Uma introdução ao LaTeX

> In fact, my main conclusion after spending ten years of my life working on the TeX project is that software is hard. It's harder than anything else I've ever had to do.
>
> Knuth, Donald (2002)

Carla Negri Lintzmayer
Centro de Matemática, Computação e Cognição
Universidade Federal do ABC
carla.negri@ufabc.edu.br

em elaboração
Junho de 2018 (última atualização)

Lista de Figuras

1.1 Documentos feitos com LaTeX: livros. 7
1.2 Documentos feitos com LaTeX: artigos. 8
1.3 Documentos feitos com LaTeX: apresentações (slides). 9
1.4 Documentos feitos com LaTeX: cartas. 10
1.5 Documentos feitos com LaTeX: crachás, cartões de visita, certificados, ... (elementos repetidos). 11
1.6 Documentos feitos com LaTeX: currículos. 12
1.7 Documentos feitos com LaTeX: jornais. 13
1.8 Documentos feitos com LaTeX: pôsteres. 14

8.1 Logotipo da UFABC. 18
8.2 Logotipo da UFABC. 19

11.1 Logotipo da UFABC. 34

12.1 Ferramenta de exportação de entrada BibTeX em um site de revista científica. 44

13.1 Exemplo de uma ferramenta de seleção de cores (link). A cor e seu código RGB estão destacadas por retângulos vermelhos. 47

Sumário

0 Sobre esse material **4**

1 O que é LaTeX e por que usá-lo **5**

2 Informações técnicas **15**

3 Criando seu primeiro documento **16**

4 Escrevendo o texto **19**

5 Formatando o texto **22**
5.1 Estilo de fonte 22
5.2 Espaçamentos 24
5.3 Criando um título simples 1

6 Criando listas **3**

7 Estruturando o texto **10**
7.1 Essa é uma seção 11
7.1.1 Essa é uma subseção 11
7.2 Outra seção 11
7.3 Essa é uma seção 12
7.3.1 Essa é uma subseção 12

8 Inserindo figuras **14**

9 Inserindo tabelas **21**

10 Modo matemático **29**

11 Referências cruzadas **32**
11.1 Sobre a vida do autor 33
11.1.1 Infância 33
11.2 Sobre o trabalho do autor 33

12 Referências bibliográficas **38**

13 Hyperlinks **45**

14 Modo matemático (extra) **48**

15 Apresentação de slides **49**

16 Pôsteres **50**

17 Mais exemplos do que pode ser feito com LaTeX **51**

18 Indo além **59**

19 Referências **60**

Índice Remissivo **60**

Capítulo 0

Sobre esse material

Esse material foi criado em junho de 2018 para um minicurso de introdução ao LaTeX ministrado na Universidade Federal do ABC durante a IV Semana do CMCC. Minha intenção, no entanto, é continuar atualizando o material e deixá-lo disponível para qualquer pessoa que tiver interesse.

A ideia do minicurso é criar um documento PDF específico projetado por mim. Criar esse documento faz com que você seja exposto a todos os conceitos básicos do LaTeX, aprendendo comandos que certamente serão necessários em qualquer outro arquivo que você venha a preparar.

Sendo assim, por um lado essa apostila está estruturada como um tutorial que o ajudará na criação do documento citado acima, de forma que, se seguida sequencialmente, então no final você terá um bom conhecimento básico porém amplo de como criar documentos em LaTeX[1].

Por outro lado, ela também contém explicações gerais sobre LaTeX, de modo a servir como fonte de pesquisa para quem já tem algum conhecimento e precisa apenas de algumas informações específicas.

Os Capítulos 1 e 2 apresentam algumas informações sobre o que é o LaTeX, alguns exemplos do que podemos fazer com ele e algumas informações técnicas básicas e interessantes.

Dos Capítulos 3 a 13 você terá informações sobre como de fato criar um documento com o LaTeX. A ideia é começar com um arquivo em branco e construir o documento que eu mencionei acima usando vários dos recursos disponíveis.

Os Capítulos 14 a 18 apresentam informações extras que foram mencionadas rapidamente durante o minicurso e aqui encontram-se mais detalhadas. São algumas informações extras sobre o modo matemático, como fazer apresentações e posters usando o LaTeX e como ir além do que está sendo mostrado nesse documento.

O Capítulo 18, inclusive, é importante para aqueles que pretendem continuar usando LaTeX. Visto que o LaTeX é muito poderoso e abrangente e que esse material de modo algum é completo, você precisa ser capaz de procurar informações em outros locais, dependendo do que estiver querendo fazer.

Por fim, o Capítulo 19 apresenta algumas referências que foram consultadas durante o preparo dessa apostila.

[1] FYI, o minicurso foi projetado para durar 4 horas.

Capítulo 1

O que é LaTeX e por que usá-lo

LaTeX é um sistema de preparação de documentos profissionais de alta qualidade tipográfica que inclui características projetadas para a produção de documentos técnicos e científicos. Foi construído por Leslie Lamport na década de 80 sobre o programa TeX criado Donald Knuth na década de 70.

Os famosos processadores de texto como LibreOffice Writer, Microsoft Word e Apple Pages são editores WYSIWYG, sigla de *what you see is what you get*, pois permitem que o usuário veja a aparência final do seu documento (como ele será impresso) enquanto está produzindo o conteúdo do mesmo.

Além disso, os documentos produzidos por esses processadores de texto possuem um formato específico (DOC, DOCX, RTF, ODT, ...) e muito provavelmente só poderão ser corretamente manipulados pelo próprio programa que os criou (já tentou abrir um arquivo DOC com um programa estilo bloco de notas?). Agora, e se o programa for atualizado para uma nova versão? Abrir um documento do Microsoft Word em uma versão diferente da que ele foi criado, por exemplo, pode causar incoerências na formatação.

Com essas duas características dos processadores WYSIWYG começamos a ver as vantagens do LaTeX. Em primeiro lugar, ele é baseado na ideia de WYSIWYM, sigla de *what you see is what you mean*, no sentido de que o conteúdo e o formato do documento ficam claramente separados, permitindo que você se foque na geração do conteúdo. Em segundo lugar, o texto a ser impresso é escrito em um arquivo texto simples juntamente com os comandos de formatação, de forma que qualquer programa estilo bloco de notas consegue manipular o arquivo. A saída final (o documento que será impresso) pode ser visualizada previamente depois do arquivo para LaTeX ser devidamente processado.

Chamamos o arquivo texto escrito para LaTeX de *código-fonte*[1]. Um código-fonte para o LaTeX possui o conteúdo do seu documento intercalado com *comandos* que vão indicar ao LaTeX o que deve ser feito com relação à formatação. Vou me adiantar aqui e já dizer que existem dois tipos de comando. O primeiro é da forma `\comando[opções]{parâmetro}`. Por exemplo, o comando `\textbf{casa}` indica que a palavra casa deve aparecer em negrito no documento final (**casa**). O segundo é da forma `\begin{comando}` *texto* `\end{comando}`, caso em que é chamado de *ambiente*. Todo texto interno ao

[1] Esse termo não é exclusivo do LaTeX: qualquer texto escrito de forma ordenada com instruções que ainda vai ser processado para ser compreendido pelo computador é chamado assim (se você sabe programação, já deve estar familiarizado com o termo).

`begin`–`end` de um ambiente é alterado pelo que quer que aquele ambiente esteja querendo formatar. Esses conceitos ficarão mais claros nos próximos capítulos, quando veremos aplicações concretas deles.

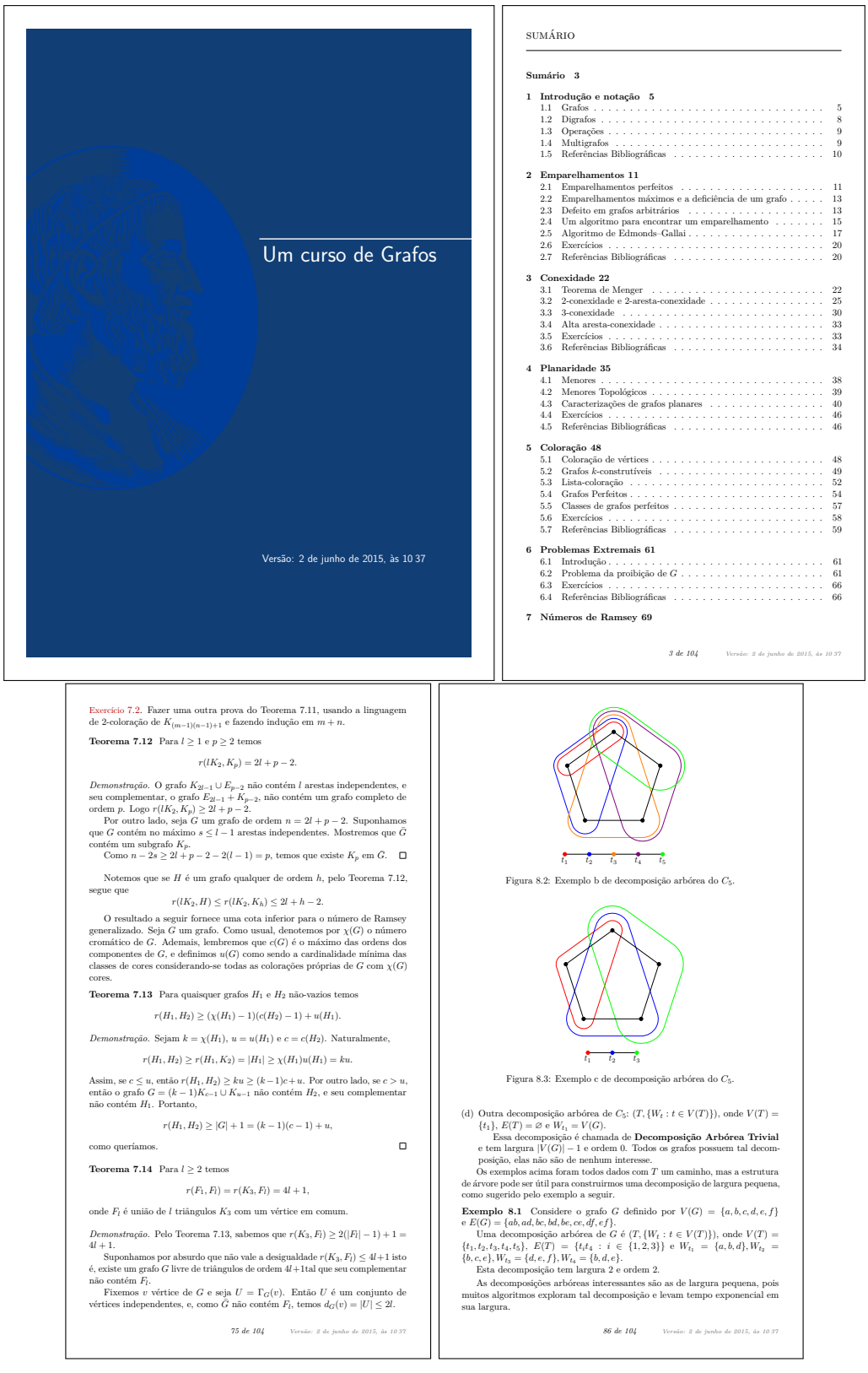
Uma vez que tivermos um código-fonte para LaTeX pronto, é necessário processá-lo para gerar o documento final. Vários formatos diferentes são possíveis, mas provavelmente o mais útil e comum é o PDF (*portable document format*), que realmente tem a aparência de como será a impressão e pode ser facilmente transferido entre computadores. Chamamos essa transformação de código-fonte para documento final de *compilação*. Não é minha intenção te explicar detalhes de como a compilação é feita e eu nem acho que é sua obrigação saber disso, mas esse termo provavelmente vai aparecer no restante desse documento e eu preciso que você saiba sobre essa ideia por trás dele: transformar código-fonte em arquivo final.

Bom, mas voltando ao objetivo desse capítulo[2], podemos ainda citar várias outras vantagens do LaTeX sobre os processadores de texto normais. A mais óbvia é com relação à aparência do documento: qualquer documento feito com LaTeX tem aparência profissional (como essa apostila, por exemplo – ela não é linda?). A inserção de fórmulas matemáticas é suportada de forma muito simples. Modificar a formatação de algo que está presente no documento todo pode ser feito com a mudança de um ou poucos comandos. Introdução de novas figuras e tabelas (e elementos "flutuantes" no geral) é muito mais simples pois os posicionamentos ficam completamente a cargo do LaTeX e não do escritor.

É claro que existem algumas desvantagens no LaTeX também. Em primeiro lugar, a curva de aprendizado do LaTeX não é mínima como a de editores de texto como Microsoft Word. Mudanças muito específicas no *layout* do documento podem requerer mais tempo do que o previsto. E em geral vai ser muito difícil fazer documentos mal estruturados e desorganizados.

Se você ainda não está convencido de que usar LaTeX é muito mais legal, vou colocar alguns exemplos nas Figuras 1.1 a 1.8 do que é possível fazer com ele.

[2]Se não ficou claro, é o de catequização.



Um curso de Grafos

Versão: 2 de junho de 2015, às 10 37

SUMÁRIO

Sumário 3

1 Introdução e notação 5
1.1 Grafos . . . 5
1.2 Digrafos . . . 8
1.3 Operações . . . 9
1.4 Multigrafos . . . 9
1.5 Referências Bibliográficas . . . 10

2 Emparelhamentos 11
2.1 Emparelhamentos perfeitos . . . 11
2.2 Emparelhamentos máximos e a deficiência de um grafo . . . 13
2.3 Defeito em grafos arbitrários . . . 13
2.4 Um algoritmo para encontrar um emparelhamento . . . 15
2.5 Algoritmo de Edmonds–Gallai . . . 17
2.6 Exercícios . . . 20
2.7 Referências Bibliográficas . . . 20

3 Conexidade 22
3.1 Teorema de Menger . . . 22
3.2 2-conexidade e 2-aresta-conexidade . . . 25
3.3 3-conexidade . . . 30
3.4 Alta aresta-conexidade . . . 33
3.5 Exercícios . . . 33
3.6 Referências Bibliográficas . . . 34

4 Planaridade 35
4.1 Menores . . . 38
4.2 Menores Topológicos . . . 39
4.3 Caracterizações de grafos planares . . . 40
4.4 Exercícios . . . 46
4.5 Referências Bibliográficas . . . 46

5 Coloração 48
5.1 Coloração de vértices . . . 48
5.2 Grafos k-construtíveis . . . 49
5.3 Lista-coloração . . . 52
5.4 Grafos Perfeitos . . . 54
5.5 Classes de grafos perfeitos . . . 57
5.6 Exercícios . . . 58
5.7 Referências Bibliográficas . . . 59

6 Problemas Extremais 61
6.1 Introdução . . . 61
6.2 Problema da proibição de G . . . 61
6.3 Exercícios . . . 66
6.4 Referências Bibliográficas . . . 66

7 Números de Ramsey 69

3 de 104 *Versão: 2 de junho de 2015, às 10 37*

Exercício 7.2. Fazer uma outra prova do Teorema 7.11, usando a linguagem de 2-coloração de $K_{(m-1)(n-1)+1}$ e fazendo indução em $m+n$.

Teorema 7.12 Para $l \geq 1$ e $p \geq 2$ temos

$$r(lK_2, K_p) = 2l + p - 2.$$

Demonstração. O grafo $K_{2l-1} \cup E_{p-2}$ não contém l arestas independentes, e seu complementar, o grafo $E_{2l-1} + K_{p-2}$, não contém um grafo completo de ordem p. Logo $r(lK_2, K_p) \geq 2l + p - 2$.

Por outro lado, seja G um grafo de ordem $n = 2l + p - 2$. Suponhamos que G contém no máximo $s \leq l - 1$ arestas independentes. Mostremos que G contém um subgrafo K_p.

Como $n - 2s \geq 2l + p - 2 - 2(l-1) = p$, temos que existe K_p em G. □

Notemos que se H é um grafo qualquer de ordem h, pelo Teorema 7.12, segue que

$$r(lK_2, H) \leq r(lK_2, K_h) \leq 2l + h - 2.$$

O resultado a seguir fornece uma cota inferior para o número de Ramsey generalizado. Seja G um grafo. Como usual, denotemos por $\chi(G)$ o número cromático de G. Ademais, lembremos que $c(G)$ é o máximo das ordens dos componentes de G, e definimos $u(G)$ como sendo a cardinalidade mínima das classes de cores considerando-se todas as colorações próprias de G com $\chi(G)$ cores.

Teorema 7.13 Para quaisquer grafos H_1 e H_2 não-vazios temos

$$r(H_1, H_2) \geq (\chi(H_1) - 1)(c(H_2) - 1) + u(H_1).$$

Demonstração. Sejam $k = \chi(H_1)$, $u = u(H_1)$ e $c = c(H_2)$. Naturalmente,

$$r(H_1, H_2) \geq r(H_1, K_2) = |H_1| \geq \chi(H_1)u(H_1) = ku.$$

Assim, se $c \leq u$, então $r(H_1, H_2) \geq ku \geq (k-1)c + u$. Por outro lado, se $c > u$, então o grafo $G = (k-1)K_{c-1} \cup K_{u-1}$ não contém H_2, e seu complementar não contém H_1. Portanto,

$$r(H_1, H_2) \geq |G| + 1 = (k-1)(c-1) + u,$$

como queríamos. □

Teorema 7.14 Para $l \geq 2$ temos

$$r(F_1, F_l) = r(K_3, F_l) = 4l + 1,$$

onde F_l é união de l triângulos K_3 com um vértice em comum.

Demonstração. Pelo Teorema 7.13, sabemos que $r(K_3, F_l) \geq 2(|F_l| - 1) + 1 = 4l + 1$.

Suponhamos por absurdo que não vale a desigualdade $r(K_3, F_l) \leq 4l + 1$ isto é, existe um grafo G livre de triângulos de ordem $4l + 1$tal que seu complementar não contém F_l.

Fixemos v vértice de G e seja $U = \Gamma_G(v)$. Então U é um conjunto de vértices independentes, e, como $\bar{G}$ não contém F_l, temos $d_G(v) = |U| \leq 2l$.

75 de 104 *Versão: 2 de junho de 2015, às 10 37*

Figura 8.2: Exemplo b de decomposição arbórea do C_5.

Figura 8.3: Exemplo c de decomposição arbórea do C_5.

(d) Outra decomposição arbórea de C_5: $(T, \{W_t : t \in V(T)\})$, onde $V(T) = \{t_1\}$, $E(T) = \varnothing$ e $W_{t_1} = V(G)$.

Essa decomposição é chamada de **Decomposição Arbórea Trivial** e tem largura $|V(G)| - 1$ e ordem 0. Todos os grafos possuem tal decomposição, elas não são de nenhum interesse.

Os exemplos acima foram todos dados com T um caminho, mas a estrutura de árvore pode ser útil para construirmos uma decomposição de largura pequena, como sugerido pelo exemplo a seguir.

Exemplo 8.1 Considere o grafo G definido por $V(G) = \{a, b, c, d, e, f\}$ e $E(G) = \{ab, ad, bc, bd, be, ce, df, ef\}$.

Uma decomposição arbórea de G é $(T, \{W_t : t \in V(T)\})$, onde $V(T) = \{t_1, t_2, t_3, t_4, t_5\}$, $E(T) = \{t_i t_4 : i \in \{1, 2, 3\}\}$ e $W_{t_1} = \{a, b, d\}, W_{t_2} = \{b, c, e\}, W_{t_3} = \{d, e, f\}, W_{t_4} = \{b, d, e\}$.

Esta decomposição tem largura 2 e ordem 2.

As decomposições arbóreas interessantes são as de largura pequena, pois muitos algoritmos exploram tal decomposição e levam tempo exponencial em sua largura.

86 de 104 *Versão: 2 de junho de 2015, às 10 37*

Figura 1.1: Documentos feitos com LaTeX: livros.

Two-Dimensional Knapsack for Circles

Carla Negri Lintzmayer[1](✉), Flávio Keidi Miyazawa[2], and Eduardo Candido Xavier[2]

[1] Center for Mathematics, Computer, and Cognition, Federal University of ABC, Santo André, Brazil
carla.negri@ufabc.edu.br

[2] Institute of Computing, University of Campinas, Campinas, Brazil
{fkm,eduardo}@ic.unicamp.br

Abstract. In this paper we consider the Two-dimensional Knapsack for Circles problem, in which we are given a set $\mathcal{C}$ of circles and want to pack a subset $\mathcal{C}' \subseteq \mathcal{C}$ of them into a rectangular bin of dimensions w and h such that the sum of the area of circles in $\mathcal{C}'$ is maximum. By packing we mean that the circles do not overlap and they are fully contained inside the bin. We present a polynomial-time approximation scheme that, for any $\epsilon > 0$, gives an approximation algorithm that packs a subset of the input circles into an augmented bin of dimensions w and $(1 + O(\epsilon))h$ such that the area packed is at least $(1 - O(\epsilon))$ times the area packed by an optimal solution into the regular bin of dimensions w and h. This result also extends to the multiple knapsack version of this problem.

Keywords: Circle packing · Two-dimensional Knapsack
Polynomial-time approximation scheme

1 Introduction

In the *Two-dimensional Knapsack* problem, we are given a set of 2D items with profits and we want to pack some of the items into one 2D recipient such that the sum of their profits is maximum. By packing we mean that the items do not overlap and they are fully contained inside the recipient. We consider the variation where the items are circles, the item profit is the circle's area, and the recipient is a rectangle.

The Two-dimensional Knapsack is a generalization of the famous *Knapsack* problem in one dimension, where we have a set of items with profits and weights and want to find a subset of items of maximum profit whose total weight is at most the capacity of the given knapsack. As such, it is also NP-hard [6]. This problem is also related to the likewise famous *Two-dimensional Bin Packing*

This work was supported by São Paulo Research Foundation (grants 2016/14132-4, 2015/11937-9, 2016/23552-7, 2016/01860-1) and National Counsel of Technological and Scientific Development (grants 306358/2014-0, 311499/2014-7, and 425340/2016-3).

© Springer International Publishing AG, part of Springer Nature 2018
M. A. Bender et al. (Eds.): LATIN 2018, LNCS 10807, pp. 741–754, 2018.
https://doi.org/10.1007/978-3-319-77404-6_54

Theorem 1. TransformOptimal *transforms a packing of $\mathcal{C}^* \subseteq \mathcal{C}$ into bin $B^*_{w\times h}$, where $p(\mathcal{C}^*) = \mathrm{OPT}_{w\times h}(\mathcal{C})$, into a structured packing of $\mathcal{C}^\star \subseteq \mathcal{C}^*$ into the same bin such that $p(\mathcal{C}^\star) \geq (1 - 64\varepsilon)\mathrm{OPT}_{w\times h}(\mathcal{C})$.*

Proof (Sketch). We account the profit loss when removing circles from $\mathcal{H}^*_t$ (at most $\varepsilon\mathrm{OPT}_{w\times h}(\mathcal{C})$), from D (at most $96\varepsilon/\pi\mathrm{OPT}_{w\times h}(\mathcal{C}) < 31\varepsilon\mathrm{OPT}_{w\times h}(\mathcal{C})$, by Lemma 3), from virtual bins in D (at most $16\varepsilon\sum_{j\geq 1} p(\mathcal{S}^*_{j-1}) \leq 16\varepsilon\mathrm{OPT}_{w\times h}(\mathcal{C})$, by Lemma 4), and from bins of grid j over $B^*_{w\times h}$ that intersect the borders of circles of $\mathcal{S}^*_{j-1}$ for all $j \geq 1$ (at most $16\varepsilon\mathrm{OPT}_{w\times h}(\mathcal{C})$, by Lemma 4). □

6 Analysis of the Algorithm

Recall that during the execution of CircleKnapsack we create sets $F^t_j \subseteq \mathsf{G}_j(B_{w\times h})$ of bins, for all $j \geq 0$, which pack a subset of circles of $\mathcal{S}_j$ using an optimal solution for 2DMK. Thus, by construction,

$$p(\text{CircleKnapsack}(\mathcal{C}, w, h, \varepsilon)) = \sum_{j\geq 0} \mathrm{OPT}^{\mathrm{2DMK}}_{w_j\times h_j}(\mathcal{S}_j, |F^t_j|). \quad (1)$$

On the other hand, TransformOptimal receives a packing of $\mathcal{C}^* \subseteq \mathcal{C}$ into $B^*_{w\times h}$ such that $p(\mathcal{C}^*) = \mathrm{OPT}_{w\times h}(\mathcal{C})$ and modify it into a structured packing containing circles from $\mathcal{C}^\star \subseteq \mathcal{C}^*$. For every $j \geq 0$, let $P^\star_j$ be the subset of $\mathsf{G}_j(B^*_{w\times h})$ that is used to pack circles from $\mathcal{S}^\star_j \cap \mathcal{C}^\star$. Thus, $\sum_{j\geq 0} p(P^\star_j) = p(\mathcal{C}^\star)$. We showed in Theorem 1 that $p(\mathcal{C}^\star) \geq (1 - 64\varepsilon)\mathrm{OPT}_{w\times h}(\mathcal{C})$. We also have

$$\sum_{j\geq 0} p(P^\star_j) \geq (1 - 64\varepsilon)\mathrm{OPT}_{w\times h}(\mathcal{C}). \quad (2)$$

Lemma 5 thus shows a relation, for $j \geq 0$, between $\mathrm{OPT}^{\mathrm{2DMK}}_{w_j\times h_j}(\mathcal{S}_j, |F^t_j|)$ and $p(P^\star_j)$. Theorem 2 at last shows a bound on the total profit of circles packed by CircleKnapsack.

Lemma 5. *For any $j \geq 0$,*

$$\sum_{k=0}^{j} \mathrm{OPT}^{\mathrm{2DMK}}_{w_k\times h_k}(\mathcal{S}_k, |F^t_k|) \geq (1 - 16\varepsilon)\sum_{k=0}^{j} p(P^\star_k). \quad (3)$$

Proof. We will show, by induction on j, that (3) is valid. When $j = 0$, $P^\star_0$ is a solution for 2DMK for input $(\mathcal{S}_0, w, h, 1)$, so $\mathrm{OPT}^{\mathrm{2DMK}}_{w\times h}(\mathcal{S}_0, 1) \geq p(P^\star_0)$, and the result follows.

Suppose now that $j \geq 1$. If $|F^t_j| \geq |P^\star_j|$, then we have $\mathrm{OPT}^{\mathrm{2DMK}}_{w_j\times h_j}(\mathcal{S}_j, |F^t_j|) \geq \mathrm{OPT}^{\mathrm{2DMK}}_{w_j\times h_j}(\mathcal{S}_j, |P^\star_j|) \geq p(P^\star_j)$. By the induction hypothesis, we have (3) valid for $j - 1$ and so the result follows.

Therefore, consider $|F^t_j| < |P^\star_j|$. As simplification, let us call B the bin of size $w \times h$ returned by CircleKnapsack and B^* the bin of size $w \times h$ returned by TransformOptimal.

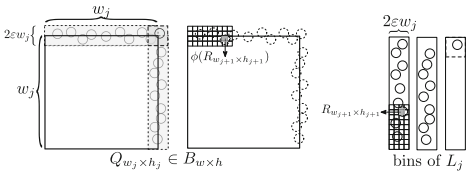


Fig. 1. Circles from $\mathcal{W}_j$ that intersect the borders of bin $Q_{w_j\times h_j}$ are removed from their current positions and placed into three temporary bins of size $2\varepsilon w_j \times h_j$.

Let $\mathcal{W}_j$ be the set of circles of $\mathcal{S}^*_j$ that intersect the lines of grid j over $B^*_{w\times h}$. In the end of the first phase, each circle of $\mathcal{W}_j$ will either be reallocated inside $B^*_{w\times h}$ or discarded. We start by temporarily removing such circles from $B^*_{w\times h}$ and placing them into a set L_j of bins in the following way. For each subbin $Q_{w_j\times h_j}$ of $\mathsf{G}_j(B^*_{w\times h})$, we only need to care for removing circles that intersect its superior and right borders (doing this for all subbins of the grid will collectively free the grid lines). See Fig. 1 for a visual depiction of the following discussion.

Since all circles of $\mathcal{W}_j$ have diameter at most εw_j, two bins of size $2\varepsilon w_j \times h_j$ comprise almost all circles that intersect the two mentioned borders of $Q_{w_j\times h_j}$. A third bin of size $2\varepsilon w_j \times 2\varepsilon w_j$ is needed to comprise circles from the top right corner. Thus, all circles from these borders can fit into three bins of size $2\varepsilon w_j \times h_j$ each, so we remove them from the current packing, place them into these bins preserving the relative positions, and add all these bins to L_j. Lemma 2 shows the total area of the bins in L_j.

Lemma 2. *For each $j \geq 1$, $\mathsf{Area}(L_j) = 6\varepsilon wh$.*

Proof. There are $\frac{w}{w_j}\frac{h}{h_j}$ subbins in $\mathsf{G}_j(B_{w\times h})$ and each of them contributes to L_j with three rectangles of area $2\varepsilon w_j h_j$ each. □

Now we create a new bin D of size $w \times 6\varepsilon h$. Note that since both w and $6\varepsilon h$ are multiple of w_j, any grid j over D perfectly divides D into bins of size $w_j \times h_j$. Also note that every $1/(2\varepsilon)$ bins of L_j form a bin of size $w_j \times h_j$. Thus, by Lemma 2, all bins of size $w_j \times h_j$ formed from L_j can be packed into D. The idea now for each of these bins is to (i) move it to D, (ii) move it back to $B^*_{w\times h}$, or (iii) discard it.

Suppose we pack all bins of size $w_j \times h_j$ formed from L_j into D and consider grid $j + 1$ over $B^*_{w\times h}$ and over D. Since $2\varepsilon w_j$ is a multiple of $w_{j+1} = \varepsilon^{2r} w_j$, we can make a bijection ϕ between the subbins of $\mathsf{G}_{j+1}(D)$ and the subbins of $\mathsf{G}_{j+1}(B^*_{w\times h})$ that are in the rectangles of size $w_j \times 2\varepsilon w_j$ around the superior and right borders of each $Q_{w_j\times h_j} \in \mathsf{G}_j(B^*_{w\times h})$ (the same ones considered to build L_j). So, for all subbin $R_{w_{j+1}\times h_{j+1}}$ in D there exists a corresponding subbin

Theorem 6. *Let $(\mathcal{C}, W, H)$ be an instance of the Two-dimensional Knapsack for Circles such that $H/W \in O(1)$ and $\mathcal{C}$ contains n circles. For any constant $0 < \epsilon \leq 1$, we can obtain, in time polynomial in n, a packing of a subset of circles from $\mathcal{C}$ into an augmented bin of size $W \times (1 + 7\epsilon)H$ such that their total profit is at least $(1 - O(\epsilon))\mathrm{OPT}_{W\times H}(\mathcal{C})$.*

Proof (Sketch). We write w, h, and ε as functions of W, H, and ϵ, respectively, such that $1/\varepsilon$ is integer multiple of 2 and $h/(w\varepsilon)$ is integer. Then we use VCircleKnapsack and analyse its result considering the original input. □

References

1. Adamaszek, A., Wiese, A.: A quasi-PTAS for the two-dimensional geometric Knapsack problem. In: Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2015), pp. 1491–1505. Society for Industrial and Applied Mathematics, Philadelphia (2015)
2. Christensen, H.I., Khan, A., Pokutta, S., Tetali, P.: Approximation and online algorithms for multidimensional bin packing: a survey. Comput. Sci. Rev. **24**, 63–79 (2017)
3. Demaine, E.D., Fekete, S.P., Lang, R.J.: Circle Packing for Origami Design Is Hard. A K Peters/CRC Press, Singapore (2010). pp. 609–626
4. Fishkin, A.V., Gerber, O., Jansen, K., Solis-Oba, R.: On packing squares with resource augmentation: maximizing the profit. In: Proceedings of the 2005 Australasian Symposium on Theory of Computing (CATS 2005), pp. 61–67. Australian Computer Society Inc., Darlinghurst (2005)
5. Fishkin, A.V., Gerber, O., Jansen, K., Solis-Oba, R.: Packing weighted rectangles into a square. In: Jędrzejowicz, J., Szepietowski, A. (eds.) MFCS 2005. LNCS, vol. 3618, pp. 352–363. Springer, Heidelberg (2005). https://doi.org/10.1007/11549345_31
6. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1979)
7. Hifi, M., M'Hallah, R.: A literature review on circle and sphere packing problems: models and methodologies. Adv. Oper. Res. **2009**, 1–22 (2009)
8. Hokama, P., Miyazawa, F.K., Schouery, R.C.S.: A bounded space algorithm for online circle packing. Inf. Process. Lett. **116**(5), 337–342 (2016)
9. Jansen, K., Solis-Oba, R.: Packing squares with profits. SIAM J. Discret. Math. **26**(1), 263–279 (2012)
10. Jansen, K., Zhang, G.: Maximizing the total profit of rectangles packed into a rectangle. Algorithmica **47**(3), 323–342 (2007)
11. Lenstra, H.W.: Integer programming with a fixed number of variables. Math. Oper. Res. **8**(4), 538–548 (1983)
12. Lodi, A., Martello, S., Monaci, M., Vigo, D.: Two-Dimensional Bin Packing Problems, pp. 107–129. Wiley, Hoboken (2013)
13. Meir, A., Moser, L.: On packing of squares and cubes. J. Comb. Theory **5**(2), 126–134 (1968)
14. Miyazawa, F.K., Pedrosa, L.L.C., Schouery, R.C.S., Sviridenko, M., Wakabayashi, Y.: Polynomial-time approximation schemes for circle and other packing problems. Algorithmica **76**(2), 536–568 (2016)
15. Szabó, P.G., Markót, M.C., Csendes, T., Specht, E., Casado, L.G., García, I.: New Approaches to Circle Packing in a Square. Springer Optimization and its Applications. Springer, New York (2007). https://doi.org/10.1007/978-0-387-45676-8

Figura 1.2: Documentos feitos com LaTeX: artigos.

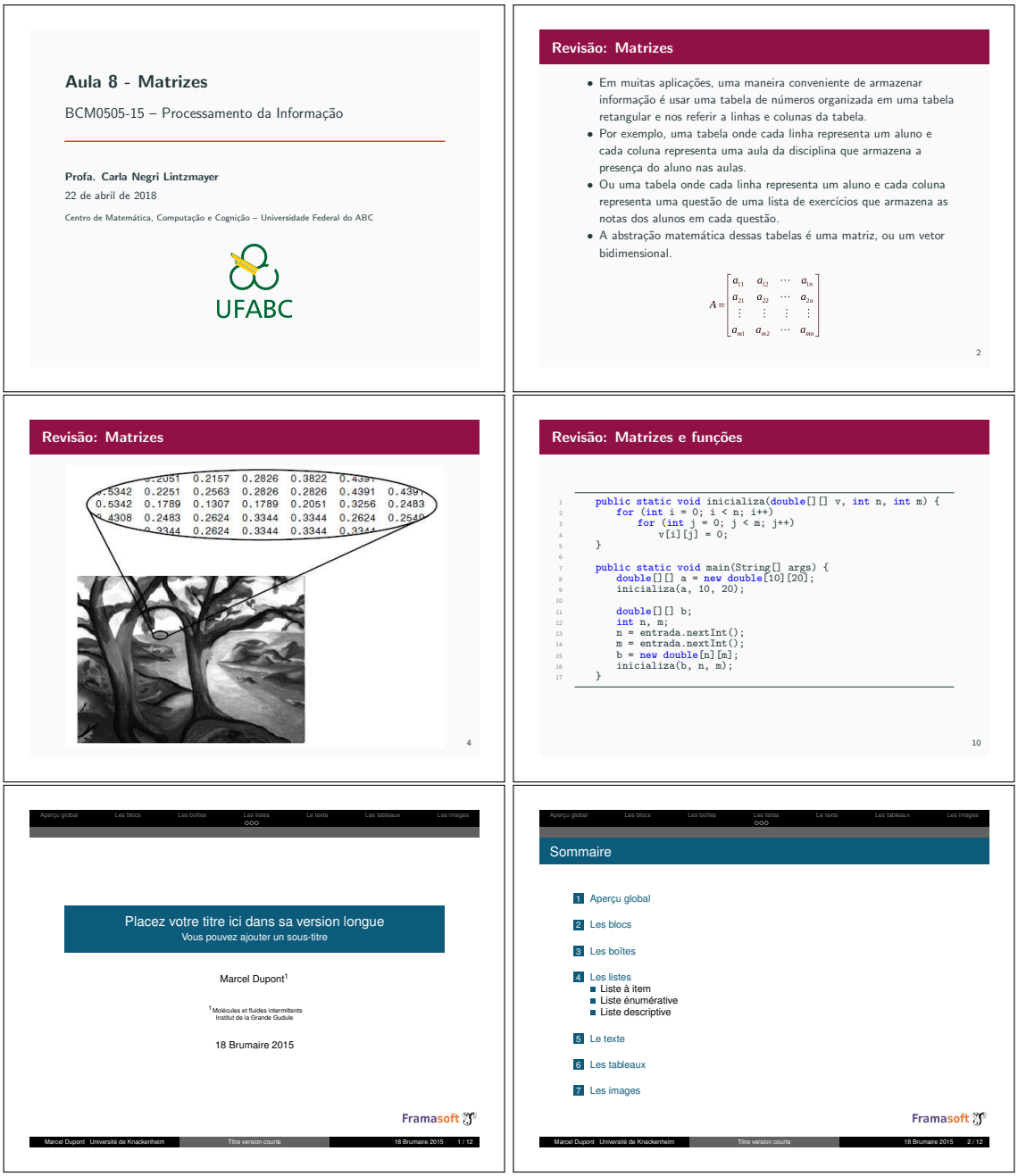


Figura 1.3: Documentos feitos com LaTeX: apresentações (slides).

Nombre Apellido Apellido
Dirección
CP, Ciudad
País
000-000-000-000
mailmailmail@gmail.com

Destinatario
Departamento, Empresa

27 de octubre de 2014

Estimado Destinatario,

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque ultrices ultricies sapien et vehicula. Sed ut quam mattis, dignissim augue et, volutpat libero. Nulla facilisi. Nulla facilisi. Suspendisse eu libero ac mi tincidunt ullamcorper eu id risus. Maecenas dignissim tortor ac tellus tristique, et hendrerit risus finibus. Morbi rhoncus, ex in placerat lacinia, justo tortor posuere ipsum, ac scelerisque urna lectus porttitor quam. Quisque quis neque massa. Aenean metus ante, lacinia vitae eros ut, porta ullamcorper mi. Sed varius bibendum interdum. Aenean at ligula iaculis nulla egestas tristique sit amet eget eros. Nam tincidunt tempor neque a sollicitudin. Integer risus sapien, tempor vel congue eget, mattis sed mi. Nullam condimentum imperdiet pulvinar.

Nam in mi sagittis, rhoncus nisl id, elementum tortor. Pellentesque id risus laoreet mi ullamcorper gravida. Nunc eget orci vitae eros efficitur consectetur a porttitor urna. Suspendisse dapibus sapien ac nulla tincidunt fermentum. Suspendisse eleifend iaculis ex et porta. Morbi at dignissim velit. Nunc non lacus ut arcu luctus dapibus. Nulla facilisis vehicula ligula non malesuada. Sed fermentum placerat mi eu finibus. Morbi nulla velit, volutpat quis efficitur in, vestibulum sed odio. In tempus, quam quis bibendum iaculis, nulla mauris rutrum leo, pellentesque molestie est lacus ut neque. Integer luctus elementum ipsum sed molestie. Vestibulum non rutrum enim. Nulla iaculis finibus elementum. Phasellus faucibus ante et pretium semper.

Morbi volutpat in ligula ut blandit. Vivamus et arcu venenatis, faucibus ante nec, suscipit dolor. Duis id molestie sapien, nec dictum ante. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut at ante sem. Ut porttitor, tortor at convallis convallis, neque mi convallis ipsum, vel ultrices enim dolor sed diam. In dictum vulputate massa id molestie. Donec sollicitudin nisi vestibulum diam varius aliquet.

Muchas gracias por su tiempo e interés y reciba un cordial saludo.

Nombre Apellido Apellido

Adjunto: CV

Figura 1.4: Documentos feitos com LaTeX: cartas.

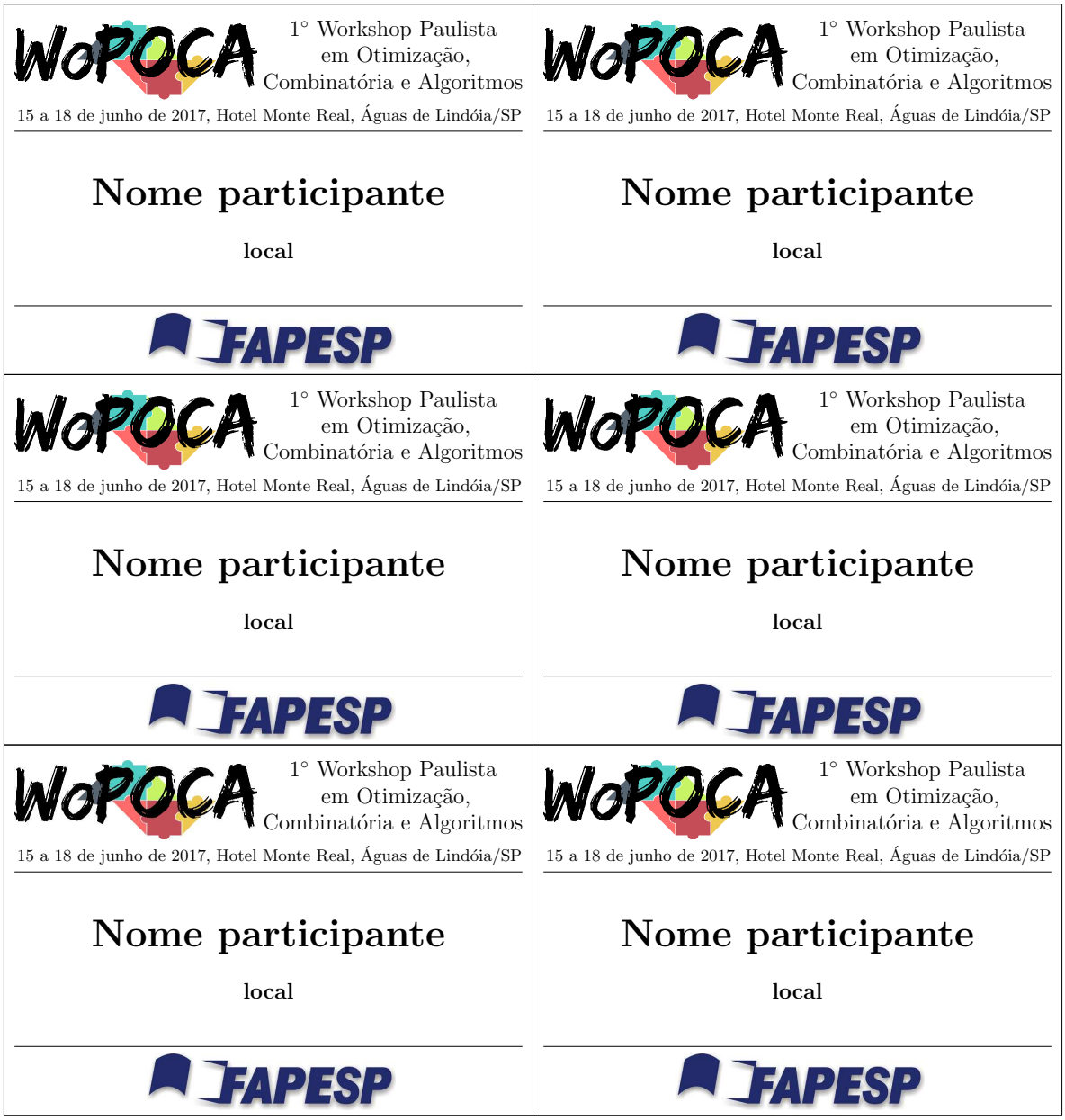


Figura 1.5: Documentos feitos com LaTeX: crachás, cartões de visita, certificados, ... (elementos repetidos).

John Doe

Resumé title

Some quote

Education

year–year **Degree**, *Institution*, City, *Grade*.
Description

year–year **Degree**, *Institution*, City, *Grade*.
Description

Master thesis

title *Title*

supervisors Supervisors

description Short thesis abstract

Experience

Vocational

year–year **Job title**, *Employer*, City.
General description no longer than 1–2 lines.
Detailed achievements:

- Achievement 1;
- Achievement 2, with sub-achievements:
  - Sub-achievement (a);
  - Sub-achievement (b), with sub-sub-achievements (don't do this!);
    - Sub-sub-achievement i;
    - Sub-sub-achievement ii;
    - Sub-sub-achievement iii;
  - Sub-achievement (c);
- Achievement 3.

year–year **Job title**, *Employer*, City.
Description line 1
Description line 2

Miscellaneous

street and number – postcode city – country
+1 (234) 567 890 • *+2 (345) 678 901* • *+3 (456) 789 012*
john@doe.org • *www.johndoe.com* • *john.doe* • *jdoe*
jdoe • *additional information*

Figura 1.6: Documentos feitos com LaTeX: currículos.

"All the News I Feel Like Printing."

Committee Times

Washington DC

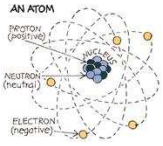
VOL.I. . . No.1 | *OCTOBER 6, 2016* | ZERO DOLLARS

Geek Designs New LaTeX Package

By MATTHEW ALLEN

The package is basically a redefinition of the `\maketitle` command. The model was the New York Times—hopefully I haven't violated any copyright laws. I also had to redefine the plain pagestyle. It kept me busy for a few nights after work. The rest is packages other people have written.

The `multicol` package allows using multiple columns without starting a new page. Using floats is not possible in a columns environment, however with the `picinpar` package, I can set a picture inside a block of text—just like you one you see here. Isn't LaTeX cool? And now we're just filling more space, and yet more space.



The Atom

Another Headline

This is just an example to fill up some space, but as long as I have your attention, I'll give some newspaper advice.

I suppose we could also show how an equation is type set:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

and there you have it.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer 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.

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, consectetuer 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.

Figura 1.7: Documentos feitos com LaTeX: jornais.

Unnecessarily Complicated Research Title

John Smith, James Smith and Jane Smith

Department and University Name

Objectives

Lorem ipsum dolor sit amet, consectetur, nunc tellus pulvinar tortor, commodo eleifend risus arcu sed odio:

- Mollis dignissim, magna augue tincidunt dolor, interdum vestibulum urna
- Sed aliquet luctus lectus, eget aliquet leo ullamcorper consequat. Vivamus eros sem, iaculis ut euismod non, sollicitudin vel orci.
- Nascetur ridiculus mus.
- Euismod non erat. Nam ultricies pellentesque nunc, ultrices volutpat nisl ultrices a.

Introduction

Lorem ipsum dolor **sit amet**, consectetur adipiscing elit. Sed commodo molestie porta. Sed ultrices scelerisque sapien ac commodo. Donec ut volutpat elit. Sed laoreet accumsan mattis. Integer sapien tellus, auctor ac blandit eget, sollicitudin vitae lorem. Praesent dictum tempor pulvinar. Suspendisse potenti. Sed tincidunt varius ipsum, et porta nulla suscipit et. Etiam congue bibendum felis, ac dictum augue cursus a. **Donec** magna eros, iaculis sit amet placerat quis, laoreet id est. In ut orci purus, interdum ornare nibh. Pellentesque pulvinar, nibh ac malesuada accumsan, urna nunc convallis tortor, ac vehicula nulla tellus eget nulla. Nullam lectus tortor, *consequat tempor hendrerit* quis, vestibulum in diam. Maecenas sed diam augue.

Lorem ipsum dolor **sit amet**, consectetur adipiscing elit. Sed commodo molestie porta. Sed ultrices scelerisque sapien ac commodo. Donec ut volutpat elit. Sed laoreet accumsan mattis. Integer sapien tellus, auctor ac blandit eget, sollicitudin vitae lorem. Praesent dictum tempor pulvinar. Suspendisse potenti. Sed tincidunt varius ipsum, et porta nulla suscipit et. Etiam congue bibendum felis, ac dictum augue cursus a.

This statement requires citation [1].

Placeholder Image

Figure 1: Figure caption

Materials

The following materials were required to complete the research:

- Curabitur pellentesque dignissim
- Eu facilisis est tempus quis
- Duis porta consequat lorem
- Eu facilisis est tempus quis

The materials were prepared according to the steps outlined below:

1. Curabitur pellentesque dignissim
2. Eu facilisis est tempus quis
3. Duis porta consequat lorem
4. Curabitur pellentesque dignissim

Methods

Lorem ipsum dolor **sit amet**, consectetur adipiscing elit. Sed laoreet accumsan mattis. Integer sapien tellus, auctor ac blandit eget, sollicitudin vitae lorem. Praesent dictum tempor pulvinar. Suspendisse potenti. Sed tincidunt varius ipsum, et porta nulla suscipit et. Etiam congue bibendum felis, ac dictum augue cursus a. **Donec** magna eros, iaculis sit amet placerat quis, laoreet id est. In ut orci purus, interdum ornare nibh. Pellentesque pulvinar, nibh ac malesuada accumsan, urna nunc convallis tortor, ac vehicula nulla tellus eget nulla. Nullam lectus tortor, *consequat tempor hendrerit* quis, vestibulum in diam. Maecenas sed diam augue.

Important Result

Lorem ipsum dolor **sit amet**, consectetur adipiscing elit. Sed commodo molestie porta. Sed ultrices scelerisque sapien ac commodo. Donec ut volutpat elit.

Mathematical Section

Nam quis odio enim, in molestie libero. Vivamus cursus mi at nulla elementum sollicitudin. Nam quis odio enim, in molestie libero. Vivamus cursus mi at nulla elementum sollicitudin.

$$E = mc^2 \tag{1}$$

Nam quis odio enim, in molestie libero. Vivamus cursus mi at nulla elementum sollicitudin. Nam quis odio enim, in molestie libero. Vivamus cursus mi at nulla elementum sollicitudin.

$$\cos^3\theta = \frac{1}{4}\cos\theta + \frac{3}{4}\cos 3\theta \tag{2}$$

Nam quis odio enim, in molestie libero. Vivamus cursus mi at nulla elementum sollicitudin. Nam quis odio enim, in molestie libero. Vivamus cursus mi at nulla elementum sollicitudin.

$$\kappa = \frac{\xi}{E_{\max}} \tag{3}$$

Nam quis odio enim, in molestie libero. Vivamus cursus mi at nulla elementum sollicitudin. Nam quis odio enim, in molestie libero. Vivamus cursus mi at nulla elementum sollicitudin.

Results

Placeholder Image

Figure 2: Figure caption

Nunc tempus venenatis facilisis. Curabitur suscipit consequat eros non porttitor. Sed a massa dolor, id ornare enim:

Nunc tempus venenatis facilisis. Curabitur suscipit consequat eros non porttitor. Sed a massa dolor, id ornare enim:

Nunc tempus venenatis facilisis. Curabitur suscipit consequat eros non porttitor. Sed a massa dolor, id ornare enim:

Nunc tempus venenatis facilisis. Curabitur suscipit consequat eros non porttitor. Sed a massa dolor, id ornare enim:

Treatments	Res. 1	Res. 2
Treatment 1	0.0003262	0.562
Treatment 2	0.0015681	0.910
Treatment 3	0.0009271	0.296

Table 1: Table caption

Conclusion

Nunc tempus venenatis facilisis. **Curabitur suscipit** consequat eros non porttitor. Sed a massa dolor, id ornare enim. Fusce quis massa dictum tortor **tincidunt mattis**. Donec quam est, lobortis quis pretium at, laoreet scelerisque lacus. Nam quis odio enim, in molestie libero. Vivamus cursus mi at *nulla elementum sollicitudin*.

Nunc tempus venenatis facilisis. Curabitur suscipit consequat eros non porttitor. Sed a massa dolor, id ornare enim.

Additional Information

Maecenas ultricies feugiat velit non mattis. Fusce tempus arcu id ligula varius dictum.

- Curabitur pellentesque dignissim
- Eu facilisis est tempus quis
- Duis porta consequat lorem

Maecenas ultricies feugiat velit non mattis. Fusce tempus arcu id ligula varius dictum.

- Curabitur pellentesque dignissim
- Eu facilisis est tempus quis
- Duis porta consequat lorem

References

[1] J. M. Smith and A. B. Jones. *Book Title*. Publisher, 7th edition, 2012.

[2] A. B. Jones and J. M. Smith. Article Title. *Journal title*, 13(52):123–456, March 2013.

Acknowledgements

Nam mollis tristique neque eu luctus. Suspendisse rutrum congue nisi sed convallis. Aenean id neque dolor. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Contact Information

- Web: http://www.my.edu/smithlab
- Email: john@smith.com
- Phone: +1 (000) 111 1111

PLACEHOLDER LOGO PLACEHOLDER LOGO

Figura 1.8: Documentos feitos com LaTeX: pôsteres.

Capítulo 2

Informações técnicas

Em geral, escrevemos código-fonte para LaTeX em arquivos com extensão ".tex". Para compilar um arquivo ".tex" e transformá-lo em um PDF, por exemplo, você precisa de um programa que faça isso.

No Windows, você pode instalar o MiKTeX. No Linux, você pode instalar o pacote `texlive-full`, disponível em praticamente todos os repositórios. No Mac OS, você pode instalar o MacTeX.

Escrever os códigos-fonte pode ser feito por qualquer editor de texto simples como Bloco de notas, Notepad++, Gedit, Vim ou Emacs. Nesse caso você precisa manualmente compilar seu arquivo por meio de um terminal. Mas você também pode usar editores próprios para LaTeX, que oferecem algumas facilidades como por exemplo um botão de compilação integrado. As opções mais famosas são TeXworks (que já vem junto na instalação do MiKTeX), TeXmaker, TeXstudio, TeXshop ou TeXpad. Você pode ver mais opções aqui.

Quando um documento ".tex" é compilado, vários arquivos serão criados com o mesmo nome porém com outra extensão (alguns deles apenas se o seu código-fonte pedir por isso, como é o caso do sumário, por exemplo). A seguir listamos alguns desses arquivos. O arquivo ".dvi" (*device independent*) está pronto para ser visualizado e pode ser transformado em PDF. O arquivo ".aux" contém as informações de referência cruzada (por exemplo, quando eu cito a Figura 1.6 ou o Capítulo 3, estou fazendo referências). O arquivo ".toc" contém os títulos das seções e capítulos (que geraram, por exemplo, o Sumário da página 3). O arquivo ".lof" contém os títulos das figuras (que geraram a Lista de Figuras da página 1). O arquivo ".lot" contém os títulos das tabelas (para gerar uma lista de tabelas). O arquivo ".idx" contém o conteúdo do índice remissivo (veja as últimas páginas). O arquivo ".log" gera um relatório da compilação feita, contendo erros e avisos. O arquivo ".bbl" é escrito pelo BibTeX (serve para gerar as referências bibliográficas).

Outra solução mais simples é utilizar ferramentas online. Esse é o caso do minicurso (meu objetivo não é que você aprenda a mexer em alguma ferramenta de edição, e sim escrever código LaTeX). O que eu recomendo é o Overleaf, onde você inclusive pode ter outros colaboradores escrevendo o mesmo documento. Uma característica bem legal do Overleaf é a quantidade enorme de *templates* prontos que ele oferece. Um template é um modelo de documento semipronto que já contém vários comandos de formatação, bastando inserir o conteúdo. Os exemplos que eu mostrei no final do Capítulo 1 foram retirados do Overleaf.

Capítulo 3

Criando seu primeiro documento

No decorrer dessa apostila, teremos vários exemplos como esse a seguir, onde acima da linha pontilhada mostrarei como deve é o código-fonte e abaixo da linha mostrarei como é a saída esperada após a compilação.

O exemplo a seguir é o documento mais simples possível que pode ser criado.

```
\documentclass{article}

\begin{document}
    Seu primeiro texto em \LaTeX{}!
\end{document}
```

Seu primeiro texto em LaTeX!

Códigos em LaTeX são bem estruturados. Eles sempre devem começar com o comando

```
\documentclass[opções]{estilo}
```

O `documentclass` é um comando que indica qual é o tipo de documento que você vai criar. No exemplo anterior, o estilo do documento era um artigo (*article*), porém existem outras possibilidades[1]:

`article`	artigos em revistas/conferências, documentações, convites, relatórios curtos, ...
`book`	livros
`letter`	cartas
`report`	relatórios longos, livros pequenos, teses, dissertações, ...
`beamer`	apresentações

Quanto às opções desse comando, elas especificam algum comportamento do estilo escolhido, como tamanho da fonte ou do papel. Você pode usar várias opções nesse comando e para isso basta separá-las por vírgula. Correndo o risco de ser redundante, as opções são opcionais: se você não detalhar

[1] Na verdade, é possível criar seu próprio estilo de documento. Em geral congressos, revistas ou universidades fornecem seus próprios estilos. Veremos mais sobre isso em futuros capítulos.

nenhuma opção (como era o caso do nosso primeiro exemplo básico), existem valores padrão para cada uma delas. As opções possíveis são as seguintes:

`10pt`, `11pt`, `12pt`	indica o tamanho da fonte no documento todo (`10pt` é o padrão)
`a4paper`, `letterpaper`, `executivepaper`, ...	define tamanho do papel (`letterpaper` é o padrão)
`twocolumn`	o texto é disposto em duas colunas (padrão é uma)
`titlepage`, `notitlepage`	indica se uma nova página deve ser criada depois do título ou não
`twoside`, `oneside`	indica se a paginação é dupla ou simples (não relacionado com a impressora)
`landscape`	indica que o documento deve ter as páginas em formato paisagem
`openright`, `openany`	indica se os capítulos devem começar apenas em páginas ímpares ou na próxima página disponível (a classe `article` não permite capítulos)
`fleqn`	mostra as fórmulas alinhadas à esquerda (padrão é centro)
`leqno`	coloca a numeração das fórmulas à esquerda (padrão é à direita)

O exemplo a seguir incrementa o anterior, informando que o tamanho do papel será A4, a fonte terá tamanho 12pt e que o documento vai ter uma página separada para o título.

```
\documentclass[a4paper, 12pt, titlepage]{article}

\begin{document}
    Seu primeiro texto em \LaTeX{}!
\end{document}
```

Seu primeiro texto em LaTeX!

As opções e mesmo o estilo de um documento podem ser modificados *a qualquer momento*, mesmo depois que o texto todo tiver sido escrito. Começamos desde já a observar a flexibilidade permitida pelo LaTeX.

A segunda parte importante que todo documento latex deve ter é o ambiente

```
\begin{document}
...
\end{document}
```

dentro do qual deve aparecer o seu texto. Chamamos a região entre o `\begin{document}` e o `\end{document}` de *corpo do documento*. Qualquer coisa que aparecer depois de `\end{document}` será completamente ignorada pelo compilador.

O documento simples que criamos acima é realmente o mais simples possível. Acontece que escrito dessa forma um código LaTeX não fornece muito mais oportunidades para elaborar o documento.

Por exemplo, se trocarmos o texto do documento simples (que atualmente é "Seu primeiro texto em LaTeX!") por "Este é o seu primeiro texto em LaTeX!". Você verá que o texto no documento gerado é

Este o seu primeiro texto em LaTeX!

isto é, o caractere com acento não aparece.

Em geral, quando escrevemos documentos em línguas que não Inglês, o LaTeX precisa ser configurado apropriadamente para entender regras de separação de sílabas da linguagem nova, entender caracteres diferentes e gerar corretamente textos que são gerados automaticamente.

A parte do arquivo entre `\documentclass{}` e `\begin{document}` é chamada de **preâmbulo**. É no preâmbulo do documento que nós vamos indicar essas novas regras ao LaTeX. Toda vez que precisamos de uma configuração nova que não é suportada pelo LaTeX básico, devemos indicar no preâmbulo o uso de um novo *pacote*. Um pacote nada mais é do que um arquivo que contém vários comandos complicados e que nos fornece comandos simples que permitem fazer documentos mais elaborados. Alguns pacotes já vêm instalados por padrão, mas muitos outros podem ser encontrados separadamente. Novos pacotes são criados pelos usuários de LaTeX a todo momento, inclusive. Não é escopo dessa apostila, mas você mesmo pode criar um pacote eventualmente!

Pacotes são inseridos no preâmbulo usando o comando

```
\usepackage[opções]{pacote}
```

Os principais pacotes que existem fornecem comandos para inserir figuras, para colorir o texto, para traduzir textos que são gerados automaticamente no documento, para permitir acentuação de caracteres. Nós veremos cada um deles e alguns outros conforme eles forem sendo necessários.

Voltando ao foco: para gerar os caracteres com acentuação que eu mencionei acima, precisamos do pacote `inputenc` com a opção `utf8`[2]. Outro pacote importante que vai traduzir textos automáticos e cuidar da separação de sílabas é o `babel` com opção `brazil`.

```
\documentclass[a4paper, 12pt, titlepage]{article}
\usepackage[utf8]{inputenc}
\usepackage[brazil]{babel}

\begin{document}
    Esse é seu primeiro texto em \LaTeX{}!
\end{document}
```

Esse é seu primeiro texto em LaTeX!

De modo geral, comandos colocados no preâmbulo de um documento servem para determinar algum comportamento global, que vai afetar o documento todo. Alguns dos comandos que são colocados no preâmbulo também podem ser colocados no corpo do documento, mas nesse caso eles irão determinar um comportamento local apenas (comandos `usepackage` devem sempre vir no preâmbulo).

[2] Se você ainda estiver tendo problemas, talvez sua opção deva ser `latin1`.

Capítulo 4

Escrevendo o texto

Quando você insere um texto simples em LaTeX e vê o documento final, vai perceber que o número de espaços inseridos entre as palavras no código fonte não faz diferença. Além disso, um parágrafo novo só é criado quando você deixa pelo menos uma linha em branco entre dois blocos de texto (novamente, mais de uma linha em branco não vai fazer diferença).

```
Esse    é  um   texto simples   com  vários   espaços,
inclusive   com quebra    de  linha   depois de uma vírgula.
Esse não é   um parágrafo novo, apenas uma frase nova
no primeiro parágrafo.

Esse é um parágrafo novo.




Esse também, e o espaço entre ele e o anterior é o mesmo que entre o anterior e
o primeiro.
```

Esse é um texto simples com vários espaços, inclusive com quebra de linha depois de uma vírgula. Esse não é um parágrafo novo, apenas uma frase nova no primeiro parágrafo.
Esse é um parágrafo novo.
Esse também, e o espaço entre ele e o anterior é o mesmo que entre o anterior e o primeiro.

Isso acontece porque o LaTeX se preocupa em deixar o seu texto visualmente correto e justificado com as margens e os espaços necessários para isso não são controlados por você. Se você julgar necessário uma quebra de linha, pode fazer isso explicitamente usando o comando \\.

```
Esse    é  um   texto simples   com  vários   espaços.
Esse não é   um parágrafo novo, apenas uma frase nova no primeiro parágrafo.\\
Sem o \\ essa frase deveria continuar na mesma linha da anterior.
% Esse é um comentário, que não aparece no documento final
```

Esse é um texto simples com vários espaços. Esse não é um parágrafo novo, apenas uma frase nova no primeiro parágrafo.
Sem o \\ essa frase deveria continuar na mesma linha da anterior.

Apesar de eu ter dito que você pode escrever qualquer texto simples sem formatação em LaTeX, existem alguns caracteres que são *reservados* por terem um significado especial. São os caracteres

```
\  #  $  %  ^  &  _  {  }  ~
```

O caractere \, por exemplo, indica ao LaTeX que um comando está começando. Os caracteres { e } também são usados por comandos. O caractere % indica o início de um comentário, e o texto que vem após ele e vai até o final de uma linha é completamente ignorado no documento final. O caractere ~ insere um pequeno espaço onde for inserido. Os outros caracteres farão sentido para você nos próximos capítulos. Assim, se você quiser escrevê-los literalmente no seu texto deve usar os comandos

```
\textbackslash  \#  \$  \%  \^{}  \&  \_  \{  \}  \~{}
```

Note que como já vimos, \\ indica uma quebra de linha forçada e por isso o caractere \ tem um comando especial diferente dos outros.

Outro caractere que não funciona tão bem no LaTeX são as aspas. Para fazer citações entre aspas você deve colocar seu texto entre `` (dois sinais de crase sem espaço entre eles) e '' (duas aspas simples sem espaço entre elas). Veja o exemplo.

```
Texto com o caractere aspas: "não fica bom".

Texto feito corretamente: ``agora sim''.
```

Texto com o caractere aspas: "não fica bom".
Texto feito corretamente: "agora sim".

Uma outra formação de caracteres que fica melhor com comandos são os três pontos. O comando `\ldots` deve ser utilizado, gerando "...", ao invés de três pontos seguidos, que geram "...".

Traços em LaTeX podem ser feitos de quatro formas diferentes, cada um com um significado. Apenas um traço "`-`" é usado para separação de sílabas e palavras hifenizadas. Dois traços "`--`" criam um traço maior "–" e são usados normalmente para mencionar intervalos ("veja as páginas 4–10" ao invés de "4-10"). Três traços "`---`" criam um traço ainda maior "—" que são usados para desvios em frases ("temos um assunto — paramos para explicar algo — e voltamos ao assunto"). E por fim existe o sinal de menos − (feito no ambiente matemático) que será tratado melhor no Capítulo 10.

Também é possível inserir no texto outros símbolos diferentes, que não existem em uma tecla do teclado, por exemplo. Alguns dos mais simples que não precisam de nenhum pacote para funcionar são ¶ (comando `\P`), © (comando `\copyright`), £ (comando `\pounds`), † (comando `\dag`), æ (comando `\ae`). O pacote `textcomp` fornece ainda € (comando `\texteuro`), ° (comando `\textdegree`) e ℃ (comando `\textcelsius`). Milhares são ainda os símbolos que podem ser inseridos no modo matemático, o que veremos melhor no Capítulo 10. Mas ainda existem vários outros para o modo texto (veja esse arquivo).

Pode ser que seja útil para você gerar um documento que tenha um texto que apareça exatamente da forma como ele está sendo escrito. Para criar os exemplos dessa apostila, por exemplo, eu precisei muito disso (no primeiro exemplo desse capítulo eu precisei mostrar os espaçamentos que são ignorados pelo LaTeX). O ambiente `verbatim` permite esse tipo de coisa.

```
\begin{verbatim}
    Esse texto    sai exatamente    como
    escrito por estar
    dentro do ambiente      verbatim.
    Qualquer caractere é válido aqui: \ # $ % & _ ^ ~
\end{verbatim}
```

```
    Esse texto    sai exatamente    como
    escrito por estar
    dentro do ambiente      verbatim.
    Qualquer caractere é válido aqui: \ # $ % & _ ^ ~
```

Um outro comando simples que já posso mencionar nessa altura da apostila serve para inserir notas de rodapé no texto. O comando é o `\footnote{texto}` que, onde inserido, cria uma nota de rodapé já devidamente enumerada. Vou inserir aqui[1] uma nota de rodapé. O código-fonte dessa última frase encontra-se a seguir.

```
Vou inserir aqui\footnote{Com qualquer informação.} uma nota de rodapé.
```

Faça você mesmo

Crie um documento simples (o mais simples possível) com o texto de introdução sobre a vida de Ramanujan que está disponível aqui (apenas a parte da introdução). Corrija tudo que for necessário para que um documento PDF seja gerado, considerando o que foi ensinado nesse capítulo.

[1] Com qualquer informação.

Capítulo 5

Formatando o texto

Nesse capítulo veremos quatro maneiras diferentes de fazer o que eu estou chamando de formatação de texto. Digo "o que eu estou chamando" porque talvez a última delas não faça tanto sentido assim. As três primeiras são sobre como modificar a forma do texto (negrito, itálico, fonte diferente, tamanho de fonte), seu espaçamento e seu alinhamento. A quarta é sobre como inserir um cabeçalho simples no documento.

5.1 Estilo de fonte

As três formas mais básicas de formatar um texto é inserindo termos em negrito, itálico e sublinhado. Os comandos LaTeX que fazer isso são `\textbf{texto}`, `\textit{texto}` e `\underline{texto}`, respectivamente. Eles podem ser combinados entre si.

```
Esse termo \textbf{em negrito}, este \textit{em itálico}, este
\textbf{\textit{em ambos}}, este \underline{sublinhado}, este
\underline{\textbf{\textit{combinando vários}}}.
```

Esse termo **em negrito**, este *em itálico*, este ***em ambos***, este <u>sublinhado</u>, este <u>***combinando vários***</u>.

O LaTeX fornece ainda o comando `\emph{texto}`, que dá ênfase em um texto. Em condições normais, ele vai se comportar como o `textit`, mas eles não são a mesma coisa. Veja o exemplo a seguir.

```
Esse texto está \emph{com ênfase}.
Agora veja o que acontece \textit{com esse texto em itálico \emph{quando
combinado} com a ênfase}.
```

Esse texto está *com ênfase*. Agora veja o que acontece *com esse texto em itálico* quando combinado *com a ênfase*.

Outras formas de destacar termos pode ser modificando a família da fonte. Você deve conhecer as fontes Times New Roman, Courier New, Helvetica ou Comic Sans. Mas em geral podemos classificá-las em três famílias: serif, sans serif e monoespaçadas. Os comandos LaTeX para transformar o texto em cada uma dessas três famílias são `\textrm{texto}`, `\textsf{texto}` e `\texttt{texto}`.

```
Texto \textrm{com fonte sans}, \textsf{com fonte sans serif} e \texttt{com fonte
monoespaçada}.
```

Texto com fonte sans, com fonte sans serif e `com fonte monoespaçada`.

As famílias em si são definidas pelos comandos `\rmfamily`, `\sffamily` e `\ttfamily`. Se algum desses comandos for colocado no meio do corpo do documento, a fonte do restante do documento será modificada para a respectiva família Se você precisar modificar o estilo do documento todo, basta colocar o comando `\renewcommand{\familydefault}{familia}` no preâmbulo, onde `familia` pode ser um dos três comandos que eu mencionei.

Uma outra possibilidade de formatação de texto é dada pelo comando `\textsc{texto}`, que gera TEXTOS DO TIPO SMALL CAPITALS.

Uma terceira forma de modificar a formatação do texto é trocando o tamanho das fontes. Veja as possibilidades no exemplo a seguir.

```
{\tiny texto minúsculo} \\
{\scriptsize texto muito pequeno} \\
{\footnotesize texto pequeno} \\
{\small texto menor que o normal} \\
{\normalsize texto normal} \\
{\large texto maior que o normal} \\
{\Large texto ainda maior} \\
{\LARGE texto grande} \\
{\huge texto muito grande} \\
{\Huge o maior texto possível}
```

texto minúsculo

texto muito pequeno

texto pequeno

texto menor que o normal

texto normal

texto maior que o normal

texto ainda maior

texto grande

texto muito grande

o maior texto possível

5.2 Espaçamentos

Para modificar o espaçamento do texto, vamos precisar do pacote `setspace`. Assim, adicione ao preâmbulo do seu documento o comando

```
\usepackage{setspace}
```

O espaçamento entre linhas pode ser modificado com os comandos `\singlespacing`, `\onehalfspacing`, `\doublespacing` ou `\linespread{fator}`, que indicam, respectivamente, espaçamento simples, um e meio, duplo ou então algum fator específico. Quando colocados no preâmbulo, modificam o espaçamento entre linhas do documento todo. Eles também podem ser utilizados no formato de ambientes, como por exemplo `\begin{doublespace}` ... `\end{doublespace}`.

```
\documentclass[a4paper, 12pt, titlepage]{article}
\usepackage[utf8]{inputenc}
\usepackage{setspace}

\begin{document}
    Esse parágrafo \\
    tem espaçamento \\
    simples.
\end{document}

\begin{verbatim}
\documentclass[a4paper, 12pt, titlepage]{article}
\usepackage[utf8]{inputenc}
\usepackage{setspace}

\onehalfspacing

\begin{document}
    Esse parágrafo \\
    tem espaçamento \\
    de um e meio.
\end{document}
```

Esse parágrafo
tem espaçamento
simples.

Esse parágrafo
tem espaçamento
de um e meio.

O espaçamento entre parágrafos pode ser modificado com o comando

```
\setlength{\parskip}{valor}
```

onde `valor` deve indicar um valor e sua medida, por exemplo 1cm, 10pt, 4em. Esse comando também deve ser colocado no preâmbulo para modificar o espaçamento entre parágrafos do documento inteiro.

Podemos também modificar o tamanho da indentação dos parágrafos (espaço entre a margem esquerda e a primeira linha do parágrafo). Para isso podemos usar o comando

```
\setlength{\parindent}{valor}
```

Os comandos `\noindent` e `\indent` podem ser utilizados no corpo do documento, logo antes da primeira linha do parágrafo, para evitar ou forçar a indentação do mesmo, respectivamente.

Existem quatro ambientes que podemos usar para modificar o posicionamento do texto. São eles o `center`, o `flushleft`, o `flushright` e o `minipage`. Os três primeiros transformam a justificação do texto que estiver dentro deles para centralizado, alinhado à esquerda e alinhado à direita, respectivamente. O último serve para criar colunas. Veja os exemplos a seguir.

```
Esse parágrafo está alinhado normalmente.
\begin{center}
    Este está centralizado.
\end{center}
\begin{flushleft}
    Este está justificado à esquerda.
\end{flushleft}
\begin{flushright}
    Este está justificado à direita
\end{flushright}

\begin{minipage}{0.3\textwidth}
    Essa coluna tem largura igual à 30\% da largura do texto (entre as margens).
\end{minipage}
~~
\begin{minipage}{0.2\textwidth}
    Essa coluna tem largura 20\% do texto.
\end{minipage}
```

Esse parágrafo está alinhado normalmente.

Este está centralizado.

Este está justificado à esquerda.

Este está justificado à direita.

Essa coluna tem largura igual à 30% da largura do texto (entre as margens).

Essa coluna tem largura 20% do texto.

5.3 Criando um título simples

O LaTeX fornece alguns comandos básicos para criação de um título simples no documento. Os comandos são `\title{título do texto}`, `\author{nome do autor}` e `\date{data do documento}`. Você deve colocá-los no preâmbulo do seu documento devidamente preenchidos. Para gerar o cabeçalho, pode utilizar o comando `\maketitle`, que deve ser colocado no corpo do documento, no local onde deseja-se criar o cabeçalho.

```
\documentclass[a4paper, 12pt, titlepage]{article}
\usepackage[utf8]{inputenc}
\title{Exemplo de título simples}
\author{Carla Negri Lintzmayer}
\date{\today}

\begin{document}
    \maketitle

    Seu documento começa aqui.
\end{document}
```

Exemplo de título simples

Carla Negri Lintzmayer

19 de junho de 2018

Seu documento começa aqui.

Uma outra característica importante em alguns documentos científicos é a existência de um *resumo*. Para isso o LaTeX disponibiliza o ambiente `abstract`.

```latex
\documentclass[a4paper, 12pt, titlepage]{article}
\usepackage[utf8]{inputenc}

\begin{document}
    \begin{abstract}
        O resumo do documento vai aqui.
    \end{abstract}

    Seu documento começa aqui.
\end{document}
```

Resumo

O resumo do documento vai aqui.

Seu documento começa aqui.

Faça você mesmo

Vá ao nosso documento sobre o Ramanujan, crie um título simples, o resumo e faça todas as formatações necessárias aprendidas nesse capítulo (veja o documento final). Em seguida, crie um cabeçalho com mais detalhes, como o existente no documento final.

Capítulo 6

Criando listas

Listas são formas de organizar a informação de uma forma clara. Existem três ambientes para criar listas em LaTeX: `itemize`, `enumerate` e `description`.

O ambiente `itemize` serve para criar listas simples, não ordenadas. Seu formato básico é mostrado a seguir.

```
Uma lista não ordenada:
\begin{itemize}
    \item Um item
    \item Outro item
    \item Cada novo item cria um bullet novo!
\end{itemize}
```

Uma lista não ordenada:

- Um item
- Outro item
- Cada novo item cria um bullet novo!

O ambiente `enumerate` serve para criar listas ordenadas. Seu formato básico é mostrado a seguir.

```
Uma lista ordenada:
\begin{enumerate}
    \item Primeiro item
    \item Segundo item
    \item Cada novo item incrementa o valor da entrada anterior!
\end{enumerate}
```

Uma lista ordenada:

1. Primeiro item
2. Segundo item
3. Cada novo item incrementa o valor da entrada anterior!

Por fim, o ambiente `description` serve para criar listas descritivas[1], que em geral também não são ordenadas. Seu formato básico é mostrado a seguir.

```
Uma lista descritiva:
\begin{description}
    \item[Valor 1] Descrição do valor 1
    \item[Valor 2] Descrição do valor 2
    \item[Valor] Cada novo item indica um novo valor a ser descrito
\end{description}
```

Uma lista descritiva:

Valor 1 Descrição do valor 1

Valor 2 Descrição do valor 2

Valor Cada novo item indica um novo valor a ser descrito

O texto que segue o `\item` em cada ambiente anterior pode ser de qualquer tamanho e conter quantos parágrafos você quiser.

Você também pode inserir uma lista dentro de outra lista, seja do mesmo tipo ou de tipos variados, mas até uma profundidade de 4 níveis. Essa construção é chamada de "aninhamento de listas".

[1] Desculpe, não consegui pensar em uma explicação melhor. De qualquer forma, algumas coisas ficam claras apenas com exemplos.

```
Listas aninhadas:
\begin{enumerate}
    \item Existem outras informações sobre esse item que eu quero mostrar de
    forma organizada:
    \begin{enumerate}
        \item Por exemplo, essa
        \item E essa
    \end{enumerate}
    \item Esse item só precisa de uma descrição maior, que certamente vai ocupar
    mais de um parágrafo.

    Como por exemplo esse novo parágrafo.
    \item E esse item também precisa de outras informações organizadas:
    \begin{itemize}
        \item Como essa, que por sua vez também precisa de mais informações:
        \begin{itemize}
            \item Tipo essa
        \end{itemize}
    \end{itemize}
\end{enumerate}
```

Listas aninhadas:

1. Existem outras informações sobre esse item que eu quero mostrar de forma organizada:
   (a) Por exemplo, essa
   (b) E essa
2. Esse item só precisa de uma descrição maior, que certamente vai ocupar mais de um parágrafo.

   Como por exemplo esse novo parágrafo.
3. E esse item também precisa de outras informações organizadas:
   - Como essa, que por sua vez também precisa de mais informações:
     - Tipo essa

Por padrão, no ambiente `itemize` o primeiro nível terá o símbolo $\bullet$, o segundo terá o símbolo $-$, o terceiro terá o símbolo $*$ e o quarto terá o símbolo $\cdot$. É possível modificá-los com o comando

```
\renewcommand{nível_do_rótulo}{novo_símbolo}
```

onde `nível_do_rótulo` pode ser `\labelitemi`, `\labelitemii`, `\labelitemiii` ou `\labelitemiv` para os níveis 1, 2, 3 ou 4, respectivamente, e `novo_símbolo` pode ser qualquer símbolo desejado, inclusive do ambiente matemático. Veja o exemplo a seguir.

```latex
\begin{minipage}{0.45\textwidth}
Símbolos padrão:
\begin{itemize}
    \item Nível 1
    \begin{itemize}
        \item Nível 2
        \begin{itemize}
            \item Nível 3
            \begin{itemize}
                \item Nível 4
            \end{itemize}
        \end{itemize}
    \end{itemize}
\end{itemize}
\end{minipage}
\begin{minipage}{0.45\textwidth}
Símbolos modificados:
\renewcommand{\labelitemi}{$\star$}
\renewcommand{\labelitemii}{\dag}
\renewcommand{\labelitemiii}{$\clubsuit$}
\renewcommand{\labelitemiv}{$\rightsquigarrow$}
\begin{itemize}
    \item Nível 1
    \begin{itemize}
        \item Nível 2
        \begin{itemize}
            \item Nível 3
            \begin{itemize}
                \item Nível 4
            \end{itemize}
        \end{itemize}
    \end{itemize}
\end{itemize}
\end{minipage}
```

Símbolos padrão:

- Nível 1
  - Nível 2
    - Nível 3
      - Nível 4

Símbolos modificados:

$\star$ Nível 1

† Nível 2

$\clubsuit$ Nível 3

$\rightsquigarrow$ Nível 4

De forma similar, por padrão o ambiente `enumerate` terá no primeiro nível a numeração arábica (1, 2, ...), no segundo nível letras minúsculas, no terceiro nível numeração por algarismos romanos minúsculos (i, ii, ...) e no quarto nível letras maiúsculas. É possível modificá-los com o comando `renewcommand` similar ao mostrado acima, mas onde `nível_do_rótulo` pode ser `\theenumi`, `\theenumii`, `\theenumiii` ou `\theenumiv` para os nívels 1, 2, 3, ou 4, respectivamente. No caso de `novo_símbolo`, apenas cinco possibilidades estão disponíveis: `\alph` (letras minúsculas), `\Alph` (letras maiúsculas), `\arabic` (números arábicos 1, 2, ...), `\roman` (algarismos romanos minúsculos i, ii, iii, ...) e `\Roman` (algarismos romanos maiúsculos I, II, ...). Cada um deve ser seguido de {`enumX`}, onde `X` pode ser i, ii, iii ou iv (um para cada nível da lista). Veja o exemplo a seguir.

```latex
\begin{minipage}{0.45\textwidth}
Numeração padrão:
\begin{enumerate}
    \item Nível 1
    \begin{enumerate}
        \item Nível 2
        \begin{enumerate}
            \item Nível 3
            \begin{enumerate}
                \item Nível 4
            \end{enumerate}
        \end{enumerate}
    \end{enumerate}
\end{enumerate}
\end{minipage}
\begin{minipage}{0.45\textwidth}
Numeração modificada:
\renewcommand{\theenumi}{\Roman{enumi}}
\renewcommand{\theenumii}{\roman{enumii}}
\renewcommand{\theenumiii}{\Alph{enumiii}}
\renewcommand{\theenumiv}{\alph{enumiv}}
\begin{enumerate}
    \item Nível 1
    \begin{enumerate}
        \item Nível 2
        \begin{enumerate}
            \item Nível 3
            \begin{enumerate}
                \item Nível 4
            \end{enumerate}
        \end{enumerate}
    \end{enumerate}
\end{enumerate}
\end{minipage}
```

Numeração padrão:

1. Nível 1
   (a) Nível 2
      i. Nível 3
         A. Nível 4

Numeração modificada:

I. Nível 1
   (i) Nível 2
      A. Nível 3
         a. Nível 4

Esses comandos de alteração de símbolo/numeração podem ser colocados no preâmbulo para modificar a formatação da lista no documento todo ou então logo antes do ambiente, para modificar apenas aquela ocorrência, como foi o caso dos exemplos.

Ainda é possível alterar a forma como os rótulos aparecem. Por exemplo, alguns aparecem com "( )" ao redor e outros aparecem com "." ao final. O pacote `enumerate` permite esse tipo de personalização[2].

Faça você mesmo

Vá ao nosso documento sobre o Ramanujan e escreva a parte sobre a vida dele, que contém alguns eventos importantes descritos em uma lista. Escreva também os itens pelos quais ele é conhecido, na seção de resultados notáveis.

[2]Enquanto não escrevo sobre ele aqui, você pode ver esse texto.

Capítulo 7

Estruturando o texto

Qualquer documento bem organizado está separado em capítulos e/ou seções. O LaTeX oferece meios de fazer isso de maneira simples, para que você não tenha que se preocupar com numeração de nada.

Existem 7 níveis de profundidade possíveis, mas nem todos estão disponíveis para todas as classes de documento (definidas pelo `\documentclass`). Os comandos que definem as divisões são

`\part{Título}`	define uma parte (disponível para os tipos `book` e `report` apenas)
`\chapter{Título}`	define um capítulo (disponível para os tipos `book` e `report` apenas)
`\section{Título}`	define uma seção
`\subsection{Título}`	define uma subseção
`\paragraph{Título}`	define um parágrafo
`\subparagraph{Título}`	define um subparágrafo

Quando um comando desses indicados acima aparece, o LaTeX o substitui pelo conteúdo de `Título` devidamente formatado para aquele nível de divisão. O controle de onde um capítulo/seção/subseção começa ou acaba é, portanto, seu: é onde você coloca o respectivo comando.

```
\section{Essa é uma seção}
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.

\subsection{Essa é uma subseção}
Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id,
vulputate a, magna.

\section{Outra seção}
Etiam  lobortisfacilisis sem.  Nullam nec mi et neque pharetra sollicitudin.
```

7.1 Essa é uma seção

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

7.1.1 Essa é uma subseção

Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

7.2 Outra seção

Etiam lobortisfacilisis sem. Nullam nec mi et neque pharetra sollicitudin.

Se por algum motivo você não quiser que as partes do seu documento apareçam enumeradas, basta utilizar a versão "estrela" deles: `\part*{Título}`, `\chapter*{Título}`, `\section*{Título}` e `\subsection*{Título}`.

```
\section{Essa é uma seção}
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

\subsection{Essa é uma subseção}
Curabitur dictum gravida mauris.

\subsection*{Outra subseção, mas sem número}
Etiam  lobortisfacilisis sem.
```

7.3 Essa é uma seção

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

7.3.1 Essa é uma subseção

Curabitur dictum gravida mauris.

Outra subseção, mas sem número

Etiam lobortisfacilisis sem.

Uma vez que você estruturou seu documento com os comandos citados acima, você pode criar um sumário para o mesmo, como o que aparece na página 3. Para isso, basta colocar o comando

```
\tableofcontents
```

no local onde o sumário deve aparecer (logo após a capa do seu documento e antes do primeiro capítulo/seção do mesmo).

Capítulos/seções sem numeração (por exemplo, os gerados na versão com asterisco ou então no caso das referências, como veremos no Capítulo 12) não aparecem no sumário. É possível adicioná-los à lista mesmo assim, por meio do comando

```
\addcontentsline{lista}{tipo_da_divisão}{Título}
```

onde `lista` é o tipo da lista onde o item deve aparecer (no caso do sumário, o tipo é `toc`, de *table of contents*), `tipo_da_divisão` é o nível do item (*section*, *chapter*, etc.) e `Título` é o texto que deve aparecer na lista.

Pode ser também que o título do seu capítulo/seção/subseção seja grande demais para aparecer no sumário ou para aparecer no cabeçalho (caso o seu documento esteja gerando cabeçalho na página). Você pode sugerir um título mais curto nesse caso: o título original, longo, só vai aparecer quando o capítulo/seção/subseção de fato começarem. Para isso, você coloca o título curto entre "[ ]" logo antes do título longo: `\part[Título curto]{Título longo}`, `\chapter[Título curto]{Título longo}`, `\section[Título curto]{Título longo}` e `\subsection[Título curto]{Título longo}`.

Outra forma de organizar seu documento é utilizando arquivos separados, um para cada parte (capítulo/seção), por exemplo. Para isso, o LaTeX oferece os comandos `\input` e `\include`. Arquivos

inseridos por meio desses comandos não devem conter comandos que aparecem no preâmbulo e muito menos `\begin{document}` e `\end{document}`[1].

Faça você mesmo

Vá ao nosso documento sobre o Ramanujan e adicione as seções e subseções necessárias. Adicione também o sumário.

[1] Enquanto eu não escrevo melhor sobre isso, você pode ver mais detalhes aqui.

Capítulo 8

Inserindo figuras

Você pode inserir figuras externas no seu texto (digo externas porque é possível criar figuras com LaTeX, o que você vai ver nos capítulos finais). O LaTeX facilita muito a inserção desse tipo de "objeto flutuante", pois você não precisa se preocupar com o posicionamento dele no texto, a menos que isso seja extremamente importante para você.

Para inserir figuras precisamos do pacote `graphicx`. Sendo assim, você deve inserir no preâmbulo o comando

```
\usepackage{graphicx}
```

O comando que permite de fato a inserção da figura é

```
\includegraphics[opções]{caminho_da_figura}
```

onde `caminho_da_figura` contém o caminho do diretório no qual a figura se encontra juntamente com seu nome. Por exemplo, a figura pode estar no mesmo diretório do seu arquivo principal ".tex", de modo que basta dizer qual é o nome da figura. No exemplo abaixo, estou supondo que existe um diretório de nome `figs` no mesmo diretório onde está o arquivo ".tex", no qual se encontra a figura que quero adicionar ao texto.

```
\includegraphics{figs/logo-ufabc}
```

Note como a figura ficou enorme. Isso se dá porque você não especificou mais nada sobre ela, o que pode ser feito usando as opções do comando `\includegraphics`. As opções possíveis são

`width`	para alterar a largura da figura
`height`	para alterar a altura da figura
`angle`	para rotacionar a figura
`scale`	para mudar a escala da figura

As opções `width` e `height` aceitam valores específicos (como 2cm ou 50pt) ou então unidades relativas a elementos do documento (como `0.5\textwidth`, indicando 50% da largura do texto). Se apenas um desses dois for especificado, o outro vai ser escalado proporcionalmente (faz mais sentido especificar apenas um deles, visualmente falando). No exemplo abaixo, a caixa colorida tem largura `\textwidth` (pois ela vai da borda esquerda à borda direita da página) e que eu peço que a figura tenha 30% desse valor.

```
\includegraphics[width=0.3\textwidth]{figs/logo-ufabc}
```

A opção `scale` simplesmente aceita números (por exemplo, se `scale=2.5` é dado como opção, queremos que a imagem seja escalada para duas vezes e meia seu tamanho original). A opção `angle` aceita números também, que indicam o ângulo que a figura será rotacionada. Valores positivos indicam rotação no sentido anti-horário e valores negativos indicam rotação no sentido horário.

```
\includegraphics[width=0.3\textwidth, angle=-45]{figs/logo-ufabc}
```

Como você deve ter percebido no exemplo anterior, mais de uma opção pode ser fornecida, bastando separá-las por vírgulas.

Repare no exemplo a seguir como o comando `\includegraphics` age quando existe texto ao seu redor.

```
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.

\includegraphics[width=0.1\textwidth]{figs/logo-ufabc}

Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id,
vulputate a, magna.
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

UFABC

Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

Provavelmente essa não é a forma como você gostaria de inserir figuras, não é mesmo? Você quer destacá-las e inserir legendas nas mesmas. Para isso existe o ambiente `figure`:

```
\begin{figure}[opções]
    \centering
    \includegraphics[opções]{caminho_da_figura}
    \caption{Legenda.}
\end{figure}
```

onde o comando `\centering` indica que tudo o que estiver dentro daquele ambiente onde ele foi posicionado (no caso, no ambiente `figure`), deve ficar centralizado horizontalmente na página, e o comando `\caption{}` nos permite adicionar uma legenda à figura.

```
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.

\begin{figure}
    \centering
    \includegraphics[width=0.1\textwidth]{figs/logo-ufabc}
    \caption{Logotipo da UFABC.}
\end{figure}

Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id,
vulputate a, magna.
```

Figura 8.1: Logotipo da UFABC.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.
Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

Usar o ambiente `figure` faz com que o LaTeX posicione a figura da melhor maneira que se encaixe com o restante do documento. Isso pode não ser exatamente onde você inseriu o ambiente: veja no exemplo anterior que o ambiente estava entre os dois parágrafos e a figura foi posicionada acima de ambos (em geral, o LaTeX tenta posicionar objetos flutuantes no topo das páginas).

Se você julgar necessário controlar o posicionamento da figura, você pode usar as opções disponíveis para o ambiente `figure`, que são

`h`	do inglês *here*, posiciona a figura o mais próximo possível de onde o comando foi colocado
`t`	do inglês *top*, posiciona no topo da página
`b`	do inglês *bottom*, posiciona no fim da página
`p`	do inglês *page*, posiciona em uma página nova, apenas para objetos flutuantes
`!`	combinado aos anteriores, obriga o LaTeX a ignorar o que ele acha ser um bom posicionamento

No caso desse ambiente, usar mais de uma opção é possível e não há necessidade de utilizar vírgulas para separá-las.

```
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.

\begin{figure}[h!]
    \centering
    \includegraphics[width=0.1\textwidth]{figs/logo-ufabc}
    \caption{Logotipo da UFABC.}
\end{figure}

Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id,
vulputate a, magna.
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

Figura 8.2: Logotipo da UFABC.

Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

Veja que o LaTeX cuida da numeração das figuras que são colocadas no texto sem você pedir por isso. Assim, você pode referenciar as figuras escrevendo frases do tipo "veja a Figura 8.2".

Agora note que se você precisar trocar a ordem de duas figuras, basta trocar a ordem dos ambientes `figure` que as incluem. No entanto isso vai trocar a ordem da numeração das figuras. É por isso que nós **nunca** escrevemos textos do tipo "veja a Figura 8.2" escrevendo o valor 8.2 em si. Para isso, fazemos *referências cruzadas*, que serão explicitamente abordadas no Capítulo 11.

Existem ainda formas de criar subfiguras. Por exemplo, usando o pacote `subcaption` que fornece o ambiente `subfigure`[1].

Dependendo do documento que você está criando, pode ser que seja necessário criar uma lista de figuras (por exemplo, sua tese, dissertação, trabalho de graduação, um livro, um relatório grande), como a que aparece na página 1. Para isso, basta colocar o comando

```
\listoffigures
```

no local onde a lista deve aparecer (em geral, isso é logo antes do sumário).

[1] Enquanto não escrevo sobre ele aqui, você pode ver esse texto.

Faça você mesmo

Vá ao nosso documento sobre o Ramanujan e adicione a foto dele que aparece na seção sobre sua vida.

Capítulo 9

Inserindo tabelas

Outro tipo de "objeto flutuante" que podemos inserir no texto são as tabelas. Não há necessidade de usar nenhum pacote para isso.

O ambiente `tabular` é utilizado para criar linhas horizontais e verticais e criar de fato uma tabela

```
\begin{tabular}[opções]{especificação}
    conteúdo da tabela
\end{tabular}
```

Veremos a seguir detalhes das `opções`, da `especificação` e de como escrever o `conteúdo da tabela`. Primeiro veremos um exemplo simples.

```
\begin{tabular}{ccc}
 Aaa & Bb & C \\
 D & Eeee & Ff \\
\end{tabular}
```

Aaa	Bb	C
D	Eeee	Ff

Não usamos nenhuma opção (que, como era de se esperar, são opcionais) e no campo de especificação colocamos `ccc`, que indica que devem existir 3 colunas na tabela, cada uma com o conteúdo centralizado (daí a letra 'c'). A primeira linha tem o conteúdo "A" na primeira célula, o conteúdo "B" na segunda e o conteúdo "C" na terceira. O LaTeX sabe disso porque eles estão sendo separados pelo símbolo & (por isso ele é proibido no texto e, se quisermos usá-lo literalmente precisamos escrever \&). O fim de uma linha é indicado por \\ (por isso não dá para usar \\ quando queremos o caractere \ e precisamos usar o comando `\textbackslash`).

Agora que você sabe um pouquinho mais sobre a parte especificação, vamos detalhar o que ela pode conter. Em primeiro lugar, ela precisa indicar quantas colunas a tabela terá. Uma coluna vai

ser indicada por uma letra, que por sua vez representa o alinhamento daquela coluna. Os símbolos disponíveis para especificação das colunas das tabelas são

`c`	da palavra *center*, para centralizar
`l`	da palavra *left*, para justificar à esquerda
`r`	da palavra *right*, para justificar à direita
`p{valor}`	para indicar que a coluna vai ter largura "`valor`" com texto verticalmente alinhado ao topo
`|`	para indicar uma linha vertical

Veja o exemplo anterior novamente, com outras especificações agora:

```
\begin{tabular}{|c||l|r|}
 Aaa & Bb & C \\
 D & Eeee & Ff \\
\end{tabular}
```

Aaa	Bb	C
D	Eeee	Ff

E ainda é possível inserir linhas horizontais com o comando `\hline`. Ele deve sempre aparecer após o comando \\, que indica o fim de linha (significando que você deseja colocar uma barra horizontal abaixo da linha que acabou de escrever). Você pode inserir quantos comandos desse você quiser (caso queira criar várias linhas horizontais – por exemplo, para diferenciar o cabeçalho da tabela). Repare que a primeira linha horizontal da tabela é o primeiro comando dentro do ambiente.

```
\begin{tabular}{|c||l|r|}
 \hline
 \textbf{Col1} & \textbf{Col2} & \textbf{Col3} \\ \hline\hline
 Aaa & Bb & C \\ \hline
 D & Eeee & Ff \\ \hline
\end{tabular}
```

Col1	**Col2**	**Col3**
Aaa	Bb	C
D	Eeee	Ff

Existe também o comando `\cline{i-j}`, que serve para criar uma linha horizontal parcial, que vai da coluna i até a coluna j.

```
\begin{tabular}{|c||l|r|}
 \hline
 \textbf{Col1} & \textbf{Col2} & \textbf{Col3} \\ \hline\hline
 Aaa & Bb & C \\ \hline
 D & Eeee & Ff \\ \hline
\end{tabular}
```

Col1	**Col2**	**Col3**
Aaa	Bb	C
	b	c
D	Eeee	Ff

A opção de especificação `p{valor}` é normalmente usada quando o conteúdo da coluna é muito grande e ultrapassa a margem da página (o LaTeX por padrão não cuida disso).

```
\begin{tabular}{c|l}
    \hline
    col1 & col2 \\ \hline\hline
    Lorem & ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
    vestibulum ut, placerat ac, adipiscing vitae, felis. \\ \hline
    Curabitur & dictum gravida mauris. Nam arcu libero, nonummy eget,
    consectetuer id, vulputate a, magna. \\ \hline
\end{tabular}
\vspace{0.5cm}
\begin{tabular}{c|p{0.6\textwidth}}
    \hline
    col1 & col2 \\ \hline\hline
    Lorem & ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
    vestibulum ut, placerat ac, adipiscing vitae, felis. \\ \hline
    Curabitur & dictum gravida mauris. Nam arcu libero, nonummy eget,
    consectetuer id, vulputate a, magna. \\ \hline
\end{tabular}
```

col1	col2
Lorem	ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing
Curabitur	dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

col1	col2
Lorem	ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.
Curabitur	dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

É possível ainda gerar tabelas um pouco mais complexas, por exemplo se for necessário que apenas uma ou outra linhas combinem várias colunas em uma só. Ou então se for necessário que apenas uma ou outra colunas combinem várias linhas em uma só. Uma forma simples de unir várias colunas em uma só é usando o comando

```
\multicolumn{número de colunas a unir}{especificação da nova
coluna}{conteúdo da célula}
```

Enquanto eu não escrevo mais detalhes sobre isso aqui, você pode olhar aqui ou então aqui.

Agora repare no exemplo a seguir como o ambiente `tabular` age quando existe texto ao seu redor.

```
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.
\begin{tabular}{|c||l|r|}
 \hline
 \textbf{Col1} & \textbf{Col2} & \textbf{Col3} \\ \hline\hline
 Aaa & Bb & C \\ \hline
 D & Eeee & Ff \\ \hline
\end{tabular}
Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id,
vulputate a, magna.
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

Col1	**Col2**	**Col3**
Aaa	Bb	C
D	Eeee	Ff

Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

Se a sua intenção era realmente colocar uma tabela no meio do texto, você pode usar as opções do ambiente `tabular` para indicar o comportamento desse texto ao redor. As opções são

`c`	da palavra *center*, para que o texto fique no centro (o que é o padrão)
`b`	da palavra *bottom*, para que o texto fique para baixo
`t`	da palavra *top*, para que o texto fique para cima

```
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.
\begin{tabular}[b]{|c||l|r|}
 \hline
 \textbf{Col1} & \textbf{Col2} & \textbf{Col3} \\ \hline\hline
 Aaa & Bb & C \\ \hline
 D & Eeee & Ff \\ \hline
\end{tabular}
Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id,
vulputate a, magna.
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

Col1	**Col2**	**Col3**
Aaa	Bb	C
D	Eeee	Ff

Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

Normalmente, no entanto, queremos que as tabelas fiquem em destaque e tenham legendas[1]. Para isso existe o ambiente `table`:

```
\begin{table}[opções]
    \centering
    \caption{Legenda.}
    \begin{tabular}[opções]{especificação}
        conteúdo da tabela
    \end{tabular}
\end{table}
```

onde o comando `\centering` indica que tudo o que estiver dentro daquele ambiente onde ele foi posicionado (no caso, no ambiente `table`), deve ficar centralizado horizontalmente na página, e o comando `\caption{}` nos permite adicionar uma legenda à tabela (em geral, legendas de tabelas ficam acima das mesmas).

```
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.
\begin{table}
    \centering
    \caption{Exemplo de uma tabela em \LaTeX{}.}
    \begin{tabular}{|c||l|r|}
     \hline
     \textbf{Col1} & \textbf{Col2} & \textbf{Col3} \\ \hline\hline
     Aaa & Bb & C \\ \hline
     D & Eeee & Ff \\ \hline
    \end{tabular}
\end{table}
Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id,
vulputate a, magna.
```

Tabela 9.1: Exemplo de uma tabela em LaTeX.

Col1	**Col2**	**Col3**
Aaa	Bb	C
D	Eeee	Ff

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.
Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

Note que o ambiente `table` indica ao LaTeX que ele deve posicionar a tabela da melhor maneira que

[1] A partir daqui, muitas coisas vão ficar super parecidas com o que eu já falei no Capítulo 8, sobre figuras. Daqui a pouco voltamos a uma discussão específica para tabelas.

se encaixe com o restante do documento. Isso pode não ser exatamente onde você inseriu o ambiente: veja no exemplo anterior que o ambiente estava entre dois parágrafos e a tabela foi posicionada acima de ambos (em geral, o LaTeX tenta posicionar objetos flutuantes no topo das páginas).

Se você julgar necessário controlar o posicionamento da tabela, você pode usar as opções disponíveis para o ambiente `table`, que são

`h`	do inglês *here*, posiciona a tabela o mais próximo possível de onde o comando foi colocado
`t`	do inglês *top*, posiciona no topo da página
`b`	do inglês *bottom*, posiciona no fim da página
`p`	do inglês *page*, posiciona em uma página nova, apenas para objetos flutuantes
`!`	combinado aos anteriores, obriga o LaTeX a ignorar o que ele acha ser um bom posicionamento

```
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.
\begin{table}[h!]
    \centering
    \caption{Exemplo de uma tabela em \LaTeX{}.}
    \begin{tabular}{|c||l|r|}
     \hline
     \textbf{Col1} & \textbf{Col2} & \textbf{Col3} \\ \hline\hline
     Aaa & Bb & C \\ \hline
     D & Eeee & Ff \\ \hline
    \end{tabular}
\end{table}
Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id,
vulputate a, magna.
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

Tabela 9.2: Exemplo de uma tabela em LaTeX.

Col1	**Col2**	**Col3**
Aaa	Bb	C
D	Eeee	Ff

Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

Veja que o LaTeX cuida da numeração das tabelas que são colocadas no texto sem você pedir por isso. Assim, você pode referenciar as tabelas escrevendo frases do tipo "veja a Tabela 9.2".

Agora note que se você precisar trocar a ordem de duas tabelas, basta trocar a ordem dos ambientes `table` que as incluem. No entanto isso vai trocar a ordem da numeração das tabelas. É por isso que nós **nunca** escrevemos textos do tipo "veja a Tabela 9.2" escrevendo o valor 9.2 em si. Para isso, fazemos *referências cruzadas*, que serão explicitamente abordadas no Capítulo 11.

Vimos que o LaTeX não se preocupa se uma coluna é grande demais e ultrapassa a margem direita da página e que para arrumar isso basta utilizar o especificador `p{...}`. Mas caso você realmente necessite de uma tabela mais longa horizontalmente falando, é possível usar o ambiente `landscape` fornecido pelo pacote `lscape`, que rotaciona a tabela na página para mantê-la no formato paisagem.

Esse problema citado acima acontece porque, por padrão, o LaTeX mantém o ambiente `table` em uma única página, mesmo que isso signifique ultrapassar também a margem inferior dd mesma. Caso você realmente necessite de uma tabela mais longa verticalmente falando, é possível usar o ambiente `longtable`, fornecido pelo pacote `longtable`.

Dependendo do documento que você está criando, pode ser que seja necessário criar uma lista de tabelas (por exemplo, sua tese, dissertação, trabalho de graduação, um livro, um relatório grande). Para isso, basta colocar o comando

```
\listoftables
```

no local onde a lista deve aparecer (em geral, isso é logo antes do sumário, após a lista de figuras).

Faça você mesmo

Vá ao nosso documento sobre o Ramanujan e adicione a tabela da seção sobre o quadrado mágico. Adicione também a tabela na seção sobre cultura popular.

Para finalizar, ainda é possível personalizar altura e largura das linhas e colunas, a distância entre o texto e as bordas da tabela, e várias outras coisas. Não cabe a essa apostila ensinar tantos detalhes assim, porque isso começa a depender exatamente de como é a tabela específica que você quer fazer. Uma dica que eu deixo é para você procurar ferramentas nas quais você possa manualmente fazer a tabela (algo parecido com as planilhas eletrônicas) e que criem o código para LaTeX automaticamente para você.

Capítulo 10

Modo matemático

A forma como o LaTeX permite a inserção de notações matemáticas no texto é de fato o que o faz mais apropriado para documentos científicos.

Existem basicamente duas formas de mostrar uma expressão matemática: no meio do texto, o que chamamos de modo *inline*, e em destaque, o que chamamos de modo *display*.

Existem duas formas de delimitar o modo *inline*: com \$...\$ ou com \\(...\\). Tudo o que estiver dentro desses delimitadores será considerado no modo matemático. Veja exemplos a seguir.

```
Note a diferença entre as letras x, i ou n escritas normalmente e as letra $x$,
$i$ ou $n$ escritas em modo matemático.

Números negativos como -3 ficam bem melhores em modo matemático: $-3$.

O Teorema de Pitágoras diz que $x^2 + y^2 = z^2$, onde $x$ e $y$ são os catetos
e $z$ é a hipotenusa. Podemos então dizer que $z = \sqrt{x^2 + y^2}$.
```

Note a diferença entre as letras x, i ou n escritas normalmente e as letra x, i ou n escritas em modo matemático.
Números negativos como -3 ficam bem melhores em modo matemático: -3.
O Teorema de Pitágoras diz que $x^2 + y^2 = z^2$, onde x e y são os catetos e z é a hipotenusa. Podemos então dizer que $z = \sqrt{x^2 + y^2}$.

Existem três formas de delimitar o modo *display*: com \$\$...\$\$, com \\[...\\] ou então com o ambiente `\begin{equation}`...`\end{equation}`. O ambiente `equation`, em particular, enumera as equações. Veja exemplos a seguir.

```
O Teorema de Pitágoras diz que $$x^2 + y^2 = z^2,$$ onde $x$ e $y$ são os catetos
e $z$ é a hipotenusa.

Uma equação $a x^2 + b x + c = 0$ tem como solução
\begin{equation}
    x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
\end{equation}
```

O Teorema de Pitágoras diz que

$$x^2 + y^2 = z^2,$$

onde x e y são os catetos e z é a hipotenusa.
Uma equação $ax^2 + bx + c = 0$ tem como solução

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{10.1}$$

Veja que o LaTeX cuida da numeração das equações que são colocadas no ambiente `equation` sem você pedir por isso. Assim, você pode referenciar as equações escrevendo frases do tipo "veja a Equação (10.1)".

Agora note que se você precisar trocar a ordem de duas equações, basta trocar a ordem dos ambientes `equation` que as incluem. No entanto isso vai trocar a ordem da numeração das equações. É por isso que nós **nunca** escrevemos textos do tipo "veja a Equação (10.1)" escrevendo o valor (10.1) em si. Para isso, fazemos *referências cruzadas*, que serão explicitamente abordadas no Capítulo 11.

Note que os espaços gerados dentro das fórmulas, sejam elas *inline* ou *display* são ignorados pelo LaTeX. Os símbolos matemáticos $+$, $-$, $=$, $<$, $>$, $/$, $:$, $!$, $'$, $|$, $[$, $]$, $($, $)$ e letras podem ser digitados normalmente no teclado, mas outros símbolos necessitam de comandos. As chaves "{ }" servem para agrupar logicamente partes da fórmula e não são impressas. Quando você precisar incluir chaves em uma fórmula, deve usar \{ e \}. Existem dois documentos muito legais com símbolos matemáticos (esse tem vários comandos que são usados frequentemente enquanto esse é **muito** mais completo). O pacote `amsmath` certamente será utilizado pela maioria dos comandos matemáticos, então pode incluí-lo no seu preâmbulo.

O ambiente matemático do LaTeX é muito rico e poderoso e permite que muitas coisas sejam feitas. Na Tabela 10.1 eu apresento algumas, mas esse é aquele tipo de aprendizado que só se aperfeiçoa com a prática. Você também pode olhar outras referências, como essa, essa ou essa.

Se você precisar digitar um texto dentro de um ambiente matemático, vai perceber que o resultado pode não ser como você deseja. Para fazer isso, existe o comando `\mbox{ texto }`. Veja o exemplo a seguir.

Tabela 10.1: Descrição de alguns comandos matemáticos. Nessa tabela, `a` e `b` podem ser qualquer outra expressão.

Descrição	**Comando**	**Resultado**
Frações	`\frac{a}{b}`	$\frac{a}{b}$
Raízes	`\sqrt{a}`	$\sqrt{a}$
Sobrescrito	`a^b`	a^b
Subscrito	`a_b`	a_b
Somatório	`\sum_{a}^b`	$\sum_{a}^{b}$
Letras gregas	`\alpha`, `\beta`, `\gamma`, `\pi`	α, β, γ, π
Comparações	`>`, `<`, `\geq`, `\leq`	$>$, $<$, $\geq$, $\leq$
Limite	`\lim_{a \rightarrow b}`	$\lim_{a \rightarrow b}$
Piso e teto	`\lfloor` e `\rfloor`, `\lceil` e `\rceil`	$\lfloor$ e $\rfloor$, $\lceil$ e $\rceil$

```
Seja $F_n$ o $n$-ésimo número de Fibonacci.
Temos que $F_0 = 0$, $F_1 = 1$ e
$$F_n = F_{n-1} + F_{n-2} para todo n \geq 2.$$
ou, melhor dizendo,
$$F_n = F_{n-1} + F_{n-2} \mbox{ para todo } n \geq 2.$$
```

Seja F_n o n-ésimo número de Fibonacci. Temos que $F_0 = 0$, $F_1 = 1$ e

$$F_n = F_{n-1} + F_{n-2} paratodon \geq 2.$$

ou, melhor dizendo,

$$F_n = F_{n-1} + F_{n-2} \text{ para todo } n \geq 2.$$

Faça você mesmo

Vá ao nosso documento sobre o Ramanujan e adicione o texto da seção de resultados notáveis, formatando as fórmulas corretamente.

Capítulo 11

Referências cruzadas

Como visto nos capítulos anteriores, o LaTeX automaticamente enumera seções, capítulos, figuras, tabelas e mesmo equações. O motivo disso é para que possamos fazer referências a elas no restante do texto. Como todos esses objetos são "flutuantes", no sentido de que você pode mudá-los de lugar a qualquer momento e inserir novos deles entre os que já existem, não podemos escrever textos do tipo "veja a Figura 10", "esse assunto apareceu na página 9" ou "tal assunto foi visto na Seção 5" colocando os números literalmente.

Para que esse tipo de problema não ocorra, o LaTeX oferece comandos simples que nos ajudam a referenciar objetos corretamente. Você vai ter apenas a obrigação de lembrar qual é o objeto que deseja referenciar (ao invés de se preocupar onde ele está) e o LaTeX vai se preocupar em enumerar corretamente os objetos e as referências que você fizer.

Quando você precisa referenciar um objeto, deve colocar um comando `\label{rótulo}` próximo do mesmo, onde `rótulo` é um nome que você deve dar àquele objeto que não tenha caracteres especiais como acentos (normalmente usamos nomes representativos — isso ficará mais claro nos exemplos). O importante é que um rótulo só pode ser usado em um objeto, pois caso contrário você vai erroneamente mandar o LaTeX referenciar algo indesejado.

No caso de capítulos/seções/subseções, o comando `\label` deve ser colocado logo após o comando que os define. Veja os exemplos a seguir.

```
\section{Sobre a vida do autor}
\label{sec:vida}

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.

\subsection{Infância}
\label{subsec:infancia}

Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id,
vulputate a, magna.

\section{Sobre o trabalho do autor}
\label{sec:trabalho}

Etiam  lobortisfacilisis sem.  Nullam nec mi et neque pharetra sollicitudin.
```

11.1 Sobre a vida do autor

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

11.1.1 Infância

Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

11.2 Sobre o trabalho do autor

Etiam lobortisfacilisis sem. Nullam nec mi et neque pharetra sollicitudin.

No caso de figuras e tabelas, o comando `\label` deve ser colocado logo após o comando `\caption` dos ambientes `\figure` e `\table`. Isso é necessário porque é o comando `\caption` que gera a numeração desses objetos.

```
\begin{figure}[h!]
    \centering
    \includegraphics[width=0.1\textwidth]{figs/logo-ufabc}
    \caption{Logotipo da UFABC.}
    \label{fig:logotipo}
\end{figure}

\begin{table}[h!]
    \centering
    \caption{Exemplo de uma tabela em \LaTeX{}.}
    \label{tab:exemplo-tabela}
    \begin{tabular}{|c||l|r|}
     \hline
     \textbf{Col1} & \textbf{Col2} & \textbf{Col3} \\ \hline\hline
     Aaa & Bb & C \\ \hline
     D & Eeee & Ff \\ \hline
    \end{tabular}
\end{table}
```

Figura 11.1: Logotipo da UFABC.

Tabela 11.1: Exemplo de uma tabela em LaTeX.

Col1	**Col2**	**Col3**
Aaa	Bb	C
D	Eeee	Ff

E no caso de equações feitas com o ambiente `\equation`, o comando `\label` só precisa ser colocado dentro do ambiente `\equation` (procure colocar no início do mesmo, para não atrapalhar a escrita da fórmula).

```
Uma equação $a x^2 + b x + c = 0$ tem como solução
\begin{equation}
\label{eq:bhaskara}
    x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
\end{equation}
```

Uma equação $ax^2 + bx + c = 0$ tem como solução

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{11.1}$$

Nos exemplos anteriores, perceba que usei um padrão nos rótulos. Seções tinham rótulos iniciando com "sec:", figuras com "fig:", tabelas com "tab:", equações com "eq:". Isso é uma boa forma de não se perder entre os rótulos. Em primeiro lugar, você sempre vai saber que tipo de objeto aquele rótulo está referenciando. Em segundo, pode ser o caso de que faria sentido usar um mesmo nome de rótulo para objetos diferentes (por exemplo, posso ter uma seção que fala sobre Bháskara e uma equação que mostra a fórmula). Como eu disse antes, um rótulo só pode ser usado em um objeto, e o uso desses padrões impede o problema.

Uma vez que temos objetos rotulados, existem três comandos que podemos usar para referenciá-los. O comando `\ref{rótulo}` é substituído exatamente pelo número que o LaTeX deu ao objeto que rotulamos com o texto `rótulo`. Eu posso, por exemplo, mencionar as Figuras 1.1 ou 1.6 que coloquei lá no começo da apostila. Veja:

```
Temos a Seção~\ref{sec:vida}, a Seção~\ref{subsec:infancia}, a
Seção~\ref{sec:trabalho}, a Figura~\ref{fig:logotipo}, a
Tabela~\ref{tab:exemplo-tabela} e a Equação~\ref{eq:bhaskara}.
```

Temos a Seção 11.1, a Seção 11.1.1, a Seção 11.2, a Figura 11.1, a Tabela 11.1 e a Equação 11.1.

Observe que ao clicar no número, você será levado ao ponto do documento que apresenta aquele objeto.

O comando `\pageref{rótulo}` é substituído exatamente pela página onde o objeto que rotulamos com o texto `rótulo` aparece no documento final. Eu posso, por exemplo, dizer que o Capítulo 0 começou na página 4. Veja:

```
Temos a Seção~\ref{sec:vida} na página~\pageref{sec:vida}, a
Seção~\ref{subsec:infancia} na página~\pageref{subsec:infancia}, a
Seção~\ref{sec:trabalho} na página~\pageref{sec:trabalho}, a
Figura~\ref{fig:logotipo} na página~\pageref{fig:logotipo}, a
Tabela~\ref{tab:exemplo-tabela} na página~\pageref{tab:exemplo-tabela} e a
Equação~\ref{eq:bhaskara} na página~\pageref{eq:bhaskara}.
```

Temos a Seção 11.1 na página 33, a Seção 11.1.1 na página 33, a Seção 11.2 na página 33, a Figura 11.1 na página 34, a Tabela 11.1 na página 34 e a Equação 11.1 na página 35.

Por fim, temos o comando `\eqref{rótulo}` que é usado especificamente para referenciar ambientes `\equation`. Ele é fornecido pelo pacote `amsmath`.

```
O que fica melhor: Equação~\ref{eq:bhaskara} ou Equação~\eqref{eq:bhaskara}?
```

O que fica melhor: Equação 11.1 ou Equação (11.1)?

É importante notar também que você não precisa referenciar apenas objetos que já foram definidos antes no texto. Veja o exemplo a seguir.

```
Em~\eqref{eq:futura} você verá detalhes de como essa operação é feita.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.

Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id,
vulputate a, magna.

\begin{equation}
\label{eq:futura}
    \log_b(xy) = \log_b(x) + \log_b(y) \mbox{ porque } b^c \times b^d = b^{c+d}
\end{equation}
```

Em (11.2) você verá detalhes de como essa operação é feita.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.
Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

$$\log_b(xy) = \log_b(x) + \log_b(y) \text{ porque } b^c \times b^d = b^{c+d} \tag{11.2}$$

É por esse motivo que o nome desse capítulo é "referências cruzadas". Quando compilando um documento ".tex" que possui referências, é necessário fazer isso pelo menos duas vezes, para que o LaTeX as resolva corretamente: uma vez ele guarda os rótulos que foram definidos e na outra ele substitui os comandos de referência corretamente.

De forma geral, você pode colocar rótulos com o comando `\label` em vários outros objetos do texto, como por exemplo algum `\item` de um ambiente `enumerate`[1].

Faça você mesmo

Vá ao nosso documento sobre o Ramanujan e adicione rótulos à todas as seções e subseções. Conserte o texto da seção de introdução. Acrescente rótulos e referências à figura, tabela e equação que aparecem no texto.

[1] Enquanto eu não escrevo mais sobre isso, você pode ver outras informações aqui.

Capítulo 12

Referências bibliográficas

Uma outro ponto muito forte do LaTeX é a facilidade de gerenciar bibliografias. O método mais comum de gerenciar bibliografias no LaTeX é usando a ferramenta BibTeX. Com o BibTeX, a base de dados de referências é mantida em um arquivo separado, com extensão `.bib`. Antes de explicarmos os detalhes, vamos apresentar um exemplo bem sucinto de como fazer referências bibliográficas em LaTeX.

No arquivo `.bib`, teremos "entradas bibliográficas" com as nossas referências. Cada entrada tem um tipo (como por exemplo artigo de conferência, artigo de revista, livro), que deve ser devidamente declarado no arquivo `.bib`. Antes de especificar como deve ser declarada cada entrada, vamos ver um exemplo simples.

Suponha que temos um arquivo com o conteúdo a seguir que é chamado `bibliografia.bib`. Esse arquivo contém 3 entradas: um artigo, um livro e um website.

```
@article{ChudnovskyEtAl2006,
    title = {The strong perfect graph theorem},
    author = {Chudnovsky, Maria and Robertson, Neil and Seymour, Paul and
             Thomas, Robin},
    journal = {Annals of mathematics},
    pages = {51--229},
    year = {2006},
    publisher = {JSTOR}
}

@book{Harari2017,
    title = {Sapiens: a brief history of humankind},
    author = {Harari, Yuval N. and Perkins, Derek},
    year = {2017},
    publisher = {HarperCollins}
}

@misc{sapiens-solds,
```

```
    author = {Natasha Onwuemezi},
    title = {Harari's {S}apiens hits one million mark},
    url = {https://www.thebookseller.com/news/harari-hits-one-million-mark-603691}
}
```

Observe que a primeira linha de cada uma das três entradas tem formato `@tipo{rótulo,`. Aliás, cada entrada de um arquivo `.bib` tem o formato `@tipo{rotulo, parâmetros}`. Isso é obrigatório para todas as entradas bibliográficas. Por isso, no nosso arquivo principal (`.tex`), podemos referenciar esses itens bibliográficos por meio do comando `\cite{rótulo}`. Veja o exemplo a seguir.

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[brazil]{babel}
\bibliographystyle{plain}

\begin{document}

Chudnovsky, Robertson, Seymour, e Thomas provaram que um grafo é perfeito se e
somente se não contém um ciclo ímpar com pelo menos cinco vértices ou o seu
complemento como subgrafo induzido~\cite{ChudnovskyEtAl2006}.

O livro Sapiens~\cite{Harari2017} já vendeu mais de um milhão de
cópias~\cite{sapiens-solds}.

\bibliography{bibliografia}
\end{document}
```

Chudnovsky, Robertson, Seymour, e Thomas provaram que um grafo é perfeito se e somente se não contém um ciclo ímpar com pelo menos cinco vértices ou o seu complemento como subgrafo induzido [1].
O livro Sapiens [2] já vendeu mais de um milhão de cópias [3].

Referências Bibliográficas

[1] Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Annals of mathematics*, pages 51–229, 2006.

[2] Yuval N. Harari and Derek Perkins. *Sapiens: a brief history of humankind*. HarperCollins, 2017.

[3] Natasha Onwuemezi. Harari's Sapiens hits one million mark.

O comando

```
\bibliographystyle{plain}
```

indicou ao LaTeX qual é o estilo da bibliografia, que nesse caso é *plain* (existem muitos outros estilos, veja aqui, por exemplo).

Já o comando

```
\bibliography{bibliografia}
```

deve ser colocado na posição onde você deseja que a seção de referências bibliográficas seja inserida (note que ele cria o título da seção). Ele indica qual é o arquivo onde estão as entradas bibliográficas.

Voltemos agora às entradas que podem fazer parte de um arquivo `.bib` e os parâmetros de cada uma. Elas podem ser:

Article Um artigo de revista.

```
@article{rótulo,
    author = {},
    title = {},
    journal = {},
    year = {},
    volume = {},
    number = {},
    pages = {},
}
```

Book Um livro que tenha editora.

```
@book{rótulo,
    author = {},
    editor = {},
    title = {},
    publisher = {},
    year = {},
    volume = {},
    series = {},
    address = {},
    edition = {},
}
```

Inbook Parte de livro, que pode ser um capítulo ou intervalo de páginas.

```
@inbook{rótulo,
    author = {},
    editor = {},
    title = {título do livro},
    chapter = {número do capítulo},
    pages = {},
```

```
    year = {},
    volume = {},
    series = {},
    publisher = {},
    address = {},
    edition = {},
}
```

Incollection Parte de um livro que tem título próprio.

```
@incollection{rótulo,
    author = {},
    editor = {},
    title = {},
    booktitle = {},
    pages = {},
    year = {},
    publisher = {},
    address = {},
}
```

Inproceedings Artigo de anais de conferência.

```
@inproceedings{rótulo,
    author = {},
    title = {},
    booktitle = {},
    series = {},
    year = {},
    editor = {},
    volume = {},
    publisher = {},
    address = {},
}
```

Manual Documentação técnica.

```
@manual{rótulo,
    title = {},
    author = {},
    organization = {},
    address = {},
```

```
    edition = {},
    year = {},
}
```

Masterthesis Dissertação de mestrado.

```
@mastersthesis{rótulo,
    author = {},
    title = {},
    school = {},
    year = {},
}
```

phdthesis Tese de doutorado.

```
@phdthesis{rótulo,
    author = {},
    title = {},
    school = {},
    year = {},
}
```

Techreport Um relatório publicado por uma universidade ou outra instituição, geralmente numerado.

```
@techreport{rótulo,
    author = {},
    title = {},
    institution = {},
    year = {},
    number = {},
}
```

Unpublished Documento com autor e título, mas não publicado formalmente.

```
@unpublished{rótulo,
    author = {},
    title = {},
    note = {},
    year = {},
}
```

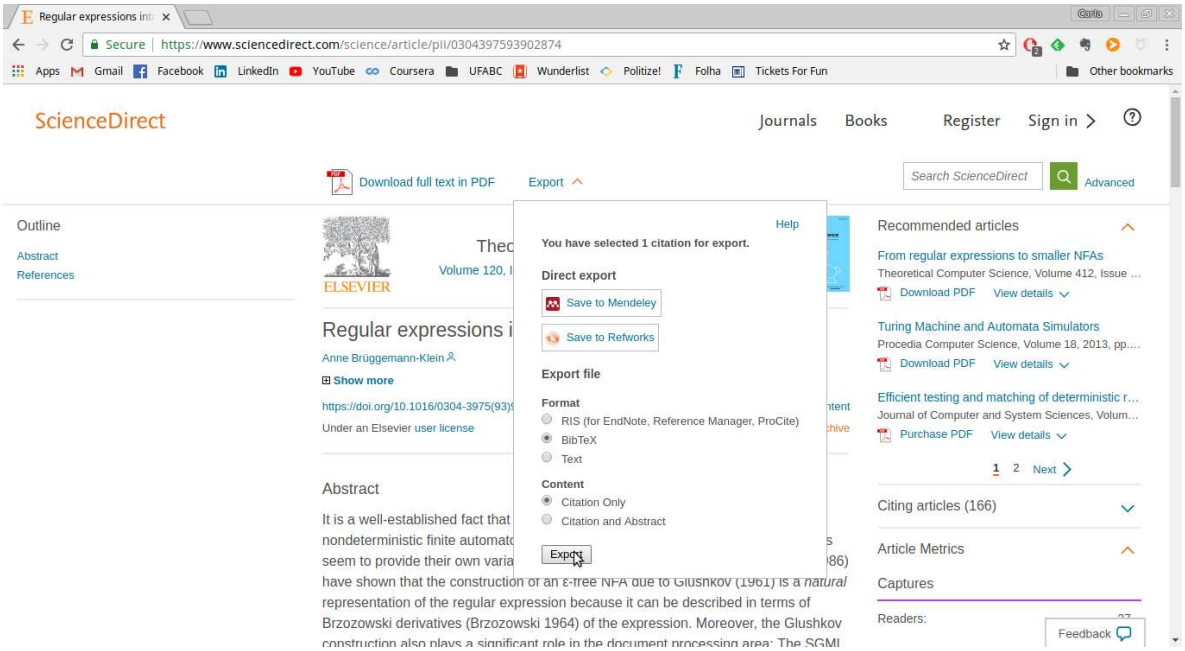


Figura 12.1: Ferramenta de exportação de entrada BibTEX em um site de revista científica.

Misc Use esse tipo quando nada mais parece apropriado.

```
@misc{rótulo,
    author = {},
    title = {},
    howpublished = {},
    month = {},
    year = {},
    note = {},
}
```

Você pode, portanto, preencher uma entrada manualmente dada a referência que você tem. Mas a grande maioria das editoras fornece uma forma de exportar a entrada já no formato para BibTEX. Veja um exemplo na Figura 12.1.

Por fim, por padrão a seção de referências bibliográficas que é criada pelo comando `\bibliography{}` não é inserida no sumário. Para fazer isso, você pode utilizar o comando

```
\addcontentsline{toc}{section}{Referências Bibliográficas}
```

que já foi mencionado no Capítulo 7.

Faça você mesmo

Vá ao nosso documento sobre o Ramanujan e crie corretamente a última seção, sobre as publicações dele.

Capítulo 13

Hyperlinks

É muito comum que documentos sejam consumidos em mídia digital, ou seja, são lidos no computador, *tablet*, ou *smartphone* e nunca chegam a serem impressos. Uma vantagem dos documentos digitais é que estes podem usufruir dos *hyperlinks* para melhorar a experiência de navegação pelo documento. Como vimos no Capítulo 11, o LaTeX provê uma forma muito simples de lidar com referências no documento e nós podemos transformar estas referências em *hyperlinks* simplesmente utilizando o pacote `hyperref`.

Se você apenas inserir o comando `\usepackage{hyperref}` no preâmbulo, o padrão do LaTeX é colocar uns quadrados vermelhos ao redor dos links, o que não é visualmente muito bonito (destaca o link, mas não é bonito). É possível personalizar isso usando o comando `\hypersetup`, que recebe uma sequência de parâmetros, separados por vírgula, no formato "`parâmetro=valor`". Veja um exemplo a seguir:

```
\hypersetup{
    colorlinks=true,
    linkcolor=blue,
    urlcolor=green,
}
```

No exemplo, o parâmetro `colorlinks=true` especifica que os *links* serão coloridos, `linkcolor=blue` especifica que os *links* de referências cruzadas (veja Capítulo 11) serão coloridos em azul e o parâmetro `urlcolor=green` especifica que os links para websites serão coloridos de verde.

Os parâmetros `linkcolor` e `urlcolor` recebem como valor o nome de uma cor em inglês, como no exemplo, *blue* (azul) e *green* (verde), respectivamente. Por padrão, o LaTeX compreende as seguintes cores: *white* (branco), *black* (preto), *yellow* (amarelo), *green* (verde), *blue* (azul), *purple* (roxo), *cyan* (azul ciano), e *magenta* (magenta).

Além das cores pré-definidas pelo LaTeX, podemos, é claro, definir nossas próprias cores. Para isso, primeiro precisamos importar o pacote `color`. Para definir uma nova cor, usamos o comando

```
\definecolor{nome-da-cor}{modelo}{definição-da-cor}
```

O parâmetro `modelo` diz qual forma você irá utilizar para definir a cor e seus possíveis valores são:

`rgb`, `RGB`, `cmyk` e `gray`.

Neste texto, vamos utilizar apenas o modelo RGB[1]. O modelo RGB mistura as cores vermelho (Red em inglês), verde (Green em inglês) e azul (Blue em inglês) para fazer a cor desejada. A intensidade de cada cor utilizada na mistura é dada por um número inteiro entre 0 e 255.

Para encontrar os valores RGB da cor pretendida, geralmente usamos uma ferramenta de seleção de cores, como a fornecida pelo website https://htmlcolorcodes.com/color-picker/ (veja a Figura 13.1).

Voltando ao comando `\definecolor`, nós colocamos os devidos valores RGB da cor pretendida no parâmetro `definição-da-cor`.

O exemplo abaixo mostra como colorir os *links* para websites do documento usando a cor de código RGB (120,10,191), que nomeamos de `MeuVioleta`.

```
\usepackage{color}
\usepackage{hyperref}

\definecolor{MeuVioleta}{RGB}{120, 10, 191}

\hypersetup{
    colorlinks=true,
    linkcolor=blue,
    urlcolor=MeuVioleta,
}
```

Como ficou claro pelo comando `\hypersetup`, podemos inserir *links* para websites. Para isso, temos os comandos `\url{website-url}` e `\href{website-url}{texto}`. O primeiro desses comandos, `\url{website-url}`, irá simplesmente exibir o texto do parâmetro `website-url` e torná-lo em um *link* clicável. Já o comando `\href{website-url}{texto}` irá exibir o texto apresentado no parâmetro `texto` e torná-lo um *link* clicável que irá levar ao endereço especificado pelo parâmetro `website-url`. Veja os exemplos a seguir.

```
Mais informações sobre o hyperlinks no \LaTeX{} podem ser encontradas em
\url{https://www.sharelatex.com/learn/Hyperlinks}
(ou procure no \href{http://www.google.com}{Google} \smiley{}).
```

Mais informações sobre o hyperlinks no LaTeX podem ser encontradas em https://www.sharelatex.com/learn/Hyperlinks (ou procure no Google ☺).

Faça você mesmo

Vá ao nosso documento sobre o Ramanujan e importe o pacote `hyperref` para habilitar os *hyperlinks* no seu texto. Faça alterações para personalizar as cores e a forma de destaque dos links. Insira comandos `\href` onde for necessário.

[1] Para mais informações sobre os outros modos, veja aqui.

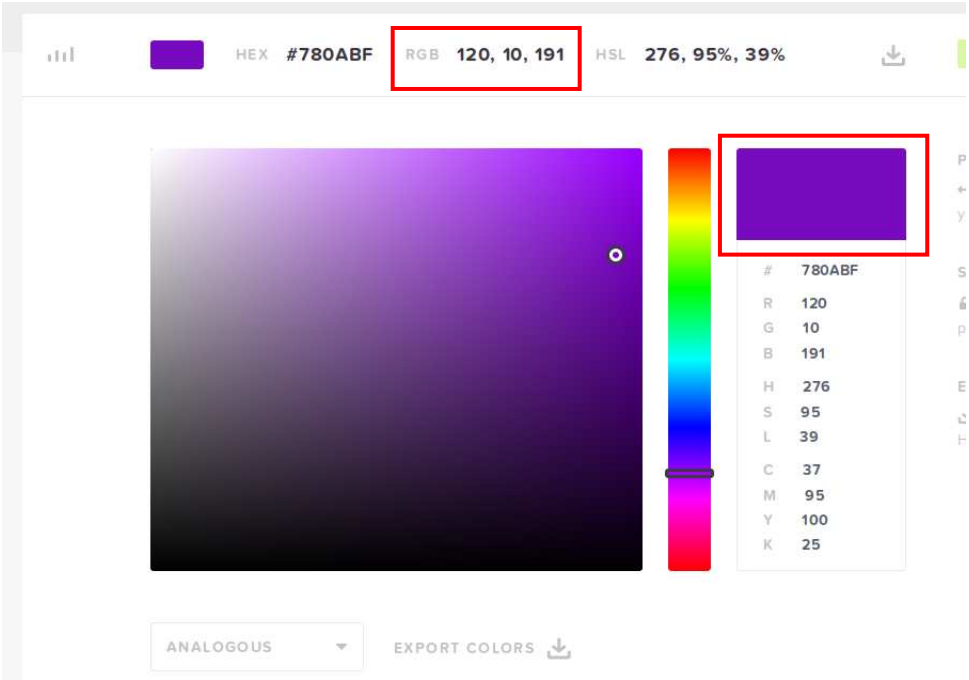


Figura 13.1: Exemplo de uma ferramenta de seleção de cores (link). A cor e seu código RGB estão destacadas por retângulos vermelhos.

Capítulo 14

Modo matemático (extra)

Em breve.

Capítulo 15

Apresentação de slides

Em breve.

Capítulo 16

Pôsteres

Em breve.

Capítulo 17

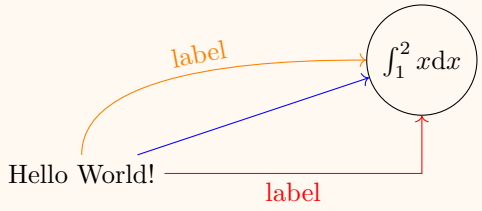
Mais exemplos do que pode ser feito com LaTeX

Eu não sou a criadora dos exemplos a seguir, então cada um deles apresenta a fonte de onde o retirei. Você também pode utilizar essas referências para pesquisar mais sobre o exemplo.

O pacote `tikz` é provavelmente o mais poderoso para criar elementos gráficos diretamente com código LaTeX. Veja mais sobre ele aqui.

Exemplo de uso do `tikz`

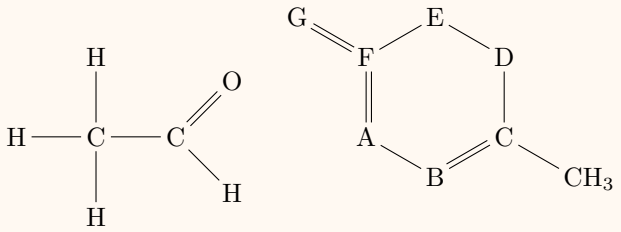
```
\begin{tikzpicture}
    \path (0,0) node(x) {Hello World!}
    (3,1) node[circle,draw](y) {$\int_1^2 x \mathrm d x$};
    \draw[->,blue] (x) -- (y);
    \draw[->,red] (x) -| node[near start,below] {label} (y);
    \draw[->,orange] (x) .. controls +(up:1cm) and +(left:1cm) ..
        node[above,sloped] {label} (y);
\end{tikzpicture}
```



O pacote `chemfig` pode ser usado para criar fórmulas químicas. Veja mais sobre ele aqui.

Exemplo de uso do `chemfig`

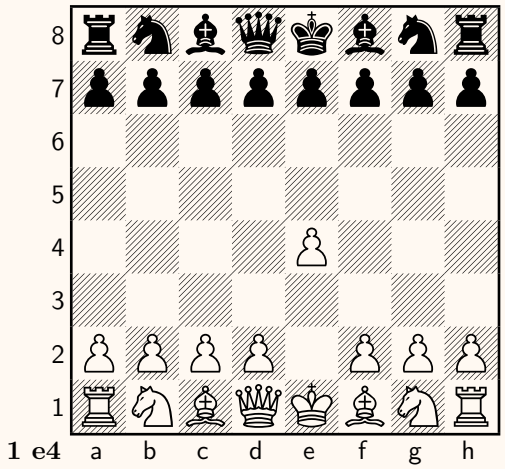
```
\chemfig{H-C(-[2]H)(-[6]H)-C(=[1]O)-[7]H}
\chemfig{A*6(-B=C(-CH_3)-D-E-F(=G)=)}
```



O pacote `skak` é usado para mostrar jogos de xadrez. Veja mais sobre ele aqui.

Exemplo de uso do `skak`

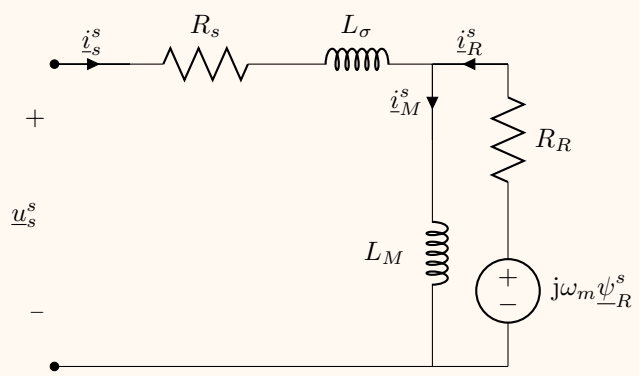
```
\newgame
\mainline{1.e4}
\showboard
```



O pacote `circuitikz` oferece comandos para desenhar circuitos eletrônicos. Veja mais sobre ele aqui.

Exemplo de uso do `circuitikz`

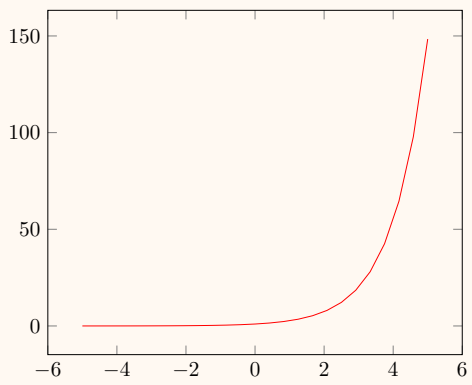
```
\begin{circuitikz}[american voltages]
 \draw
 (0,0) to [short, *-] (6,0)
 to [V, l_=$\mathrm{j}{\omega}_m \underline{\psi}^s_R$] (6,2)
 to [R, l_=$R_R$] (6,4)
 to [short, i_=$\underline{i}^s_R$] (5,4)
 (0,0) to [open, v^>=$\underline{u}^s_s$] (0,4)
 to [short, *- ,i=$\underline{i}^s_s$] (1,4)
 to [R, l=$R_s$] (3,4)
 to [L, l=$L_{\sigma}$] (5,4)
 to [short, i_=$\underline{i}^s_M$] (5,3)
 to [L, l_=$L_M$] (5,0);
 \end{circuitikz}
```

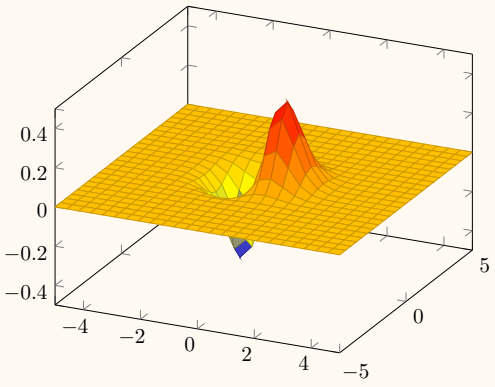


O pacote `pgfplots` é baseado no `tikz` e voltado à criação de gráficos. Veja mais sobre ele aqui.

Exemplo de uso do `pgfplots`

```
\begin{tikzpicture}
    \begin{axis}
        \addplot[color=red]{exp(x)};
    \end{axis}
\end{tikzpicture}
\hskip 5pt
\begin{tikzpicture}
    \begin{axis}
        \addplot3[surf]{exp(-x^2-y^2)*x};
    \end{axis}
\end{tikzpicture}
```

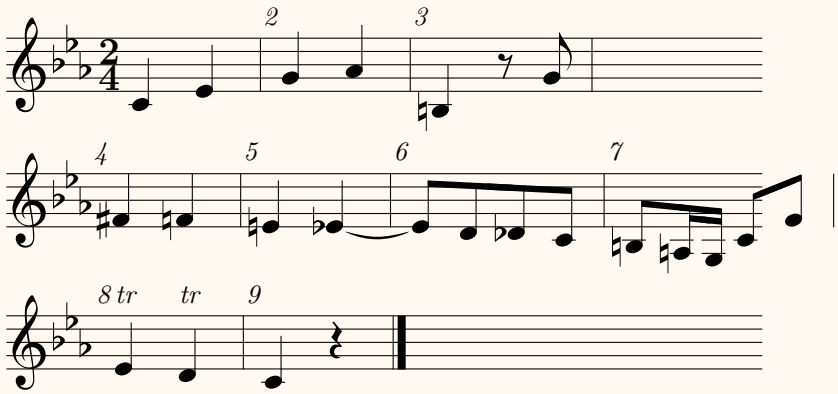




O pacote `musixtex` permite a escrita de notação musical. Veja mais sobre ele aqui.

Exemplo de uso do `musixtex`

```
 \hsize=100mm
\generalmeter{\meterfrac24}%
\parindent 0pt
\generalsignature{-3}
\nostartrule
\startpiece\bigaccid
\NOtes\qu{ce}\en\bar
\NOtes\qu{gh}\en\bar
\NOtes\qu{=b}\en
\Notes\ds\cu g\en\bar
\NOtes\qu{^f=f}\en\bar
\NOtes\qu{=e}\itied0e\qu{_e}\en\bar
\Notes\ttie0\Qqbu ed{_d}c\en\bar
\Notes\ibu0b{-2}\qb0{=b}\en
\notes\nbbu0\qb0{=a}\tqh0N\en
\Notes\Dqbu cf\en\bar
\NOtes\uptext{\it tr}\qu e\uptext{\it tr}\qu d\en\bar
\NOtes\qu c\qp\en\Endpiece
```



O pacote `listings` permite a escrita de algoritmos. Veja mais sobre ele aqui.

Exemplo de uso do `listings`

```
\begin{lstlisting}[language=C]
#include <stdio.h>

void main() {
    printf("Hello world!\n");
    return 0;
}
\end{lstlisting}
```

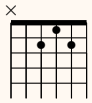
```
#include <stdio.h>

void main() {
   printf("Hello world!\n");
   return 0;
}
```

O pacote `gchords` permite criar diagramas de violão/guitarra. Veja mais sobre ele aqui.

Exemplo de uso do `gchords`

```
\begin{verse}
    \smallchords
    \newcommand{\AsevenMaj}{\chord{t}{x,n,p2,p1,p2,n}{A7+}}
    \upchord{\AsevenMaj}
\end{verse}
```

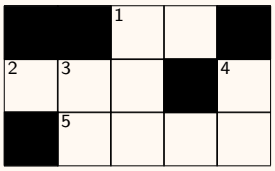


A7+

O pacote `cwpuzzle` permite criar tabuleiros estilo palavras-cruzadas. Veja mais sobre ele aqui.

Exemplo de uso do `cwpuzzle`

```
\begin{Puzzle}{5}{3}%
    |* |* |[1]E|X |* |.
    |[2]A|[3]S|T |* |[4]T|.
    |* |[5]P|A |R |T |.
\end{Puzzle}
```



O pacote `tkz-graph` permite a criação de grafos. Veja mais sobre ele aqui.

Exemplo de uso do `tikz`

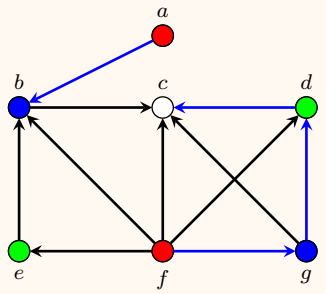
```
\begin{tikzpicture}[scale=1.9]
    \tikzstyle{part}=[blue]
    \GraphInit[vstyle=Welsh]
    \SetVertexMath
    \SetVertexLabel
    \SetVertexLabelOut

    \tikzset{VertexStyle/.append style = {minimum size = 8pt, inner sep = 0pt}}
    \Vertex[x = 2, y = -0.5, Lpos=90]{a}
    \Vertex[x = 1, y = -1, Lpos = 90]{b}
    \Vertex[x = 2, y = -1, Lpos = 90]{c}
    \Vertex[x = 3, y = -1, Lpos = 90]{d}
    \Vertex[x = 1, y = -2, Lpos = -90]{e}
    \Vertex[x = 2, y = -2, Lpos = -90]{f}
    \Vertex[x = 3, y = -2, Lpos = -90]{g}

    \tikzset{EdgeStyle/.append style = {->, >=stealth, line width = 1pt}}
    \Edge(b)(c)
    \Edge(e)(b)
    \Edge(f)(c)
    \Edge(f)(e)
    \Edge(f)(b)
    \Edge(f)(d)
    \Edge(g)(c)

    \tikzset{EdgeStyle/.append style = {color = blue}}
    \Edge(a)(b)
    \Edge(d)(c)
    \Edge(f)(g)
    \Edge(g)(d)

    \AddVertexColor{red}{a}
    \AddVertexColor{blue}{b}
    \AddVertexColor{green}{e}
    \AddVertexColor{red}{f}
    \AddVertexColor{blue}{g}
    \AddVertexColor{green}{d}
\end{tikzpicture}
```



Capítulo 18

Indo além

Agora que você conhece como o LaTeX funciona, ele não é mais tão assustador, não é mesmo? Você já conhece boa parte dos comandos básicos e das personalizações possíveis. Mas existe muito muito muito mais por aí. E com um pouco de curiosidade, você será capaz de descobrir todos os recursos que o LaTeX oferece.

Os materiais que eu cito no Capítulo 19, de referências, contêm bastante coisa também, muitas indo bem além do que eu escrevi nessa apostila. Além disso, muitas vezes basta fazer uma busca pelo que você quer fazer no seu documento para descobrir que muitas outras pessoas já tentaram isso também e, boa parte das vezes, a solução é bem simples de ser encontrada. Se possível, procure fazer a busca em inglês.

Por fim, a ideia que eu quero passar para você é: pratique. A partir de agora, faça todos os documentos que você precisa fazer usando LaTeX[1]. Cada nova funcionalidade que você descobrir, é algo que você pode reutilizar no futuro facilmente. E uma vez que você tiver um bom vocabulário em LaTeX, tudo vai se tornar muito mais fácil.

[1] Na falta de tempo para começar um arquivo novo toda vez, você pode usar um template pronto, como por exemplo os que o Overleaf provê.

Capítulo 19

Referências

Eu criei essa apostila principalmente a partir dos conhecimentos prévios que eu já tinha sobre LaTeX. No entanto eu pesquisei outras fontes para ter certeza de que estava cobrindo todos os comandos básicos possíveis e também para verificar todas as opções de cada um deles. Também precisei procurar por vários dos pacotes extras que apresento no Capítulo 17. Como não usei nenhum livro ou artigo científico para isso, é com muita tristeza que informo que meu capítulo de referências não usará o BibTeX. Apenas vou citar a seguir uma lista de sites/materiais que acabei acessando em algum momento.

- ⇝ https://www.google.com.br (esse aqui é o mais importante de todos, sem dúvida)
- ⇝ https://en.wikibooks.org/wiki/LaTeX
- ⇝ https://www.latex-project.org/
- ⇝ https://www.overleaf.com/latex/templates/
- ⇝ https://tobi.oetiker.ch/lshort/lshort.pdf
- ⇝ https://reu.dimacs.rutgers.edu/Symbols.pdf
- ⇝ http://tug.ctan.org/info/symbols/comprehensive/symbols-a4.pdf
- ⇝ https://www.sharelatex.com/learn/Learn_LaTeX_in_30_minutes
- ⇝ http://www.mat.ufmg.br/~regi/topicos/intlat.pdf
- ⇝ https://www.overleaf.com/latex/learn/free-online-introduction-to-latex-part-1

Índice Remissivo

caracteres reservados, 19
centering, 17
chapter, 10

document, 16
documentclass, 15

espaçamento, 23
    linha, 23
    parágrafo, 24

figuras, 14
    caption, 17
    figure, 17
    graphicx, 14
    includegraphics, 15
fonte, 21
    família, 22
    itálico, 21
    negrito, 21
    sublinhado, 21
    tamanho, 22

indentação, 24

listas, 3
    aninhamento, 4
    description, 3
    enumerate, 3
    itemize, 3

part, 10
preâmbulo, 17

section, 10
subfiguras, 19
subsection, 10
sumário, 12

tabelas, 21
    caption, 26
    landscape, 28
    longtable, 28
    multicolumn, 24
    table, 26
    tabular, 21

traços, 19

verbatim, 20