greater than the one in inputs we use another method : the backward mode.

3.2 Backward mode

The forward mode was the more complex in math because we needed to make sure to understand how we can obtain exact derivative. The backward mode is in fact really simple and straightforward but the catch is in the implementation which is trickier.