

UNIVERSITY OF SÃO PAULO
INSTITUTE OF MATHEMATICS AND STATISTICS
BACHELOR OF COMPUTER SCIENCE

Título do trabalho
um subtítulo

Antonio Fernando Silva e Cruz Filho
João Gabriel Andrade de Araujo Josephik

FINAL ESSAY
MAC 499 — CAPSTONE PROJECT

Supervisor: Prof. Nina S. T. Hirata

São Paulo
2024

*The content of this work is published under the CC BY 4.0 license
(Creative Commons Attribution 4.0 International License)*

[illegible]

Resumo

Antonio Fernando Silva e Cruz Filho

João Gabriel Andrade de Araujo Josephik. **Title of the document: *a subtitle*.** Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2024.

[illegible]

Palavras-chave: Palavra-chave1. Palavra-chave2. Palavra-chave3.

Abstract

Antonio Fernando Silva e Cruz Filho

João Gabriel Andrade de Araujo Josephik. **Título do trabalho: *um subtítulo***. Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2024.

[illegible]

Keywords: Keyword1. Keyword2. Keyword3.

List of Abbreviations

AI	Artificial Intelligence
ML	Machine Learning
XAI	Explainable AI
MLP	Multilayer Perceptron
CNN	Convolutional Neural Network
Conv	Convolution
IME	Institute of Mathematics and Statistics
USP	University of São Paulo

List of Figures

List of Tables

List of Programs

Contents

Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Other Section Example

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Next Section example

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Chapter 1

Background

In this chapter, we will introduce important concepts required for the understanding of this study. We begin by introducing Explainable AI (XAI) and its principles. We also introduce Neural Networks and important concepts such as Gradient Descent and Back Propagation. Finally, we introduce Convolutional Neural Networks, the main focus of this study.

1.1 Explainable AI

With the rise of Machine Learning models in the last decade in the business and academic areas, Artificial Intelligence (AI) is becoming increasingly present in important decision-making tasks. However, as AI models have become more sophisticated, particularly with the advent of Deep Learning techniques, their internal workings have often remained opaque. Explainable AI (XAI) aims to make models and their decisions more transparent, interpretable and understandable to both experts and inexperienced users.

1.1.1 What is Explainable AI?

Defining a mathematical formalization to explainability of Machine Learning is a difficult task considering the subjective nature of what one may consider "explainable". In non-mathematical terms, Explainability in AI refers to the capacity to articulate or justify the behavior of a model, focusing on methods that explain a model's decisions after they are made.

Another important concept in the area is Interpretability, which can be defined as "the degree to which a human can understand the cause of a decision" by Miller (2017).¹ In this case, however, a model's decision is understandable entirely by its inherent transparency. In other terms, the model is simple enough to be interpretable by a human directly, without the use of external techniques.

¹ Miller, Tim. "Explanation in artificial intelligence: Insights from the social sciences." arXiv Preprint arXiv:1706.07269. (2017).

Models with low complexity whose decisions are understandable by humans are defined as *Interpretable Models*. Linear Regression, Logistic Regression and Decision Tree models are examples of models classified as *Interpretable Models*. Now, models with a level of complexity that prevents humans from directly understanding their decision-making processes are referred to as *Explainable Models*. Recently popular *Deep Learning Models* are one kind of *Explainable Models* and will be the main focus of this essay, especially *Deep Convolutional Neural Networks*, explored in section 1.3.

1.1.2 Why Explainable AI is Necessary

Creating explanations to a model's decisions can yield many advantages, including increase in model trust, more ethical and fair decisions, correctly following regulatory compliances and easier model debugging.

While Debugging, Machine Learning models can have quite unpredictable behavior, detecting biases not initially noticed by humans. This can yield great performance in the training or even validation and test datasets, but poor performance in real world deployment. For example, while training an image classifier on dog and cat images one can find great accuracy on classification of dogs over green fields. However, when analyzing what regions of the image a model "sees" for the prediction, researchers may find out that the model is actually looking at the image background, since dog owners tend to take their pets for walks more often than cat owners, therefore making dog pictures with a green background more likely than pictures with cats over green fields. Visualizing regions of images that our vision model "sees" is only possible with Explainable AI techniques, such as GradCAM (Selvaraju_2019) and other Gradient Saliency methods.

1.2 Gradient Descent

Let $f : A \rightarrow B$ where $A \subseteq \mathbb{R}^n$ for $n \in \mathbb{N}$ and $B \subseteq \mathbb{R}^+$. Suppose we want to find the solution to the optimization problem

$$\operatorname{argmin}_{x \in A} f(x) \quad (1.1)$$

when $\frac{\partial f}{\partial x}$ is known for any value of x . Considering that the vector $\frac{\partial f}{\partial x}$ points to the direction of the steepest ascent of the function, the vector $-\frac{\partial f}{\partial x}$ will point to the steepest descent from the given point x . Therefore, one can define an initial random value for x and update x using $-\frac{\partial f}{\partial x}$ and a scaling factor η in order to find a local minimum of f and an approximation to the solution of given optimization problem.

We can define such method using the following formula, where x_t represents the value of x at iteration t of the algorithm:

$$x_{t+1} = x_t - \eta \frac{\partial f}{\partial x}. \quad (1.2)$$

The term η is often called the *learning rate* used in the Gradient Descent method and is often defined manually by the user.

The Gradient Descent method can be used to optimize a neural network's parameters to solve a given problem using a *loss function*.

1.3 Neural Networks

Neural Networks are proven to be universal approximators.¹ That means that Neural Networks are Machine Learning models capable of representing any continuous function, therefore making Neural networks adept at modeling a range of different complex problems. This class of models have seen a growing presence across both academic and industry landscapes. However, given the architecture of multiple hidden layers of Neural Networks creating complex internal patterns, such models are classified as Explainable Models.

In this section, the inner workings of Neural Networks will be explained, starting with the *Perceptron*, considered the fundamental building block of Neural Networks.

1.3.1 Perceptron

A Perceptron is a Machine Learning model inspired by how biological neurons work. It is a simple binary linear classifier that defines its parameters by linear combinations of points in the dataset. The Perceptron model can be described by Figure ??.

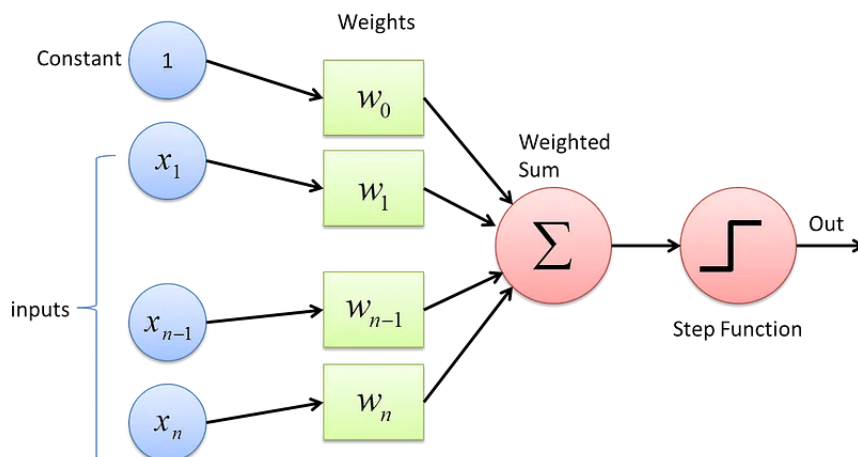


Figure 1.1: Perceptron Architecture. Font: Towards Data Science².

Where the weights w_i for $i \in \{0, 1, \dots, n\}$ are trainable parameters and the step function can be defined as $\sigma : \mathbb{R} \rightarrow \{0, 1\}$ such that

¹ Hornik, K., Stinchcombe, M., White, H. Multilayer feedforward networks are universal approximators. [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)

² <https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53>

$$\sigma(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0. \end{cases} \quad (1.3)$$

Therefore, the Perceptron model can be defined as the function $f : \mathbb{R}^n \rightarrow \{0, 1\}$ where

$$f(x) = \sigma(w_0 + \sum_{i=1}^n w_i x_i). \quad (1.4)$$

The Perceptron model updates its parameters using each sample (x, y) of the dataset with the rule

$$w_i^{t+1} = w_i^t + \eta (y - f(x))x_i \quad (1.5)$$

for $i \in \{1, \dots, n\}$ and

$$w_0^{t+1} = w_0^t + \eta (y - f(x)), \quad (1.6)$$

where η is the *learning rate* hyperparameter and t is the update iteration number.

As a linear model, the Perceptron can only model linear problems, which only represent a small subset of real world problems. As a solution, researchers started combining Perceptrons in a layered structure. The multiple layers consisting of multiple Perceptrons created a structure capable of representing more complex functions, or more specifically, *any continuous function*. Such architecture is called a *Multilayer Perceptron* (MLP). Neural Networks are considered "*softened*" MLPs,³ which means that those models use differentiable activation functions that allow for gradient-based optimization, using algorithms like the *Gradient Descent*.

1.3.2 How Neural Networks Work

Neural Networks work by using multiple Perceptrons with differentiable activation functions grouped in a layered structure in order to model complex functions. Such model's architecture can be described by Figure ?? in a network similar to a graph.

³ Terminology used by Professor Yaser Abu-Mostafa, in the famous Machine Learning Book *Learning from Data*. Source: <https://work.caltech.edu/telecourse>

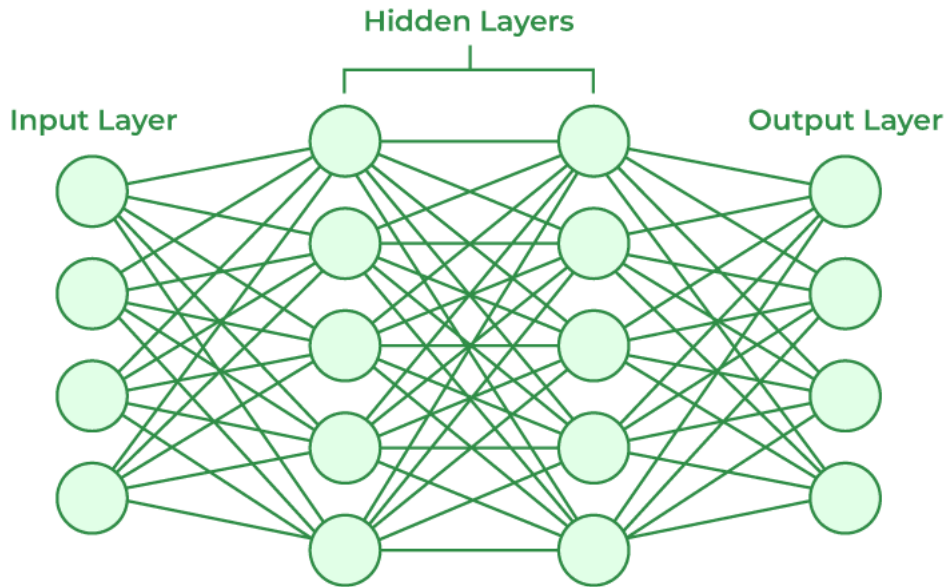


Figure 1.2: Neural Network Architecture. Font: Geeks For Geeks⁴.

Each node of such graph represents a *neuron*, which is a single Perceptron model with each edge representing an output of a neuron being passed to another neuron as input. Each edge will have a weight associated to it, representing the importance of a neuron from a previous layer. Inference in this model is called *Forward Propagation*, where the input is passed in the *Input Layer* and is computed by calculating a Perceptron's output and passing it to the next layer, until the *Output Layer* is reached. Layers in between are called *Hidden Layers*.

Mathematically, a neuron can be described by the same formula in equation ??, but with a different activation function.

1.4 Convolutional Neural Networks

1.4.1 Convolutions

First, it is important to define what a convolution is. Given two discrete one-dimensional signals f and g , their convolution $f * g$ is defined as:

$$(f * g)[n] = \sum_{i=-\infty}^{+\infty} f[n]g[n-i]$$

Given two discrete two-dimensional signals f and g , their convolution $f * g$ is calculated as:

⁴ <https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/>

$$(f * g)[m][n] = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} f[m][n] \cdot g[m-i][n-j]$$

In practice, the signal g is represented by a window (or kernel), usually square and of odd size. Thus, we can abstract convolution as the multiplication of a sliding window. The picture below illustrates this process. It is important to note that the window needs to be flipped during the convolution, although this is not illustrated in the picture.

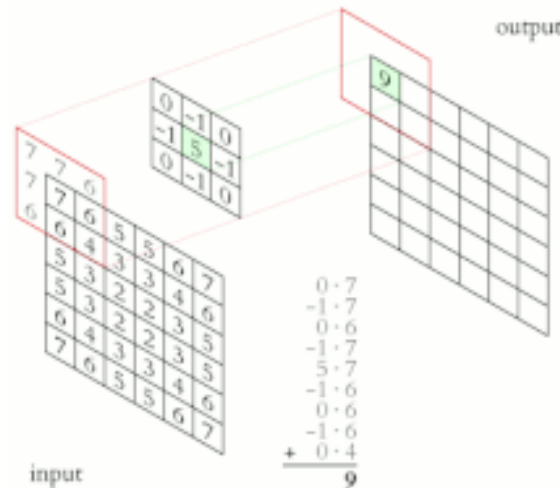


Figure 1.3: 2D Convolution

This process can be used to apply different filters to images. For example: using a 3×3 window with all weights equal to $\frac{1}{9}$, we can generate a filter that blurs the image (moving average). Below is an example of applying this filter:

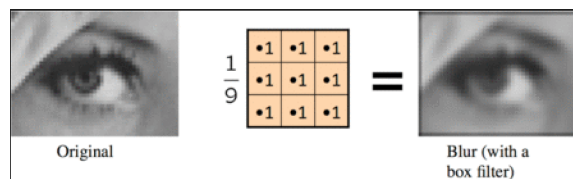


Figure 1.4: Filtered Image (3x3 Mean Filter)

It is also important to note that convolution is commonly implemented in machine learning contexts as "cross-correlation," which is a very similar operation but without the flipping of the window. Note that, since the weights are learned in our case, there is no difference. Therefore, in our context, convolution and cross-correlation are synonymous.

A pertinent question that can be asked is what happens at the edges of the image. When the window is sliding over them, what happens to the missing pixels? The process of filling in these pixels is called padding. Padding can be done with zeros, the nearest pixel, or not be done at all. Note that when there is no padding, the image decreases in size after convolution.

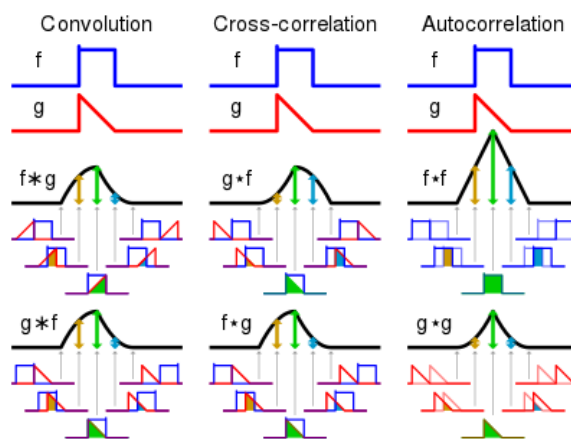


Figure 1.5: Comparison with Cross-Correlation

Another important hyperparameter that can be adjusted is the "stride." This defines how many positions the window is moved at a time. That is: a stride value different from 1 also implies a decrease in image size after convolution.

1.4.2 Convolutional networks

In order to understand the need for convolutional networks, we have to understand why it's impractical to use fully-connected networks to process images. Let's walk through an example to see that.

Consider a color image with dimensions 512×512 . If we were to process this image with a conventional neural network, the input layer would have $3 \cdot 512 \cdot 512 = 786432$ dimensions. Assume a hidden layer with only 128 neurons (which is relatively small). Just between these two layers, there would be 100663296 parameters! This is highly inefficient.

The solution to this problem is to extract features from the image, which will serve as input to the network. These features could include various aspects such as symmetry, black levels, contrast, presence or absence of patterns, etc. All of these features will serve as input to the network. As a result, we can reduce the input layer's dimensionality from several hundred thousand to just a few dozen.

However, a challenge still remains: how do we select these features? We can apply convolutions to the image to calculate interesting features, and these convolutions can be learned alongside the rest of the network! It is important to understand some essential details about these networks before proceeding. Each convolutional layer has three dimensions: height, width, and the number of channels. The input layer typically has one channel for black and white images, or three channels for color images.

Each channel in each convolutional layer combines all the channels from the previous layer. In other words, the "windows" used have weights for all the channels. These windows slide over the data from the previous layer to generate **one channel** in the next layer.

Let us consider an example. Suppose we have a network that processes color images of size 128×128 pixels. This network has 3 convolutional layers with 16, 32, and 64 channels per layer, respectively. Assume a window size of 3 for all layers. In this case, we have:

- First layer: window size is $3 \times 3 \times 3$. With 16 output channels, we will have

$$16 \cdot 3 \cdot 3 \cdot 3 = 432 \text{ parameters.}$$

- Second layer: window size is $3 \times 3 \times 16$. With 32 output channels, we will have

$$32 \cdot 3 \cdot 3 \cdot 16 = 4608 \text{ parameters.}$$

- Third layer: window size is $3 \times 3 \times 32$. With 64 output channels, we will have

$$64 \cdot 3 \cdot 3 \cdot 32 = 18432 \text{ parameters.}$$

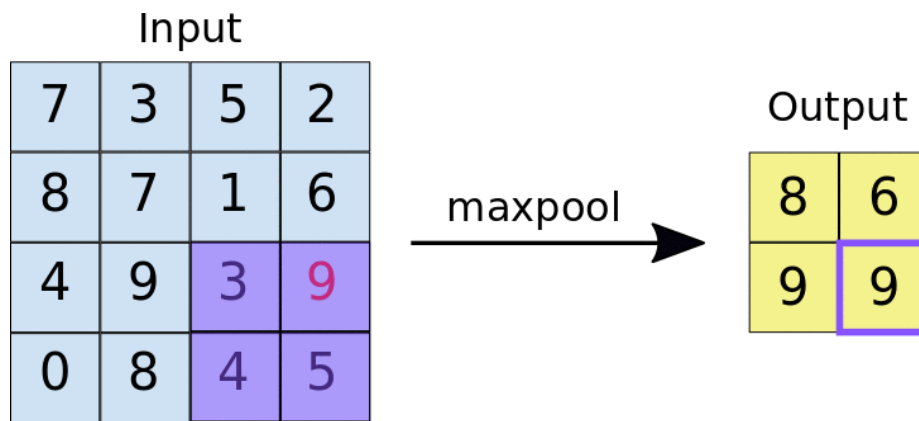
Another important detail is the output dimension of each layer. This depends on whether or not **padding** is used. **Padding** refers to how the layer behaves at the image edges. We can complete the image with zeros, the nearest pixel value, or the pixel value from the opposite edge of the image. If padding is not used, the output dimensions will decrease by $2\lfloor \frac{W}{2} \rfloor$, where W is the window size. For instance, with a window size of 3, each layer will reduce the image size by 2 pixels. If the input is 128×128 , the output of the first layer will be 126×126 , the second layer will output 124×124 , and so on.

1.4.3 Pooling

Remember, each convolution extracts a feature from the image. Therefore, when we perform another convolution using the outputs from the previous layer, we are combining features extracted from the image to compute new features. As a result, deeper layers extract more complex features from the image. For instance, the first layer may extract features like the presence of vertical straight lines, while the tenth layer may extract features like "presence of dog snouts."

Thus, the features involved gradually become less localized and more global (pertaining to the entire image). This is why it is useful to summarize information into smaller dimensions as the network deepens.

To accomplish this, we use "pooling" layers. These layers work similarly to convolutions: windows slide over the data and compute an output based on nearby pixels. However, this time, a function is used to aggregate these data. Common functions include "max" (maximum value) and "avg" (average value).

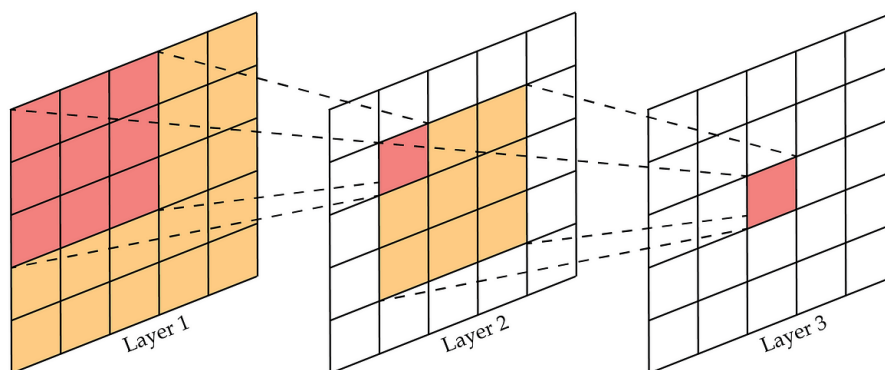


1.4.4 Receptive Field

An important concept for understanding the power of **deep convolutional networks** is the **receptive field**. This concept relates to the power that chained convolutions have.

Consider an input image. Apply a 3×3 convolution to it. Now, apply another 3×3 convolution to the output of the first convolution. Observe this output image. How much information does each pixel contain about its neighbors?

Receptive Field in Convolutional Networks



The answer is that each pixel contains information from a region of size 5×5 around it! This is the receptive field of these neurons.

A common misconception is that, since the receptive field of two 3×3 convolutions is 5×5 , two 3×3 convolutions have the same **expressive power** as a 5×5 convolution. This **is not** true. A 5×5 convolution has 25 parameters, while two chained 3×3 convolutions only have 18 parameters.

1.5 GradCAM

One of the most prominent model-specific methods to acquire explanations for classification with CNNs is GradCAM. The intuition for the method is simple. At the last convolutional layers, we have several channels that represent each a different feature.

Those features are used by the next part of the network to produce the final output. If we want to know which parts of the image are being more useful to the network, we can look at the feature maps and observe which parts of the image are generating the signal used by the rest of the network.

The problem with this approach is that the features have informations about all the output classes. How we know what features are more important to the decision? The idea behind GradCAM is to **average the feature maps weighted by the gradient** of each channel with respect to a specific class.

However, this will still highlight the regions that have a negative influence to the decision. To filter out those regions, the result is passed through ReLU. The result is a coarse heatmap of the image highlighting important regions.

The formula for the heatmap with regard to the class c is:

$$H = \text{ReLU}(\sum_k \alpha_k^c A^k)$$

Where α_k^c , the weight of the k -th feature map for the class c , is defined as:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_i^k j}$$

Where A^k is the k -th feature map, y^c is the output for the c class, and Z is the number of neurons in each map.

1.5.1 Guided Backpropagation

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer quis eros ultrices libero accumsan tempus eu in lacus. Cras magna ex, congue vitae fermentum nec, interdum ac eros. Praesent egestas risus eget turpis malesuada molestie. Suspendisse commodo sollicitudin nulla eu consectetur. Etiam bibendum, enim ut pretium elementum, lorem nisl imperdiet eros, non rhoncus enim eros eget nunc. Aliquam congue nisi commodo dui imperdiet, facilisis placerat nunc venenatis. Pellentesque pharetra leo at libero suscipit vehicula. Donec pellentesque at metus vel mattis. Donec id ex eget libero efficitur lobortis non quis neque. Aliquam malesuada urna neque. Donec ut pharetra turpis, fermentum dignissim ligula. In aliquam nisi sit amet lectus posuere volutpat. Nunc in suscipit erat. Ut et pulvinar ante, id viverra orci. Suspendisse augue justo, commodo convallis magna vitae, condimentum vestibulum nisl. Vivamus justo urna, posuere vitae nibh sit amet, egestas efficitur orci.

1.5.2 Guided GradCAM

The output of GradCAM has the dimensions of the last convolutional layer of the network, and has to be upsampled to be overlayed on top of the input image. This results in

a very coarse heatmap, with rough borders and lost details. To solve this, we can multiply (pixel by pixel) the heatmap with other simpler method, as guided-backpropagation.

Chapter 2

Usando este modelo

Não é necessário que o texto seja redigido usando \LaTeX ou este modelo, mas seu uso é fortemente recomendado, pois ele facilita diversas etapas do trabalho e o resultado final é muito bom.¹ Este modelo é distribuído com uma “colinha” dos principais comandos \LaTeX e inclui comentários explicativos para auxiliá-lo com ele, sendo composto pelos arquivos:

- `tese.tex`, `artigo.tex`, `apresentacao.tex` e `poster.tex` (exemplos de cada um desses tipos de documento);
- Capítulos, apêndices, imagens etc. deste texto de exemplo, nos diretórios `conteudo`, `figuras` e `logos` (procure os comandos `\input` e `\graphicspath` nos arquivos de exemplo mencionados acima para modificar o nome desses diretórios);
- `bibliografia.bib` (exemplo de banco de dados bibliográficos; procure o comando `\addbibresource` nos arquivos de exemplo mencionados acima para modificar o nome desse arquivo ou acrescentar outros);
- `imegoodies.sty`, que carrega várias *packages* comumente usadas. Você normalmente não vai precisar mexer nesse arquivo, mas pode fazê-lo se quiser, e também pode usá-lo em outros documentos \LaTeX (veja o Anexo ??);
- `imelooks.sty`, que carrega mais *packages* e define diversas opções relacionadas à aparência do documento, inclusive a capa da tese. Você normalmente também não vai precisar mexer nesse arquivo, mas pode fazê-lo se quiser, além de também poder usá-lo em outros documentos \LaTeX (veja o Anexo ??);
- `evenragged.sty`, `froufrou.sty`, `lstpseudocode.sty`, `texlogsieve` e `texlogsieverc` (classes e programas auxiliares ao modelo);
- `beamer-ime.[bbx|cbx]`, `plainnat-ime.[bbx|cbx]` e `plainnat-ime[-en].bst` (definições de estilos bibliográficos).

¹ O uso de um sistema de controle de versões, como mercurial (mercurial-scm.org) ou git (git-scm.com), também é altamente recomendado.

Para compilar o documento, basta executar o comando `latexmk`.² Talvez seu editor ofereça uma opção de menu para compilar o documento; sempre que possível, configure-o para utilizar o `latexmk` ao selecioná-la. \LaTeX gera diversos arquivos auxiliares durante a compilação que, em algumas raras situações, podem ficar inconsistentes (causando erros de compilação ou erros no PDF gerado, como referências faltando ou numeração de páginas incorreta no sumário). Nesse caso, é só usar o comando `latexmk -C`, que apaga todos esses arquivos auxiliares gerados, e em seguida rodar `latexmk` novamente.

Você pode mudar a língua do documento para o inglês no início de cada arquivo `.tex` de exemplo, na linha `\documentclass`. No caso do arquivo `tese.tex`, isso muda todos os textos padrão da capa e folhas de rosto.

Os arquivos deste modelo incluem vários comentários com dicas e explicações; se o que você precisa não está mencionado diretamente, é provável que haja pelo menos a indicação da *package* relacionada ao que você precisa.

Se você encontrar algum problema com o modelo, ajude a melhorá-lo! Envie um relatório de erro ou entre em contato em gitlab.com/ccsl-usp/modelo-latex.

² Você também pode usar `latexmk poster`, `latexmk apresentacao` etc.

Chapter 3

Instalação do \LaTeX

\LaTeX é, na verdade, um conjunto de programas. Ao invés de procurar e baixar cada um deles, o mais comum é baixar uma coleção com todos eles juntos. Há duas coleções desse tipo disponíveis: MiKTeX (miktex.org) e \TeX Live (www.tug.org/texlive). Ambos funcionam em Linux, Windows e macOS. Em Linux, \TeX Live costuma estar disponível para instalação junto com os demais opcionais do sistema. Em macOS, o mais popular é o MacTeX (www.tug.org/mactex/), a versão do \TeX Live para macOS. Em Windows, o mais comumente usado é o MiKTeX .

Por padrão, eles não instalam tudo que está disponível, mas sim apenas os componentes mais usados, e oferecem um gestor de pacotes que permite adicionar outros. Embora uma instalação completa do \LaTeX seja relativamente grande (perto de 5GB), em geral vale a pena instalar a maior parte dos componentes. Se você preferir uma instalação mais “enxuta”, não deixe de incluir tudo que é necessário para este modelo, como indicado no arquivo `README.md`.

Também é muito importante ter o `latexmk`. No Linux, a instalação é similar à de outros programas. No macOS e no Windows, `latexmk` pode ser instalado pelo gestor de pacotes do MiKTeX ou \TeX Live . Observe que ele depende da linguagem `perl`. No macOS, `perl` já faz parte do sistema; no Windows, \TeX Live inclui uma versão básica de `perl`, mas se você estiver usando MiKTeX será preciso instalar `perl` manualmente (www.perl.org/get.html).

3.1 Documentação sobre \LaTeX

Há muito material sobre \LaTeX na Internet, mas também há muita informação obsoleta (incluindo trechos da própria documentação oficial!). Em particular, você pode ignorar explicações sobre como converter arquivos no formato DVI gerados por \LaTeX em PDF: as versões atualmente recomendadas de \LaTeX (cf. Seção ??) geram arquivos PDF diretamente. Quanto a imagens, os formatos de arquivo PS/EPS (PostScript e Encapsulated PostScript) não são adequados para essas novas versões de \LaTeX ; elas trabalham com arquivos de imagem nos formatos PDF, PNG e JPEG. Finalmente, recursos gráficos normalmente não usam mais *packages* como `pstricks`, `eepic` ou outras tradicionalmente citadas; ao invés disso, `PGF/TikZ` é a ferramenta mais comum.

Um possível caminho para o aprendizado é começar com o Capítulo ?? deste modelo e o conteúdo em overleaf.com/learn, que tem escopo similar mas também inclui várias páginas sobre como utilizar recursos específicos, ou ainda o sítio <https://www.learnlatex.org/pt/>, em português. Após esse contato inicial, o tutorial em tug.org/twg/mactex/tutorials/ltxprimer-1.0.pdf é bastante abrangente e detalhado. Não deixe de ver também o Capítulo ?? deste modelo (e seu código-fonte), que inclui várias dicas úteis. Para os principais comandos do modo matemático, veja `texdoc undergradmath` e, para aprender a criar apresentações, veja `texdoc beamer`.

Depois que você estiver razoavelmente familiarizado com a linguagem, utilize o manual de referência que pode ser acessado em latexref.xyz ou com `texdoc latex2e` (disponível também em francês, com `texdoc latex2e-fr`, e em espanhol, com `texdoc latex2e-es`).

A documentação de referência mais importante sobre os recursos matemáticos é acessível com `texdoc amsmath`, `texdoc amsthm` e `texdoc mathtools`; `texdoc maths-symbols` agrega os símbolos matemáticos disponíveis. Para uma lista completa de todos os símbolos disponíveis com L^AT_EX, use `texdoc symbols-a4` (esse documento tem mais de 300 páginas!).

Como dito anteriormente, L^AT_EX é, na verdade, um conjunto de programas e, em geral, instalamos coleções pré-prontas com todos eles. Essas coleções (T_EXLive e MiK_TE_X) contêm também a documentação das *packages* incluídas: Basta digitar `texdoc nome-da-package` (T_EXLive) ou `mtxhelp nome-da-package` (MiK_TE_X) para ter acesso à documentação correspondente.¹ `texdoc/mtxhelp` incluem também alguns tutoriais e textos introdutórios.

Para dúvidas pontuais, o sítio tex.stackexchange.com é um fórum de perguntas e respostas sobre L^AT_EX muito útil, pois os principais desenvolvedores do sistema participam das discussões, e o sítio texfaq.org é bastante abrangente e atualizado.

Existem também diversos bons livros sobre L^AT_EX (embora em geral um tanto antigos), dos quais destacamos dois:

1. A quarta edição de “A Guide to L^AT_EX”, de Helmut Kopka e Patrick W. Daly (publicada em 2003), além de uma ótima introdução, aborda vários tópicos relativamente avançados e úteis.²
2. A segunda edição de “The L^AT_EX Companion” (publicada em 2004) é um livro quase obrigatório, pois discute em detalhes praticamente todos os recursos e *packages* importantes de L^AT_EX, servindo tanto para o aprendizado quanto como material de referência. A terceira edição, amplamente atualizada, foi lançada em 2023.



Existem inúmeras alternativas aos materiais citados acima; outros exemplos de textos introdutórios são www.maths.tcd.ie/~dwilkins/LaTeXPrimer/GSWLaTeX.pdf e www.andy-roberts.net/writing/latex. Em português, você pode consultar polignu.org/sites/polignu.org/files/latex/latex-fflch.pdf e git.febrace.org.br/material-latex/material-latex.

¹ O sítio texdoc.org também oferece acesso a esse conteúdo.

² Uma versão não-final está disponível em www2.mps.mpg.de/homes/daly/GTL/gtl_20030512.pdf.

(este precisa ser baixado e compilado). O canal [youtube.com/c/anteroneves](https://www.youtube.com/c/anteroneves) tem vários vídeos instrutivos em português. `texdoc/mthelp` incluem ainda opções como “The Not So Short Introduction to \LaTeX 2 ϵ ” (`texdoc lshort-eng`; há uma versão em português, mas não está em dia com o original) e “A Simplified Introduction to \LaTeX ” (`texdoc simplified-intro`). Versões recentes do \LaTeX incluem também o “ \LaTeX 2 ϵ via exemplos” (`texdoc latex-via-exemplos`), em português.

3.1.1 Outros recursos (avançados)

O sítio ctan.org é o repositório semi-oficial das *packages* \LaTeX e sua documentação; \TeX Live e $\text{MiK}\TeX$ são construídas a partir do que está nesse site, então a última versão estável de qualquer *package* (e da documentação acessível com `texdoc/mthelp`) em geral está ali.

`texdoc fntguide` explica como funciona a gestão de fontes de \LaTeX , e você pode ver exemplos de fontes disponíveis para \LaTeX em tug.org/FontCatalogue. $\text{Lua}\LaTeX$ e $\text{Xe}\LaTeX$ funcionam de outra maneira, permitindo também o uso das fontes comuns instaladas no seu sistema operacional (veja `texdoc fontspec`).

Minúcias sobre o funcionamento interno do sistema estão descritas em `texdoc source2e` e, sobre as classes padrão (`article`, `book` etc.), em `texdoc classes`. Você normalmente não vai usar esses documentos, mas eles podem servir para esclarecer algum detalhe. `texdoc macros2e`, `texdoc xparse` e `texdoc interface3` apresentam a linguagem de programação usada por \LaTeX , enquanto `texdoc clsguide` é um guia para a criação de novas classes e *packages*.

Quando você se tornar um usuário avançado, pode se interessar em conhecer melhor a linguagem \TeX , que está na base do \LaTeX . “The \TeX book”, de Donald Knuth (o criador do \TeX), é amplamente recomendado, mas há três livros completos a respeito que são instalados com \LaTeX : “A gentle introduction to \TeX ” (`texdoc gentle`), “ \TeX for the impatient” (`texdoc impatient`) e “ \TeX by topic” (`texdoc texbytopic`).

Chapter 4

Do zero ao mínimo com L^AT_EX

Neste capítulo, apresentamos uma visão geral sobre L^AT_EX para quem nunca trabalhou com ele antes. Se você já tem conhecimento básico ou intermediário sobre o sistema, sinta-se à vontade para ir diretamente ao Capítulo ??, que inclui diversos exemplos e dicas úteis. A intenção deste capítulo não é propriamente ensinar a usar L^AT_EX, mas sim expor seus princípios de funcionamento e principais recursos, de maneira que o leitor esteja melhor capacitado a compreender outros documentos e exemplos.

4.1 Por que L^AT_EX?

Preparar um texto para impressão envolve duas coisas:

Escrever: digitar, recortar/colar trechos, revisar etc.

Formatar: definir o tamanho da fonte, o espaçamento entre parágrafos etc.

Hoje é comum fazer essas duas coisas ao mesmo tempo, graças à visualização imediata que o computador oferece. No entanto, imagine como era o processo de produção de um livro nos anos 1970: o autor escrevia seu texto em uma máquina de escrever e enviava esse material para o editor, que era responsável pela tarefa de formatá-lo para impressão. O autor muitas vezes inseria anotações para o editor explicando coisas como “este parágrafo é uma citação”, e o editor criava algum mecanismo visual para representar isso.

Não é de se surpreender que, com o surgimento do microcomputador, os primeiros programas para criação de textos seguissem um funcionamento similar: o autor digitava e editava seu texto sem formatá-lo visualmente, apenas inserindo alguns comandos correspondentes a aspectos da formatação que ele depois revisava na versão impressa. L^AT_EX é uma ferramenta baseada nesse processo: você prepara seu texto no editor de sua preferência, insere comandos no texto que indicam a estrutura do documento e o processa com o L^AT_EX, que gera um arquivo PDF formatado. Embora seja um estilo “antigo” de trabalhar, ele é muito eficiente em vários casos. Ou seja, dependendo da situação, pode ser mais adequado trabalhar fazendo tudo ao mesmo tempo ou dividindo o trabalho nessas duas fases. De maneira geral:

- Se você precisa criar páginas diferentes entre si com *layout* definido manualmente, é melhor usar uma ferramenta que permita trabalhar visualmente, como LibreOffice Writer, MS-Word, Google Docs etc.;
- Se você precisa fazer um documento relativamente longo com estrutura regular (capítulos, seções etc.), é melhor usar ferramentas que formalizam essa estrutura (como L^AT_EX) ao invés de ferramentas visuais;
- Se você precisa fazer um documento envolvendo referências cruzadas, bibliografia relativamente extensa ou fórmulas matemáticas, é difícil encontrar outra ferramenta tão eficiente quanto L^AT_EX;
- Se você precisa criar um documento simples, ambas as abordagens funcionam bem; cada um escolhe esta ou aquela em função da familiaridade com as ferramentas;
- Se você quer que a qualidade tipográfica do resultado seja realmente excelente, é necessário usar uma ferramenta profissional, como L^AT_EX, Scribus, Adobe InDesign ou outras; processadores de texto convencionais não oferecem o mesmo nível de qualidade dessas ferramentas.¹

4.2 Visão geral

Com L^AT_EX, você prepara o texto (incluindo as indicações de estrutura) em um editor de textos qualquer, salva como arquivo de texto puro (".txt", mas é comum usar a extensão ".tex" ao invés de ".txt") e processa esse arquivo com o comando "latexmk" ("compila" o documento) para obter o PDF correspondente. Qualquer editor capaz de salvar arquivos em formato texto puro, como o bloco de notas do windows, vim, emacs etc. pode ser usado. Programas como LibreOffice Writer, MS-Word etc. também funcionam, mas possivelmente vão gerar dores de cabeça porque vão tentar formatar algumas coisas automaticamente (e de maneira incompatível com L^AT_EX).

Em geral, é recomendável usar editores projetados especificamente para trabalhar com L^AT_EX; eles utilizam cores para distinguir o texto dos comandos de formatação, automatizam o processo de compilação do documento (veja a Seção ??) e oferecem outras comodidades. O mais usado atualmente é o T_EXstudio, que é software livre e funciona em Windows, macOS e Linux. O editor Visual Studio Code (code.visualstudio.com) é voltado para programadores e tem uma interface às vezes peculiar para outros usuários, mas em conjunto com a *package* LaTeX Workshop (do editor, não do L^AT_EX), é uma boa opção. O mesmo vale para o editor emacs (www.gnu.org/software/emacs) e sua *package* AUCT_EX. Ainda outra possibilidade são os editores *online*; dentre eles, o overleaf (www.overleaf.com) é o mais usado.

Um documento L^AT_EX é dividido em duas partes: o *preâmbulo*, onde você coloca comandos de configuração para o documento, e o *corpo* do documento em si, que contém o texto propriamente dito. O preâmbulo é onde você define as características do resultado tipográfico esperado para o documento como um todo: tipo e tamanho da fonte a usar,

¹ A maior diferença (mas não a única) é o algoritmo que divide cada parágrafo em uma série de linhas: T_EX (desde 1982) e Adobe InDesign (desde 1999) analisam cada parágrafo como um todo, ao invés de uma linha por vez, para obter espaçamentos mais homogêneos e menos palavras hifenizadas.

posição dos títulos e subtítulos na página etc. O corpo, por sua vez, consiste no texto e em alguns comandos indicativos da estrutura.

Dado que configurar o preâmbulo é um tanto complexo e que mesmo no corpo do texto às vezes há comandos especiais (para a geração da bibliografia ou tabelas, por exemplo), usar algum documento existente como base para criar seu texto em geral é uma boa ideia. O IME/USP oferece um conjunto de modelos adequados para teses/dissertações, artigos, apresentações e pôsteres (gitlab.com/ccsl-usp/modelo-latex) que pode ser adaptado para outros usos e outras instituições. Há também uma família de modelos (www.abntex.net.br) que procura seguir as normas da ABNT para diversos tipos de documentos científicos, e algumas publicações científicas fornecem modelos de acordo com suas diretrizes.

4.3 Estrutura de um documento L^AT_EX

O preâmbulo L^AT_EX começa com a definição da *classe* a ser utilizada, que determina boa parte da configuração do documento. As principais classes são `book`, `article` e `beamer` (para apresentações); você pode saber mais sobre elas (e outras) em qualquer texto introdutório sobre L^AT_EX na Internet (veja a Seção ??).² A seguir, são carregadas várias *packages* (“*plugins*”) que acrescentam funcionalidades ou modificam as classes padrão; qualquer documento L^AT_EX utiliza várias delas. A classe é definida com o comando `\documentclass{nome-da-classe}`; *packages* são carregadas com o comando `\usepackage{nome-da-package}`. Classes e *packages* podem receber opções adicionais entre colchetes (`\usepackage[opção1,opção2...]{nome-da-package}`); a documentação de cada *package* e classe (veja a Seção ??) detalha as opções disponíveis.

L^AT_EX ignora quebras de linha e trata sequências de vários espaços como se fossem apenas um. Isso significa que você pode usar quebras de linha e espaços no texto que está digitando como “dicas visuais” da estrutura do texto durante a edição. É muito comum fazer isso com listas de itens, por exemplo (veja a Seção ??). Uma ou mais linhas em branco sinalizam o fim de um parágrafo e o início de outro. O caractere “%” indica que o restante da linha é um comentário, ou seja, um trecho de texto que não tem nenhum efeito sobre o resultado final do documento. Comentários podem ser usados como lembretes sobre alguma decisão, para indicar um parágrafo que ainda precisa de revisão etc. Por conta desse significado especial, para inserir um caractere % “normal” no texto é preciso digitar “\%”.

Como mencionado anteriormente, L^AT_EX divide o trabalho de produção de um texto entre a preparação do conteúdo e a definição da forma de apresentação. Assim, os comandos usados durante a produção do conteúdo procuram expressar o *significado* de cada elemento, e não sua aparência. Por exemplo, para realçar uma palavra é comum usar texto *em itálico*; embora exista um comando especificamente para gerar textos em itálico em L^AT_EX, o recomendado é que se utilize o comando `\emph` (“ênfático”), pois em alguns casos pode ser melhor utilizar **negrito**, `VERSALETE` ou outro mecanismo para dar ênfase a uma palavra. Essa é uma orientação geral para a escrita de textos com L^AT_EX: procure definir a estrutura, não a aparência.

² Algumas revistas acadêmicas têm suas próprias classes; por exemplo, a AMS (American Mathematical Society) disponibiliza as classes `amsart`, `amsbook` e `amspc`; você pode usá-las para seus trabalhos mesmo que não pretenda publicar com a AMS, veja `texdoc Author_Handbook_Journals`.

Um exemplo de documento L^AT_EX simples (lembre-se, “%” indica um comentário):

```
% O documento começa com o preâmbulo
% Vamos usar a classe "book" com fonte no tamanho 11pt
\documentclass[11pt]{book}
% Vamos escrever em português do Brasil
\usepackage[brazilian]{babel}
% Finaliza o preâmbulo e inicia o conteúdo:
\begin{document}
% Estas linhas não imprimem nada, apenas definem as
% informações que serão usadas por "\maketitle" a seguir
\author{Fulano de Tal}
\title{Começando a usar o \LaTeX{}}
% Cria um bloco ou página de título com os dados acima
\maketitle
% Capítulos, seções etc. são numerados automaticamente
\chapter{Cheguei!}
Oi, Galera!
% É preciso sinalizar o final do documento
\end{document}
```

Esse exemplo mostra como definir o nome de um capítulo. Existem também os comandos `\section`, `\subsection`, `\subsubsection` e `\paragraph` (a classe `book` inclui também `\part`, um nível acima de `\chapter`). Usar o nome do comando seguido de um asterisco (`\chapter*` etc.) faz o capítulo/seção não ser numerado (nem considerado na contagem de capítulos, seções etc.) nem incluído no sumário. Este modelo ainda define `\chapter**`, `\section**`, `\subsection**` e `\subsubsection**`, que eliminam a numeração mas incluem o capítulo ou seção no sumário (úteis para a introdução, por exemplo).

4.4 Executando L^AT_EX e comandos auxiliares

Depois de escrever o arquivo `.tex`, é preciso *compilá-lo*, ou seja, processá-lo para gerar o PDF desejado. Isso envolve executar, além do próprio L^AT_EX (veja a Seção ??), alguns programas auxiliares (em geral, `biber` ou `bibtex` e `makeindex`). Nesse processo, L^AT_EX quase sempre precisa ser executado três ou mais vezes antes de gerar o PDF final.³ Por conta dessa complexidade, é comum utilizar alguma ferramenta para automatizar o processamento. Existem diversas opções, mas a mais comum é o `latexmk`, que é capaz de identificar automaticamente os passos necessários para a geração do documento, executando os programas na ordem correta quantas vezes forem necessárias.⁴ Assim, embora seja possível gerar o PDF executando apenas `pdflatex nome-do-arquivo.tex`, acostume-se a compilar o documento sempre com `latexmk -pdf nome-do-arquivo.tex`. Note que editores especializados em L^AT_EX costumam ter uma opção de menu para a compilação do documento; dentre as configurações possíveis do editor, prefira sempre a que simplesmente aciona `latexmk`.

³ A cada vez, ele gera uma nova versão intermediária do arquivo PDF, mas essas versões têm defeitos, como citações e referências cruzadas incorretas ou sumário inexistente.

⁴ É possível personalizar o comportamento de `latexmk` com o arquivo de configuração `latexmkrc`.

4.5 Mais sobre estrutura

Para criar listas de itens, você pode fazer⁵:

```
\begin{itemize}
  \item Pedra
  \item Papel
  \item Tesoura
\end{itemize}
```

Além de “itemize”, há também “enumerate” (auto-explicativo) e “description”:

```
\begin{description}
  \item[Pedra:] perde para papel;
  \item[Papel:] perde para tesoura;
  \item[Tesoura:] perde para pedra.
\end{description}
```

Citações curtas normalmente são incluídas no fluxo normal do texto e colocadas entre aspas; para citações mais longas, use `\begin{quote}` ou `\begin{quotation}` (este último é mais adequado para citações com vários parágrafos). A package `csquotes` acrescenta recursos sofisticados para citações.

Para poesia, use `\begin{verse}` (a package `verse` acrescenta vários recursos ao comando `verse`). Estrofes são separadas por uma linha em branco e versos são separados por `*`. O asterisco é opcional; ele instrui \LaTeX a manter as linhas na mesma página.

Para inserir uma nota de rodapé, use o comando `\footnote{texto da nota}`.

4.6 Figuras e tabelas (*floats*)

É possível utilizar `\includegraphics` para acrescentar figuras ao texto (nos formatos PDF, PNG e JPEG), mas normalmente elas não são inseridas diretamente. A razão é que, se você simplesmente inserir uma figura em qualquer lugar, ela pode ser grande demais para o espaço disponível na página, o que forçará \LaTeX a deixar um espaço em branco e colocá-la na página seguinte. O mesmo vale para tabelas (criadas com `\begin{tabular}`). Para contornar esse problema, \LaTeX possui *floats*, que são blocos com algum conteúdo cuja localização é flexível: \LaTeX procura colocar um *float* “perto” de onde ele foi definido, mas não necessariamente no lugar exato.

Ao invés de um único comando como “`\begin{float}`” a ser usado tanto para figuras quanto para tabelas, \LaTeX define `\begin{figure}` e `\begin{table}`. Ele faz isso porque, assim como com capítulos e seções, \LaTeX também numera figuras e tabelas — mas, para isso, ele precisa saber qual é o tipo de cada *float*.⁶ À parte isso, o conteúdo de um *float* pode ser qualquer coisa mas, em geral, é `\includegraphics` ou `\begin{tabular}` respectivamente.

⁵ Observe o uso de espaços no início das linhas com `\item` para deixar a estrutura visualmente mais clara durante a edição.

⁶ É possível criar outros tipos de *float* também: como pode ser visto no Capítulo ??, este modelo define o tipo `program`.

Uma consequência importante (e proposital) dos tipos diferentes de *floats* é que L^AT_EX garante que a sequência das figuras e a sequência das tabelas sejam respeitadas (a Figura 6 nunca aparece depois da Figura 7). No entanto, isso *não* se aplica a *floats* de tipos diferentes, ou seja, se você definiu a Figura 5, a Tabela 3 e a Figura 6, elas podem aparecer no documento na ordem “Figura 5, Tabela 3, Figura 6”, “Figura 5, Figura 6, Tabela 3” ou “Tabela 3, Figura 5, Figura 6”.

4.7 Referências cruzadas

É comum que um trecho do texto faça referência a outro trecho (“como discutimos no Capítulo X...”). Isso pode ser feito diretamente, mas se você reorganizar o documento ou acrescentar seções, a numeração pode mudar. Para evitar esse problema, você pode gerar essas referências automaticamente com o par de comandos `\label{nome-sugestivo}` e `\ref{nome-sugestivo}` (para o número da seção/capítulo) ou `\pageref{nome-sugestivo}` (para o número da página).

Esse mecanismo também é muito útil para figuras e tabelas. Dentro do *float*, além da figura em si, em geral é uma boa ideia acrescentar uma legenda com `\caption`. Além disso, é possível inserir um `\label` dentro da legenda para que se possa fazer referência à figura/tabela no texto (com os comandos `\ref` e `\pageref`).

4.8 Referências bibliográficas e bibliografia

A geração de bibliografias no L^AT_EX é feita através da package `biblatex` e do programa auxiliar `biber`⁷ e envolve três passos:

1. A criação de um banco de dados, no formato “.bib”, das obras de interesse. Esse banco de dados pode incluir obras que não vão ser de fato referenciadas no documento final. Isso significa que você pode criar um único banco de dados e utilizá-lo em todos seus documentos.⁸
2. A inserção de referências às obras ao longo do texto, usando diferentes comandos dependendo do caso: `\cite`, `\citet`, `\citep` etc. Como já mencionado, esses comandos estão descritos na documentação da package `natbib` (**natbib**).
3. A escolha do estilo bibliográfico (usando as opções da package `biblatex`) que formata as citações ao longo do texto e gera a bibliografia automaticamente através do comando `\printbibliography`. Normalmente, apenas as obras efetivamente citadas são incluídas na lista de referências, mas é possível forçar a inclusão de uma obra sem citá-la explicitamente com o comando `\nocite`.

O banco de dados é um arquivo de texto contendo uma *entrada* para cada item da bibliografia e, em cada entrada, uma série de *campos* com os dados (título, autor etc.). A entrada inclui também uma *chave*, que é usada para inserir as citações no texto. Há

⁷ Antigamente, usava-se a package `natbib` e o comando auxiliar `bibtex`. O funcionamento geral dos dois mecanismos é similar e o formato do banco de dados de ambos é o mesmo.

⁸ É comum criar bancos de dados desse tipo separados por assunto, mas isso não é necessário.

vários tipos de entrada (para artigos, livros, sítios web etc.) e, para cada tipo, uma lista de campos possíveis (considere que periódicos normalmente incluem o número do volume, mas teses não). O exemplo abaixo é um livro cuja chave é “dissertjourney”; ele pode ser citado com o comando `\cite{dissertjourney}`:

```
@book{dissertjourney,
  author    = {Carol M. Roberts},
  title     = {The Dissertation Journey},
  publisher = {Corwin},
  year      = 2010,
  edition   = 2,
  location  = {Thousand Oaks, CA},
}
```

Em alguns casos, \LaTeX troca as letras maiúsculas definidas em `\title` para minúsculas. Para evitar que isso afete siglas ou nomes próprios, basta colocá-los entre chaves (“Automated Application-Level Checkpointing of {MPI} Programs”).

Os campos `author` e `publisher` podem incluir uma lista de nomes separados por `and`; `biblatex` reconhece que cada nome é composto por nome e sobrenome, às vezes com partículas como “de”, “dos” ou “von” e, dependendo do estilo bibliográfico, pode abreviar nomes, mudar sobrenomes para caixa alta etc. Isso evidentemente não funciona quando o autor é, na verdade, uma instituição; nesses casos, basta colocar o nome inteiro da instituição entre chaves (“{Universidade de São Paulo — Sistema Integrado de Bibliotecas}”) para que `biblatex` não faça alterações desse tipo. Se o nome é longo, pode ser interessante definir o campo `shortauthor`.

A fonte mais detalhada de informações sobre o banco de dados é a documentação da package `biblatex` (**`biblatex`**), mas o material ali é um tanto denso. Há muito material introdutório ao formato “.bib” e ao `bibtex` disponível *online*, e você pode se inspirar em exemplos para criar seu banco de dados bibliográfico. Além disso, ferramentas como Zotero ou Mendeley (o uso de uma delas é altamente recomendado!) podem exportar para o formato .bib. Observe que `biblatex` oferece recursos bastante sofisticados para o tratamento de referências e bibliografias. Se você precisar de alguma funcionalidade especial, consulte a documentação do pacote ou a Internet; é quase certeza que `biblatex` oferece uma solução.

4.9 Fórmulas matemáticas

A diagramação de fórmulas matemáticas tem regras específicas: letras são interpretadas como variáveis e espaços em branco são ignorados (\LaTeX usa o contexto da fórmula para definir o espaçamento). Assim, para criar fórmulas em \LaTeX , é preciso usar um comando para iniciar o modo matemático. Isso pode ser feito de duas formas:

- Pequenas fórmulas no meio do texto ($e^{i\pi} + 1 = 0$) são inseridas com `$fórmula$` (e, portanto, para inserir um caractere \$ normal no texto, é preciso usar `\$`).

- Fórmulas mais longas ou que devem aparecer em um parágrafo separado são inseridas com `\[fórmula]` (ou `\begin{displaymath}`).

L^AT_EX é capaz de oferecer uma boa solução para praticamente qualquer problema de diagramação para matemática; basta ler a documentação.

4.10 Formatação manual

Às vezes é preciso inserir formatação de forma manual; os comandos mais importantes são: `\emph` (texto *ênfatisado*, em geral *itálico*), `\texttt` (texto teletype, imitando um terminal de texto ou uma impressora), `\textit` (*itálico*), `\textbf` (**negrito**), `\textsf` (fonte sem serifa), `\textsc` (texto VERSALETE — nem todas as fontes oferecem essa possibilidade), `\normalsize` (tamanho normal), `\small` (tamanho reduzido), `\footnotesize` (ainda menor), `\scriptsize` (ainda menor), `\tiny` (ainda menor), `\large` (tamanho aumentado), `\Large` (ainda maior), `\LARGE` (ainda maior), `\Huge` (ainda maior), `\vspace{\baselineskip}` (deixa uma linha em branco), `\begin{center}` (centraliza parágrafos), `\begin{flushleft}` (alinha parágrafos à esquerda), `\begin{flushright}` (alinha parágrafos à direita),⁹ `\-` (sugere um possível local para hifenização localizada), `\linebreak[0–4]` (sugere um possível local para mudar de linha; o número indica quão forte é a sugestão, ou seja, 4 faz a mudança obrigatória. Por ser uma mudança de linha “normal”, se o parágrafo é justificado, a linha é justificada normalmente), `\newline` ou `\\` (força uma quebra de linha; por ser uma quebra forçada, a linha *não* é justificada nesse caso), `\pagebreak[0–4]` (sugere um possível local para mudar de página; como `\linebreak`, o número indica quão forte é a sugestão. Por ser uma mudança de página “normal”, o texto da página é espalhado verticalmente de maneira a fazer a última linha alinhada com o final das demais páginas) e `\newpage` (força uma quebra de página; por ser uma quebra forçada, o final da página *não* é alinhado com o final das demais páginas nesse caso).

Mas, como discutido na Seção ??, não é recomendável usar esses comandos ao longo do texto: o ideal em L^AT_EX é expressar o significado de cada elemento, não a sua forma de apresentação, pois isso permite que você faça alterações na formatação com mais facilidade. Assim, quando os recursos pré-definidos do L^AT_EX (`\itemize`, `\chapter` etc.) não forem suficientes, o mais adequado é definir comandos novos, em geral usando os comandos de formatação mencionados acima. Esse é um tópico avançado, mas você pode consultar o início do arquivo L^AT_EX deste capítulo para alguns exemplos simples.

4.11 Versões do L^AT_EX

Assim como há packages para o L^AT_EX, o próprio L^AT_EX é, na verdade, um conjunto de extensões para o programa T_EX. Assim, se você encontrar referências a “T_EX” ou a “plain T_EX”, basta saber que esse é o sistema que funciona “por baixo” do L^AT_EX.

L^AT_EX é um sistema em evolução (desde os anos 80!). Uma das consequências disso é que há, na verdade, quatro versões diferentes dele:

⁹ É altamente recomendável carregar a package `ragged2e` (já incluída neste modelo) e utilizar `Center`, `FlushLeft` e `FlushRight` ao invés de `center`, `flushleft` e `flushright`.

1. \LaTeX “tradicional”, que gera arquivos em formato DVI que, por sua vez, precisam ser convertidos para o formato PDF. Essa versão não é capaz de usar as fontes instaladas no sistema; ela só pode usar fontes adaptadas para uso com o \LaTeX . Hoje em dia não há boas razões para usar essa versão.
2. $\text{pdf}\text{\LaTeX}$, que gera arquivos PDF e dá suporte a alguns recursos avançados de tipografia adicionais. É a versão mais usada hoje em dia, embora também só possa usar as fontes adaptadas para uso com o \LaTeX .
3. $\text{Xe}\text{\LaTeX}$ que, além dos recursos do $\text{pdf}\text{\LaTeX}$, opera internamente em UTF-8 (ou seja, funciona melhor com múltiplas línguas) e pode funcionar não só com as fontes adaptadas para o \LaTeX como também com as fontes instaladas no sistema. $\text{Xe}\text{\LaTeX}$ foi muito importante ao ser lançado, mas atualmente a comunidade está mais empenhada em evoluir o sistema com $\text{Lua}\text{\LaTeX}$.
4. $\text{Lua}\text{\LaTeX}$, que oferece os mesmos recursos que o $\text{Xe}\text{\LaTeX}$ e também pode ser estendido internamente com mais facilidade (através da linguagem de programação Lua).

Todas essas versões são instaladas quando você instala \LaTeX na sua máquina. Em geral, se você pretende escrever apenas com línguas no alfabeto latino e não pretende usar fontes diferentes das disponíveis por padrão, qualquer das três versões modernas ($\text{pdf}\text{\LaTeX}$, $\text{Xe}\text{\LaTeX}$ e $\text{Lua}\text{\LaTeX}$) é adequada; $\text{pdf}\text{\LaTeX}$ é um pouco mais rápido, mas $\text{Lua}\text{\LaTeX}$ gera arquivos PDF um pouco menores. Se você pretende usar outros alfabetos, gostaria de escolher fontes diferentes ou precisa de recursos tipográficos específicos (`texdoc fontspec`, `texdoc unicode-math`), use $\text{Lua}\text{\LaTeX}$.

4.12 Limitações do \LaTeX

Como qualquer ferramenta, \LaTeX tem limitações e características indesejáveis:

- A linguagem é muito prolixa: é bastante tedioso escrever coisas como “`\begin{itemize}`” etc. Linguagens como `asciidoc` (asciidoctor.org), `markdown` (commonmark.org), `bookdown` (bookdown.org) e `reStructuredText` (sphinx-doc.org) operam de maneira similar a \LaTeX , mas sua sintaxe é bem mais enxuta. Elas funcionam muito bem para a geração de páginas web, mas \LaTeX oferece mais recursos e geralmente produz resultados impressos melhores.
- \LaTeX gera muitas mensagens pouco importantes durante o processamento do documento, o que dificulta a identificação de problemas (o programa auxiliar `texlogsieve`, incluído com versões recentes de \LaTeX , pode minimizar esse incômodo). Além disso, quando ocorrem erros durante esse processamento, as mensagens explicativas muitas vezes são confusas ou, pior, não indicam o problema real que causou a falha.
- \LaTeX procura ser uma linguagem *declarativa*, ou seja, os comandos buscam expressar o que se deseja e não como fazer algo (“este texto é um título” e não “pule duas linhas, selecione uma fonte maior, escreva este texto, pule mais duas linhas e selecione a fonte de tamanho padrão”). No entanto, ela é insuficiente em algumas situações, obrigando o usuário a utilizar vários comandos, às vezes obscuros, para obter resultados relativamente simples.

- Há diversas *packages* para personalizar os aspectos básicos da formatação final do documento, como o tipo de fonte, tamanho dos títulos das seções, espaçamento etc. No entanto, quando se quer fazer modificações maiores, é preciso lidar com partes complexas da linguagem e diversos comportamentos surpreendentes.
- Às vezes há incompatibilidades entre *packages*; em alguns casos, isso pode ser contornado mudando a ordem em que elas são carregadas, mas em outros pode simplesmente não ser possível combiná-las.
- A colocação automática dos *floats* e o algoritmo que encontra as quebras de página em geral funcionam bem, mas às vezes é possível obter resultados melhores manualmente (veja as Seções ?? e ??).
- As classes padrão (*book*, *article* etc.) não foram criadas para serem facilmente modificadas, o que deu origem a inúmeras *packages* voltadas para possibilitar a personalização de diversos aspectos da apresentação final do documento. Esse mecanismo não é ideal, por diversas razões. Por conta disso, existe um conjunto de versões alternativas dessas classes (*scrbook* no lugar de *book*, *scrartcl* no lugar de *article* etc.) chamado KOMA-Script, com mais recursos e mais possibilidades de customização. A classe *memoir* tem o mesmo objetivo, mas procura dar suporte a livros e artigos com uma única classe. Ambas abordagens são muito boas, mas a maioria dos modelos usados por revistas e outras publicações é baseada nas classes padrão. A versão 3 de L^AT_EX está em desenvolvimento com vistas a resolver boa parte dos problemas atuais do sistema, mas ainda deve demorar muitos anos para ficar pronta. ConT_EXt é um “irmão mais novo” de L^AT_EX com diversas vantagens, mas com sintaxe diferente e que ainda não é tão popular.

Chapter 5

Exemplos e dicas de L^AT_EX

Neste capítulo, apresentamos exemplos comuns com alguma complexidade e, principalmente, pequenas dicas para evitar surpresas indesejáveis. Mesmo que você já conheça L^AT_EX, vale a pena analisar este material, incluindo o código-fonte do capítulo. Se você ainda não conhece nada sobre L^AT_EX, o Capítulo ?? e outros materiais citados na Seção ?? apresentam os conceitos básicos.

5.1 Bibliografia e referências

A documentação do pacote biblatex (**biblatex**) é bastante extensa e explica (nas Seções 2.1.1 e 2.2.2) os diversos tipos de documento suportados, bem como o significado de cada campo. Na prática, às vezes é preciso fazer escolhas sobre o que incluir na descrição de um item bibliográfico e muitas vezes é mais fácil aprender copiando exemplos já existentes, como estes (consulte o arquivo `bibliografia.bib` para ver como foi criado o banco de dados e a bibliografia na página ?? para ver o resultado impresso):

- @Book: **Knuth:96**.
- @Article (em periódico): **floats2014**.
- @InProceedings (ou @Conference): **alves03:simi**.
- @InCollection (capítulo de livro ou coletânea): **bobaoglu93:concepts**.
- @PhdThesis: **garcia01:PhD**.
- @MastersThesis: **schmidt03:MSc**.
- @Techreport: **alvisi99:analysisCIC**.
- @Manual: **biblatex**.
- @Misc: **gridftp**.
- @Online (para referência a artigo *online*): **fowler04:designDead**.
- @Online (para referência a página web): **FSF:GNU-GPL**.

A maioria das revistas científicas ainda utiliza bibtex e não biblatex, mas isso não faz muita diferença na prática: latexmk identifica automaticamente qual sistema usar durante a geração do documento e os modelos normalmente já incluem os comandos `\bibliography`, `\printbibliography`, `\bibliographystyle` etc. conforme o caso. O único detalhe importante

se refere a datas: biblatex prefere o uso do campo “date” para definir ano, mês etc. No entanto, se você quiser garantir compatibilidade tanto com biblatex quanto com bibtex, use os campos “year” e “month”. Ambos reconhecem diversos formatos para o campo “month”, mas apenas um funciona corretamente com os dois: o nome do mês em inglês, abreviado com três letras minúsculas e sem chaves, ou seja:

```
...
author = {Fulano de Tal},
year = {2011},
month = oct,
title = {Um título grandioso},
...
```

Para citar material *online*, há três casos:

- Para citar publicações *online* que se enquadram em formatos tradicionais (como um ebook ou a versão online de um artigo científico, independentemente de a revista existir ou não no formato impresso), use o tipo correspondente (@article, @book, @inproceedings etc.) e acrescente o campo url no arquivo .bib, aceito por todos os tipos de documento do bibtex/biblatex.
- Para citar materiais essencialmente *online* que possuem título e autor definidos (como uma postagem ou comentário em blog ou uma mensagem de email para uma lista de discussão), use o tipo @online de biblatex. Bibtex, por padrão, não tem um tipo específico para isso; com ele, normalmente usa-se o tipo “misc” e seu campo “howpublished” para especificar que se trata de um recurso *online*.
- Se o que você quer citar não é propriamente uma ideia (que normalmente possui um autor e faz parte de um texto com título etc.) mas sim um sítio (como uma empresa, produto ou repositório de software), pode ser mais adequado colocar a referência apenas como nota de rodapé e não na lista de referências. Outra opção é criar uma segunda lista de referências especificamente para recursos *online* desse tipo (biblatex permite criar múltiplas bibliografias).

5.2 Identificando problemas

Como mencionado na Seção ??, L^AT_EX gera um grande volume de mensagens informativas durante o processamento, o que torna mais difícil encontrar problemas. O arquivo de configuração latexmkrc deste modelo utiliza o programa texlogsieve para filtrar essas mensagens, apresentando apenas as mais importantes para o usuário; considere usá-lo em outros trabalhos com L^AT_EX também.

5.3 Modo matemático

O modo matemático do L^AT_EX tem sintaxe própria, mas ela não é complicada e há bastante documentação *online* a respeito. Por exemplo, “massa e energia são grandezas relacionadas pela Equação $E = mc^2$, definida inicialmente por Einstein”, ou ainda “equações de segundo grau (Equação ??) são estudadas no ensino médio. As raízes de uma equação de

segundo grau podem ser encontradas por (??) — a fórmula de Bháskara. O valor do discriminante Δ (Equação ??) determina se a equação tem zero, uma ou duas raízes reais distintas”.

$$ax^2 + bx + c = y \quad \forall x \in \mathbb{R} \quad (5.1)$$

$$y = 0 \Leftrightarrow x = \frac{-b \pm \sqrt{\Delta}}{2a} \Leftrightarrow x \text{ é raiz da equação} \quad (5.2)$$

$$\Delta \text{ (delta)} = b^2 - 4ac \quad (5.3)$$

Para inserir um espaço explicitamente no modo matemático, use `\quad` ou `\enspace`. Para inserir texto “normal” em uma fórmula matemática, use `\text{texto}` (para texto de fato) ou `\mathit{texto}` (para nomes de variáveis ou funções com mais de uma letra). Pode ser necessário deixar um espaço no início do texto para evitar que ele fique colado com o caractere matemático que o antecede.

Para recursos mais sofisticados, incluindo frações com múltiplas linhas, matrizes, sistemas de equações alinhadas, setas, acentos etc., procure a documentação das packages `amsmath` e `mathtools`. Para teoremas, lemas, conjecturas etc., leia a documentação das packages `amsthm` e `thmtools` e decida de quais tipos de estrutura você vai precisar no seu documento. Aqui criamos três: “Pegadinha”, “Teorema” e “Conjectura” (observe as numerações):

Pegadinha 1 :-) $1 = 0$

Proof. Tomemos dois números, a e b , tais que $a = b + 1$.

$$\begin{aligned} a &= b + 1 \\ (a - b)a &= (a - b)(b + 1) \\ a^2 - ab &= ab + a - b^2 - b \\ a^2 - ab - a &= ab - b^2 - b \\ a(a - b - 1) &= b(a - b - 1) \\ \cancel{a(a - b - 1)} &= \cancel{b(a - b - 1)} \\ a &= b \\ b + 1 &= b \\ 1 &= b - b \\ 1 &= 0 \end{aligned}$$

□

Teorema 1. *É sempre possível colorir os vértices de um grafo sem que dois vértices adjacentes tenham a mesma cor usando no máximo quatro cores diferentes.*

Proof. A demonstração do Teorema ?? é um exercício a cargo do leitor.

Conjectura (?) 1: *Dado qualquer inteiro $n > 2$, não existem inteiros positivos a , b e c tais que $a^n + b^n = c^n$.*

Proof. Este espaço é muito pequeno para apresentá-la.

Teorema 2. $P \neq NP$

Proof. **P** tem apenas uma letra, enquanto **NP** tem duas letras. □

5.4 Quebras de página

O algoritmo que L^AT_EX usa para quebrar páginas funciona bem, minimizando linhas órfãs ou viúvas e garantindo uma distribuição homogênea do texto na página, mas não é excelente. Assim, se houver quebras de página ruins no seu texto final, pode ser útil modificá-las manualmente. Uma técnica usada por editores profissionais é mudar ligeiramente a altura do texto impresso em algumas páginas, melhorando a distribuição geral do texto. Para isso, ao invés de comandos como `\pagebreak` ou `\newpage`, o mais adequado é usar `\enlargethispage{\baselineskip}` (ou `-1\baselineskip`). Esse comando instrui L^AT_EX a fazer a página ligeiramente maior (ou menor), tornando possível acomodar mais uma linha de texto (ou uma linha a menos). Em documentos frente e verso, lembre-se de sempre garantir que a página adjacente também tenha seu tamanho modificado para que a alteração não seja tão perceptível. Um outro truque às vezes útil é aplicar o comando `\looseness=1` (ou `-1`) a um parágrafo, que faz L^AT_EX tentar reorganizar as quebras de linha de maneira a fazer o parágrafo ter uma linha a mais (ou a menos), se isso for possível.

5.5 Figuras, gráficos e outros *floats*

Evidentemente, L^AT_EX permite inserir figuras no texto; além disso, ele também permite girá-las e criar subfiguras (com sublegendas), como no exemplo da Figura ??, que inclui as subfiguras ?? e ??.

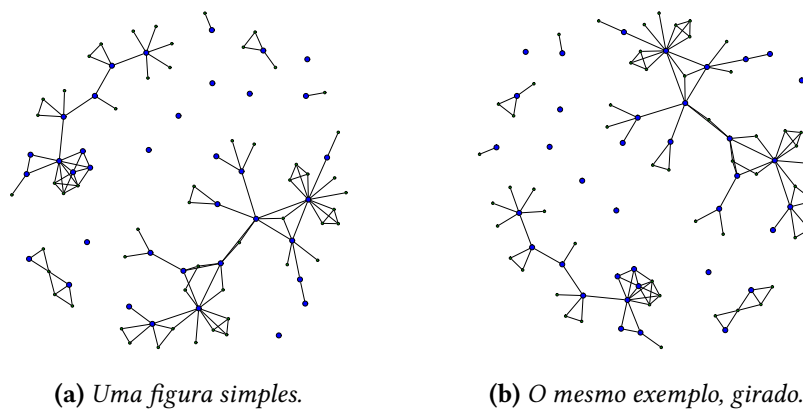


Figure 5.1: Exemplo de subfiguras.

Floats em geral incluem uma legenda e um *label*. Prefira sempre colocar o comando `\label` de uma figura ou tabela dentro do comando `\caption`; não fazê-lo muitas vezes funciona, mas às vezes causa problemas.

Você pode carregar arquivos de imagem de um subdiretório usando `dir/img.pdf`, mas é mais fácil modificar o comando `\graphicspath` próximo ao início de cada arquivo `.tex` de exemplo.

Para centralizar uma imagem mais larga que o texto da página, use a *package* `adjustbox` (incluída neste modelo):

```
\begin{figure}
\adjustbox{center}{\includegraphics[width=1.2\textwidth]{img.pdf}}
\caption{...}
\end{figure}
```

Uma “figura”, na verdade, pode ser qualquer tipo de conteúdo ilustrativo (um exemplo interessante é o cronograma mostrado na Figura ??) mas, com a *package* `float`, também é possível definir ambientes específicos para cada tipo de conteúdo adicional (cada um com numeração independente), como é o caso do Programa ??. Há mais informações e dicas sobre recursos específicos para inclusão de código-fonte e pseudocódigo no Anexo ??.¹

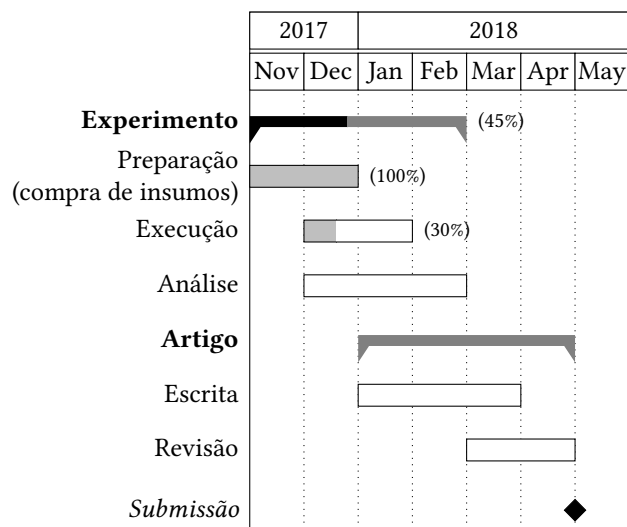


Figure 5.2: Exemplo de cronograma.

\LaTeX também é capaz de gerar ilustrações e diagramas diretamente, mas usar esses recursos em geral não é trivial. Em particular, a *package* `tikz` oferece bons mecanismos para a criação de figuras (incluindo funções pré-prontas para formas geométricas, grafos, matrizes etc.) e é fácil usá-la para traçar linhas ou curvas simples.

¹ Observe que o nome do Anexo (“??”) foi impresso em uma linha separada, o que não é muito bom visualmente. Para evitar que isso aconteça (não só no final do parágrafo, mas em qualquer quebra de linha), utilize um espaço não-separável para fazer referências a figuras, tabelas, seções etc. ou antes de símbolos: “... no Anexo~\ref{ap:pseudocode}”, “O discriminante é denotado por~\Delta\$”.

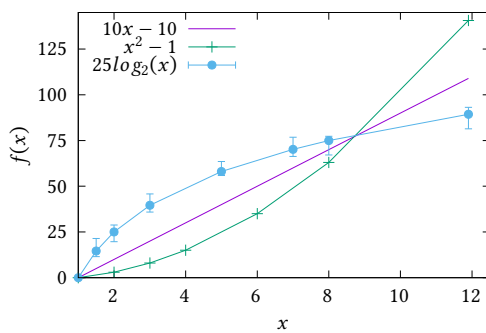
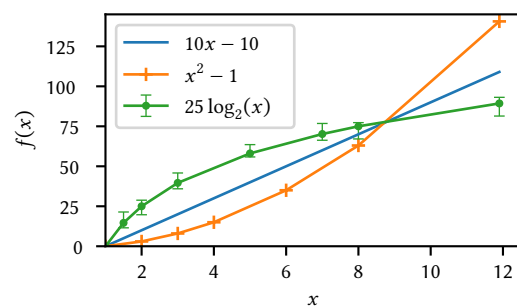
Program 5.1 Exemplo de laço em Java.

```

1  for (i = 0; i < 20; i++)
2  {
3      // Comentário
4      System.out.println("Mensagem...");
5  }

```

Gráficos de dados ou funções matemáticas de excelente qualidade podem ser gerados com a *package* `pgfplots` (há um exemplo comentado neste arquivo; experimente des-comentar para ver o resultado). Também é possível importar gráficos gerados por `matplotlib`, `gnuplot` e `R` como qualquer outra imagem, mas nesse caso a fonte usada nesses gráficos provavelmente será diferente do corpo do texto. Felizmente, isso pode ser solucionado: `Gnuplot` (com o *driver* `lua tikz`²), `matplotlib` (com o *backend* `PGF`³) e `R` (com `tikzDevice`⁴) são capazes de exportar gráficos de dados na forma de comandos para `tikz`⁵: o resultado pode ser visto na Figura ??.

(a) *gnuplot*.(b) *matplotlib*.**Figure 5.3:** Exemplos de gráficos gerados externamente

Note que a colocação automática dos *floats* em geral funciona bem, mas às vezes pode ser melhorada. Isso acontece porque L^AT_EX decide o posicionamento de cada *float* individualmente, sem levar em conta os próximos *floats*, e nunca reavalia essa decisão. No exemplo da Seção ??, se a ordem “Figura 5, Tabela 3, Figura 6” for aceitável, esse vai ser o resultado, mesmo que a ordem “Tabela 3, Figura 5, Figura 6” seja melhor. Apenas se não for possível encontrar um lugar aceitável para a Figura 5 imediatamente (ou seja, na página atual) é que L^AT_EX processa os *floats* seguintes e, depois, procura novamente um lugar para ela. Por isso, depois que seu trabalho estiver finalizado, vale a pena avaliar se a colocação dos *floats* pode ser melhorada; se sim, mudar o lugar em que eles são definidos no documento (veja algumas dicas em **floats2014**) pode fazer L^AT_EX gerar um resultado melhor (mas lembre-se que isso só faz sentido depois que o documento estiver pronto, pois qualquer mudança no texto pode mudar totalmente a posição final dos *floats*).

² gnuplot.info/docs_5.5/loc20850.html

³ matplotlib.org/users/pgf.html

⁴ cran.r-project.org/package=tikzDevice

⁵ Você pode se interessar também pela *package* `gnuplottex`.

5.6 Tabelas

Talvez você precise organizar a apresentação da informação na forma de tabelas⁶; um exemplo simples é a Tabela ?? Para um resultado visual excelente, não deixe de ler a documentação da *package* booktabs.

Código	Abreviatura	Nome completo
A	Ala	Alanina
C	Cys	Cisteína
...
W	Trp	Triptofano
Y	Tyr	Tirosina

(a) Com linhas de cores alternadas.

Código	Abreviatura	Nome completo
A	Ala	Alanina
C	Cys	Cisteína
...
W	Trp	Triptofano
Y	Tyr	Tirosina

(b) Com cabeçalhos girados.

Table 5.1: Exemplos de tabelas (códigos, abreviaturas e nomes dos aminoácidos).

Normalmente, o fim de cada linha de uma tabela é indicado por `\`. No entanto, se sua tabela causar erros misteriosos, experimente usar `\tabularnewline` ao invés de `\`.

Se a tabela tem muitas linhas e, portanto, não cabe em uma única página, é possível fazê-la continuar ao longo de várias páginas com a *package* longtable, como é o caso da Tabela ?. Nesse caso, a tabela não é um *float* e, portanto, ela aparece de acordo com a sequência normal do texto. Se, além de muito longa, a tabela for também muito larga, você pode usar o comando *landscape* (da *package* pdfscape) em conjunto com *longtable* para imprimi-la em modo paisagem ao longo de várias páginas. A Tabela ? tem essa configuração comentada; experimente des-comentar as linhas correspondentes.⁷ Ela também demonstra o uso da *package* siunitx para alinhar as colunas numéricas pelo separador decimal.

	Ângulo		Função					
	graus	rads	sen	cos	tan	cotan	sec	cosec
0	0.0000	0.0000	1.0000	0.0000	-		1.0000	-
2	0.0349	0.0349	0.9994	0.0349	28.6363	1.0006	28.6537	
4	0.0698	0.0698	0.9976	0.0699	14.3007	1.0024	14.3356	
6	0.1047	0.1045	0.9945	0.1051	9.5144	1.0055	9.5668	
8	0.1396	0.1392	0.9903	0.1405	7.1154	1.0098	7.1853	

continua →

Table 5.2: Exemplo de tabela com valores numéricos.

⁶ Para defini-las com \LaTeX , pode valer a pena usar o sítio www.tablesgenerator.com.

⁷ Observe que, nesse caso, vai sempre haver uma quebra de página no texto para fazer a tabela começar em uma página em modo paisagem.

	Ângulo		Função					
	graus	rads	sen	cos	tan	cotan	sec	cosec
10	0.1745	0.1736	0.9848	0.1763	5.6713	1.0154	5.7588	
12	0.2094	0.2079	0.9781	0.2126	4.7046	1.0223	4.8097	
14	0.2443	0.2419	0.9703	0.2493	4.0108	1.0306	4.1336	
16	0.2793	0.2756	0.9613	0.2867	3.4874	1.0403	3.6280	
18	0.3142	0.3090	0.9511	0.3249	3.0777	1.0515	3.2361	
20	0.3491	0.3420	0.9397	0.3640	2.7475	1.0642	2.9238	
22	0.3840	0.3746	0.9272	0.4040	2.4751	1.0785	2.6695	
24	0.4189	0.4067	0.9135	0.4452	2.2460	1.0946	2.4586	
26	0.4538	0.4384	0.8988	0.4877	2.0503	1.1126	2.2812	
28	0.4887	0.4695	0.8829	0.5317	1.8807	1.1326	2.1301	
30	0.5236	0.5000	0.8660	0.5774	1.7321	1.1547	2.0000	
32	0.5585	0.5299	0.8480	0.6249	1.6003	1.1792	1.8871	
34	0.5934	0.5592	0.8290	0.6745	1.4826	1.2062	1.7883	
36	0.6283	0.5878	0.8090	0.7265	1.3764	1.2361	1.7013	
38	0.6632	0.6157	0.7880	0.7813	1.2799	1.2690	1.6243	
40	0.6981	0.6428	0.7660	0.8391	1.1918	1.3054	1.5557	
42	0.7330	0.6691	0.7431	0.9004	1.1106	1.3456	1.4945	
44	0.7679	0.6947	0.7193	0.9657	1.0355	1.3902	1.4396	
46	0.8029	0.7193	0.6947	1.0355	0.9657	1.4396	1.3902	
48	0.8378	0.7431	0.6691	1.1106	0.9004	1.4945	1.3456	
50	0.8727	0.7660	0.6428	1.1918	0.8391	1.5557	1.3054	
52	0.9076	0.7880	0.6157	1.2799	0.7813	1.6243	1.2690	
54	0.9425	0.8090	0.5878	1.3764	0.7265	1.7013	1.2361	
56	0.9774	0.8290	0.5592	1.4826	0.6745	1.7883	1.2062	
58	1.0123	0.8480	0.5299	1.6003	0.6249	1.8871	1.1792	
60	1.0472	0.8660	0.5000	1.7321	0.5774	2.0000	1.1547	
62	1.0821	0.8829	0.4695	1.8807	0.5317	2.1301	1.1326	
64	1.1170	0.8988	0.4384	2.0503	0.4877	2.2812	1.1126	
66	1.1519	0.9135	0.4067	2.2460	0.4452	2.4586	1.0946	
68	1.1868	0.9272	0.3746	2.4751	0.4040	2.6695	1.0785	
70	1.2217	0.9397	0.3420	2.7475	0.3640	2.9238	1.0642	
72	1.2566	0.9511	0.3090	3.0777	0.3249	3.2361	1.0515	
74	1.2915	0.9613	0.2756	3.4874	0.2867	3.6280	1.0403	
76	1.3265	0.9703	0.2419	4.0108	0.2493	4.1336	1.0306	
78	1.3614	0.9781	0.2079	4.7046	0.2126	4.8097	1.0223	
80	1.3963	0.9848	0.1736	5.6713	0.1763	5.7588	1.0154	
82	1.4312	0.9903	0.1392	7.1154	0.1405	7.1853	1.0098	
84	1.4661	0.9945	0.1045	9.5144	0.1051	9.5668	1.0055	
86	1.5010	0.9976	0.0698	14.3007	0.0699	14.3356	1.0024	
88	1.5359	0.9994	0.0349	28.6363	0.0349	28.6537	1.0006	
90	1.5708	1.0000	0.0000	-	0.0000	-	1.0000	

Table 5.2: Exemplo de tabela com valores numéricos.

Tabelas mais complexas são um tanto trabalhosas em \LaTeX ; a Tabela ?? mostra como construir uma tabela em forma de ficha. Além de complexa, ela é larga e, portanto, deve ser impressa em modo paisagem. No entanto, usamos um outro mecanismo para girar a tabela: o comando `sidewaystable` (da *package* `rotating`). Com esse mecanismo, ela continua sendo um *float* (e, portanto, não força quebras de página no meio do texto), mas sempre é impressa em uma página separada.

Resumindo:

- Se uma tabela cabe em uma página, defina-a como um *float* (`\begin{table}`);
- Se cabe em uma página mas é muito larga e precisa ser impressa em modo paisagem, use `sidewaystable` (que também é um *float*);
- Se não cabe em uma página por ser muito longa, use `longtable`;
- Se não cabe em uma página por ser muito longa e precisa ser impressa em modo paisagem por ser muito larga, use `longtable` em conjunto com `landscape`. Nesse caso, vai haver uma quebra de página no texto para que a tabela inicie em uma nova página em modo paisagem.

5.7 Caracteres especiais

Um espaço não-separável é indicado pelo caractere til (“~”) e é possível forçar uma quebra de linha com “\”. Aspas tipográficas (“ ” e ‘ ’) são inseridas com “` ” e “` ’”. Os principais símbolos matemáticos estão listados em `texdoc undergradmath` e você pode consultar a lista completa de símbolos disponíveis com `texdoc symbols-a4` ou em www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf. Uma outra maneira de encontrar símbolos é usar este sítio: detexify.kirelabs.org/classify.html.

5.8 Línguas e hifenização

Detalhes sobre o suporte a diferentes línguas estão descritos na documentação da *package* `babel`. Para mudar temporariamente a língua do documento (por conta de uma citação, por exemplo), use `\foreignlanguage{língua}{texto}` (isso altera a hifenização das palavras e algumas convenções tipográficas, como espaços antes de pontuação em francês). Para trocar completamente de língua (o que inclui “captions”, ou seja, palavras geradas automaticamente como “Capítulo”, “Sumário” etc.), use `\selectlanguage{língua}`. Se você quiser apenas desativar temporariamente a hifenização, faça `\begin{hyphenrules}{nohyphenation}` (para uma única palavra, `\mbox{palavra}` é mais simples). \LaTeX em geral é capaz de hifenizar palavras de maneira excelente, mas você pode “ensiná-lo” a hifenizar uma palavra de maneira diferente do padrão acrescentando ao preâmbulo `\babelhyphenation{al-gu-mas pa-la-vras}` (todas as línguas) ou `\babelhyphenation[língua, língua...]{al-gu-mas pa-la-vras}` (uma ou mais línguas específicas).

Experimento número:	1	Data:					jan 2017
Título:	Medições iniciais						
Tipo de experimento:	Levantamento quantitativo						
Locais	São Paulo	Rio de Janeiro	Porto Alegre	Recife	Manaus	Brasília	Rio Branco
Valores obtidos	0.2	0.3	0.2	0.7	0.5	0.1	0.4

Table 5.3: Exemplo de tabela similar a uma ficha.

Appendix A

Perguntas frequentes sobre o modelo

- **Não consigo decorar tantos comandos!**

Use a colinha que é distribuída juntamente com este modelo (gitlab.com/ccsl-usp/modelo-latex/raw/main/pre-compilados/colinha.pdf?inline=false).

- **Estou tendo problemas com caracteres acentuados.**

Versões modernas de \LaTeX usam UTF-8, mas arquivos antigos podem usar outras codificações (como ISO-8859-1, também conhecido como latin1 ou Windows-1252). Nesses casos, use `\usepackage[latin1]{inputenc}` no preâmbulo do documento. Você também pode representar os caracteres acentuados usando comandos \LaTeX : `\a` para á, `\c{c}` para cedilha etc., independentemente da codificação usada no texto.¹

- **É possível resumir o nome das seções/capítulos que aparece no topo das páginas e no sumário?**

Sim, usando a sintaxe `\section[mini-titulo]{titulo enorme}`. Isso é especialmente útil nas legendas (*captions*) das figuras e tabelas, que muitas vezes são demasiadamente longas para a lista de figuras/tabelas.

- **Existe algum programa para gerenciar referências em formato bibtex?**

Sim, há vários. Uma opção bem comum é o JabRef; outra é usar Zotero ou Mendeley e exportar os dados deles no formato .bib.

- **Posso usar pacotes \LaTeX adicionais aos sugeridos?**

Com certeza! Você pode modificar os arquivos o quanto desejar, o modelo serve só como uma ajuda inicial para o seu trabalho.

¹ Você pode consultar os comandos desse tipo mais comuns em en.wikibooks.org/wiki/LaTeX/Special_Characters. Observe que a dica sobre o pingo do i *não* é mais válida atualmente; basta usar `\i`.

Annex A

As packages imegoodies e imelooks

Este modelo inclui as *packages* imegoodies e imelooks, que você pode querer usar em outros documentos \LaTeX .

imegoodies inclui um grande número de *packages* que são comumente usadas e bastante úteis. Em geral, você pode incluí-la em seus documentos sem que isso cause problemas de compatibilidade. Se, no entanto, algo não funcionar, você pode editar o arquivo para eliminar a *package* responsável pelo problema se ela não for necessária. imegoodies ainda inclui vários comentários explicativos sobre as *packages* carregadas.

imelooks também inclui um grande número de *packages*, mas estas são relacionadas mais explicitamente à aparência do documento (fontes, cores, margens etc.). Você também pode utilizá-la em outros documentos se quiser se aproximar da aparência deste modelo. imelooks reconhece diversos parâmetros que ativam/desativam aspectos específicos:

- `fonts` carrega as fontes deste modelo (libertinus e sourcecodepro), além de outros pequenos ajustes relacionados. Esta opção é sempre ativada por padrão; para desativá-la, use `nofonts`
- `spacing` utiliza os espaçamentos definidos neste modelo (margens, espaço entre parágrafos, indentação da primeira linha do parágrafo etc.). Esta opção é sempre ativada por padrão; para desativá-la, use `nospacing`
- `captions` e `footnotes` fazem respectivamente as legendas (das figuras e tabelas) e as notas de rodapé de acordo com este modelo. Estas opções são sempre ativadas por padrão; para desativá-las, use `nocaptions` e `nofootnotes`
- `autohttp` acrescenta o prefixo `http://` a URLs criadas com `\url` que não incluam o *schema*. Esta opção é sempre ativada por padrão; para desativá-la, use `noautohttp`
- `hidelinks`, `borderlinks` e `colorlinks` definem a aparência dos hiperlinks. `hidelinks` faz os hiperlinks sem nenhuma formatação especial; `borderlinks` faz os hiperlinks serem enviados por um quadrado colorido (apenas na tela; o quadrado não é impresso); `colorlinks` faz o texto dos hiperlinks ser colorido. A opção `colorlinks` é sempre ativada por padrão

- `biblatex` carrega a *package* `biblatex` e os estilos bibliográficos deste modelo. Esta opção é sempre ativada por padrão; para desativá-la, use `nobiblatex`
- `raggedbib` faz a bibliografia (com `biblatex`) ser formatada com alinhamento à esquerda ao invés de justificado. Esta opção é sempre ativada por padrão, exceto quando o estilo bibliográfico é `plainnat-ime` (usado nas teses); para desativá-la, use `noraggedbib`; para ativá-la incondicionalmente, use `raggedbib`
- `bibstyle=?` seleciona um estilo bibliográfico específico. O estilo padrão é `numeric`, exceto em pôsteres e apresentações (`beamer-ime`) e *reports* (`plainnat-ime`)
- `listings` carrega a *package* `listings` e diversas configurações relacionadas usadas neste modelo. Esta opção é sempre ativada por padrão; para desativá-la, use `nolistings`
- `greeny`, `bluey`, `sandy` ativam esquemas de cores diferentes para pôsteres e apresentações (o padrão é `bluey`)
- `beamer` **desativa** algumas *packages* que são incompatíveis com a classe `beamer` (note que as opções `slides` e `presentation`, discutidas abaixo, já fazem isso)
- `presentation` (ou `slides`) e `poster` ativam as opções relevantes para, respectivamente, apresentações com `beamer` ou pôsteres com `tcolorbox`
- `report` ativa as opções relevantes para documentos com capítulos (cabeçalhos das páginas, características do sumário etc.)
- `thesis` ativa a opção `report` e também define o que é necessário para a geração da capa das teses de acordo com este modelo
- `resumoabstract` define os comandos `resumo` e `abstract` de acordo com este modelo. Esta opção é ativada por padrão com `report`; para desativá-la, use `noresumoabstract`
- `brazilian` verifica se a língua portuguesa está ativa no documento e, em caso negativo, gera um erro. Esta opção é ativada por padrão com a opção `thesis`; para desativá-la, use `nobrazilian`

Annex B

Código-fonte e pseudocódigo

Com a *package listings*, programas podem ser inseridos diretamente no arquivo, como feito no caso do Programa ??, ou importados de um arquivo externo com o comando `\lstinputlisting`, como no caso do Programa ??.

Program B.1 Máximo divisor comum (arquivo importado).

```

1  FUNCTION euclid( $a, b$ )  $\triangleright$  The g.c.d. of  $a$  and  $b$ 
2       $r \leftarrow a \bmod b$ 
3      while  $r \neq 0$   $\triangleright$  We have the answer if  $r$  is 0
4           $a \leftarrow b$ 
5           $b \leftarrow r$ 
6           $r \leftarrow a \bmod b$ 
7      end
8      return  $b$   $\triangleright$  The g.c.d. is  $b$ 
9  end
```

Trechos de código curtos (menores que uma página) podem ou não ser incluídos como *floats*; trechos longos necessariamente incluem quebras de página e, portanto, não podem ser *floats*. Com *floats*, a legenda e as linhas separadoras são colocadas pelo comando `\begin{program}`; sem eles, utilize o ambiente `programruledcaption` (atenção para a colocação do comando `\label{}`, dentro da legenda), como no Programa ??¹:

Program B.2 Máximo divisor comum (em português).

```

1  FUNCAO euclides( $a, b$ )  $\triangleright$  O máximo divisor comum de  $a$  e  $b$ 
2       $r \leftarrow a \bmod b$ 
3  enquanto  $r \neq 0$   $\triangleright$  Atingimos a resposta se  $r$  é zero
4           $a \leftarrow b$ 
5           $b \leftarrow r$ 
```

cont \longrightarrow

¹ listings oferece alguns recursos próprios para a definição de *floats* e legendas, mas neste modelo não os utilizamos.

```

→ cont
6       $r \leftarrow a \bmod b$ 
7      fim
8  devolva  $b \triangleright$  O máximo divisor comum é  $b$ 
9  fim

```

Além do suporte às várias linguagens incluídas em listings, este modelo traz uma extensão para permitir o uso de pseudocódigo, útil para a descrição de algoritmos em alto nível. Ela oferece diversos recursos:

- Comentários seguem o padrão de C++ (`//` e `/* ... */`), mas o delimitador é impresso como “ \triangleright ”.
- “`:=`”, “`<>`”, “`<=`”, “`>=`” e “`!=`” são substituídos pelo símbolo matemático adequado.
- É possível acrescentar palavras-chave além de “if”, “and” etc. com a opção “`morekeywords={pchave1,pchave2}`” (para um trecho de código específico) ou com o comando `\lstset{morekeywords={pchave1,pchave2}}` (como comando de configuração geral).
- É possível usar pequenos trechos de código, como nomes de variáveis, dentro de um parágrafo normal com `\lstinline{blah}`.
- “`$...$`” ativa o modo matemático em qualquer lugar.
- Outros comandos \LaTeX funcionam apenas em comentários; fora, a linguagem simula alguns pré-definidos (`\textit{}`, `\texttt{}` etc.).
- O comando `\label` também funciona em comentários; a referência correspondente (`\ref`) indica o número da linha de código. Se quiser usá-lo numa linha sem comentários, use `/// \label{blah}`; “`///`” funciona como `//`, permitindo a inserção de comandos \LaTeX , mas não imprime o delimitador (\triangleright).
- Para suspender a formatação automática, use `\noparse{blah}`.
- Para forçar a formatação de um texto como função, identificador, palavra-chave ou comentário, use `\func{blah}`, `\id{blah}`, `\kw{blah}` ou `\comment{blah}`.
- Palavras-chave dentro de comentários não são formatadas automaticamente; se necessário, use `\func{}`, `\id{}` etc. ou comandos \LaTeX padrão.
- As palavras “Program”, “Procedure” e “Function” têm formatação especial e fazem a palavra seguinte ser formatada como função. Funções em outros lugares *não* são detectadas automaticamente; use `\func{}`, a opção “`functions={func1,func2}`” ou o comando “`\lstset{functions={func1,func2}}`” para que elas sejam detectadas.
- Além de funções, palavras-chave, strings, comentários e identificadores, há “`specialidentifiers`”. Você pode usá-los com `\specialid{blah}`, com a opção “`specialidentifiers={id1,id2}`” ou com o comando “`\lstset{specialidentifiers={id1,id2}}`”.