Basics of Mathematics in Machine Learning II

Toni Karvonen

Exactum B326 — University of Helsinki
toni.karvonen@helsinki.fi

March 11, 2024

Contents

1 Introduction

These are lecture notes for the course *Basics of Mathematics in Machine Learning II* (MAT11015) lectured in Spring 2024. Th notes approximately contain the material in Chapter 5 and Sections 6.1–6.4 and 7.1 of *Mathematics for Machine Learning* by Deisenroth, Faisal and Oong (freely available at https://mml-book.github.io/). We cover the following three topics:

I *Univariate calculus* — Section 2

II *Vector calculus* — Section 3

III *Discrete probability* — Section 4

Topics **I** and **II** cover the tools and results from mathematical analysis that are essential for machine learning. The primary learning objective of these topics is to (a) understand the basics of mathematical optimisation and (b) be able to implement some optimisation methods. Our approach to calculus is not completely rigorous, as we do not properly define some of the notions that we encounter or use and occasionally sweep certain finer points under the rug. A rigorous treatment would require *much* more time than we have. Those interested should consider some of the following courses:

- *Raja-arvot* (MAT11003)
- *Differentiaalilaskenta* (MAT11004)
- *Integraalilaskenta* (MAT11005)
- *Calculus IA: Limits and differentiation* (MAT11006)
- *Calculus IB: Integration* (MAT11007)
- *Advanced calculus* (MAT11008)
- *Vektorianalyysi I* (MAT21003)
- *Vektorianalyysi II* (MAT21020)

Topic **III** covers the basics definitions, concepts and results of discrete probability theory. The objective of this topic is to provide the essentials upon which to build on subsequent courses on statistical machine learning and data science. We have time to cover only the bare minimum about probability, and you should seriously consider taking a course dedicated to probability theory, such as

- *Todennäköisyyslaskenta I* (MAT12003)

1.1 Motivation

Let us begin by motivating the these topics with three simple examples.

Ordinary Least Squares. Suppose that we have observations (or outputs) $y_1, \dots, y_n \in \mathbb{R}$ at some distinct locations (or inputs) $x_1, \dots, x_n \in \mathbb{R}$. This is our *training data* (suom. *opetusjoukko*). To name a few examples, the locations could describe spatial coordinates of some physical sensors, they could be time instances [in which case with a *time series* (suom. *aikasarja*)], or they could be different parameters that an experiment is performed with. Given these data, we want to predict what the observation might be at a location that is not included in the dataset. To do this, we can postulate that observations have linear relationship to the locations:

$$y_i = a + bx_i + \varepsilon_i \quad \text{for} \quad i = 1, \dots n, \tag{1.1}$$

where a and b determine the relationship between x_i and y_i and the residual ε_i accounts for observation noise or influence on y_i of sources other than x_i. Our task is to find a "good" or

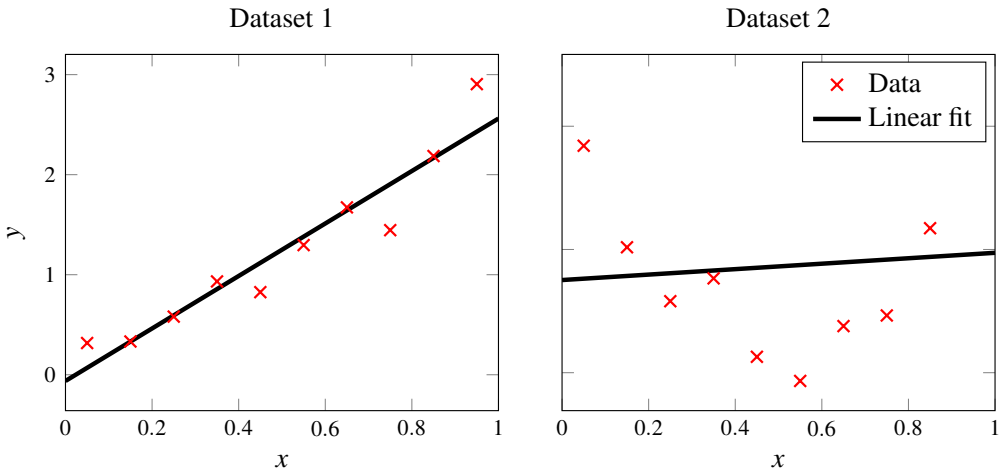

Figure 1: The linear fits $f(x) = a^* + bx^*$ for two different datasets. Here a^* and b^* are computed from (1.3). The fit is quite good for the first dataset but completely useless for the second.

"optimal" values for a and b. This is called *linear regression* (suom. *lineaarinen regressio*). One way to achieve this is to select a and b that minimise the *sum of squared residuals*

$$L(a,b) = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (y_i - a - bx_i)^2. \tag{1.2}$$

The function L is an important example of a *loss function* (suom. *tappiofunktio*), or *cost function*. The selection of a loss function is arbitrary (e.g., we could alternatively try to minimise $\sum_{i=1}^{n} |\varepsilon_i|$), but the *quadratic* loss function in (1.2) is mathematically extremely convenient, yielding the *ordinary least squares* method (suom. *pienimmän neliösumman menetelmä*). During the course we will learn how to minimise L and that (as long as $n \geq 2$)

$$\begin{bmatrix} a^* \\ b^* \end{bmatrix} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}, \quad \text{where} \quad \mathbf{X} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \in \mathbb{R}^{n\times 2} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n, \tag{1.3}$$

solve this minimisation problem *uniquely*, in that $L(a^*,b^*) < L(a,b)$ for any other pair (a,b). The *linear fits* $f(x) = a^* + bx^*$ produced by linear regression are depicted in Figure 1 for two rather different datasets. The figure shows that the linear fit can be quite useful if the data indeed exhibit a linear trend, as is the case for Dataset 1. That is, for an input x_0 not contained in the training data it seems plausible that $f(x_0)$ would be close to the corresponding output, y_0. However, the fit is useless if no such trend is present, as is the case for Dataset 2. This is because the relationship (1.1) between x_i and y_i that we have postulated is not particularly expressive, having only two parameters, a and b.

Neural Networks. At their core, *neural networks* (suom. *neuroverkko*) consist of nothing more than (i) a flexible postulate like (1.1) for the relationship between the inputs and outputs; (ii) selection of some parameters by minimising a loss function; and (iii) computation of a fit that is then used to predict the output at unobserved inputs.[1] Let us look at a simple two-layer network based on the *sigmoid activation function* (suom. *aktivaatiofunktio*)

$$\sigma(x) = \frac{1}{1+e^{-x}}. \tag{1.4}$$

[1]The following article gives a decent introduction to neural networks: HIGHAM & HIGHAM (2019). Deep learning: An introduction for applied mathematicians. *SIAM Review* 61(4):860–891.

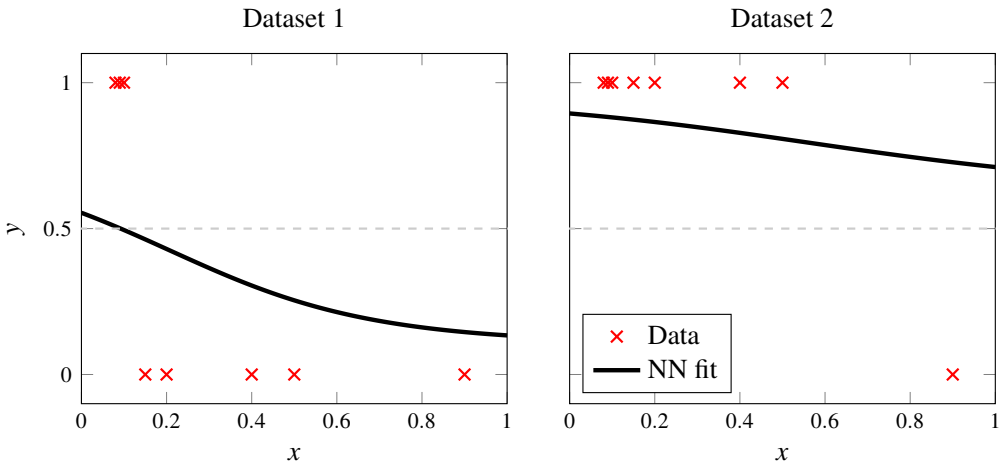

Figure 2: The neural network (NN) fit $f(x) = F(x \mid w_1^*, w_2^*, b_1^*, b_2^*) = \sigma(w_2^*\sigma(w_1^* x + b_1^*) + b_2^*)$ for two different datasets. Here w_1^*, w_2^*, b_1^* and b_2^* have been computed using gradient descent. The datasets consist of $n = 8$ *labels* $y \in \{0, 1\}$ (i.e., points labelled $y = 1$ are in a category A and points those labelled $y = 0$ in category B). After computing the neural network fit f, we may classify a new point by, for example, saying it is in category A if $f(x) > \frac{1}{2}$ and in category B if $f(x) \leq \frac{1}{2}$. This *decision boundary* (suom. *päätöspinta*) is plotted as a dashed grey line. Such classification works for Dataset 1 (at least barely), but not for Dataset 2. This is understandable, as the neural network we have used is extremely simple.

Let $w_1, w_2 \in \mathbb{R}$ be *weights* (suom. *paino*) and $b_1, b_2 \in \mathbb{R}$ *biases* (suom. *vakiotermi*). Inspired by (1.1), we postulate that the training outputs $0 \leq y_i \leq 1$ are related to the inputs x_i via the equation

$$y_i = \sigma(w_2\sigma(w_1 x_i + b_1) + b_2) + \varepsilon_i \quad \text{for} \quad i = 1, \ldots, n. \tag{1.5}$$

Define

$$F(x \mid w_1, w_2, b_1, b_2) = \sigma(w_2\sigma(w_1 x + b_1) + b_2), \tag{1.6}$$

so that (1.5) becomes $y_i = F(x_i \mid w_1, w_2, b_1, b_2) + \varepsilon_i$. As in the case of linear regression, we may define a quadratic loss function

$$L(w_1, w_2, b_1, b_2) = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (y_i - F(x_i \mid w_1, w_2, b_1, b_2))^2 \tag{1.7}$$

and select the weights w_1, w_2 and biases b_1, b_2 by minimising this loss function. But we hit a roadblock. For, unlike in the case of ordinary least squares method that provides the simple expression (1.3) for the minimisers of the loss function (1.2), no such nice formula is available now. Instead, we have to device an *optimisation method* to minimise (1.7). Because in real problems there are thousands of weights and biases and the training data sets are huge, it is essential that this method be efficient, in that it should call the loss function as few times as possible. During the course we will learn how to use *derivatives* and *gradients* to describe the direction and rate of change of functions and how to implement the *gradient descent* (suom. *gradienttimenetelmä*) algorithm, the workhorse of machine learning. Results of selecting the parameters w_1, w_2, b_1 and b_2 using gradient descent are displayed in Figure 2.

Statistical Inference and Probability. Much of machine learning and data analysis is based on *statistical inference* (suom. *tilastollinen päättely*). That is, given some data one seeks to infer properties of an underlying probability distribution and exploit these properties to

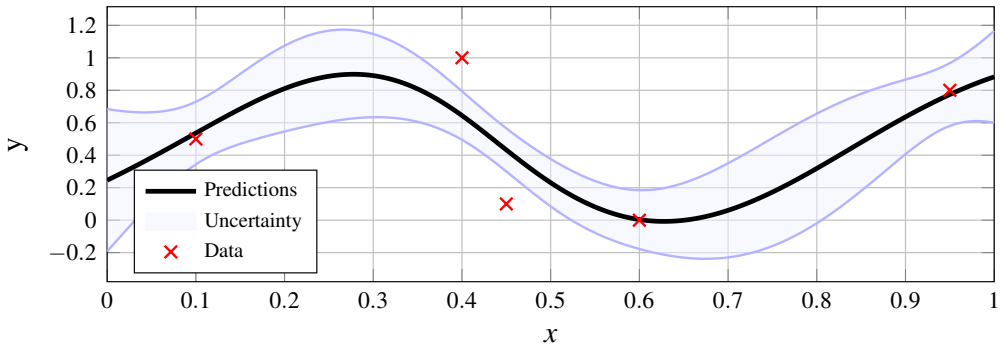

Figure 3: Statistical inference and prediction based on *Gaussian processes* (suom. *gaussinen prosessi*). Given some data, we construct a certain *statistical model* (suom. *tilastollinen malli*) that attempts to represent the process that generated the data (i.e., some relationship between x and y). Using this model and statistical assumptions about noise present in the data we can perform predictive inference. The predictions are uncertain, as represented by the shaded region. While our best prediction for the data value at, say, $x = 0.7$ is $y \approx 0.05$, we acknowledge that there is significant uncertainty in this prediction, considering "it very likely" that $y \in [-0.2, 0.4]$.

make predictions. The foundations of statistical inference are in the *theory of probability* (suom. *todennäköisyysteoria*), which provides a mathematical language for decision-making and uncertainty. Figure 3 illustrates a certain statistical inference and prediction method popular in machine learning and spatial statistics. During the course we will learn the basic definitions, concepts and results of probability theory.

1.2 Notational Conventions

Number sets. The sets $\mathbb{R}$, $\mathbb{N}$, and $\mathbb{Z}$ of real numbers, natural numbers, and integers are

$$\mathbb{R} = (-\infty, \infty), \quad \mathbb{N} = \{1, 2, \ldots\}, \quad \text{and} \quad \mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\},$$

respectively. Note that this definition excludes zero from $\mathbb{N}$ and that the infinities $-\infty$ and ∞ are not included in $\mathbb{R}$.

Scalars, vectors and matrices. When talking about scalars, vectors, and matrices, we will use plain font (i.e., x or X) for scalars, bold lowercase font for vectors (i.e., $\mathbf{x}$), and bold uppercase font for matrices (i.e., $\mathbf{X}$). Note that, depending on the dimensions, $\mathbf{x}$ may happen to be a scalar and $\mathbf{X}$ a vector (or even a scalar). But x will never be a vector or a matrix.

Constants and functions. The constant $e \approx 2.718$ is *Euler's constant*, or *Napier's constant* (suom. *Neperin luku*). Moreover, $\exp(x) = e^x$. Euler's constant is the base of the natural logarithm. The natural logarithm of x is always denoted $\log(x)$ in these notes. It is the inverse function of e^x, satisfying $\log(e^x) = x \log(e) = x$. We do not use logarithms with other bases.

1.3 Preliminaries on Functions

Let us recall some basic notation and concepts that we will use throughout the course.

Definition 1.1 (Function). Let X and Y be sets. A *function* (suom. *funktio*) f from X to Y, denoted

$$f \colon X \to Y, \tag{1.8}$$

assigns to every element $x \in X$ exactly one element $f(x) \in Y$. The function f *maps* (suom. *kuvaa*) the *input* x to the *output* $f(x)$. This input-output relationship is often written as

$$x \mapsto f(x). \tag{1.9}$$

While there can be only one output $f(x)$ for each $x \in X$, each output does not have to correspond to a unique input. That is, different inputs can yield the same output [i.e., there can be $x_1 \neq x_2$ such that $f(x_1) = f(x_2)$]. Examples of such functions include any constant function from $\mathbb{R}$ to $\mathbb{R}$ given by $f(x) = c$ for some fixed $c \in \mathbb{R}$ or the quadratic function $f(x) = x^2$ from $\mathbb{R}$ to $\mathbb{R}$, which satisfies $f(1) = f(-1) = 1$.

In this course we mostly focus on functions $f\colon \mathbb{R}^d \to \mathbb{R}$ (i.e., $X = \mathbb{R}^d$ and $Y = \mathbb{R}$) that map vectors $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ to real numbers $f(\mathbf{x}) \in \mathbb{R}$. Examples of such functions are the bivariate polynomial $f(\mathbf{x}) = f(x_1, x_2) = x_1^2 x_2^3 + x_1 - 3$ (i.e., $f\colon \mathbb{R}^2 \to \mathbb{R}$) and the *ReLU activation function*

$$f(\mathbf{x}) = \max\{0, a + \mathbf{w} \cdot \mathbf{x}\} = \max\{0, a + \textstyle\sum_{i=1}^d w_i x_i\} = \begin{cases} 0 & \text{if} \quad a + \mathbf{w} \cdot \mathbf{x} < 0, \\ a + \mathbf{w} \cdot \mathbf{x} & \text{otherwise,} \end{cases} \tag{1.10}$$

where $a \in \mathbb{R}$ and $\mathbf{w} \in \mathbb{R}^d$ are fixed (i.e., $f\colon \mathbb{R}^d \to \mathbb{R}$). However, a function can be something much more complicated: A computer program that maps user input, such as text or parameter values, to some output, such as a text response or the result of a physics simulation, is also a function.

1.4 Function Composition

Complicated functions are often formed by *composing* simple functions.

Definition 1.2 (Function composition). Let $g\colon X \to Y$ and $h\colon Y \to Z$ be functions. Their *composite function* (suom. *yhdistetty funktio*) $f = h \circ g$ is a function from X to Z given by

$$f(x) = h(g(x)). \tag{1.11}$$

To compute $f(x) = h(g(x))$ we (i) first map x through g and (ii) after this map the output $g(x)$ through h. We can also compose more than two functions, so that

$$f = r \circ h \circ g \quad \text{is given by} \quad f(x) = (r \circ h \circ g)(x) = r(h(g(x))). \tag{1.12}$$

One can think recursively:

$$f = r \circ h \circ g = r \circ f_1, \quad \text{where} \quad f_1 = h \circ g. \tag{1.13}$$

Example 1.3. Define the functions

$$g(x) = 3\pi, \quad h(x) = \log(1+x), \quad r(x) = 2 + x^{-3} \quad \text{and} \quad l(x) = x, \tag{1.14}$$

all from $\mathbb{R}$ to $\mathbb{R}$. Then

$$f(x) = (h \circ g)(x) = h(g(x)) = h(3\pi) = \log(1 + 3\pi), \tag{1.15a}$$
$$f(x) = (g \circ h)(x) = g(h(x)) = g(\log(1+x)) = 3\pi, \tag{1.15b}$$
$$f(x) = (h \circ r)(x) = h(r(x)) = h(2 + x^{-3}) = \log(3 + x^{-3}), \tag{1.15c}$$
$$f(x) = (r \circ l)(x) = r(l(x)) = r(x) = 2 + x^{-3}, \tag{1.15d}$$
$$f(x) = (r \circ l \circ h)(x) = r(l(h(x))) = r(h(x)) = r(\log(1+x)) = 2 + \log(1+x)^{-3}. \tag{1.15e}$$

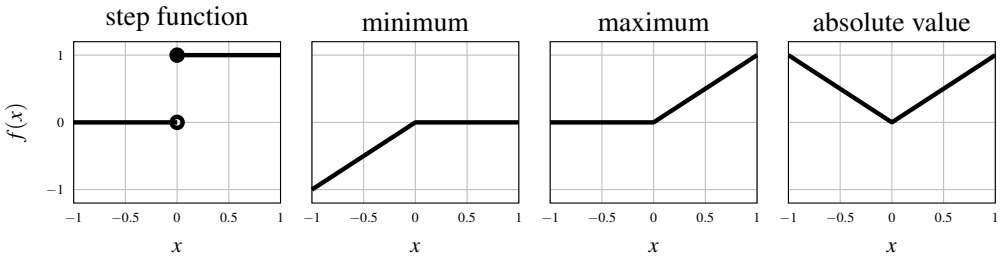

Figure 4: Four piecewise defined functions. From left to right: The step function in (1.19), the minimum and maximum functions in (1.20), and the absolute value function in (1.21).

Note that (1.15a) and (1.15b) are simply the *constant functions* $x \mapsto \log(1+3\pi)$ and $x \mapsto 3\pi$, which map every $x \in \mathbb{R}$ to the constants $\log(1+3\pi)$ and 3π, respectively.

Example 1.4. We can write the ReLU activation function (1.10) as the composition

$$f = h \circ g, \tag{1.16}$$

where

$$g(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} \quad \text{and} \quad h(x) = \max\{0, a+x\} \tag{1.17}$$

are functions from $\mathbb{R}^d$ to $\mathbb{R}$ and from $\mathbb{R}$ to $\mathbb{R}$, respectively. The composition is *not* unique. We could have alternatively selected $g(\mathbf{x}) = a + \mathbf{w} \cdot \mathbf{x}$ and $h(x) = \max\{0, x\}$.

Example 1.5. Consider the following idiotic piece of pseudocode:

```
1: procedure FUNC(scalar x, integer n ≥ 1)
2:     if n ≥ 1 then
3:         return 2 × FUNC(x, n − 1)
4:     else
5:         return x
6:     end if
7: end procedure
```

Given an integer constant $n \geq 1$, this procedure computes $f_n(x) = \text{FUNC}(x, n) = 2^n x$ by using the n-fold composition

$$f_n(x) = \underbrace{(g \circ \cdots \circ g)}_{n \text{ times}}(x), \quad \text{where} \quad g(x) = 2x. \tag{1.18}$$

For example, when $n = 3$ we have $f_n(x) = (g \circ g \circ g)(x) = g(g(g(x))) = 2(2(2x)) = 2^3 x = 8x$.

1.5 Piecewise defined functions

We will occasionally encounter *piecewise defined functions* (suom. *paloittain määritelty funktion*), such as the ReLU activation function (1.10). Typical piecewise defined functions include the *step function* (suom. *porrasfunktio*)

$$f(x) = \begin{cases} 0 & \text{if } x < 0, \\ 1 & \text{if } x \geq 0, \end{cases} \tag{1.19}$$

the *maximum* and *minimum*

$$pf(x) = \min\{0,x\} = \begin{cases} x & \text{if } x < 0, \\ 0 & \text{if } x \geq 0 \end{cases} \quad \text{and} \quad f(x) = \max\{0,x\} = \begin{cases} 0 & \text{if } x < 0, \\ 1 & \text{if } x \geq 0, \end{cases} \tag{1.20}$$

and the *absolute value* (suom. *itseisarvo*)

$$f(x) = |x| = \begin{cases} -x & \text{if } x < 0, \\ x & \text{if } x \geq 0. \end{cases} \tag{1.21}$$

These four functions are plotted in Figure 4.

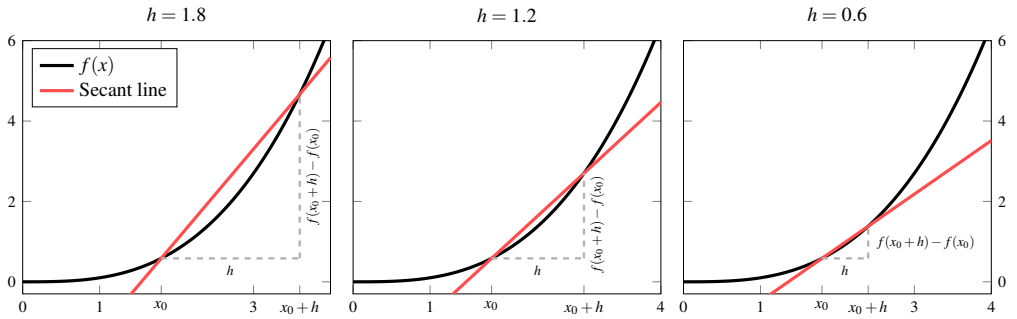

Figure 5: Secant lines for the function $f(x) = \frac{1}{10}x^3$ and $h = 1.8$, $h = 1.2$, and $h = 0.6$ at the point $x_0 = 1.8$. Observe how the secant line "barely touches" the graph of f when $h = 0.6$.

2 Calculus

In this section we consider *univariate* functions $f\colon \mathbb{R} \to \mathbb{R}$, which map reals to reals. *Multivariate* functions will be considered in Section 3. Our treatment of differentiation is by necessity very brief, cursory, and non-rigorous. If you need more practice with differentiation, you may wish to look at, for example, Chapter 3 in *Calculus: Volume 1* by Strang and Herman (freely available at https://openstax.org/details/books/calculus-volume-1).

2.1 Differentiation

We are interested in how to describe and compute the *rate of change* of a function. Having access to a quantity describing how a function changes is very useful in optimisation: When we want to minimise a function, it is natural to move towards the direction in which the function is decreasing.

Definition 2.1 (Difference quotient). Let $h \in \mathbb{R}$. The expression

$$\frac{f(x+h) - f(x)}{h} \tag{2.1}$$

is called the *difference quotient* (suom. *erotusosamäärä*) of the function f at point x.

The difference quotient measures the *slope* (suom. *kulmakerroin*) of the *secant line* (suom. *sekantti*) passing through the points $f(x+h)$ and $f(x)$. As h tends to zero, denoted $h \to 0$, this secant line tends to the *tangent* (suom. *tangentti*), the line that "barely touches" the curve defined by f at $(x, f(x))$; see Figure 5. The slope of the tangent line is the derivative of f at x.

Definition 2.2 (Derivative). The *derivative* (suom. *derivaatta*) of a function f at point x is

$$f'(x) = \frac{\mathrm{d}f}{\mathrm{d}x}(x) = \frac{\mathrm{d}}{\mathrm{d}x}f(x) = \lim_{h\to 0}\frac{f(x+h) - f(x)}{h}. \tag{2.2}$$

Computation of the derivative $f'(x)$ is called *differentiation* (suom. *derivointi*). The derivative is not necessarily well-defined at every $x \in \mathbb{R}$; we shall discuss this in Remark 2.6.

While there is a proper mathematical definition[2] for what $\lim_{h\to 0}$ means in (2.2), we will be

[2]The mathematically rigorous definition of the limit, the so-called ε-δ definition by Bolzano, Cauchy and Weierstrass from the 1800s, goes as follows: A function $g\colon \mathbb{R} \to \mathbb{R}$ has the limit c at a point $x_0 \in \mathbb{R}$, denoted $\lim_{x\to x_0} g(x) = c$, if for every $\varepsilon > 0$ one can find $\delta > 0$ (which can depend on ε) such that $|g(x) - c| < \varepsilon$ for all x such that $|x - x_0| < \delta$. You can learn this type of rigorous calculus on the courses *Raja-arvot* (MAT11003) and *Calculus IA: Limits and Differentiation* (MAT11006).

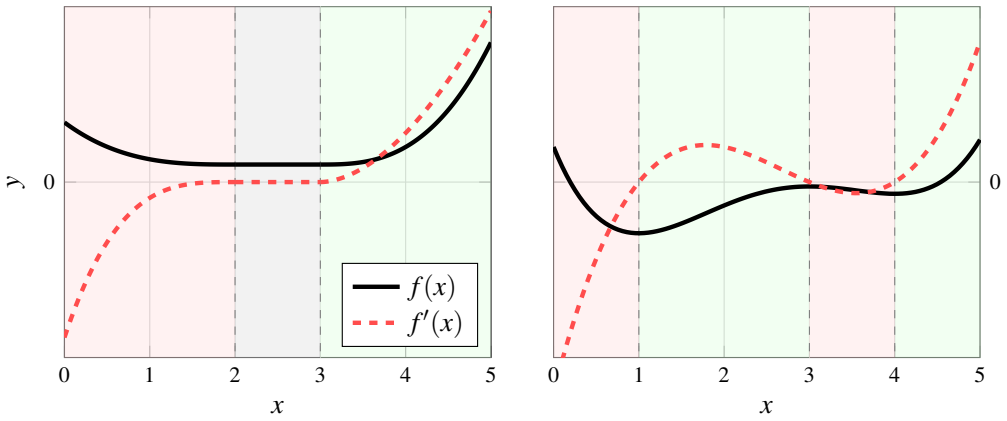

Figure 6: Two functions and their derivatives. The functions are decreasing on the red intervals ($[0,2]$ for the first function and $[0,1]$ and $[3,4]$ for the second), constant on the grey interval ($[2,3]$ for the first function), and increasing on the green intervals ($[3,5]$ for the first functino and $[1,3]$ and $[4,5]$ for the second). When the functions decrease, their derivatives are negative; when they are constant, the derivatives are zero; and when they increase, the derivatives are positive.

content with the following informal definition: Let g be a function and x_0 a point. Then

$$g(x) \to c \text{ as } x \to x_0, \text{ or } \lim_{x\to x_0} g(x) = c, \text{ or } \text{“}g(x) \text{ tends to } c \text{ as } x \text{ tends to } x_0\text{”}, \tag{2.3}$$

means, *very* roughly speaking, that $g(x)$ can be made arbitrarily close (or even equal) to c by taking x sufficiently close to x_0. In the context of Equation (2.2) this means that $c = f'(x)$ if the difference quotient $g(h) = [f(x+h) - f(x)]/h$ can be made arbitrarily close to c by taking h sufficiently close to 0. Note that "h is close to 0" permits h to be negative. That is, we are considering $h \in \mathbb{R}$ such that the absolute value $|h|$ is close to zero.

Working with limits is intuitive in many cases. For example, it should be clear that $\lim_{h\to 0} h^n = 0$ for any $n \in \mathbb{N}$ since a number close to zero raised to a positive power remains close to zero; we will see this principle in action in Examples 2.3 and 2.5. Similarly,

$$\lim_{h\to 0} \frac{1+h^n}{e^h} = 1 \tag{2.4}$$

because $1+h^n$ tends to 1 and e^h tends to $e^0 = 1$. However, sometimes such reasoning fails.

Leibniz's notation $\mathrm{d}f/\mathrm{d}x$ for derivative is suggestive: To compute derivative we divide a small difference, $\mathrm{d}f$, in the values of f [i.e., $f(x+h) - f(h)$] by a small difference, $\mathrm{d}x$, in x [i.e., $h = (x+h) - x$]. While Leibniz's notation can be treacherous, its formal manipulation allows one to easily recall many basic differentiation formulae.

Example 2.3. Let $a \in \mathbb{R}$. For the *constant function* $f(x) = a$ we have

$$f'(x) = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h} = \lim_{h\to 0} \frac{a-a}{h} = \lim_{h\to 0} 0 = 0. \tag{2.5}$$

Because a constant function does not change, its derivative (which describes the rate of change) should indeed be zero. For the *linear function* $f(x) = ax$ we have

$$f'(x) = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h} = \lim_{h\to 0} \frac{a(x+h) - ax}{h} = \lim_{h\to 0} \frac{ah}{h} = a. \tag{2.6}$$

For the *quadratic function* $f(x) = ax^2$ we have

$$f'(x) = \lim_{h\to 0} \frac{a(x+h)^2 - ax^2}{h} = \lim_{h\to 0} \frac{ax^2 + 2ahx + ah^2 - ax^2}{h} = \lim_{h\to 0}(2ax + ah) = 2ax. \quad (2.7)$$

Derivative describes both the *rate* and *direction* of change. If $f'(x)$ is positive, the function f is increasing at x; if the derivative is negative, the function is decreasing. If $f'(x) = 0$, the function "does not change", though what this precisely means depends on the situation (we shall return to this in Section 2.5). Figure 6 illustrates each of these cases.

Result 2.4 (DIFFERENTIATION OF ELEMENTARY FUNCTIONS). Let $a \in \mathbb{R}$ be a constant.

$$\begin{aligned}
f(x) = ax^n &\implies f'(x) = anx^{n-1} \quad [n \in \mathbb{R}] && (2.8)\\
f(x) = c &\implies f'(x) = 0 \quad [c \in \mathbb{R}] && (2.9)\\
f(x) = a\log(x) &\implies f'(x) = ax^{-1} \quad [x > 0] && (2.10)\\
f(x) = ae^x &\implies f'(x) = ae^x && (2.11)\\
f(x) = a\sin(x) &\implies f'(x) = a\cos(x) && (2.12)\\
f(x) = a\cos(x) &\implies f'(x) = -a\sin(x) && (2.13)
\end{aligned}$$

In Example 2.3 we derived (2.8) for $n \in \{0,1,2\}$. The proof is similar for general $n \in \mathbb{N}\cup\{0\}$.

Example 2.5. Let $f(x) = x^n$ for $n \in \mathbb{N}$. First, observe that

$$\begin{aligned}
(x+h)(x+h)(x+h) = (x+h)(x^2+2xh+h^2) &= x(x^2+2xh+h^2) + h(x^2+2xh+h^2)\\
&= x^3 + 2x^2h + xh^2 + hx^2 + h(2xh+h^2)\\
&= x^3 + 3x^2h + h^2a_3,
\end{aligned}$$

where $a_3 = 3x+h$. Multiplying this with $x+h$ gives us

$$\begin{aligned}
(x+h)^4 = x(x^3+3x^2h+h^2a) + h(x^3+3x^2h+h^2a) &= x^4 + 3x^3h + xh^2a + hx^3 + 3x^2h^2 + h^3a\\
&= x^4 + 4x^3h + h^2a_4,
\end{aligned}$$

where $a_4 = xa + 3x^2 + ah$. Iterating this procedure gives us $(x+h)^n = x^n + nx^{n-1}h + h^2a_n$. Therefore

$$\begin{aligned}
f'(x) = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h} = \lim_{h\to 0}\frac{(x+h)^n - x^n}{h} &= \lim_{h\to 0}\frac{x^n + nx^{n-1}h + h^2a_n - x^n}{h}\\
&= \lim_{h\to 0}(nx^{n-1} + ha_n)\\
&= nx^{n-1},
\end{aligned}$$

which is (2.8). The general form of $(x+h)^n$ is given by the *binomial theorem* (suom. *binomikaava*), which states that

$$(x+h)^n = \sum_{k=0}^{n}\binom{n}{k}x^{n-k}h^n, \quad \text{where} \quad \binom{n}{k} = \frac{n!}{k!(n-k)!}. \quad (2.14)$$

Here $n! = 1\cdot 2\cdots(n-1)\cdot n$ is the *factorial* (suom. *kertoma*) and we define $0! = 1$. The term $nx^{n-1}h$ that played a crucial role in computation of the derivative is the $k = 1$ term in the binomial theorem since $1! = 1$ and $n!/(n-1)! = n$.

Above we have implicitly assumed that f is *differentiable* (suom. *derivoituva*) at x. That is, that the limit in (2.2) in fact exists. The following remark shows that this need not be the case.

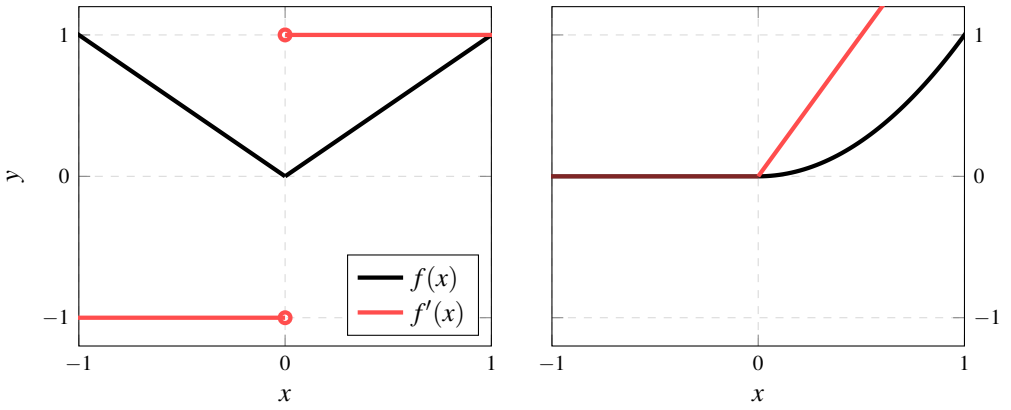

Figure 7: *Left*: The absolute value function $f(x) = |x|$. *Right*: The function in (2.17). We see that the derivative of the absolute value function has a discontinuity at $x = 0$ and is not well-defined at this point. The derivative of the piecewise-defined function in (2.17) is well-defined everywhere. However, its *second* derivative (see Definition 2.7) would have a discontinuity at $x = 0$.

Remark 2.6. Consider the absolute value function $f(x) = |x|$ plotted in Figure 4. Let us try to differentiate it at $x = 0$. For any $h > 0$ the difference quotient in (2.1) is

$$\frac{f(0+h) - f(0)}{h} = \frac{|h| - |0|}{h} = \frac{h}{h} = 1. \tag{2.15}$$

But for $h < 0$ we get (since in this case $|h| = -h$)

$$\frac{f(0+h) - f(0)}{h} = \frac{|h| - |0|}{h} = \frac{-h}{h} = -1. \tag{2.16}$$

This means that there is no single number c to which the difference quotient would tend to as $|h|$ is taken closer and closer to zero. The other three functions in Figure 4 behave similarly at $x = 0$. It should be clear from the figure that the rate of change of any of these functions at $x = 0$ cannot be defined unambiguously, as the rate depends on whether one approaches from left or right. Note however that a function being piecewise defined does not imply that it is not differentiable at the changepoint. For example, the function

$$f(x) = \begin{cases} 0 & \text{if } x < 0, \\ x^2 & \text{if } x \geq 0. \end{cases} \tag{2.17}$$

is differentiable at $x = 0$. By (2.8), its derivative is $f'(x) = 0$ if $x < 0$ and $f'(x) = 2x$ if $x > 0$. Since its derivatives to the left and right of 0 are equal at 0, the function is differentiable also at $x = 0$ and we may write its piecewise defined derivative as

$$f'(x) = \begin{cases} 0 & \text{if } x < 0, \\ 2x & \text{if } x \geq 0. \end{cases} \tag{2.18}$$

Figure 7 shows the functions discussed in this example along with their derivatives.

The derivative itself is a function that can be differentiated.

Definition 2.7 (HIGHER-ORDER DERIVATIVES). The nth derivative of a function f is defined recursively as

$$f^{(n)}(x) = \frac{\mathrm{d}}{\mathrm{d}x} f^{(n-1)}(x), \quad \text{where } f^{(0)} = f. \tag{2.19}$$

Note that $f = f^{(0)}$ and $f' = f^{(1)}$. The second derivative $f^{(2)}$ is usually denoted f''. The nth derivative is also written as

$$f^{(n)}(x) = \frac{\mathrm{d}^n f}{\mathrm{d}x^n}(x) = \frac{\mathrm{d}^n}{\mathrm{d}x^n} f(x). \tag{2.20}$$

Just as the derivative describes how the function changes, the nth derivative describes how the $(n-1)$th derivative changes.

Example 2.8. Consider the polynomial $f(x) = x^5$. By (2.8) with $n = 5$ and $a = 1$, the derivative of this polynomial is

$$f'(x) = f^{(1)}(x) = 5x^{5-1} = 5x^4. \tag{2.21}$$

Applying (2.8) to f', now with $n = 4$ and $a = 4$, then yields

$$f''(x) = f^{(2)}(x) = \frac{\mathrm{d}}{\mathrm{d}x} f'(x) = 5 \cdot 4x^{4-1} = 5 \cdot 4x^3. \tag{2.22}$$

Carrying on in this manner, we obtain the fifth derivative

$$f^{(5)}(x) = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1x^0 = 5!, \tag{2.23}$$

where $5! = 120$ is the factorial from the end of Example 2.5. Because the 5th derivative is a constant function, Equation (2.9) implies that $f^{(6)}$ and all subsequent derivatives are zero.

Example 2.9. Consider the function $f(x) = \sin(x)$. From (2.12) we get

$$f'(x) = f^{(1)}(x) = \cos(x). \tag{2.24}$$

Equation (2.13) then yields

$$f''(x) = f^{(2)}(x) = \frac{\mathrm{d}}{\mathrm{d}x} f'(x) = -\sin(x). \tag{2.25}$$

Similarly,

$$f^{(3)}(x) = \frac{\mathrm{d}}{\mathrm{d}x} f''(x) = -\cos(x) \tag{2.26}$$

and

$$f^{(4)}(x) = \frac{\mathrm{d}}{\mathrm{d}x} f^{(3)}(x) = -(-\sin(x)) = \sin(x). \tag{2.27}$$

Here we observe a pattern emerge:

$$f^{(2n)}(x) = (-1)^n \sin(x) \quad \text{and} \quad f^{(2n+1)}(x) = (-1)^n \cos(x) \tag{2.28}$$

for every $n \geq 0$.

2.2 Differentiation Rules

Result 2.4 provided derivatives of elementary functions. The following differentiation rules are used to differentiate more complicated functions formed out of these elementary functions.

Result 2.10 (LINEARITY). Let $f(x) = ag(x) + bh(x)$ for functions g and h and constants $a, b \in \mathbb{R}$. Then

$$f'(x) = ag'(x) + bh'(x). \tag{2.29}$$

Result 2.11 (PRODUCT RULE). Let $f(x) = g(x)h(x)$. The *product rule* (suom. *tulon derivoimissääntö*) states that

$$f'(x) = g'(x)h(x) + g(x)h'(x). \tag{2.30}$$

Example 2.12. The product rule (2.30) and Result 2.4 yield the following derivatives:

- Let $f(x) = xe^x = g(x)h(x)$ for $g(x) = x$ and $h(x) = e^x$. By (2.8) and (2.11), $g'(x) = 1$ and $h'(x) = e^x$. Therefore (2.30) gives

$$f'(x) = g'(x)h(x) + g(x)h'(x) = 1 \cdot e^x + x \cdot e^x = e^x(1+x). \tag{2.31}$$

- Let $f(x) = e^x \sin(x) = g(x)h(x)$ for $g(x) = e^x$ and $h(x) = \sin(x)$. By (2.11) and (2.12), $g'(x) = e^x$ and $h'(x) = \cos(x)$. Therefore (2.30) gives

$$f'(x) = g'(x)h(x) + g(x)h'(x) = e^x \cdot \sin(x) + e^x \cdot \cos(x) = e^x[\sin(x) + \cos(x)]. \tag{2.32}$$

- Let $f(x) = e^{2x} = e^x \cdot e^x = g(x)h(x)$ for $g(x) = e^x$ and $h(x) = e^x$. By (2.11), $g'(x) = e^x$ and $h'(x) = e^x$. Therefore (2.30) gives

$$f'(x) = g'(x)h(x) + g(x)h'(x) = e^x \cdot e^x + e^x \cdot e^x = 2e^{2x}. \tag{2.33}$$

Result 2.13 (QUOTIENT RULE). Let $f(x) = g(x)/h(x)$. The *quotient rule* (suom. *osamäärän derivoimissääntö*) states that

$$f'(x) = \frac{g'(x)h(x) - g(x)h'(x)}{h(x)^2} \quad \text{if} \quad h(x) \neq 0. \tag{2.34}$$

Example 2.14. The quotient rule (2.34) and Results 2.4 and 2.10 yield the following derivatives:

$$f(x) = \frac{x}{e^x} \implies f'(x) = \frac{e^x - xe^x}{e^{2x}} \qquad [g(x) = x \text{ and } h(x) = e^x]$$

$$f(x) = \frac{\cos(x)}{1+x^2} \implies f'(x) = -\frac{(1+x^2)\sin(x) + 2x\cos(x)}{(1+x^2)^2} \qquad [g(x) = \cos(x) \text{ and } h(x) = 1+x^2]$$

$$f(x) = e^{-x} \implies f'(x) = -e^{-x} \qquad [g(x) = 1 \text{ and } h(x) = e^x]$$

2.3 Chain Rule

Recall from Section 1.4 that $f = h \circ g$ denotes the composite function given by $f(x) = h(g(x))$. The *chain rule* is used to differentiate composite functions.

Result 2.15 (CHAIN RULE). Let $f = h \circ g$. The *chain rule* (suom. *ketjusääntö*) states that

$$f'(x) = (h \circ g)'(x) = (h' \circ g)(x)g'(x) = h'(g(x))g'(x). \tag{2.35}$$

The procedure to apply chain rule to compute $f'(x)$ goes as follows:

1. Compute the derivative h' of h.
2. Compute $h'(g(x))$, the value of this derivative at $g(x)$.
3. Multiply by $g'(x)$, the derivative of g at x.

Leibniz's notation provides an easy mnemonic for the chain rule. By treating $\mathrm{d}f$, $\mathrm{d}x$, and other such quantities as numbers, we can formally write

$$f' = \frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\mathrm{d}f}{\mathrm{d}g} \cdot \frac{\mathrm{d}g}{\mathrm{d}x}. \tag{2.36}$$

Here we first differentiate f with respect to g, which means that we treat g as the argument to f. Because $f = h \circ g = h(g)$, this gives us

$$\frac{\mathrm{d}f}{\mathrm{d}g} = \frac{\mathrm{d}h}{\mathrm{d}g}(g) = h'(g) = h' \circ g. \tag{2.37}$$

From (2.36) we then obtain the chain rule in (2.35):

$$f'(x) = \frac{\mathrm{d}f}{\mathrm{d}x}(x) = \frac{\mathrm{d}f}{\mathrm{d}g}(x) \cdot \frac{\mathrm{d}g}{\mathrm{d}x}(x) = (h' \circ g)(x)g'(x) = h'(g(x))g'(x). \tag{2.38}$$

Again, these are purely formal computations (i.e., we treat and manipulate the derivative as if it were the quetient of two numbers, forgetting the presence of a limit in its definition). Although the result is correct, Equation (2.36) is *not* a rigorous mathematical proof of the chain rule.

Example 2.16. Let $h(x) = \log(x)$ and $g(x) = x^n$. By the chain rule and (2.8) and (2.10),

$$f'(x) = (h \circ g)'(x) = h'(g(x))g'(x) = \frac{1}{g(x)} \cdot g'(x) = \frac{1}{x^n} \cdot nx^{n-1} = \frac{n}{x}. \tag{2.39}$$

Note that we obtain the same derivative from $\log(x^n) = n\log(x)$ and Result 2.10.

Example 2.17. Let $f(x) = \exp(-x^2)$. Set $h(x) = \exp(x)$ and $g(x) = -x^2$, so that $f = h \circ g$. Then the chain rule and Results 2.4 and 2.10 yield

$$f'(x) = h'(g(x))g'(x) = \exp(g(x)))g'(x) = \exp(-x^2) \cdot (-2x) = -2x\exp(-x^2). \tag{2.40}$$

Example 2.18. Let $f = h(ax)$ for a function h and a constant $a \in \mathbb{R}$. By setting $g(x) = ax$, we obtain from the chain rule, Equation (2.8) with $n = 1$ and Result 2.10 that

$$f'(x) = h'(g(x))g'(x) = ah'(ax). \tag{2.41}$$

Note that this differs from computing $h'(ax)$, the derivative of h at point ax.

Example 2.19. The derivative of the logarithm in (2.10) can be derived from the chain rule and $\mathrm{d}e^x/\mathrm{d}x = e^x$. Let $g(x) = e^x$ and $h(x) = \log(x)$. Then $f(x) = h(g(x)) = \log(e^x) = x$, so that $f'(x) = 1$ by (2.8). But we can alternatively apply the chain rule, which gives

$$f'(x) = h'(g(x))g'(x) = \left(\frac{\mathrm{d}h}{\mathrm{d}x}(e^x)\right)e^x = h'(e^x)\,e^x. \tag{2.42}$$

Since we know that (2.42) equals 1, division by e^x gives us

$$h'(e^x) = e^{-x}. \tag{2.43}$$

Note that this equation says the the derivative of the logarithm h' equals e^{-x} when evaluated at e^x. By selecting $x = \log(z)$, so that $e^{\log z} = z$ and $e^{-x} = e^{-\log z} = z^{-1}$, we obtain

$$h'(z) = z^{-1}, \tag{2.44}$$

which is (2.10).

2.4 Automatic Differentiation

While it is good to have some grasp of the basic differentiation rules, the derivatives involved in applications quickly become practically impossible (or at least extremely tedious) to compute by hand. It would be preferable to have computer take care of differentiation. This is possible in three different ways.

1. Numerical differentiation. In *numerical differentiation* the limit $h \to 0$ in the definition (2.2) of derivative is replaced with an h close to zero. That is, the derivative is approximation as the difference quotient

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}, \qquad \text{where} \qquad h \approx 0. \tag{2.45}$$

This approach is conceptually straightforward and simple to implement (at least in a very naive way). However, numerical differentiation provides only an approximation to the derivative, can introduce round-off errors, and does not scale well to higher-order derivatives and the multivariate setting.

2. Symbolic differentiation. Symbolic computation packages can be used to perform *symbolic differentiation*, which automates the process of computing symbolic expressions for derivatives. For example, inputting (here sinh and cosh are the *hyperbolic* sine and cosine functions)

```
differentiate sin(log(x^(sqrt(exp(-cos(x)))) + x^2) + cosh(x))
```

to WolframAlpha yields the lovely expression

$$\cos\left(\log(x^{\sqrt{\exp(-\cos(x))}} + x^2) + \cosh(x)\right) \times \left(\frac{x^{\sqrt{\exp(-\cos(x))}}[x^{-1}\sqrt{\exp(-\cos(x))} + \frac{1}{2}\log(x)\sin(x)\sqrt{\exp(-\cos(x))}\,] + 2x}{x^{\sqrt{\exp(-\cos(x))}} + x^2} + \sinh(x)\right)$$

for the derivative of the function

$$f(x) = \sin\left(\log(x^{\sqrt{\exp(-\cos(x))}} + x^2) + \cosh(x)\right). \tag{2.46}$$

This is quite impressive and useful if one needs to study how the derivative behaves as a part of some mathematical problem (or when it is part of an exercise to compute a derivative). However, computing such symbolic expression is computationally expensive and *completely unnecessary*. For in practice it is not a symbolic expression such as this that is needed *but values of a derivative at certain points*. To compute $f'(x_0)$ at a given point x_0 (e.g., $x_0 = 3$) it is not necessary to have access to a full symbolic expression for the derivative.

3. Automatic differentiation. *Automatic differentiation*[3] utilises the fact that complicated functions reduce to a sequence of compositions and arithemtic operations involving the elementary functions whose derivatives are available in Result 2.4. The chain rule and other differentiation rules then permit algorithmically straightforward computation of derivatives of arbitrarily complicated functions. For example, consider a function f given by

$$f(x) = h(g(h(x)g(x)) + h(x)). \tag{2.47}$$

We may write this function as

$$f(x) = h(r(x)), \tag{2.48}$$

where

$$r(x) = u(x) + h(x), \quad u(x) = g(s(x)), \quad \text{and} \quad s(x) = h(x)g(x). \tag{2.49}$$

By applying the differentiation rules from Sections 2.2 and 2.3 we can write the derivative of f as

$$f'(x) = h'(r(x))r'(x), \tag{2.50}$$

where

$$r'(x) = u'(x) + h'(x), \quad u'(x) = g'(s(x))s'(x) \quad \text{and} \quad s'(x) = h'(x)g(x) + h(x)g'(x). \tag{2.51}$$

If h and g are elementary functions appearing in Result 2.4, we can easily compute $s'(x_0)$ at any point x_0 we want. Subsequently we can use $u'(x_0) = g'(s(x_0))s'(x_0)$ to obtain the derivative $u'(x_0)$, which then gives us $r'(x_0)$. Finally, we can use $r'(x_0)$ in (2.50) to compute $f'(x_0)$. To obtain $f'(x_1)$ we simply repeat the process with $x = x_1$. At no point do we require access to a symbolic expression for $f'(x)$ or the intermediate derivatives $r'(x)$, $u'(x)$ and $s'(x)$.

Implementing very basic automatic differentiation is surprisingly simple. The following Python code suffices to differentiate the function in (2.46). This is the so-called *forward mode* of automatic differentiation.

```
import numpy as np

class fDf:
    def __init__(self, val, deriv):
        self.val = val
        self.deriv = deriv
    def __add__(self, u):
        return fDf(self.val + u.val, self.deriv + u.deriv)
    def __mul__(self, u):
        return fDf(self.val * u.val, self.deriv * u.val + self.val * u.deriv)
    def __rmul__(self, c):
        return fDf(c * self.val, c * self.deriv)

def exp(u):
    return fDf(np.exp(u.val), np.exp(u.val) * u.deriv)
```

[3]The following article gives a good introduction to automatic differentiation: NEIDINGER (2010). Introduction to automatic differentiation and MATLAB object-oriented programming. *SIAM Review* 52(3):545–563.

```
def sin(u):
    return fDf(np.sin(u.val), np.cos(u.val) * u.deriv)
def log(u):
    return fDf(np.log(u.val), (1.0 / u.val) * u.deriv)
def sqrt(u):
    return fDf(np.sqrt(u.val), (0.5 / np.sqrt(u.val)) * u.deriv)
def cosh(u):
    return fDf(np.cosh(u.val), np.sinh(u.val) * u.deriv)

x = fDf(3.0, 1.0)
fDfx = sin(log(exp(log(x) * sqrt(exp(-1.0*cos(x)))) + x * x) + cosh(x))
Dfx = fDfx.deriv
```

We first define a class `fDf` which holds the value and derivative of a function at a point of interest. By overloading addition and multiplication we can perform arithmetic on this class. For example, for two instances `f` and `g` of the class `fDf`, which store $f(x)$ and $g(x)$ as well as $f'(x)$ and $g'(x)$ at some point x, the multiplication `f * g` is an instance of `fDf` that stores the function value $f(x)g(x)$ and the derivative $f'(x)g(x)+f(x)g'(x)$ at x that has been computed using the product rule (Result 2.11). We then define a number of functions that operate on instances of `fDf`. Each of these functions calls the corresponding `numpy` to perform function evaluation and implements the chain rule (Result 2.15) for derivative evaluation. For example, `exp` assumes that it is given a function $u(x)$ and its derivative $u'(x)$ at some point x and uses these to compute the function evaluation $e^{u(x)}$ and, via the chain rule, the derivative evaluation $e^{u(x)}u'(x)$. Wit this code we can differentiate any function that involves sums, products and compositions of exp, sin, cos, log, square root and cosh. At the end we specify that we want to differentiate the function in (2.46) at $x_0 = 3$. The line `x = fDf(3.0, 1.0)` does this by defining that the innermost function in automatic differentiation is $u(x) = x$ and that $u(x_0) = 3$ and $u'(x_0) = 1$ since $u'(x) = 1$. Note that in defining the function being differentiated we use $x^{u(x)} = \exp(u(x)\log(x))$ in order to avoid having to define a differentiation rule for $x^{u(x)}$.

What makes automatic differentiation very powerful is the fact that the function being differentiated does not have to a simple algebraic expression. For example, with the above code we could easily differentiate functions that have been defined using loops and conditional statements. Of course, one should not use home-made automatic differentiation routines. One popular option is `JAX` (https://jax.readthedocs.io/), the usage of which to differentiate (2.46) is illustrated by the following code.

```
import jax.numpy as jnp
from jax import grad

def f(x):
    return jnp.sin(jnp.log(jnp.exp(jnp.log(x) *
        jnp.sqrt(jnp.exp(-1.0*jnp.cos(x)))) + x**2) + jnp.cosh(x))

Df = grad(f)
x = 3.0
Dfx = Df(x)
```

The derivative `Dfx = Df(x)` is now equal to the derivative `Dfx = fDfx.deriv` computed by our earlier code. The crucial JAX function `grad` performs automatic differentiation. Its name stands for "gradient", the multivariate generalisation of derivative that we shall learn about in Section 3. We will return to JAX in **??**. Figure 8 shows the function in (2.46) and its derivative that has been computed using `JAX`.

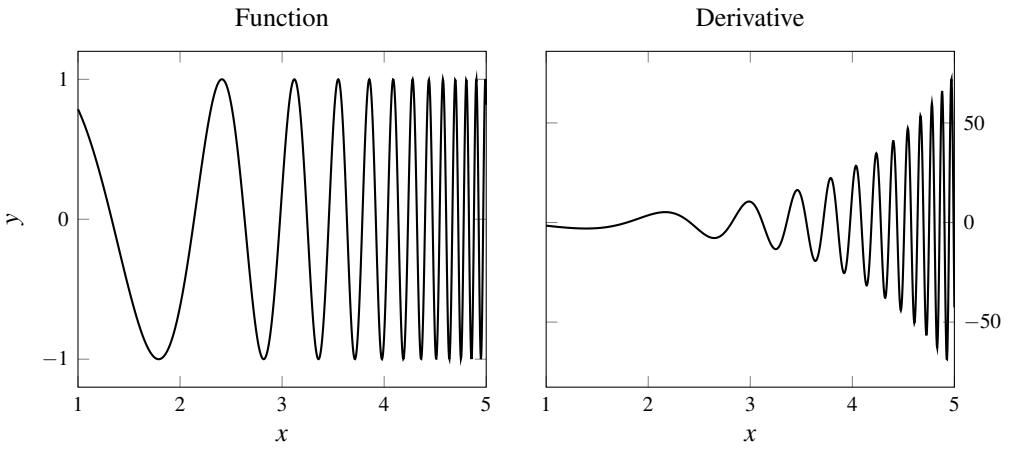

Figure 8: The function in (2.46) and its derivative that has been computed using `JAX`.

2.5 Univariate Local Optimisation

The purpose of Sections 2.1 to 2.4 has been to review some prerequisites for *local optimisation* (suom. *paikallinen optimointi*) of univariate functions. The goal of optimisation is to find point (or points) x^* at which a function f of interest attains its smallest or largest value. In machine learning f is usually a loss function that is to be minimised (recall the examples in Section 1.1). We therefore focus on minimisation. Note that this comes at no loss of generality because

$$\text{maximising } f \quad \text{is equivalent to} \quad \text{minimising } -f.$$

Definition 2.20 (Local and global minima). Let $f\colon \mathbb{R} \to \mathbb{R}$ be a function. A point $x^* \in \mathbb{R}$ is a

- *global minimum point* of f if $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}$;
- *local minimum point* of f if there exists $\varepsilon > 0$ such that $f(x^*) \leq f(x)$ for every point x in the interval $[x^* - \varepsilon, x^* + \varepsilon]$ of length 2ε centered at x^*.

If x^* is a global minimum point, $f(x^*)$ is called the *global minimum* (suom. *globaali minimi*). If x^* is a local minimum point, $f(x^*)$ is called a *local minimum* (suom. *paikallinen minimi*).

Global and local maxima are defined analogously, with "$\geq$" replacing "$\leq$" in the definitions. Minima and maxima are collectively known as *extrema* (suom. *ääriarvo*) and minimum and maximum points as *extremal points* (suom. *ääriarvopiste*).

A function attains its smallest possible value at a global minimum point. At a local minimum point a function attains a value that is smaller or equal to its "nearby" values. The definition of a local minimum point says simply that there is *some* interval around x^* on which $f(x) \leq f(x^*)$. Note that a global minimum point is also a local minimum point. Figure 9 shows global and local minima of a particular function. We focus on local optimisation as opposed to *global optimisation*. That is, we are happy find a local minimum point. Finding a global minimum point is significantly more challenging. Points at which the derivative of a function is zero and local minimum and maximum points are intimately connected.

Definition 2.21 (Critical point). A point $x^* \in \mathbb{R}$ is a *critical point* (suom. *kriittinen piste*), or a *stationary point*, of differentiable a function f if $f'(x^*) = 0$.

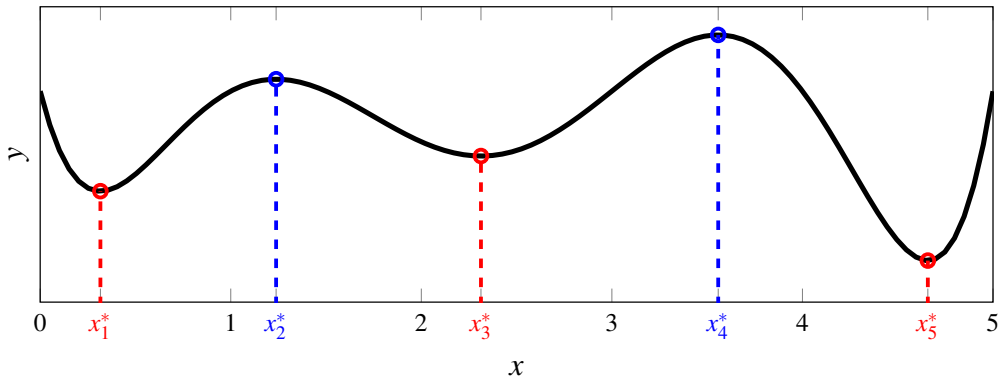

Figure 9: The local minimum points (red; x_1^*, x_3^*, and x_5^*) and maximum points (blue; x_2^* and x_4^*) of a certain function. The local minimum point x_5^* is also a *global* minimum point. Although this cannot be determined from the figure, this function has no global maximum point as it tends to ∞ as $x \to -\infty$ and $x \to \infty$.

Example 2.22. Consider the functions $f(x) = \cos(x)$ and $g(x) = \frac{1}{3}x^3 - x$. Equation (2.13) gives $f'(x) = -\sin(x)$. From trigonometry we know that $\sin(x) = 0$ if and only if $x = \pi n$ for n an integer (i.e., $n \in \mathbb{Z}$). Therefore $f'(x) = 0$ if and only if $x = \pi n$ for an integer n. These are thus the critical points of $f(x) = \cos(x)$. That is, the cosine has an infinite number of critical points. Equation (2.8) gives $g'(x) = x^2 - 1$, so that $g'(x) = 0$ if and only if $x = 1$ or $x = -1$. Therefore the polynomial $g(x) = \frac{1}{3}x^3 - x$ has two critical points, $x^* = -1$ and $x^* = 1$.

Result 2.23 (FERMAT'S THEOREM). Every local extremal point x^* of a differentiable function f is a critical point, in that $f'(x^*) = 0$.

We can see Result 2.23 in action in Figure 9: at each extremal point the tangent line would be completely horizontal and the derivative hence zero. This result furnishes the basic principle behind local optimisation methods: find a point at which the derivative is zero. However, Result 2.23 only says that extremal points are critical points, not that every critical point is an extremal point. Whether or not a critical point is an extremal point can be usually determined by examining the derivative between critical points.

Result 2.24 (CLASSIFICATION OF CRITICAL POINTS). Let f be a differentiable function. Suppose that $x_1^* < x_2^* < x_3^*$ are critical points of f.

1. If $f'(x) > 0$ for all $x_1^* < x < x_2^*$ and $f'(x) > 0$ for all $x_2^* < x < x_3^*$, then x_2^* is *not an extremal point*.
2. If $f'(x) < 0$ for all $x_1^* < x < x_2^*$ and $f'(x) < 0$ for all $x_2^* < x < x_3^*$, then x_2^* is *not an extremal point*.
3. If $f'(x) > 0$ for all $x_1^* < x < x_2^*$ and $f'(x) < 0$ for all $x_2^* < x < x_3^*$, then x_2^* is a *local maximum point*.
4. If $f'(x) < 0$ for all $x_1^* < x < x_2^*$ and $f'(x) > 0$ for all $x_2^* < x < x_3^*$, then x_2^* is a *local minimum point*.

Recall that the sign of its derivative tells us whether a function is increasing or decreasing. In Cases 1 and 2 of Result 2.24 the function is increasing (Case 1) or decreasing (Case 2) both to

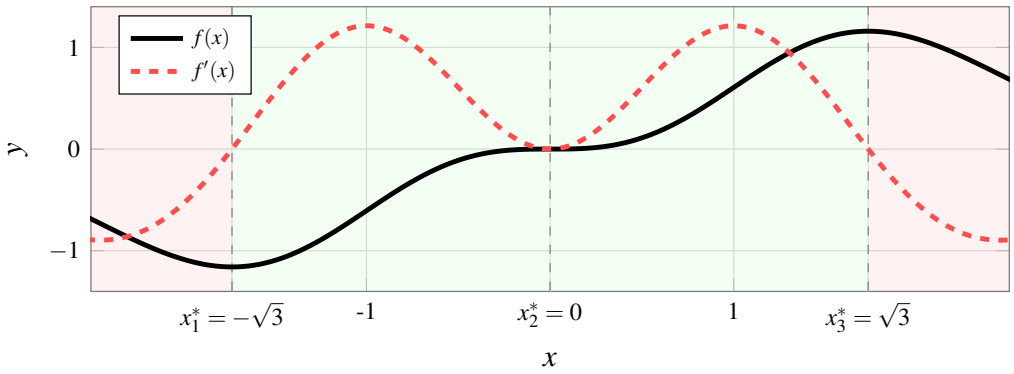

Figure 10: The function $f(x) = x^3 \exp(-\frac{1}{2}x^2)$ and its derivative in (2.53). The function is increasing (i.e., derivative is positive) on the green interval and decreasing (i.e., derivative is negative) on the two red intervals. The derivative vanishes at x_1^*, x_2^*, and x_3^*, which means that these are critical points of f. The critical points x_1^* and x_3^* are extremal points, but the critical point $x_2 = 0$ is not because f is increasing both the left and right of x_2^*.

the left and right of the critical point x_2^*. Therefore x_2^* cannot be an extremal point. In Case 3 the function is increasing to the left of x_2^* and decreasing to the right of it, meaning that it must attain a local maximum at x_2^*. In Case 4 the function is decreasing to the left and increasing to the right, so that it must have a local minimum at x_2^*. The following example and Figure 10 illustrate this.

Example 2.25. Let us classify the critical points of the function

$$f(x) = x^3 \exp(-\tfrac{1}{2}x^2). \tag{2.52}$$

Using the differentiation rules in Sections 2.1 to 2.3 we compute

$$f'(x) = (3x^2 - x^4)\exp(-\tfrac{1}{2}x^2). \tag{2.53}$$

Since the function $\exp(-\frac{1}{2}x^2)$ is everywhere positive, $f'(x) = 0$ if and only if $3x^2 - x^4 = 0$. The equation

$$3x^2 - x^4 = x^2(3 - x^2) = 0 \tag{2.54}$$

has the three solutions $x = 0$, $x = \sqrt{3}$, and $x = -\sqrt{3}$. The critical points of f are therefore

$$x_1^* = -\sqrt{3}, \quad x_2^* = 0, \quad \text{and} \quad x_3^* = \sqrt{3}. \tag{2.55}$$

Let us begin with x_2^*. The sign of the derivative (2.53) is equal to the sign of $3x^2 - x^4$. By writing this as $x^2(3 - x^2)$, we observe that

$$f'(x) > 0 \quad \text{for all} \quad -\sqrt{3} < x < 0 \quad \text{and} \quad 0 < x < \sqrt{3}. \tag{2.56}$$

Therefore f' does *not* change sign at $x_2^* = 0$ and thus x_2^* is not an extremal point by Case 1 of Result 2.24. For $x_1^* = -\sqrt{3}$ and $x_3^* = \sqrt{3}$ we note that

$$f'(x) < 0 \quad \text{for all} \quad x < -\sqrt{3} \quad \text{and } x > \sqrt{3}. \tag{2.57}$$

Therefore f is decreasing to the left of x_1^* and increasing to the right of it, meaning that x_1^* is a local (in fact, global) minimum point. Conversely, f is increasing to the left of x_3^* and increasing to the right of it, meaning that x_3^* is a local (in fact, global) maximum point. Figure 10 makes an attemtp at showing what is going on.

Result 2.26 (**MINIMUM OF A POLYNOMIAL**). Let $P(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} + a_n x^n$ be a polynomial of degree n. Suppose that $a_n \neq 0$. Then P has a global minimum if and only if n is even and $a_n > 0$.

When minimising a function it is natural to move to a direction in which the function decreases. This basic idea is behind the following *gradient descent* (suom. *gradienttimenetelmä*) algorithm. In the multivariate version of this algorithm (**??**) derivatives are replaced by gradients, their generalistions to higher dimensions.

Algorithm 2.27 (**UNIVARIATE GRADIENT DESCENT**). The following algorithm attempts to find a local minimum point of a function f with derivative f':

Require: function handle f', initial point x_0, learning rate $\eta > 0$, tolerance $t > 0$, maximum number of iterations $n_{\max} \in \mathbb{N}$

1: $n \leftarrow 0$
2: **while** $|f'(x_n)| \geq t$ **and** $n \leq n_{\max}$ **do**
3: $\quad x_{n+1} \leftarrow x_n - \eta f'(x_n)$
4: $\quad n \leftarrow n + 1$
5: **end while**
6: **return** x_{n+1}

In this very basic form the gradient descent algorithm is quite simple. The loop on lines 2–5 is executed until the derivative becomes smaller than some user-specified tolerance (e.g., $t = 10^{-6}$) or a maximum number of iterations is reached (which is to protect against infinite loops). Line 3 is the heart of the algorithm: at each iteration we move a step towards the direction opposite to the direction the derivative is pointing. For example, if $f(x_n) > 0$, the function is increasing at x_n, which means that we want to move to the left (i.e., take x_{n+1} smaller than x_n). The *learning rate* (suom. *oppimisnopeus*), or *step size*, η controls how large steps we take at each iteration. If η is too large, the algorithm may not work properly, "jumping around" too much. If the function has a local minimum, $f'(x_n)$ will hopefully get closer and closer to zero, so that the steps taken [i.e., $-\eta f'(x_n)$] become smaller and smaller.

Since the derivative is close to zero at the point returned by the algorithm (provided that tolerance t is small), this point is close to a critical point. Of course, a critical point is not necessarily a local extremal point, as we recall from Result 2.24 and Example 2.25. However, it is unlikely (though still possible) for gradient descent to terminate at a non-minimal critical point since at such a point the derivative does not change sign, so that gradient descent is consequently more likely than not to simply "jump over" a non-extremal critical point. Figure 11 shows gradient descent in action.

Remark 2.28. Critical points and Result 2.23 are useful only when f is differentiable. Consider the absolute value function $f(x) = |x|$ As we saw in Remark 2.6, one cannot differentiate this function at $x = 0$. However, it is clear that f has a global minimum point $x^* = 0$.

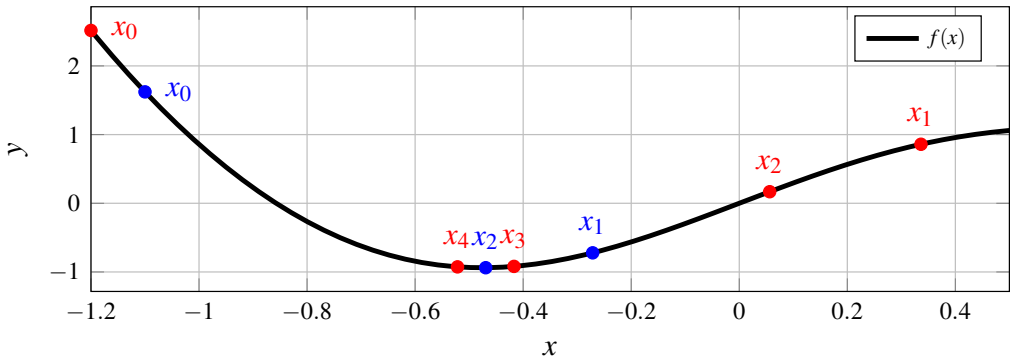

Figure 11: Gradient descent for the function $f(x) = x^4 + \sin(3x)$ with global minimum point $x^* \approx -0.47562$. The sequence of red points was produced by gradient descent with $x_0 = -1.2$ and $\eta = 0.16$; that of blue points with $x_0 = -1.1$ and $\eta = 0.1$. Observe that the point x_{n+1} is always to the right of x_n if f is decreasing at x_n and to the left of x_{n+1} if f is increasing at x_n (e.g., x_2 is to the left of x_1 since f is increasing at x_1).

2.6 Linearisation and Taylor Series

We can use the derivative at a point x_0 to form the linear approximation

$$f(x) \approx L(x) = f(x_0) + f'(x_0)(x - x_0) \tag{2.58}$$

to f around x_0. This approximation can be expected to be accurate only locally. That is, only for x close to x_0 [observe that the right-hand side reduces to $f(x_0)$ when $x = x_0$]. Finding the linear approximation L is called *linearisation* (suom. *linearisaatio*).

Example 2.29. We consider linearisation around $x_0 = 1$.

- Consider the function $f(x) = x^2$. Then $f(x_0) = 1^2 = 1$ and $f'(x_0) = 2 \cdot 1 = 2$ by (2.8). Therefore

$$L(x) = f(x_0) + f'(x_0)(x - x_0) = 1 + 2(x - 1) = 2x - 1. \tag{2.59}$$

- Consider the function $f(x) = \sin(3x)$. Then $f(x_0) = \sin(3 \cdot 1) \approx 0.1411$ and $f'(x_0) = 3\cos(3 \cdot 1) \approx -2.970$ by (2.12) and the chain rule (2.35). Therefore

$$L(x) = f(x_0) + f'(x_0)(x - x_0) = \sin(3) + 3\cos(3)(x - 1). \tag{2.60}$$

- Consider the function $f(x) = \exp(-3x)$. Then $f(x_0) = \exp(-3 \cdot 1) \approx 0.04978$ and $f'(x_0) = -3\exp(-3 \cdot 1) \approx -0.1494$ by (2.11) and the chain rule (2.35). Therefore

$$L(x) = f(x_0) + f'(x_0)(x - x_0) = e^{-3} - 3e^{-3}(x - 1). \tag{2.61}$$

These three functions and their linearisations are displayed in Figure 12.

Taylor polynomials generalise linearisation. Let $n \geq k$. By iterating (2.8), we get

$$\frac{\mathrm{d}^k}{\mathrm{d}x^k} x^n = n \frac{\mathrm{d}^{k-1}}{\mathrm{d}x^{k-1}} x^{n-1} = \cdots = n(n-1) \times \cdots \times (n-k+1) x^{n-k} = \frac{n!}{(n-k)!} x^{n-k}, \tag{2.62}$$

where $n! = 1 \cdot 2 \cdots (n-1) \cdot n$ is the factorial (note that $0! = 1$). Here it may be helpful to recall Example 2.8. If $k > n$, the kth derivative of x^n is zero since the nth derivative of x^n is and the

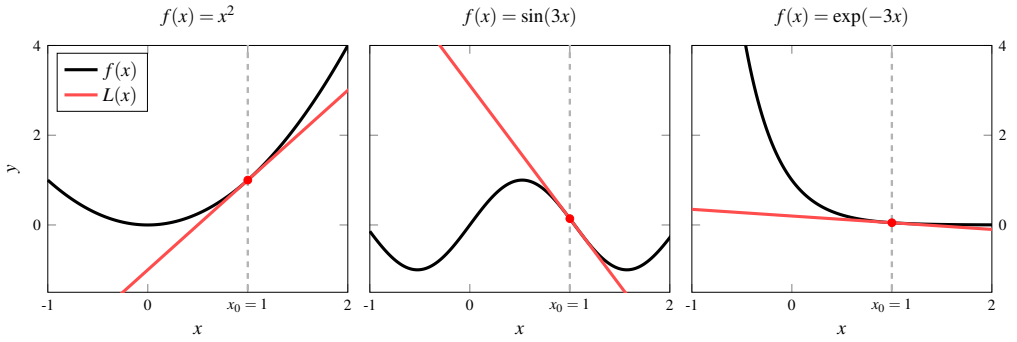

Figure 12: Three functions and their linearisations (2.58) around $x_0 = 1$. Observe how the linearisations are quite accurate [i.e., $L(x) \approx f(x)$] for x close to x_0 but approximate the function extremely badly further away from x_0.

derivative of a constant is zero. Under the notational convention $0^0 = 1$, we may write any monomial $P(x) = x^n$ as the series[4]

$$P(x) = \sum_{k=0}^{\infty} \frac{P^{(k)}(0)}{k!} x^k = \sum_{k=0}^{n} \frac{n!}{k!(n-k)!} (y^{n-k}|_{y=0}) x^k + \sum_{k=n+1}^{\infty} \frac{0}{k!} x^k = x^n, \tag{2.63}$$

where the last equation follows from the fact that $y^{n-k}|_{y=0} = 0$ if $k \neq n$. The expansion (2.63) extends to any polynomial of the form $P(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} + a_n x^n$ by linearity of differentiation. In fact, this expansion extends to many non-polynomial functions as well.

Definition 2.30 (TAYLOR POLYNOMIAL AND SERIES). The *Taylor polynomial* (suom. *Taylorin polynomi*) of degree n of a function f around a point x_0 is the function T_n given by

$$T_n(x) = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k. \tag{2.64}$$

The *Taylor series* (suom. *Taylorin sarja*) of a function f around a point x_0 is the Taylor polynomial of degree $n = \infty$:

$$T_\infty(x) = T_\infty(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k. \tag{2.65}$$

These definitions assume that f has sufficiently many derivatives. Taylor series around $x_0 = 0$ is sometimes called *Maclaurin series*.

Roughly speaking, an infinitely many times differentiable function is *analytic* (suom. *analyyttinen*) if it is equal to its Taylor series around some point x_0: $f(x) = (T_\infty f)(x)$ for all $x \in \mathbb{R}$.[5] The elementary function that one usually encounters are analytic.

[4] We do not attempt to provide a rigorous definition for $\sum_{k=1}^{\infty}$.

[5] The precise definition of an analytic function is somewhat more subtle: An infinitely differentiable function f is analytic on $\mathbb{R}$ if for every $x_0 \in \mathbb{R}$ there is $\varepsilon > 0$ (which may depend on x_0) such that $f(x)$ equals its the Taylor series $T_\infty(x)$ around x_0 for all x such that $|x - x_0| < \varepsilon$. The point is that the point x_0 around which the Taylor series is developed may depend on x if the Taylor series is to equal $f(x)$.

Example 2.31. Let $f(x) = e^x$ be the exponential function, which is analytic. From (2.11) we get $f^{(k)}(x) = e^x$ for every $k \geq 0$. Therefore we have the Maclaurin series (i.e., $x_0 = 0$)

$$e^x = \sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} x^k = \sum_{k=0}^{\infty} \frac{e^0}{k!} x^k = \sum_{k=0}^{\infty} \frac{1}{k!} x^k. \tag{2.66}$$

Example 2.32. Let $f(x) = \sin(x)$ be the sine function, which is analytic. Recall Example 2.9. From (2.12) and (2.13) we get $f^{(2k)}(x) = (-1)^k \sin(x)$ and $f^{(2k+1)}(x) = (-1)^k \cos(x)$. Because $\sin(0) = 0$ and $\cos(0) = 1$, we obtain the Maclaurin series

$$\sin(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} x^k = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1}. \tag{2.67}$$

In a similar manner,

$$\cos(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} x^{2k}. \tag{2.68}$$

By setting $n = 1$ we obtain the Taylor polynomial

$$T_1(x) = \frac{f^{(0)}(x_0)}{0!}(x - x_0)^0 + \frac{f^{(1)}(x_0)}{1!}(x - x_0)^1 = f(x_0) + f'(x_0)(x - x_0), \tag{2.69}$$

which is precisely the linear approximation L_{x_0} in (2.58). A special case of a more general *Taylor's theorem* now allows us to bound linearisation error.

Result 2.33 (LINEARISATION ERROR). Suppose that f is a differentiable function whose second derivative f'' is continuous. Let L be the linear approximation in (2.58). Then

$$|f(x) - L(x)| \leq \frac{C}{2}(x - x_0)^2, \tag{2.70}$$

where C is the maximum of $|f''(y)|$ over all y between x_0 and x.

As long as the second derivative of f is well-defined and not too large (i.e., f is sufficiently "nice"), linearisation error scales (at most) as a square of the distance to the linearisation point x_0.

2.7 Integration

The integral

$$\int_a^b f(x)\,\mathrm{d}x \tag{2.71}$$

gives the area between points a and b and bounded by the graph of f and the x-axis. When f takes negative values, the corresponding area is subtracted.[6] Figure 13 provides a simple illustration. One can think of integral as a "continuous" version of summation. Let $x_k = (b-a)k/n$ and $\Delta x = 1/n$ For "nice enough" functions it holds that

$$\sum_{k=1}^{n} f(x_k)\Delta x \approx \int_a^b f(x)\,\mathrm{d}x, \tag{2.72}$$

and the left-hand side converges to the intaegral as $n \to \infty$. Here we provide a very brief overview of integration. Integration is the reverse of differentiation.

[6]The rigorous definition (or definitions) of integral are outside the scope of this course. Equation (2.72) is the starting point to define the *Riemann integral*, while defining the more general and flexible *Lebesgue integral* requires the machinery of *measure theory* (suom. *mittateoria*).

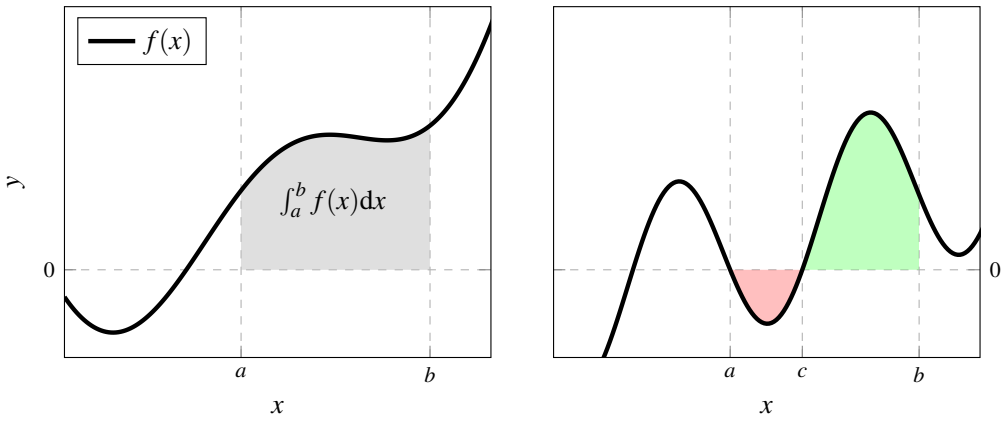

Figure 13: The integral of the function on the left from a to b is simply the area of the region (grey) between a and b bounded by the graph of f and the x-axis. When the function takes negative values on the domain of integration $[a,b]$, as in the right panel, the interal is obtained by subtracting the below x-axis (red) from the area above the axis (green).

Result 2.34 (FUNDAMENTAL THEOREM OF CALCULUS). Suppose that f is a continuous function and F a function such that $F'(x) = f(x)$. The *fundamental theorem of calculus* (suom. *analyysin peruslause*) states that

$$F(b) - F(a) = \int_a^b f(x)\,\mathrm{d}x. \tag{2.73}$$

Note that this result applies also to functions that change sign, such as the function in the second panel of Figure 13.

Result 2.35 (LINEARITY OF INTEGRATION). Let $f(x) = ag(x) + bh(x)$ for functions g and h and constants $a, b \in \mathbb{R}$. Then

$$\int_a^b f(x)\,\mathrm{d}x = \int_a^b [ag(x) + bh(x)]\,\mathrm{d}x = a\int_a^b g(x)\,\mathrm{d}x + b\int_a^b h(x)\,\mathrm{d}x. \tag{2.74}$$

The fundamental theorem of calculus provides a way to compute integrals that is conceptually straightforward: find a function F whose derivative equals f and evaluate this function at the end points a and b of integration. Unfortunately, to find this *antiderivative*, or *indefinite integral* (suom. *määräämätön integraali*), is much more difficult than differentiating a function. For example, it is easy to use the chain rule to compute that the derivative of $\exp(-\frac{1}{2}x^2)$ is $-x\exp(-\frac{1}{2}x^2)$, which we essentially did this in Example 2.17. However, $\exp(-\frac{1}{2}x^2)$ admits *no* elementary antiderivative. That is, a function F such that $F'(x) = \exp(-\frac{1}{2}x^2)$ cannot be expressed in terms of the functions appearing in Result 2.4 using a finite number of arithmetic operations.

The following result collects antiderivatives of the elementary functions in Result 2.4. If F is an indefinite integral of f, so is $F(x) + C$ for any constant C because the derivative of a constant function is zero.

Result 2.36 (ANTIDERIVATIVES OF ELEMENTARY FUNCTIONS). Let F be a function such that $F'(x) = f(x)$.

$$f(x) = x^n \quad \Longrightarrow \quad F(x) = \frac{1}{n+1}x^{n+1} + C \quad [n \in \mathbb{R}, n \neq -1] \tag{2.75}$$
$$f(x) = x^{-1} \quad \Longrightarrow \quad F(x) = \log(x) + C \tag{2.76}$$
$$f(x) = e^x \quad \Longrightarrow \quad F(x) = e^x + C \tag{2.77}$$
$$f(x) = \sin(x) \quad \Longrightarrow \quad F(x) = -\cos(x) + C \tag{2.78}$$
$$f(x) = \cos(x) \quad \Longrightarrow \quad F(x) = \sin(x) + C \tag{2.79}$$

Example 2.37. Let us verify two of the formulae in Result 2.36. Equation (2.75) follows from (2.8) since

$$\frac{\mathrm{d}}{\mathrm{d}x}F(x) = \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{n+1}x^{n+1} + C\right) = \frac{1}{n+1} \cdot \frac{\mathrm{d}}{\mathrm{d}x}x^{n+1} = \frac{1}{n+1}(n+1)x^n = x^n. \tag{2.80}$$

Equation (2.78) follows from (2.79) since

$$\frac{\mathrm{d}}{\mathrm{d}x}F(x) = \frac{\mathrm{d}}{\mathrm{d}x}[-\cos(x) + C] = -\frac{\mathrm{d}}{\mathrm{d}x}\cos(x) = -(-\sin(x)) = \sin(x). \tag{2.81}$$

Example 2.38. We can now use Results 2.34 and 2.36 to compute integrals. We may always ignore the constant C in Result 2.36 because it gets cancelled when computing $F(b) - F(a)$ in Result 2.34. For example,

$$\int_a^b x^n \,\mathrm{d}x = \frac{1}{n+1}b^{n+1} - \frac{1}{n+1}a^{n+1} = \frac{1}{n+1}(b^{n+1} - a^{n+1}) \tag{2.82}$$

for any $a < b$ by (2.75) and

$$\int_0^{2\pi} \sin(x) \,\mathrm{d}x = -\cos(2\pi) - (-\cos(0)) = -1 - (-1)) = 0 \tag{2.83}$$

by (2.78) and $\cos(2\pi) = 1$. To understand why the latter integral is zero you may want to examine the graph of $\sin(x)$ and recall that areas below the x-axis are to be subtracted when computing the integral (see also the right panel in Figure 13).

Integration by substitution or *change of variables* (suom. *sijoitusmenetelmä* tai *muuttujanvaihto*) is a powerful technique to compute integrals. Integration by substitution is very useful in probability theory (though during this course we will not get that far enough in probability to witness this).

Result 2.39 (INTEGRATION BY SUBSTITUTION). Let f be continuous and g a differentiable function such that g' is continuous. Then

$$\int_{g(a)}^{g(b)} f(x) \,\mathrm{d}x = \int_a^b f(g(y))g'(y) \,\mathrm{d}y. \tag{2.84}$$

Proof. Let F be an antiderivative of f. The chain rule (2.35) then gives

$$(F \circ g)'(y) = F'(g(y))g'(y) = f(g(y))g'(y). \tag{2.85}$$

Therefore Result 2.34 yields

$$\int_a^b f(g(y))g'(y)\,\mathrm{d}y = (F\circ g)(b) - (F\circ g)(a) = F(g(b)) - F(g(a)), \tag{2.86}$$

which, by proceeding from left to right in (2.73), is equal to the integral

$$\int_{g(a)}^{g(b)} f(x)\,\mathrm{d}x \tag{2.87}$$

because $F'(x) = f(x)$. □

It is not particularly intuitive to use Result 2.39 to perform integration by parts. The following technique based on Leibniz's notation

$$f'(x) = \frac{\mathrm{d}f}{\mathrm{d}x}(x) \tag{2.88}$$

for differentiation is usually easier to use. Suppose that a function f we want to integration can be written as $f(x) = g'(x)h(g(x))$ for some functions g and h. Then

$$\int_a^b f(x)\,\mathrm{d}x = \int_a^b h(g(x))g'(x)\,\mathrm{d}x. \tag{2.89}$$

We may employ the substitution $u = g(x)$. Following (2.88), we now write

$$g'(x) = \frac{\mathrm{d}u}{\mathrm{d}x} \tag{2.90}$$

and subsequently engage in notational abuse to rearrange this equation as

$$\mathrm{d}x = \frac{1}{g'(x)}\,\mathrm{d}u. \tag{2.91}$$

We may formally substitute $g'(x)^{-1}\,\mathrm{d}u$ for $\mathrm{d}x$ in (2.89). However, we also need to take care to adjust the limits of integration. Since $x = a$ and $x = b$ at the original limits, the substitution $u = g(x)$ results in new integration limits $g(a)$ and $g(b)$. In this way we get

$$\int_a^b f(x)\,\mathrm{d}x = \int_a^b h(g(x))g'(x)\,\mathrm{d}x = \int_{g(a)}^{g(b)} h(u)g'(x)\cdot\frac{1}{g'(x)}\,\mathrm{d}u = \int_{g(a)}^{g(b)} h(u)\,\mathrm{d}u. \tag{2.92}$$

If h is, for example, one of the elementary functions with closed-form antiderivatives in Result 2.36, we can use (2.73) to compute the integral. The following two examples illustrate integration by substitution.

Example 2.40. Suppose that we want to compute

$$\int_1^3 f(x)\,\mathrm{d}x = \int_1^3 x\exp(x^2)\,\mathrm{d}x. \tag{2.93}$$

Make the substitution $u = x^2$, so that

$$\frac{\mathrm{d}u}{\mathrm{d}x} = 2x. \tag{2.94}$$

This gives $\mathrm{d}u = 2x\,\mathrm{d}x$. Because

$$\int_1^3 f(x)\,\mathrm{d}x = \int_1^3 x\exp(x^2)\,\mathrm{d}x = \frac{1}{2}\int_1^3 2x\exp(x^2)\,\mathrm{d}x = \frac{1}{2}\int_1^3 \exp(x^2)\cdot 2x\,\mathrm{d}x, \tag{2.95}$$

we can substitute $u = x^2$ and $\mathrm{d}u = 2x\,\mathrm{d}x$ into this integral. The integration limits $x = 1$ and $x = 3$ become $u = 1^2 = 1$ and $u = 3^2 = 9$ due to the substitution $u = x^2$. This gives

$$\int_1^3 f(x)\,\mathrm{d}x = \frac{1}{2}\int_1^3 \exp(x^2)\cdot 2x\,\mathrm{d}x = \frac{1}{2}\int_1^9 e^u\,\mathrm{d}u. \tag{2.96}$$

By (2.77) and (2.73),

$$\frac{1}{2}\int_1^9 e^u\,\mathrm{d}u = \frac{1}{2}(e^9 - e^1). \tag{2.97}$$

Thus we have

$$\int_1^3 x\exp(x^2)\,\mathrm{d}x = \frac{e}{2}(e^8 - 1). \tag{2.98}$$

Example 2.41. Suppose that we want to compute

$$\int_{-2}^0 f(x)\,\mathrm{d}x = \int_{-2}^0 \left(5 + \frac{x}{3}\right)^5 \mathrm{d}x. \tag{2.99}$$

In principle, we could expand $(5 + x/3)^5$ and then apply (2.75). But this is tedious and integration by substitution provides an easier way. Make the substitution $u = 5 + x/3$, so that

$$\frac{\mathrm{d}u}{\mathrm{d}x} = \frac{1}{3}. \tag{2.100}$$

This gives $\mathrm{d}x = 3\,\mathrm{d}u$, which we may subistute for $\mathrm{d}x$ in (2.99). The integration limits $x = -2$ and $x = 0$ become $u = 5 + (-2)/3 = \frac{13}{3}$ and $u = 5 + 0/3 = 5$ due to the substitution $u = 5 + x/3$. Thus

$$\begin{aligned}\int_{-2}^0 f(x)\,\mathrm{d}x = \int_{-2}^0 \left(5 + \frac{x}{3}\right)^5 \mathrm{d}x = \int_{\frac{13}{3}}^5 3u^5\,\mathrm{d}u = \frac{3}{5+1}\left(5^{5+1} - \left(\frac{13}{3}\right)^{5+1}\right) \\ = \frac{3\,281\,908}{729}\end{aligned} \tag{2.101}$$

by (2.75) and (2.73) [also recall (2.82)].

3 Vector Calculus

4 Discrete Probability