



**Universidade Federal de Uberlândia**

FAMAT - FACULDADE DE MATEMÁTICA

---

**APOSTILA DE  $\text{\LaTeX}$**

---

**Daniel Cariello  
Evaneide Alves Carneiro  
Germano Abud de Rezende**

**1º semestre 2011**

---

# CONTEÚDO

<b>1</b>	<b>Conhecendo o <math>\text{\LaTeX}</math></b>	<b>5</b>
1.1	Um pouco de História . . . . .	5
1.2	Porque o $\text{\LaTeX}$ ? . . . . .	6
1.3	Instalação do $\text{\LaTeX}$ . . . . .	6
1.4	Estrutura Básica de um Arquivo $\text{\LaTeX}$ . . . . .	6
<b>2</b>	<b>Formatação de Textos. Comandos.</b>	<b>10</b>
2.1	Formatação de Textos . . . . .	10
2.1.1	Tamanho e Estilo da Fonte . . . . .	10
2.1.2	Posição do Texto . . . . .	12
2.1.3	Espaçamento . . . . .	12
2.1.4	Listas . . . . .	15
<b>3</b>	<b>Fórmulas Matemáticas</b>	<b>18</b>
3.1	Fracções, Expoentes, Índices, Raízes e Pontos . . . . .	18
3.1.1	Fracções . . . . .	18
3.1.2	Expoentes e Índices . . . . .	19
3.1.3	Raízes . . . . .	19
3.1.4	Pontos . . . . .	19
3.2	Textos Dentro de Fórmulas . . . . .	20
3.3	Delimitadores . . . . .	21
3.4	Alguns Símbolos Muito Usados . . . . .	21
3.5	Fontes e Símbolos . . . . .	22
3.5.1	Fontes Especiais no Modo Matemático . . . . .	22
3.5.2	Símbolos e Fórmulas em Negrito ou com Contornos . . . . .	22
3.5.3	Funções . . . . .	24
3.6	Somatórios, Produtórios, Limites, Derivadas e Integrais . . . . .	24

3.6.1	Somatórios e Produtórios . . . . .	24
3.6.2	Limites, Derivadas e Integrais . . . . .	25
3.7	Matrizes . . . . .	26
<b>4</b>	<b>Classes de Documentos. Layout</b>	<b>28</b>
4.1	Classes de Documentos e Opções . . . . .	28
4.2	Layout do Documento . . . . .	30
4.2.1	Parágrafo e Espaçamentos. Quebras . . . . .	30
4.2.2	Caracteres Especiais . . . . .	30
4.3	Ambientes Básicos . . . . .	31
4.4	Tabelas . . . . .	31
4.4.1	Juntando Colunas . . . . .	32
4.4.2	Linhas Múltiplas e Omissão de Linhas . . . . .	32
4.4.3	Igualando Largura das Colunas . . . . .	33
4.5	Ambiente equation . . . . .	34
4.6	Ambiente Minipage . . . . .	35
4.7	Ambiente Quote e similares . . . . .	37
4.8	Ambiente Verbatim . . . . .	38
<b>5</b>	<b>Figuras e algumas dicas</b>	<b>40</b>
5.1	Inserindo Figuras no Texto . . . . .	40
5.1.1	Figuras Flutuantes . . . . .	40
5.1.2	Imagens Geradas pelo L <sup>A</sup> T <sub>E</sub> X . . . . .	40
5.1.3	Imagens Geradas por Editores Gráficos Externos . . . . .	41
5.2	Empilhando Símbolos . . . . .	43
5.3	Hifenização . . . . .	44
5.4	Dicionário . . . . .	44
<b>6</b>	<b>Beamer (Slides)</b>	<b>45</b>
<b>7</b>	<b>Criando Comandos e Ambientes</b>	<b>49</b>
7.1	Criando Comandos . . . . .	49
7.1.1	Comandos sem Parâmetros . . . . .	49
7.1.2	Comandos com Parâmetros . . . . .	50
7.1.3	Comandos Duais . . . . .	50
7.1.4	Redefinindo Comandos . . . . .	51
7.1.5	O Comando <i>def</i> . . . . .	51
7.2	Criando Ambientes . . . . .	52
7.2.1	Ambientes sem Argumentos . . . . .	52
7.2.2	Ambientes com Argumentos . . . . .	52
7.2.3	Redefinindo Ambientes . . . . .	53

<b>8</b>	<b>Dividindo o Documento</b>	<b>54</b>
8.1	Arquivo Mestre . . . . .	54
8.2	Restringindo Arquivos na Saída . . . . .	55
8.3	Dividindo um Capítulo . . . . .	57
8.4	Capítulos ou Seções com Título Longo . . . . .	58
<b>9</b>	<b>Bibliografia e Índice Remissivo</b>	<b>59</b>
9.1	Referências Bibliográficas . . . . .	59
9.1.1	O Ambiente thebibliography . . . . .	59
9.2	Criando o Índice Remissivo . . . . .	60
9.2.1	Criando Sub-entradas . . . . .	62
<b>10</b>	<b>Diagramas, Tabelas, Caixas Gráficas</b>	<b>63</b>
10.1	Diagramas . . . . .	63
10.1.1	Diagramas Simples . . . . .	63
10.2	Símbolos e Fórmulas . . . . .	64
10.3	Índices . . . . .	65
10.4	Tabelas . . . . .	65
10.4.1	Tabelas Longas . . . . .	65
10.4.2	Listas de Figuras e Tabelas . . . . .	67
10.5	Caixas Gráficas . . . . .	69
10.5.1	Tamanho do Texto . . . . .	69
10.5.2	Rotação do Texto . . . . .	70

---

---

# CAPÍTULO 1

---

## CONHECENDO O L<sup>A</sup>T<sub>E</sub>X

---

---

### 1.1 Um pouco de História

---

A História do T<sub>E</sub>X teve início em 1978, quando , um professor de Ciência da Computação da Stanford University, recebeu um *preprint* do segundo volume do seu livro *The Art of Computer Programming*. Kunuth ficou decepcionado com a baixa qualidade tipográfica do documento recebido da editora, e pensou que ele, enquanto cientista da computação, deveria poder fazer algo a respeito.

Resolveu, assim, levar adiante uma idéia: implementar um sistema computacional eficiente para editoração. A tarefa que ele inicialmente estimou que duraria 6 meses, levou 10 anos para ser concluída, e vários outros cientistas o ajudaram.

Em 1985, Kunuth distribuiu o T<sub>E</sub>X juntamente com o macro (configuração) que ele usou para editar o seu livro. O sistema T<sub>E</sub>X era então formado por um conjunto de programas para processar um documento de forma automática e eficiente, mas tinha um problema: eram necessários conhecimentos de editoração para produzir documentos de qualidade. Com o intuito de permitir diagramação dos documentos científicos de qualidade profissional sem a necessidade de conhecimentos específicos de editoração, foi iniciado um projeto para desenvolver macros (configurações) novos para o sistema T<sub>E</sub>X. O grupo foi coordenado pelo matemático Leslie Lamport do DEC(Digital Equipment Corporation - Compaq). O projeto foi concluído em 1985 e recebeu o nome de L<sup>A</sup>T<sub>E</sub>X.

Resumindo:

O L<sup>A</sup>T<sub>E</sub>X é um pacote de macros do T<sub>E</sub>X que permite ao autor diagramar e imprimir seus trabalhos como documento de altíssima qualidade tipográfica, usando *layout* profissional predefinido. O L<sup>A</sup>T<sub>E</sub>X pode ser usado para produzir todos os tipos de documentos, desde uma simples carta até livros completos.

A versão atual do L<sup>A</sup>T<sub>E</sub>X é a 2 $\epsilon$  e a do T<sub>E</sub>X é 3,14159 e está convergindo para  $\pi$ . O projeto L<sup>A</sup>T<sub>E</sub>X 3 está em andamento.

---

## 1.2 Porque o L<sup>A</sup>T<sub>E</sub>X?

---

Os programas de processamento de texto podem ser divididos em duas classes: A primeira classe é formada pelos programas do tipo WYSWYG (“what you see is what you get”), que são aqueles onde o usuário digita na tela o documento exatamente como ele será impresso ou visualizado. Na segunda classe, à qual pertence o L<sup>A</sup>T<sub>E</sub>X, o processamento do texto se dá em duas etapas: o usuário digita um *input* ou arquivo fonte em um editor de texto e em seguida o submete a um formatador de textos(L<sup>A</sup>T<sub>E</sub>X), que retorna o *output* ou arquivo de saída para ser impresso ou visualizado.

Programas pertencentes à segunda classe podem parecer mais complicados à primeira vista, mas ao começar a usá-los logo percebemos várias vantagens:

- Escrita de complexas fórmulas matemáticas usando comandos. Por exemplo,  $\int_0^2 5^x dx$  é conseguida com o comando

```
\int_{0}^{2} 5^x dx.
```

- Numeração automática de teoremas, definições, fórmulas, etc.
- Mudança na formatação de todo um documento com apenas a mudança de alguns comandos.
- Outras que você perceberá com o uso!!!

Além das vantagens citadas acima, não podemos deixar de ressaltar que o L<sup>A</sup>T<sub>E</sub>X é um *freeware*, com implementações disponíveis para todos os sistemas operacionais mais famosos. Assim, tudo o que fizermos aqui no curso você pode e deve fazer no seu computador, além de usá-lo para treinar com outros textos(por exemplo, um exercício que você fez na aula de Teoria dos Números ou um teorema de Cálculo). A próxima sessão será sobre a instalação dos programas.

**Curiosidade:** O nome T<sub>E</sub>X na verdade é formado pelas letras gregas  $\tau\epsilon\chi$  (tau, épsilon, chi), que são as iniciais da palavra tecnologia. A pronúncia correta é *téqui*, *latéqui*, etc.

---

## 1.3 Instalação do L<sup>A</sup>T<sub>E</sub>X

---

Para trabalhar com o L<sup>A</sup>T<sub>E</sub>X, você precisa instalar uma distribuição L<sup>A</sup>T<sub>E</sub>X, um editor de texto e visualizadores e processadores de texto para arquivos .pdf e .ps. Aqui no curso usaremos a distribuição MiK<sub>T</sub>E<sub>X</sub>, o editor T<sub>E</sub>XMaker para Windows. Para o Linux a distribuição padrão é o T<sub>E</sub>XLive e usaremos o editor T<sub>E</sub>XMaker. Também são muito utilizados os editores T<sub>E</sub>XNicCenter (Windows) e Kile (Linux).

---

## 1.4 Estrutura Básica de um Arquivo L<sup>A</sup>T<sub>E</sub>X

---

Um arquivo L<sup>A</sup>T<sub>E</sub>X contém o texto a ser processado e comandos que indicam como fazê-lo. Em geral um comando inicia-se com `\`.

O *input* mais simples para um arquivo L<sup>A</sup>T<sub>E</sub>X é mostrado abaixo:

**Exemplo 1.4.1.** Abra o T<sub>E</sub>X<sup>N</sup>ic Center - File - New e digite:

```
\documentclass{article}
\begin{document}
Simples... Muito simples!!!
\end{document}
```

Salve o arquivo como *Exemplo1*. Gere o arquivo de visualização (.dvi, .pdf ou .ps) e veja o *output*.

Um documento em L<sup>A</sup>T<sub>E</sub>X se divide em duas partes: a primeira, antes do comando `\begin{document}`, é o *preâmbulo* e a segunda, iniciada com `\begin{document}` e encerrada com `\end{document}` é o *corpo* do documento. No preâmbulo são definidas opções para o processamento do texto, como por exemplo: tipo de documento, tipo de papel, tamanho da letra base, etc. Ele deve necessariamente começar com `\documentclass{<estilo>}`, onde as opções para <estilo> são: *article*, *book*, *letter* ou *report*. No corpo do documento está o texto a ser processado e comandos com efeito local.

Voltemos ao exemplo (1.4.1). Tente incluir no corpo do arquivo a frase *É tão simples*. O que acontece no *output*?

Resolvemos o problema incluindo no preâmbulo o comando `\usepackage[latin1]{inputenc}`, que indica que usaremos o pacote (aprenderemos sobre pacotes ao longo do curso) *inputenc* com a opção *latin1*. No Linux a codificação padrão é UTF-8, assim a opção correta para o pacote seria *utf8*. Isso permitirá acentuar as palavras diretamente o teclado. Volte novamente ao exemplo anterior, inclua no preâmbulo o comando acima, processe e veja o *output*.

Veremos a seguir um exemplo um pouco mais detalhado

**Exemplo 1.4.2.** Salve como *Exemplo2* o seguinte:

```
\documentclass[a4paper,10pt,twoside]{report}

%*****
%INSTRUÇÕES SOBRE USO DE PACOTES
%*****
\usepackage[utf8]{inputenc} % Permite usar acentuação direto do teclado. No windows use lat

%*****
%FORMATAÇÃO DAS PÁGINAS
%*****

\setlength{\textwidth}{15cm} % LARGURA DO TEXTO
\setlength{\textheight}{22cm} % ALTURA DO TEXTO

\begin{document}
O Teorema Fundamental da Aritmética diz que todo inteiro maior do
```

que 1 pode ser representado como um produto de fatores primos. Tal representação é única, a menos da ordem.

```
\end{document}
```

No exemplo (1.4.2), apareceu no preâmbulo o comando

```
\documentclass[a4paper,10pt,twoside]{report}
```

que indica que o nosso texto será processado em tamanho A4, com letra base de 10pt(poderia ser 11pt ou 12pt) e será impresso nos dois lados do papel. Além disso, o estilo do documento é *report*. O que aparece entre colchetes são as **opções** do documento. Você não precisa especificar as opções para o arquivo. Quando você não o faz, é usado o *default* (padrão), que depende do estilo do documento.

Outra novidade do exemplo acima são os comentários. Tudo o que vem depois de um sinal de % na mesma linha é entendido pelo L<sup>A</sup>T<sub>E</sub>X como um comentário - não faz parte do texto a ser processado. Também definimos o tamanho desejado para a altura e largura do texto.

Nas próximas aulas aprenderemos sobre fórmulas matemáticas, mas já podemos começar a treinar. Uma fórmula matemática no meio do texto começa e termina com \$. Já se quisermos dar destaque à fórmula, deixando-a sozinha na linha e centralizando-a usamos \$\$ no início e no fim. Veja o próximo exemplo.

#### Exemplo 1.4.3. *O texto*

As raízes da equação do segundo grau  $ax^2 + bx + c = 0$  são

$$x = \frac{-b \pm \sqrt{\Delta}}{2a},$$

onde  $\Delta = b^2 - 4ac$ . Chamaremos essas raízes de  $x_0$  e  $x_1$ .

é resultado do seguinte código fonte:

```
\documentclass[a4paper,12pt,twoside]{report}
```

```
%*****
```

```
%INSTRUÇÕES SOBRE USO DE PACOTES
```

```
%*****
```

```
\usepackage[utf8]{inputenc} % Permite usar acentuação direto do teclado
```

```
%*****
```

```
%FORMATAÇÃO DAS PÁGINAS
```

```
%*****
```

```
\setlength{\textwidth}{15cm} % LARGURA DO TEXTO
```

```
\setlength{\textheight}{22cm} % ALTURA DO TEXTO
```

```
\begin{document}
```



As raízes da equação do segundo grau  $ax^2 + bx + c = 0$  são  $x = \frac{-b \pm \sqrt{\Delta}}{2a}$ , onde  $\Delta = b^2 - 4ac$ .  
 Chamaremos essas raízes de  $x_0$  e  $x_1$ .  
 $\end{document}$

**Exercício 1.4.1.** *Produza o seguinte texto:*

As propriedades básicas das operações de adição e multiplicação são dadas a seguir: Quaisquer que sejam os números reais  $a$  e  $b$  tem-se:

$$a + b = b + a$$

$$ab = ba$$

$$(a + b) + c = a + (b + c)$$

$$a + 0 = a, a1 = a$$

$$a + (-a) = 0, a\frac{1}{a} = 1$$

**Teste 1.4.1.** *Produza o seguinte texto:*

**Decaimento das Substâncias Radioativas:** Certas substâncias, como o rádio e o urânio, se transformam espontaneamente em outras substâncias através da emissão de partículas subatômicas. Essas substâncias são chamadas de radioativas. As substâncias radioativas podem ser muito úteis, mas também podem ser perigosas. O gás radônio, por exemplo, produzido pela decomposição do rádio na crosta terrestre, é radioativo e pode constituir um perigo para a saúde.

Se chamarmos de  $y$  a quantidade de uma substância radioativa, o valor de  $y$  em função do tempo será dado por uma equação da forma

$$y = y_0 b^t.$$

Na equação acima,  $t$  representa o tempo, medido em unidades apropriadas (anos, dias, minutos, etc., dependendo da substância). O valor da base  $b$  varia de acordo com a substância considerada e satisfaz às desigualdades  $0 < b < 1$ . A constante  $y_0$  é a quantidade inicial da substância.

No caso do gás radônio, por exemplo,  $t$  é normalmente medido em dias e a base é aproximadamente 0,835, de modo que a equação de decaimento do gás radônio é

$$y = y_0(0,835)^t,$$

onde  $y_0$  é a quantidade inicial.

---

# CAPÍTULO 2

---

## FORMATAÇÃO DE TEXTOS. COMANDOS.

---

### 2.1 Formatação de Textos

---

Agora aprenderemos como mudar o tamanho, o estilo e a cor das letras em um texto, além de outros elementos de formatação, como parágrafos, espaçamento, etc.

#### 2.1.1 Tamanho e Estilo da Fonte

Você pode optar pelos seguintes tamanhos de letra em seu texto:

Comando	Display
<code>{\tiny o menor}</code>	o menor
<code>{\scriptsize muito pequeno}</code>	muito pequeno
<code>{\footnotesize bem pequeno}</code>	bem pequeno
<code>{\small pequeno}</code>	pequeno
<code>{\normalsize tamanho padrão}</code>	tamanho padrão
<code>{\large um pouco grande}</code>	um pouco grande
<code>{\Large maior}</code>	maior
<code>{\LARGE grande}</code>	grande
<code>{\huge bem grande}</code>	bem grande
<code>{\Huge o maior}</code>	o maior

Tabela 2.1: Tamanhos das Letras

Você pode digitar o comando como na tabela (2.1) ou ir ao Menu e escolher o tamanho desejado.

**Observação 2.1.1.** *Nos exemplos dados daqui por diante, o código fonte não conterá o preâmbulo, apenas o corpo do documento. Se for necessário usar algum comando novo no preâmbulo, o mesmo será citado na seção correspondente.*

**Exemplo 2.1.1.** *Vamos produzir o texto abaixo.*

Se uma função  $f$  é diferenciável em um ponto  $a$ , ela é contínua em  $a$ . A recíproca NÃO é verdadeira.

Código Fonte:

Se uma função  $f$  é diferenciável em um ponto  $a$ , ela é contínua em  $a$ . A recíproca {\Large NÃO} é verdadeira.

**Exemplo 2.1.2.** *Produza o texto a seguir. Use o comando  $\vec{v}$  para obter  $\vec{v}$ .*

Um autovalor de uma matriz quadrada  $A$  é um escalar  $c$  tal que  $A\vec{v} = c\vec{v}$  se verifica para algum vetor  $\vec{v}$  não-nulo. Neste caso, dizemos que  $\vec{v}$  é um autovetor da matriz  $A$ .

Código Fonte:

Um autovalor de uma matriz quadrada  $A$  é um escalar  $c$  tal que  $A\vec{v}=c\vec{v}$  se verifica para algum vetor  $\vec{v}$  não-nulo. Neste caso, dizemos que  $\vec{v}$  é um autovetor da matriz  $A$ .

Agora vamos aprender um pouco sobre o formato da fonte. Vejamos a tabela a seguir.

Comando	Display
<code>\textit{Itálico}</code>	<i>Itálico</i>
<code>\textsl{Inclinado}</code>	<i>Inclinada</i>
<code>\textbf{Negrito}</code>	<b>Negrito</b>
<code>\emph{Destacado}</code>	<i>Destacado</i>

Tabela 2.2: Estilos de Fonte

A fonte padrão do L<sup>A</sup>T<sub>E</sub>X é a CM(Computer Modern), que é a mais usada e apropriada para fins científicos. Mais adiante no curso, aprenderemos como modificar a fonte no documento.

### 2.1.2 Posição do Texto

O alinhamento padrão é *Justificado*. Se você desejar mudá-lo para *centralizado*, *alinhado à esquerda* ou *alinhado à direita*, seguem os comandos no exemplo abaixo. Você pode selecionar o texto e escolher na barra de ferramentas qual alinhamento você quer aplicar ao mesmo.

**Exemplo 2.1.3.** *Os comandos*

```
\begin{flushright} Alinhado à direita  
\end{flushright}
```

```
\begin{center} Centralizado \end{center}
```

```
\begin{flushleft} Alinhado à esquerda \end{flushleft}
```

*produzem, respectivamente:*

Alinhado à direita

Centralizado

e

Alinhado à esquerda

**Observação 2.1.2.** *Se você desejar que o parágrafo inicie sem espaçamento na margem esquerda, use o comando `\noindent`.*

### 2.1.3 Espaçamento

O espaçamento entre as linhas no documento é especificado pelo comando `\linespread{valor}` no preâmbulo, onde *valor* é o valor numérico real, em relação ao espaçamento normal. Por exemplo, `\linespread{1.5}` produzirá um texto com espaçamento de um e meio e `\linespread{2}` com espaço duplo.

Os comandos `\hspace{medida}` e `\vspace{medida}` inserem um espaço horizontal e vertical, respectivamente, na medida desejada. Por exemplo, `\hspace{10cm}` insere um espaço horizontal de 10cm no texto.

Segue uma tabela de exemplos de uso de alguns comandos relacionados ao espaçamento no texto.

Comando	Display
<code>\$A + \ B\$</code>	$A + B$
<code>\$A + \quad B\$</code>	$A + B$
<code>\$A + \qquad B\$</code>	$A + B$
<code>\$A + \hspace{5cm} B\$</code>	$A + B$
<code>\$A + \phantom{B} + C\$</code>	$A + C$

Tabela 2.3: Alguns Comandos

**Exemplo 2.1.4.** *Vamos produzir o texto abaixo.*

Até os Gênios se Enganam!

Em 1640, **Pierre de Fermat** conjecturou que os números da forma  $F_n = 2^{2^n} + 1, n = 1, 2, 3, \dots$  eram números primos. Mas Fermat foi traído por seus cálculos. Em 1732, **Euler**, com sua usual habilidade em lidar com números muito grandes, mostrou que

$$2^{2^5} + 1 = 6.700.417 * 671.$$

Os números da forma  $2^{2^n} + 1$  ficaram conhecidos como **números de Fermat**, e os números primos desta forma como **primos de Fermat**. Até o momento, mesmo com todo o avanço computacional, não se conseguiu encontrar outros primos de Fermat, além dos cinco primeiros que ele mesmo conhecia.

Mas os números também enganaram Euler e, é claro, enganam muita gente ainda hoje. No caso de Euler, ele conjecturou que, se  $n \geq 3$ , e se  $k$  é um número inteiro positivo, então é necessário, *pelo menos*, a soma de  $n$  n-ésimas potências inteiras  $a_1^n + a_2^n + \dots + a_n^n$  para escrever a potência  $k^n$ .

Em 1966, num artigo do Boletim da Sociedade Matemática Americana [Lander e Parkin, 1966], um simples exemplo põe por terra a conjectura de Euler: os matemáticos L. J. Lander e T. R. Parkin, mostraram que

$$144^5 = 27^5 + 84^5 + 110^5 + 133^5.$$

Dessa história se tira a lição de que, vez em quando, temos também o direito de ousar em nossas opiniões sem ter medo de errar, pois, até os gênios se enganam...

Texto extraído do livro *Um Convite à Matemática*, de Daniel C. M. Filho.

Código Fonte:

```
\begin{center}Até os Gênios se Enganam!
\end{center}
```

```
Em 1640, \textbf{Pierre de Fermat} conjecturou que os números da
```

forma  $F_n = 2^{2^n} + 1$ ,  $n = 1, 2, 3, \dots$  eram números primos. Mas Fermat foi traído por seus cálculos. Em 1732, `\textbf{Euler}`, com sua usual habilidade em lidar com números muito grandes, mostrou que

$$2^{2^5} + 1 = 6.700.417 \cdot 671.$$

Os números da forma  $2^{2^n} + 1$  ficaram conhecidos como `\textbf{números de Fermat}`, e os números primos desta forma como `\textbf{primos de Fermat}`. Até o momento, mesmo com todo o avanço computacional, não se conseguiu encontrar outros primos de Fermat, além dos cinco primeiros que ele mesmo conhecia.

Mas os números também enganaram Euler e, é claro, enganam muita gente ainda hoje. No caso de Euler, ele conjecturou que, se  $n \geq 3$ , e se  $k$  é um número inteiro positivo, então é necessário, `\textit{pelo menos}`, a soma de  $n$   $n$ -ésimas potências inteiras  $a_1^n + a_2^n + \dots + a_n^n$  para escrever a potência  $k^n$ .

Em 1966, num artigo do Boletim da Sociedade Matemática Americana [Lander e Parkin, 1966], um simples exemplo põe por terra a conjectura de Euler: os matemáticos L. J. Lander e T. R. Parkin, mostraram que

$$144^5 = 27^5 + 84^5 + 110^5 + 133^5.$$

Dessa história se tira a lição de que, vez em quando, temos também o direito de ousar em nossas opiniões sem ter medo de errar, pois, até os gênios se enganam...

`\begin{flushright}`Texto extraído do livro `\textit{Um Convite à Matemática}`, de Daniel C. M. Filho.`\end{flushright}`

**Exercício 2.1.1.** *Volte ao exemplo anterior e trabalhe com os comandos `\linespread`, `\hspace`, `\vspace` e `\noindent`.*

### 2.1.4 Listas

Existem três tipos básicos de listas no  $\text{\LaTeX}$ , que são: enumerada, com marcadores e descritiva.

Uma lista enumerada é produzida pelo ambiente `enumerate`. Por exemplo, os comandos abaixo produzem o texto a seguir:

```
\begin{enumerate}
  \item Primeiro item da lista enumerada.
  \item Segundo item da lista enumerada.
  \item Terceiro item da lista enumerada.
\end{enumerate}
```

1. Primeiro item da lista enumerada.
2. Segundo item da lista enumerada.
3. Terceiro item da lista enumerada.

Já o ambiente `itemize` produz uma lista onde os itens são precedidos por marcadores. O código

```
\begin{itemize}
  \item Primeiro item.
  \item Segundo item.
  \item Terceiro item.
\end{itemize}
```

produz

- Primeiro item.
- Segundo item.
- Terceiro item.

O marcador de itens padrão é •, mas ele pode ser configurado de acordo com a sua preferência. O texto:

- Marcador padrão.
- ♣ Marcador personalizado.
- ✠ Outro marcador personalizado.

foi produzido por:

```
\begin{itemize}
  \item Marcador padrão.
  \item [$\clubsuit$] Marcador personalizado.
  \item [$\maltese$] Outro marcador personalizado.
\end{itemize}
```

 foi produzido por:

Uma lista descritiva é uma lista onde cada item é formado pela palavra ou símbolo e na frente uma explicação. É produzida pelo ambiente `description`. Por exemplo, a lista

**enumerada** Cada item recebe uma enumeração.

$\infty$  Infinito.

♠ Espada.

cos Cosseno.

ln Logaritmo Natural.

é produzida por:

```
\begin{description}
  \item [enumerada] Cada item recebe uma enumeração.
  \item [{$\infty$}] Infinito.
  \item [{$\spadesuit$}] Espada.
  \item [{$\cos$}] Cosseno.
  \item [{$\ln$}] Logaritmo Natural.
\end{description}
```

As listas pode ser combinadas como no exemplo a seguir:

1. Este item contém uma lista
  - Primeiro item da sub lista.
  - Segundo item da sublista.
2. (a) Item 1 da segunda sub lista.
  - i. Sub item.
- (b) Item 2 da segunda sub lista

**Exemplo 2.1.5.** *O código fonte abaixo produz o texto a seguir.*

Código Fonte:

```
\begin{center}
Propriedades do Valor Absoluto
\end{center}

\begin{enumerate}
\item Multiplicação:  $|ab| = |a||b|$ 
\item Divisão:  $|\frac{a}{b}| = \frac{|a|}{|b|}$ ,  $b \neq 0$ 
\item Potenciação:  $|a^n| = |a|^n$ 
\item Radiciação:  $|\sqrt{a^2}| = |a|$ 
\end{enumerate}
```



Texto:

### Propriedades do Valor Absoluto

1. Multiplicação:  $|ab| = |a||b|$
2. Divisão:  $|\frac{a}{b}| = \frac{|a|}{|b|}, b \neq 0$
3. Potenciação:  $|a^n| = |a|^n$
4. Radiciação:  $\sqrt{a^2} = |a|$

**Teste 2.1.1.** *Produza o texto abaixo:*

### Uma *Prova* Falaciosa

Consideremos dois números  $a$  e  $b$  com  $a = b$ . Então:

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

\* Onde está o erro?

- ✓ Quando passamos do penúltimo para o último passo, dividimos a equação por  $a^2 - ab$ . Como  $a = b$ , então  $a^2 - ab = 0$  e essa divisão não é permitida.

**Teste 2.1.2.** *Produza o texto a seguir:*

### Propriedades das Funções Seno e Cosseno

1. As funções seno e cosseno têm, ambas, domínio igual ao conjunto dos números reais e imagem  $[-1, 1]$ .
2. Elas são periódicas de período  $2\pi$ , ou seja, para todo inteiro  $k$ :
  - $\cos(x) = \cos(x + 2k\pi)$
  - $\sin(x) = \sin(x + 2k\pi)$
3. A função cosseno é par e a função seno é ímpar.

**Observação 2.1.3.** *Ao usar o menu Formulas - Funções Matemáticas para gerar a função seno obtemos “sin”, porque em inglês a palavra é sine. Quando aprendermos a criar comandos, saberemos como contornar esse problema.*

---

## CAPÍTULO 3

---

# FÓRMULAS MATEMÁTICAS

Já aprendemos no Capítulo 1 que uma fórmula matemática no meio do texto começa e termina com `$` e se quisermos dar destaque à mesma para que ela apareça sozinha na linha e centralizada começamos e terminamos com `$$`. Veremos a seguir alguns elementos que aparecem frequentemente em fórmulas matemáticas.

---

### 3.1 Fracções, Expoentes, Índices, Raízes e Pontos

---

#### 3.1.1 Frações

Para incluir frações no texto, podemos usar os comandos descritos na tabela a seguir:

Comando	Display
<code>a/b</code>	$a/b$
<code>a/(b + c)</code>	$a/(b + c)$
<code>\frac a b</code>	$\frac{a}{b}$
<code>\frac a b+c</code>	$\frac{a}{b} + c$
<code>\frac{a}{b+c}</code>	$\frac{a}{b+c}$

Tabela 3.1: Frações

**Observação 3.1.1.** Note que o comando que aparece nas linhas 3 e 4 da tabela acima serve para digitar apenas frações com um caractere no numerador e um caractere no denominador. Quando não é este o caso, devemos escrever o numerador e o denominador entre chaves, ou seja, usar `\frac{numerador}{denominador}`, como na linha 5. Uma outra opção é o comando `\dfrac{numerador}{denominador}` que pode ser utilizado para que as frações não sejam ajustadas a altura da linha ou para frações contínuas.

### 3.1.2 Expoentes e Índices

O símbolo  $\wedge$  em uma fórmula indica que o próximo caractere é um expoente e  $_$  indica que é um índice. Quando o expoente ou o índice possui mais de um caractere, eles devem ser postos entre chaves. Quando índice e expoente ocorrem juntos, a ordem que digitamos não interfere na saída. Vejamos uma tabela com exemplos de uso.

Comando	Display
<code>5^x</code>	$5^x$
<code>5^{\{2x\}}</code>	$5^{2x}$
<code>5^2x</code>	$5^2x$
<code>a_1</code>	$a_1$
<code>a_{\{13\}}</code>	$a_{13}$
<code>a_13</code>	$a_13$
<code>a_2^5</code>	$a_2^5$
<code>a^5_2</code>	$a_2^5$
<code>a_{\{52\}}^{\{x^2\}}</code>	$a_{52}^{x^2}$

Tabela 3.2: Expoentes e índices

### 3.1.3 Raízes

Produzimos raízes com o comando `\sqrt[n]{radicando}`. Se o argumento `[n]` for omitido, é gerada a raiz quadrada. Vejamos exemplos:

Comando	Display
<code>\sqrt{x^2 + 2}</code>	$\sqrt{x^2 + 2}$
<code>\sqrt[4]{x^2 + 1}</code>	$\sqrt[4]{x^2 + 1}$
<code>\sqrt[n]{9 - x}</code>	$\sqrt[n]{9 - x}$
<code>\sqrt[5]{2x}</code>	$\sqrt[5]{2x}$
<code>\sqrt{2x}</code>	$\sqrt{2x}$

Tabela 3.3: Raízes

### 3.1.4 Pontos

Algumas vezes necessitamos inserir pontos (reticências) em uma fórmula matemática. Os comandos para tal são:

Comando	Display	Descrição
<code>\cdot</code>	$\cdot$	Ponto centralizado
<code>\ldots</code>	$\dots$	Três pontos
<code>\cdots</code>	$\cdots$	Três pontos centralizados
<code>\vdots</code>	$\vdots$	Três pontos na vertical
<code>\ddots</code>	$\ddots$	Três pontos na diagonal

Tabela 3.4: Pontos

**Exemplo 3.1.1.** *A fórmula*

$$(x + a)^n = x^n + na x^{n-1} + \frac{n(n-1)}{2!} a^2 x^{n-2} + \frac{n(n-1)(n-2)}{3!} a^3 x^{n-3} + \cdots + na^{n-1} x + a^n$$

*foi conseguida com*

`$$ (x + a)^n = x^n + na x^{n-1} + \frac{n(n-1)}{2!} a^2 x^{n-2} + \frac{n(n-1)(n-2)}{3!} a^3 x^{n-3} + \cdots + na^{n-1} x + a^n $$`

### 3.2 Textos Dentro de Fórmulas

Para inserir textos dentro de fórmulas matemáticas, usamos o comando `\text{}` ou `\mbox{}`. A tabela abaixo ilustra o uso desse comando.

Comando	Display
<code>x_1 = \frac{42}{6} e x_2=-7</code>	$x_1 = \frac{42}{6} e x_2 = -7$
<code>x_1 = \frac{42}{6} \mbox{ e } x_2=-7</code>	$x_1 = \frac{42}{6} \text{ e } x_2 = -7$
<code>x_1 = \frac{42}{6} \quad \mbox{e} \quad x_2=-7</code>	$x_1 = \frac{42}{6} \text{ e } x_2 = -7$

Tabela 3.5: Texto em Fórmulas

**Exemplo 3.2.1.** *O texto*

Se denotarmos

$$P = \{x \in U; P(x) \text{ é válida}\},$$

então  $\exists x \in U$  tal que  $P(x)$  vale acarreta  $P \neq \emptyset$ .  
foi produzido pelo código:

Se denotarmos `$$P = \{x \in U; P(x) \mbox{ é válida}\},$$` então  
`$\exists x \in U$ \textit{tal que} $P(x)$ \textit{vale} acarreta $P \neq \emptyset.$`

### 3.3 Delimitadores

O tamanho do delimitador(chaves, colchetes, parênteses) pode ser ajustado automaticamente para caber a fórmula digitada. Para isso, digitamos o comando `\left` à esquerda e `\right` à direita, seguidos dos delimitadores desejados. O ponto pode ser usado como delimitador branco, que pode ser útil quando queremos usar delimitador apenas de um lado da expressão. Por exemplo, o código

```
$$x_1 = (\frac{42}{6}), x_1 = \left(\frac{42}{6}\right), x_1 = \left[\frac{42}{6}\right. \\ \left.\right] \text{ e } \left.\frac{x^2}{6}\right|_1^3$$
```

produz

$$x_1 = \left(\frac{42}{6}\right), x_1 = \left(\frac{42}{6}\right), x_1 = \left[\frac{42}{6}\right] \text{ e } \left.\frac{x^2}{6}\right|_1^3$$

**Observação 3.3.1.** Para usar chaves como delimitadores, usamos o comando `\{`.

### 3.4 Alguns Símbolos Muito Usados

Alguns símbolos, como  $>$ ,  $<$ ,  $+$ ,  $-$  e  $=$  podem ser digitados diretamente do teclado. Outros não. Segue uma tabela com alguns símbolos que necessitam de comandos e aparecem com frequência:

Símbolo	Comando	Símbolo	Comando	Símbolo	Comando
$\leq$	<code>\le</code>	$\sim$	<code>\sim</code>	$\infty$	<code>\infty</code>
$\geq$	<code>\ge</code>	$\neq$	<code>\neq</code>	$\forall$	<code>\forall</code>
$\subset$	<code>\subset</code>	$\approx$	<code>\approx</code>	$\exists$	<code>\exists</code>
$\subseteq$	<code>\subseteq</code>	$\cong$	<code>\cong</code>	$\nexists$	<code>\nexists</code>
$\supset$	<code>\supset</code>	$\simeq$	<code>\simeq</code>	$\cap$	<code>\cap</code>
$\supseteq$	<code>\supseteq</code>	$\equiv$	<code>\equiv</code>	$\cup$	<code>\cup</code>
$\in$	<code>\in</code>	$\cong$	<code>\cong</code>	$\bigcap$	<code>\bigcap</code>
$\notin$	<code>\notin</code>	$\perp$	<code>\perp</code>	$\bigcup$	<code>\bigcup</code>
$\nless$	<code>\nless</code>	$\times$	<code>\times</code>	$\pm$	<code>\pm</code>
$\ngtr$	<code>\ngtr</code>	$\emptyset$	<code>\emptyset</code>	$\mp$	<code>\mp</code>

Tabela 3.6: Símbolos

**Teste 3.4.1.** *Produza o seguinte texto:*

### Sentenças Equivalentes

*Exemplo de uma Sentença Enunciada de Quatro Maneiras Diferentes*

Dois números complexos são raízes da equação  $ax^2 + bx + c = 0, a \neq 0$  se, e somente se, um deles for  $\frac{-b+\sqrt{b^2-4ac}}{2a}$  e o outro for  $\frac{-b-\sqrt{b^2-4ac}}{2a}$ .

Poderíamos escrever:

Uma condição necessária e suficiente para que dois números complexos sejam raízes da equação  $ax^2 + bx + c = 0, a \neq 0$  é que um deles seja  $\frac{-b+\sqrt{b^2-4ac}}{2a}$  e o outro seja  $\frac{-b-\sqrt{b^2-4ac}}{2a}$ .

Usando a linguagem dos conjuntos:

Se  $R = \{\text{raízes complexas da equação } ax^2 + bx + c = 0, a \neq 0\}$  e  $S = \left\{ \frac{-b+\sqrt{b^2-4ac}}{2a}, \frac{-b-\sqrt{b^2-4ac}}{2a} \right\}$ , então  $S = R$ . Observe que  $S \subset R$  e  $R \subset S$ .

Ou ainda:

As condições abaixo são equivalentes:

- i) Dois números complexos  $x_1$  e  $x_2$  são raízes da equação  $ax^2 + bx + c = 0, a \neq 0$ ;
- ii) Um dos números complexos  $x_1$  ou  $x_2$  é igual a  $\frac{-b+\sqrt{b^2-4ac}}{2a}$  e o outro é igual a  $\frac{-b-\sqrt{b^2-4ac}}{2a}$ .

Texto extraído do livro *Um Convite à Matemática*, de Daniel C. M. Filho.

Continuando nosso estudo de fórmulas matemáticas, aprenderemos a mudar a fonte na fórmula, criar matrizes, escrever somatórios, etc. Alguns pacotes fundamentais para a produção de textos matemáticos com qualidade são distribuídos pela AMS - American Mathematical Society. Começaremos a aula incluindo no preâmbulo do documento os comandos `\usepackage{amssymb}` e `\usepackage{amsmath}`. O primeiro permite usar várias fontes e símbolos matemáticos adicionais e o segundo permite incrementar ambientes matemáticos.

## 3.5 Fontes e Símbolos

### 3.5.1 Fontes Especiais no Modo Matemático

Para especificar a fonte dentro de uma fórmula matemática, usamos os comandos listados na tabela a seguir:

A fonte usada para representar os conjuntos numéricos através de letras é a fonte `\mathbb{}`. Assim, `$$\mathbb{R}, \mathbb{N}, \mathbb{C}$$` produz

$$\mathbb{R}, \mathbb{N}, \mathbb{C}.$$

### 3.5.2 Símbolos e Fórmulas em Negrito ou com Contornos

Para que um símbolo apareça em negrito, usamos o comando `\boldsymbol{símbolo}` ou `\pmb{símbolo}`.

Comando	Display	Tipo de Fonte
<code>\mathrm{ax \cong b (mod m)}</code>	$ax \equiv b(\text{mod } m)$	romano reto
<code>\mathsf{ax \equiv b (mod\ m)}</code>	$ax \equiv b(\text{mod } m)$	sem enfeites nas pontas da letras
<code>\mathtt{ax \equiv b (mod\ m)}</code>	$ax \equiv b(\text{mod } m)$	fonte de máquina de escrever
<code>\mathbf{ax \equiv b (mod\ m)}</code>	$\mathbf{ax \equiv b(mod\ m)}$	negrito
<code>\mathit{ax \equiv b (mod\ m)}</code>	$ax \equiv b(mod\ m)$	itálico
<code>\mathnormal{ax \equiv b (mod\ m)}</code>	$ax \equiv b(mod\ m)$	normal
<code>\mathcal{AX \equiv B (MOD\ M)}</code>	$\mathcal{AX} \equiv \mathcal{B}(\mathcal{MOD\ M})$	caligráfica - só funciona para letras maiúsculas

Tabela 3.7: Fontes no Modo Matemático

Para deixar uma fórmula matemática em negrito, usamos o comando `\mathversion{bold}`. Todas as fórmulas escritas a partir daí ficarão em negrito. Para que as fórmulas voltem a aparecer sem negrito, usamos o comando `\mathversion{normal}`.

**Exemplo 3.5.1.** *O texto abaixo foi produzido com o código-fonte a seguir:*

$$\Delta y = f(x + \Delta x) - f(x)$$

$$f(x) = (x - 2)^{\frac{2}{3}}$$

$$f(x) \leq f(c), \forall x \in \mathbb{R}$$

Código-Fonte:

```
\mathversion{bold}
$$\Delta y = f(x + \Delta x) - f(x)$$
$$f(x) = (x - 2)^{\frac{2}{3}}$$
\mathversion{normal}
$$f(x) \pmb{\leq} f(c), \ \ \forall \ \text{forall} \ x \ \text{in} \ \mathbb{R}$$
```

Para colocar contorno na fórmula usamos o comando `\boxed{}`.

Por exemplo, `\boxed{x^2 + y^2 = z^2}` produz  $\boxed{x^2 + y^2 = z^2}$ .

**Exercício 3.5.1.** *Produza o texto abaixo:*

O conjunto de todos os números  $x$  que satisfazem  $a < x < b$  é chamado de **intervalo aberto** e denotado por  $(a, b)$ . Assim,

$$(a, b) = \{x \in \mathbb{R} | a < x < b\}$$

O **intervalo fechado** de  $a$  até  $b$  é o intervalo  $(a, b)$  mais os dois extremos  $a$  e  $b$  e é denotado por  $[a, b]$ . Logo,

$$[a, b] = \{x \in \mathbb{R} | a \leq x \leq b\}$$

### 3.5.3 Funções

Nomes provenientes de abreviaturas, tais como nomes de funções (trigonométricas, por exemplo), devem ser escritas com a fonte `\mathrm{}` - romano reto. No `TeXNic`, no Menu *Math - Functions*, existem macros pré-definidas para grande parte das funções conhecidas. Assim, para escrever  $\cos x$ , por exemplo, não é necessário digitar `\mathrm{cos} x`, basta ir até o menu citado acima e clicar na função cosseno, ou digitar diretamente o comando `\cos x`.

**Exemplo 3.5.2.** Os comandos `\ln(x)`, `\max \{f(x), g(x)\}`, `\log_5(x)` geram:

$$\ln(x), \max\{f(x), g(x)\}, \log_5(x).$$

**Observação 3.5.1.** Algumas funções, como o seno, por exemplo, têm abreviaturas diferentes em inglês e em português. Assim, não podemos fazer com o seno o que citamos acima com o cosseno, pois o comando vai aparecer  $\sin x$ . Aprenderemos depois como definir novos comandos e criaremos um para a função seno. Por enquanto, se necessário, usaremos `\matrm{sen}`.

---

## 3.6 Somatórios, Produtórios, Limites, Derivadas e Integrais

---

### 3.6.1 Somatórios e Produtórios

O comando que gera o símbolo de somatório é `\sum` e um produtório é gerado com `\prod`. Inserimos os limitantes usando `_` e `^`, os mesmos símbolos usados para gerar índices e expoentes. Existem quatro estilos de aparência de um somatório ou produtório. Vejamos os exemplos:



Comando	Display
<code>\sum_{i=1}^n \left(\frac{1}{n}\right)</code>	$\sum_{i=1}^n \left(\frac{1}{n}\right)$
<code>\sum\limits_{i=1}^n \left(\frac{1}{n}\right)</code>	$\sum_{i=1}^n \left(\frac{1}{n}\right)$
<code>\displaystyle \sum_{i=1}^n \left(\frac{1}{n}\right)</code>	$\sum_{i=1}^n \left(\frac{1}{n}\right)$
<code>\scriptstyle \sum_{i=1}^n \left(\frac{1}{n}\right)</code>	$\sum_{i=1}^n \left(\frac{1}{n}\right)$
<code>\prod_{i=1}^n \left(\frac{1}{n}\right)</code>	$\prod_{i=1}^n \left(\frac{1}{n}\right)$
<code>\prod\limits_{i=1}^n \left(\frac{1}{n}\right)</code>	$\prod_{i=1}^n \left(\frac{1}{n}\right)$
<code>\displaystyle \prod_{i=1}^n \left(\frac{1}{n}\right)</code>	$\prod_{i=1}^n \left(\frac{1}{n}\right)$
<code>\scriptstyle \prod_{i=1}^n \left(\frac{1}{n}\right)</code>	$\prod_{i=1}^n \left(\frac{1}{n}\right)$

Tabela 3.8: Somatórios e Produtórios

**Observação 3.6.1.** O modo `\displaystyle` é como aparece quando a fórmula está entre duas cifras.

### 3.6.2 Limites, Derivadas e Integrais

Geramos limites com o comando `\lim` e integrais com o comando `\int`. A inserção de limitantes na integral segue a mesma lógica da inserção dos mesmos no somatório. Para limites, em geral usamos `\lim_{x \rightarrow a}`. A notação de derivada é conseguida simplesmente colocando uma aspa '. Segue uma tabela de exemplos.

Comando	Display
<code>\lim_{x \rightarrow 2} 2x</code>	$\lim_{x \rightarrow 2} 2x$
<code>\lim\limits_{x \rightarrow 2} 2x</code>	$\lim_{x \rightarrow 2} 2x$
<code>\displaystyle \lim_{x \rightarrow 2} 2x</code>	$\lim_{x \rightarrow 2} 2x$
<code>\scriptstyle \lim_{x \rightarrow 2} 2x</code>	$\lim_{x \rightarrow 2} 2x$
<code>\int_{2}^5 x^2 dx</code>	$\int_2^5 x^2 dx$
<code>\int\limits_{2}^5 x^2 dx</code>	$\int_2^5 x^2 dx$
<code>\displaystyle \int_{2}^5 x^2 dx</code>	$\int_2^5 x^2 dx$
<code>\scriptstyle \int_{2}^5 x^2 dx</code>	$\int_2^5 x^2 dx$
<code>f'(x) = 5x^3</code>	$f'(x) = 5x^3$

Tabela 3.9: Limites, Derivadas e Integrais

### 3.7 Matrizes

Para inserir uma matriz usamos o ambiente `array`. As colunas são separadas pelo símbolo `&` e as linhas por `\\`. É necessário dizer qual será o alinhamento das colunas, por exemplo, começar uma matriz com `\begin{array}{clrc}` diz que a matriz tem 4 colunas, onde a primeira e a última são centralizadas, a segunda é alinhada à esquerda (left) e a terceira à direita (right). Lembre-se que uma matriz é um objeto matemático, logo deve vir entre cifrões. Para inserir delimitadores(chaves, parênteses, etc), usamos os comandos `\left` antes de `\begin{array}` e `\right` depois de `\end{array}`, seguidos do delimitador desejado.

**Exemplo 3.7.1.** *A matriz*

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 12 & 356 & 275 & 43 & 57 & 1042 \end{bmatrix}$$

foi produzida por:

```

 $\left[\begin{array}{ccclrr}$ 
1 & 2 & 3 & 4 & 5 & 6 \\
12 & 356 & 275 & 43 & 57 & 1042
\end{array}\right]

```

**Teste 3.7.1.** *Produza o seguinte texto:*

#### Definição da Integral Definida

Se  $f$  for uma função definida no intervalo fechado  $[a, b]$ , então a **integral definida** de  $f$  de  $a$  até  $b$ , denotada por  $\int_a^b f(x)dx$ , será dada por:

$$\int_a^b f(x)dx = \lim_{\|\Delta\| \rightarrow 0} \sum_{i=1}^n f(c_i)\Delta x_i,$$

se esse limite existir.

#### Teorema Fundamental do Cálculo

Seja  $f$  uma função contínua no intervalo fechado  $[a, b]$ , e seja  $x \in [a, b]$ . Se  $F$  for a função definida por

$$F(x) = \int_a^x f(t)dt$$

então

$$F'(x) = f(x).$$

#### Um Exemplo de Rotação

A rotação de ângulo  $\theta$  no  $\mathbb{R}^3$ , tendo como eixo fixo o eixo  $z$  é o operador cuja matriz na base canônica é:

$$\begin{pmatrix} \cos \theta & \operatorname{sen} \theta & 0 \\ -\operatorname{sen} \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

---

# CAPÍTULO 4

---

## CLASSES DE DOCUMENTOS. LAYOUT

Hoje aprenderemos um pouco mais sobre os tipos de documentos gerados pelo  $\text{\LaTeX}$  e sobre o *layout* dos mesmos.

---

### 4.1 Classes de Documentos e Opções

---

Já aprendemos no Capítulo 1 que na primeira linha do preâmbulo dos nossos documentos em  $\text{\LaTeX}$  temos o comando `\documentclass{<estilo>}`. Vejamos um pouco mais sobre as opções para *estilo*.

article	artigo
book	livro
letter	carta
report	relatório
slides	transparências
proc	artigo para anais de congresso
amsart	artigo da AMS
amsbook	livro da AMS
amsproc	artigo para anais de congresso da AMS

Tabela 4.1: Opções para Estilo do Documento

O que aparece entre colchetes logo depois do comando `\documentclass` são as opções do documento, que podem especificar tamanho do papel, tamanho base das letras, orientação do papel,

etc. Estas opções podem ser colocadas separadas por vírgula: Veja a seguir a descrição de algumas opções:

- Tamanho das letras: 10pt (padrão), 11pt ou 12pt. O estilo slides usa um tamanho fixo de aproximadamente 20pt.
- Tipo de papel: letterpaper (padrão), legalaper, executivepaper, a4paper, b4paper, b5paper.
- Orientação do papel: report (padrão) ou landscape(paisagem)
- Número de colunas: onecolumn (padrão), twocolumn.
- Uso do papel: oneside (um lado. Padrão para article e report), ou twoside (frente/verso. Padrão para book).
- Alinhamento de equações: fleqn (equações alinhadas à esquerda). O padrão é centralizada.
- Enumeração de equações: leqno (enumeração de equação à esquerda). O padrão é na direita.

**Observação 4.1.1.** *Os dois últimos itens acima referem-se ao ambiente `equation`, que aprenderemos na próxima aula.*

**Observação 4.1.2.** *Como já sabemos do Capítulo 1, as opções para o documento não são obrigatórias. Caso não as especifiquemos, serão usadas todas as opções padrão para o estilo do documento que escolhermos.*

**Exemplo 4.1.1.** *Digite o seguinte código-fonte, salve e compile para ver a saída.*

```
\documentclass[a4paper,twocolumn,12pt]{book}
\usepackage[latin1]{inputenc}
\usepackage{amssymb,amsmath}

\setlength{\textwidth}{15cm}    % LARGURA DO TEXTO
\setlength{\textheight}{22cm}   % ALTURA DO TEXTO

\begin{document}
```

Dizemos que um ponto  $x_0$  é `\emph{ponto de mínimo absoluto}` de uma função  $f$  se  $f(x_0) \leq f(x)$ , para todo  $x$  no domínio de  $f$ . Neste caso, dizemos que  $f(x_0)$  é o `\emph{valor mínimo}`. Analogamente, dizemos que um ponto  $x_0$  é `\emph{ponto de máximo absoluto}` de uma função  $f$  se  $f(x_0) \geq f(x)$ , para todo  $x$  no domínio de  $f$ . Neste caso, dizemos que  $f(x_0)$  é o `\emph{valor máximo}`. Os pontos de máximo e mínimo absolutos também são chamados extremos absolutos de  $f$ .

```
\end{document}
```

**Exercício 4.1.1.** *Volte ao exemplo anterior e modifique ou inclua algumas opções.*

**Exercício 4.1.2.** *Mude o estilo do exemplo anterior para slides, compile e veja a saída.*

---

## 4.2 Layout do Documento

---

### 4.2.1 Parágrafo e Espaçamentos. Quebras

O primeiro parágrafo do capítulo ou seção não costuma ser indentado (empurrado para direita), pois não há necessidade de distinguir do parágrafo anterior. Do segundo parágrafo em diante são indentadas, pois tem a necessidade de distinguir do parágrafo anterior. Quando desejarmos que o parágrafo não seja empurrado para a direita, usamos antes dele o comando `\noindent`.

A quebra de linha ou espaço em branco extra (mais de um espaço) são ignorados, mas a linha em branco é interpretada como mudança de parágrafo. Porém, mais de uma linha em branco são interpretadas como um único parágrafo e não como vários parágrafos.

Se desejar pular várias linhas usando parágrafo em seguida, use linhas em branco, intercalado com `\`. Por exemplo,

```
\  
  
\  
  
\
```

significa que foram dados três parágrafos seguidos.

A quebra de linhas é feito pelo `\newline` ou `\\` para terminar a linha, mas para que o alinhamento fique justificado, usa-se o `\linebreak`. A quebra de página é feita pelo `\newpage`.

### 4.2.2 Caracteres Especiais

O Apóstrofo é aberto com crase e fechado com apóstrofo. Já as aspas, são abertas com duas crases seguidas e fechadas com dois apóstrofes seguidos. No caso das aspas citado acima, o LaTeX troca a sequência de caracteres duas crases ou dois apóstrofes por um novo caracter. Isto é denominado de “ligadura”. Outros exemplos de ligadura são: `--` que é travessão, `---` que é travessão longo, `?‘` e `!‘` que são ponto de interrogação e exclamação de *ponta cabeça*.

Os seguintes caracteres especiais são conseguidos colocando-se `\` antes deles: `$`, `#`, `%`, `&`, `_`, `{`, `}`. Caracteres especiais podem ser escritos no texto: `\$`, `\#`, `\%`, `\&`, `\_`, `\{`, `\}`.

Os logotipos `TEX`, `LATEX`, etc também podem ser inseridos no documento, mas lembre-se que os comandos do T<sub>E</sub>X são sensíveis ao maiúsculo/minúsculo, ou seja, para conseguir T<sub>E</sub>X ou L<sup>A</sup>T<sub>E</sub>X devemos digitar `\TeX` ou `\LaTeX`.

Em geral, os espaços depois do comando são ignorados. Para inserir espaços depois do comando, basta colocar um par de chaves ou `\` após o comando. Por exemplo, para produzir L<sup>A</sup>T<sub>E</sub>X sim, escreva `\LaTeX{} sim` ou `\LaTeX\ sim`.

**Exemplo 4.2.1.** *O código-fonte*

O `\LaTeX` é um pacote de macros do `\TeX` que permite ao autor diagramar e imprimir seus trabalhos como documento de altíssima qualidade tipográfica, usando `\textit{layout}` profissional predefinido. O `\LaTeX` pode ser usado para produzir todos os tipos de documentos, desde uma simples carta até livros completos.

A versão atual do `\LaTeX` é a  $2\epsilon$  e a do `\TeX` é  $3,14159$  e está convergindo para  $\pi$ . O projeto `\LaTeX` 3 está em andamento.

*Produz*

O `\LaTeX` é um pacote de macros do `\TeX` que permite ao autor diagramar e imprimir seus trabalhos como documento de altíssima qualidade tipográfica, usando *layout* profissional predefinido. O `\LaTeX` pode ser usado para produzir todos os tipos de documentos, desde uma simples carta até livros completos.

A versão atual do `\LaTeX` é a  $2\epsilon$  e a do `\TeX` é  $3,14159$  e está convergindo para  $\pi$ . O projeto `\LaTeX` 3 está em andamento.

---

## 4.3 Ambientes Básicos

---

Uma parte importante dos textos em `\LaTeX` é constituída por ambientes (em inglês, *environments*). Em geral um ambiente começa com o comando `\begin{<nome do ambiente>}` e termina com `\end{<nome do ambiente>}`. Exemplos de ambientes que já foram usados aqui no curso são: *array*(criar matrizes), *center*(centralizar o texto), *enumerate*(criar uma lista enumerada) e *itemize*(criar uma lista com itens). Hoje aprenderemos a construir tabelas e a trabalhar com o ambiente *equation*.

---

## 4.4 Tabelas

---

Uma tabela no modo texto é produzida pelo ambiente `tabular` e a tabela no modo matemático é produzida pelo ambiente `array`, que já estudamos. Esses dois ambientes apresentam a mesma sintaxe e mesma funcionalidade.

O argumento obrigatório desses ambientes é o “alinhamento” das colunas que deve ser especificado com *l*(à esquerda - left), *c*(centralizado - center), *r*(à direita - right) ou *p*{largura}(Nesta opção, o texto fica justificado e a coluna fica com a largura especificada. A largura deve conter a unidade de medida, como por exemplo, 10cm). Para traçar uma linha vertical entre colunas ou na borda, usa-se o `|`. Cada coluna é separada por `&`, e a mudança de linha é feito pelo `\\`. Para traçar uma linha horizontal, usa-se o comando `\hline`. Vejamos um exemplo:

**Exemplo 4.4.1.** *Digite o seguinte código-fonte, compile e veja a saída.*

```
\begin{tabular}{|l|c|r|p{5cm}|}
\hline
1 & 2 & 3 & 4\\ \hline
11 & 12 & 13 & 14\\ \hline
111 & 112 & 113 & 114 \\ \hline
\end{tabular}
```

A saída é:

1	2	3	4
11	12	13	14
111	112	113	114

#### 4.4.1 Juntando Colunas

Para juntar mais de uma célula, usa-se o comando `multicolumn`. A sintaxe desse comando é `multicolumn{n}{formato}{texto}`, onde **n** é o número de células a serem mescladas, **formato** é o alinhamento e as bordas verticais da célula e o terceiro argumento é o texto que será colocado na célula.

No exemplo abaixo, usamos o `multicolumn` para juntar as cinco colunas da primeira linha, centralizando os dados e traçando linhas verticais antes e depois da célula.

**Exemplo 4.4.2.** *O código-fonte abaixo produz a tabela a seguir:*

```
\begin{tabular}{|c|c|c|c|c|}
\hline
\multicolumn{5}{|c|}{Horário da Monitoria} \\ \hline
SEG & TER & QUA & QUI & SEX \\ \hline
12:00 & 14:00 & 11:30 & 17:30 & 18:00 \\ \hline
\end{tabular}
```

Horário da Monitoria				
SEG	TER	QUA	QUI	SEX
12:00	14:00	11:30	17:30	18:00

#### 4.4.2 Linhas Múltiplas e Omissão de Linhas

Para traçar mais de uma linha vertical nas bordas das células, basta usarmos o `|` tantas vezes quanto o número de linhas que queremos, e para omitir a linha vertical basta deixarmos sem `|` quando escolhermos a formatação da coluna. Analogamente, para traçar mais de uma linha horizontal, usamos o comando `\hline` no final da linha quantas vezes desejarmos e se omitirmos esse comando, tal linha ficará sem traçado embaixo. Vejamos um exemplo:

**Exemplo 4.4.3.** *Digite o código-fonte abaixo, compile e veja a saída.*



```
\begin{tabular}{|c||c|||c||c|||c|}
\hline
\multicolumn{5}{|c|}{Horário da Monitoria} \\ \hline \hline
SEG & TER & QUA & QUI & SEX \\ \hline
12:00 & 14:00 & 11:30 & 17:30 & 18:00 \\ \hline
\end{tabular}
```

A saída do código-fonte acima é:

Horário da Monitoria				
SEG	TER	QUA	QUI	SEX
12:00	14:00	11:30	17:30	18:00

**Observação 4.4.1.** Note que a primeira linha da tabela acima ficou com a borda à direita. Isso acontece porque a formatação desta linha é dada pelo segundo argumento do comando `\multicolumn`, independente do que está escrito lá em cima, logo depois do comando `\begin{tabular}`.

#### 4.4.3 Igualando Largura das Colunas

Usando o pacote `tabularx`, podemos criar tabela com larguras de colunas específicas igualadas de forma automática.

O ambiente oferecido por este pacote é `tabularx`, onde o primeiro argumento é a largura da tabela e o segundo é a especificação de formatação das colunas, mas apresenta um especificador de coluna especial: “X”. Todas as colunas especificadas por “X” terão a mesma largura. Coloque o comando `\usepackage{tabularx}` no preâmbulo do seu documento, e digite o seguinte código-fonte, cuja saída está logo a seguir:

```
\begin{tabularx}{15cm}{|X|X|c|X|c|}
\hline \multicolumn{5}{|c|}{Horário da Monitoria} \\ \hline
SEG & TER & QUA & QUI & SEX \\ \hline
12:00 & 14:00 & 11:30 & 17:30 & 18:00 \\ \hline
\end{tabularx}
```

Horário da Monitoria				
SEG	TER	QUA	QUI	SEX
12:00	14:00	11:30	17:30	18:00

**Observação 4.4.2.** Para centralizar a tabela no texto, usamos o ambiente `center`, ou seja, `\begin{center}` antes da tabela e `\end{center}` depois da tabela.

**Exercício 4.4.1.** Produza a seguinte tabela:

	$f'(x)$	Conclusão
$0 < x < 12$	+	Crescente
$x = 12$	0	Máximo Local
$x > 12$	-	Decrescente

**Observação 4.4.3.** Quando precisamos elaborar uma tabela grande, é necessário que a mesma seja quebrada em mais de uma página. Usando o pacote `longtable`, o ambiente `longtable` permite criar tabelas longas que podem ser quebradas em várias páginas. Não entraremos em detalhes sobre isso agora.

---

## 4.5 Ambiente `equation`

---

Uma fórmula com enumeração automática é criada pelo ambiente `equation` como segue:

$$h(x) = \frac{x^2 + 3}{x - 1} \quad (4.1)$$

Em geral, usamos a equação enumerada quando pretendemos citá-la mais à frente no texto. Isto é denominado de “referência cruzada”. O comando para referência cruzada é o par `\label{}` e `\ref{}`. O `\label{nome}` armazena enumeração numa variável “nome” que pode ser referenciado pelo comando `\ref{}`.

Vejamos a seguir um exemplo de referência cruzada:

A solução completa da equação de segundo grau  $ax^2 + bx + c = 0$  com  $a \neq 0$  é dado por

$$x = \frac{-b \pm \sqrt{\Delta}}{2a} \quad (4.2)$$

onde  $\Delta = b^2 - 4ac$ . A equação 4.2 é conhecida como fórmula de Baskara.

Isto foi conseguido com o Código-fonte:

A solução completa da equação de segundo grau  $ax^2+bx+c=0$  com  $a \neq 0$  é dado por

```
\begin{equation} \label{eq:baskara}
  x=\frac{-b\pm\sqrt{\Delta}}{2a}
\end{equation}
```

onde  $\Delta = b^2-4ac$ . A equação `\ref{eq:baskara}` é conhecida como fórmula de Baskara.

Quando fazemos referência a elementos que ficam longe do texto, às vezes é necessário especificarmos a página onde ele está para ajudar o leitor. Para referenciar a página armazenada no `\label`, usa-se o comando `\pageref{}` de forma similar a `\ref{}`. Por exemplo, se quisermos citar:

Pela equação 4.2 da página 34, fica evidente que a equação  $x^2 + x + 1 = 0$  não tem raiz real. Então devemos digitar:

Pela equação `\ref{eq:baskara}` da página `\pageref{eq:baskara}`, fica evidente que a equação  $x^2+x+1=0$  não tem raiz real.

**Observação 4.5.1.** Quando existe referência cruzada no documento, é necessário executar o  $\LaTeX$  duas vezes em seguida, para obtermos números corretos na referência.

**Teste 4.5.1.** *Produza o seguinte texto: (OBS: A numeração das equações no seu texto ficará diferente da numeração aqui.)*

Considere a equação de demanda:

$$p^2 + 2x - 16 = 0. \quad (4.3)$$

Como em situações econômicas normais as variáveis  $x$  e  $p$  são não negativas, quando (4.3) é resolvida para  $p$  em função de  $x$ , rejeitamos os valores negativos de  $p$ , obtendo

$$p = \sqrt{16 - 2x}. \quad (4.4)$$

Assim a função preço para a equação de demanda (4.3) é a função  $f$  para a qual  $f(x) = \sqrt{16 - 2x}$ .

Tabela de Valores				
x	0	1	2	3
f(x)	4	$\sqrt{14}$	$\sqrt{12}$	$\sqrt{10}$

## 4.6 Ambiente Minipage

O ambiente `minipage` é usado para criar uma “caixa”. O parâmetro obrigatório é a largura da “caixa”, que pode ser colocada com a unidade (por exemplo: 5cm) ou em relação à largura total do texto (por exemplo: `0.7\textwidth`, que cria a caixa com 70% da largura total do texto). Usando esse ambiente, podemos, entre outras coisas, colocar um bloco de texto centralizado ou justificado no lado direito da página. Também é possível colocar equações, fórmulas e outros elementos sem problemas dentro do `minipage`.

**Exemplo 4.6.1.** *Criando uma caixa com 60% da largura total do texto e centralizada:*

Código-Fonte:

```
\begin{center}
\begin{minipage}{0.6\textwidth}
Para exprimir que a sequência crescente  $\{a_n\}$  é ilimitada superiormente,
escrevemos  $\lim_{x \rightarrow \infty} a_n = \infty$  e dizemos que  $a_n$ 
tende ao infinito quando  $n$  cresce indefinidamente.
\end{minipage}
\end{center}
```

Saída:

Para exprimir que a sequência crescente  $\{a_n\}$  é ilimitada superiormente, escrevemos

$$\lim_{x \rightarrow \infty} a_n = \infty$$

e dizemos que  $a_n$  tende ao infinito quando  $n$  cresce indefinidamente.

**Exemplo 4.6.2.** *Mesmo texto anterior com 12cm de largura e alinhado à direita:*

Código-Fonte:

```
\begin{flushright}
\begin{minipage}{12cm}
Para exprimir que a sequência crescente  $\{a_n\}$  é ilimitada superiormente,
escrevemos  $\lim_{x \rightarrow \infty} a_n = \infty$  e dizemos que  $a_n$ 
tende ao infinito quando  $n$  cresce indefinidamente.
\end{minipage}
\end{flushright}
```

Saída:

Para exprimir que a sequência crescente  $\{a_n\}$  é ilimitada superiormente,  
escrevemos

$$\lim_{x \rightarrow \infty} a_n = \infty$$

e dizemos que  $a_n$  tende ao infinito quando  $n$  cresce indefinidamente.

O ambiente `minipage` é importante para colocar um parágrafo de texto dentro dos comandos que não aceitam parágrafos. Por exemplo, `\fbox{}` coloca moldura no elemento, mas não aceita parágrafos de texto. Então, como colocar moldura no texto com várias linhas? É simples: colocar tudo no `minipage` e colocar dentro do `fbox`.

**Exemplo 4.6.3.** *Para incluir moldura no texto anterior, combinamos o `minipage` com o comando `\fbox`:*

Código-Fonte:

```
\begin{flushright}
\fbox{\begin{minipage}{12cm}
Para exprimir que a sequência crescente  $\{a_n\}$  é ilimitada superiormente,
escrevemos  $\lim_{x \rightarrow \infty} a_n = \infty$  e dizemos que  $a_n$ 
tende ao infinito quando  $n$  cresce indefinidamente.
\end{minipage}}
\end{flushright}
```

Saída:

Para exprimir que a sequência crescente  $\{a_n\}$  é ilimitada superiormente,  
escrevemos

$$\lim_{x \rightarrow \infty} a_n = \infty$$

e dizemos que  $a_n$  tende ao infinito quando  $n$  cresce indefinidamente.

---

## 4.7 Ambiente Quote e similares

---

O ambiente `quote` é usado para escrever citações, exemplos e frases importantes. Por exemplo, Hamlet disse:

“Ser ou não ser, eis a questão.”

foi criado com:

Hamlet disse:

```
\begin{quote}
‘‘Ser ou não ser, eis a questão.’’\end{quote}
```

Quando a porção de texto citada é grande, pode ser útil que os parágrafos sejam *indentados* (tabulados para direita). Neste caso, usamos o ambiente `quotation` ao invés de `quote`. Por exemplo, vejamos uma explicação de como funcionam os ambientes similares a `quote`.

**Exemplo 4.7.1.** *Uma explicação sobre ambientes similares a `quote`.*

```
\begin{quotation}
‘‘Existem dois ambientes similares: os ambientes \verb"quotation" e
\verb"verse". Como o ambiente \verb"quotation" faz a indentação dos
parágrafos, ele é usado para citações longas que se estendem por
vários parágrafos. O ambiente \verb"verse" é usado em poemas onde
as quebras de linhas são importantes. As linhas são separadas por
\verb+\\\+ e por uma linha em branco no fim de cada verso’’.
\end{quotation}
```

“Existem dois ambientes similares: os ambientes `quotation` e `verse`. Como o ambiente `quotation` faz a indentação dos parágrafos, ele é usado para citações longas que se estendem por vários parágrafos. O ambiente `verse` é usado em poemas onde as quebras de linhas são importantes. As linhas são separadas por `\\` e por uma linha em branco no fim de cada verso”.

No caso de verso, a linha que for quebrado pela falta de espaços, mas que constitui a mesma linha de cima, será tabulada para a direita. O ambiente `verse` se encarrega deste serviço.

Vejamos um exemplo:

**Exemplo 4.7.2.** *No Meio do Caminho - Carlos Drummond de Andrade*

No meio do caminho tinha uma pedra  
tinha uma pedra no meio do caminho  
tinha uma pedra  
no meio do caminho tinha uma pedra.

Nunca me esquecerei desse aconteci-  
mento  
na vida de minhas retinas tão fatiga-  
das.  
Nunca me esquecerei que no meio do  
caminho  
tinha uma pedra  
tinha uma pedra no meio do caminho  
no meio do caminho tinha uma pedra.

foi produzido pelo código-fonte:

```
\begin{minipage}{0.5\textwidth}  
\begin{verse}  
No meio do caminho tinha uma pedra \\  
tinha uma pedra no meio do caminho \\  
tinha uma pedra\ \  
no meio do caminho tinha uma pedra.\ \  
\   
  
Nunca me esquecerei desse acontecimento\ \  
na vida de minhas retinas tão fatigadas.\ \  
Nunca me esquecerei que no meio do caminho\ \  
tinha uma pedra\ \  
tinha uma pedra no meio do caminho\ \  
no meio do caminho tinha uma pedra.  
\end{verse}  
\end{minipage}
```

---

## 4.8 Ambiente Verbatim

---

O ambiente `verbatim` é utilizado para inserir o texto de jeito que digitamos. Existem duas opções: os ambientes `verbatim` e `verbatim*`. No primeiro caso, o espaço é considerado e no segundo, é colocado um carácter `␣` no lugar de cada espaço. Vejamos a diferença entre o código colocado pelo `verbatim` e `verbatim*`:

**Exemplo 4.8.1.** *Diferença entre `verbatim` e `verbatim*`:*

Código-Fonte:

```
\begin{verbatim}
\begin{quote}
texto citado, texto importante ou exemplos.
\end{quote}

\begin{verbatim*}
\begin{quote}
texto citado, texto    importante ou exemplos.
\end{quote}
\end{verbatim*}
```

Saída:

```
\begin{quote}
texto citado, texto importante ou exemplos.
\end{quote}

\begin{quote}
texto citado, texto importanteo ou exemplos.
\end{quote}
```

**Teste 4.8.1.** *Produza o texto abaixo: (Extraído da RPM 64 - Pg. 35)*

Lendo o artigo citado em [1], encontrei um problema chinês proposto há cerca de 2000 anos:

*Encontrar o raio da circunferência inscrita em um triângulo retângulo de hipotenusa  $a$  e catetos  $b$  e  $c$ .*

A solução dada pelos antigos escribas chineses é

$$r = \frac{bc}{(a + b + c)}.$$

Surgiu então a idéia de propor a problema a estudantes. Começamos então propondo a estudantes de 16 a 17 anos a seguinte variante do problema:

$ABC$  é um triângulo com  $AB = 3$ ,  $BC = 5$  e  $CA = 4$ . Achar o raio da circunferência inscrita no triângulo  $ABC$ .

---

# CAPÍTULO 5

---

## FIGURAS E ALGUMAS DICAS

---

---

### 5.1 Inserindo Figuras no Texto

---

#### 5.1.1 Figuras Flutuantes

Em  $\text{\LaTeX}$ , um elemento flutuante é aquele que não precisa ser colocado obrigatoriamente na posição “digitada”, mas que será colocado na melhor posição possível em termos de aparência do documento.

Uma figura flutuante é especificada pelo ambiente `figure`. O primeiro parâmetro deste ambiente é a ordem em que o  $\text{\LaTeX}$  tenta inserir a figura.

<b>h</b>	Onde foi digitado
<b>b</b>	Na parte inferior da página
<b>t</b>	Na parte superior da página
<b>p</b>	Página separada.
<b>!</b>	Ignorar a restrição de espaçamento.

Tabela 5.1: Parâmetros do Ambiente `figure`

No ambiente flutuante, o  $\text{\LaTeX}$  tentará as posições seguindo a lista de especificação dos parâmetros e inserirá na primeira posição que satisfizer a exigência. Caso nenhuma posição seja conveniente, o  $\text{\LaTeX}$  criará uma página separada especialmente para ele.

#### 5.1.2 Imagens Geradas pelo $\text{\LaTeX}$

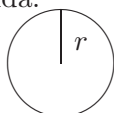
Figuras mais simples podem ser geradas usando os comandos do  $\text{\LaTeX}$  através do ambiente `picture`, mas para as mais complexas, é aconselhável desenhar no editor gráfico apropriado e inserir. O exemplo mostra o caso simples de elemento gráfico usando ambiente `picture`. Não nos aprofundaremos neste assunto aqui.



**Exemplo 5.1.1.** *Código-Fonte:*

```
\begin{picture}(40,40)
\put(20,20){\circle{40}}
\put(20,20){\line(0,1){20}}
\put(25,25){$r$}
\end{picture}
```

Saída:



### 5.1.3 Imagens Geradas por Editores Gráficos Externos

Para inserir imagens externas, é necessário usar o pacote **graphicx**.

O  $\text{\TeX}$  suporta oficialmente o formato EPS (Encapsulated Post Script) que é o formato reduzido do PS. Dependendo do  $\text{\LaTeX}$ , podem ser suportadas imagens bitmap.

Vamos gerar um documento com a imagem `distancia.eps`. A figura deve estar salva na mesma pasta do documento onde a mesma será inserida.

O código-fonte é:

```
\begin{figure}[hp]
\center
\includegraphics[width=0.5\textwidth]{distancia}
\caption{Distância}
\end{figure}
```

No exemplo acima, a largura da imagem foi ajustada para metade da largura do texto. Para controlar o tamanho e rotação da figura, usamos parâmetros opcionais no `includegraphics`.

<b>width</b>	Largura da imagem
<b>height</b>	Altura da imagem
<b>scale</b>	Ampliação
<b>angle</b>	Rotação - em graus
<b>keepaspectratio</b>	Mantém a proporção quando <b>height</b> e <b>width</b> forem especificados simultaneamente (usado sem o valor).

Tabela 5.2: Parâmetros Adicionais para Figuras

No caso de imagem bitmap, é obrigatório especificar o *height* e *width* por não conseguirmos obter seu tamanho real. O `keepaspectratio` só funciona para EPS.

**Exemplo 5.1.2.** *Digite o seguinte código-fonte, compile e veja a saída:*

```

\begin{figure}[hp]
\center
\includegraphics[width=0.5\textwidth,angle=30,scale=0.7]{distancia}
\caption{Distância}
\end{figure}

```

Saída:

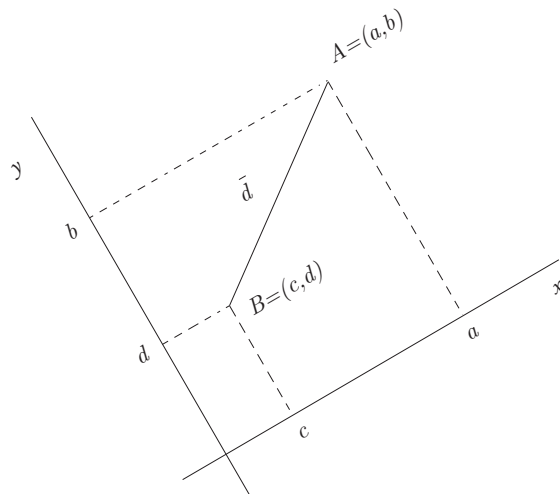


Figura 5.1: Distância

**Exercício 5.1.1.** *No exemplo anterior, inclua, nos parâmetros, um valor para altura e o `keepaspectratio`. Modifique os valores e veja o que acontece.*

Os parâmetros do comando serão aplicados na ordem. Assim, definir a altura e rotacionar 90 graus é diferente de rotacionar 90 graus e definir a altura.

Usando o `minipage`, podemos colocar duas figuras, lado a lado no exemplo a seguir:

**Exemplo 5.1.3.** *Figuras Lado a Lado:*

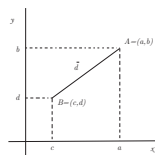


Figura 5.2: Distância

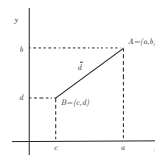


Figura 5.3: Distância

Código-Fonte:

```

\begin{figure}[hbp]
\center
\begin{minipage}{0.4\textwidth}
\center
\includegraphics[width=0.3\textwidth]{distancia}
\caption{Distância} \label{fig:lado1}
\end{minipage}
\begin{minipage}{0.4\textwidth}
\center
\includegraphics[width=0.3\textwidth]{distancia}
\caption{Distância} \label{fig:lado2}
\end{minipage}
\end{figure}

```

**Observação 5.1.1.** *Uma figura pode ser colocada ao lado de um texto se acrescentarmos no preâmbulo o comando `\usepackage{wrapfig}`. Neste caso, a inserção da figura deve ser feita no ambiente `wrapfigure` que admite vários parâmetros:*

```

\begin{wrapfigure}[nLinhas]{posicionamento}{largura}
...
\end{wrapfigure}

```

onde:

*nLinhas* = valor que define o numero de linhas do texto

*posicionamento* = *r*, *l*, *i* ou *o* para direita, esquerda, dentro e fora da página, respectivamente.

Código-Fonte:

```

\begin{wrapfigure}[6]{r}{4cm}
\centering
\includegraphics[width=4cm]{conicas.eps}
\caption{Figura ao lado de texto ({\sl wrapfigure})}
\label{fig:texto}
\end{wrapfigure}

```

**Observação 5.1.2.** *Existem programas que convertem imagens para ps ou eps, como por exemplo o Corel Draw, o jpeg2ps, o Image Magick e o GIMP.*

## 5.2 Empilhando Símbolos

O comando `stackrel` permite que símbolos sejam empilhados, conforme os exemplos a seguir:

$$f(x) \stackrel{\text{def}}{=} x^5 + x + 1$$

$$X \stackrel{f}{\rightarrow} Y$$

Código-Fonte:

```


$$f(x) \stackrel{\text{def}}{=} x^5 + x + 1$$


$$X \stackrel{f}{\longrightarrow} Y$$


```

### 5.3 Hifenização

No caso do  $\text{\LaTeX}$  não separar corretamente as sílabas de uma palavra (por exemplo, escrevendo *exe-* no final de uma linha e *mplo* no início da outra linha), pode-se “ensinar” facilmente como fazer a separação de sílabas corretamente. Para isso, basta colocar os possíveis hífenos corretos na palavra, cada um precedido de uma barra invertida. Neste caso, os hífenos que não forem realmente necessários não serão impressos. Por exemplo, se for necessário “ensinar” ao  $\text{\LaTeX}$  a separar corretamente as sílabas das palavras *matemática* e *guerreiro* basta colocar no texto `ma\te\má\ti\ca` e `guer\rei\ro`. Outra opção é acrescentar no preâmbulo, o comando

```
\hyphenation{ma-te-má-ti-ca guer-rei-ro}
```

onde as palavras são separadas apenas por espaços em branco e cada palavra traz todos os possíveis hífenos (sem barra invertida).

### 5.4 Dicionário

Para instalar um dicionário e baixe o arquivo .dic em <http://wiki.services.openoffice.org/wiki/Dictionaries>. Depois abre-se o menu Opções-¿Configurar Texmaker-¿Editor. Lá pode-se indicar a localização do dicionário que se pretende utilizar (verifique a pasta onde você salvou o arquivo).

**Observação 5.4.1.** *o arquivo pt\_PT.aff deve estar também junto do arquivo pt\_PT.dic.*

A partir de agora a base de trabalho ficou mais sólida e pode-se começar a produzir documentos com probabilidade de erros menor.

---

# CAPÍTULO 6

---

## BEAMER (SLIDES)

Beamer é uma classe de documentos  $\text{\LaTeX}$  criada para gerar apresentações com slides. O Beamer não é primeira classe desenvolvida para essa finalidade, mas atualmente é a mais utilizada. Com o Beamer, você define os slides (que são chamados de frames) e o conteúdo deles além de algumas configurações básicas da sua apresentação.

A primeira coisa a ser feita é declarar a classe de documento de um arquivo `.tex` como sendo Beamer, com o comando `\documentclass{beamer}`. Em seguida, você declara os pacotes que serão usados no seu arquivo.

```
\documentclass{beamer}
```

Agora ele está preparado para criar slides. O slide mais simples possível, pode ser criado com os comandos

```
\documentclass{beamer}
\begin{document}
\frame{Coloque aqui o texto do slide}
\end{document}
```

Alguns tipos de apresentações de slides estão prontas e podem ser utilizadas através do comando `\usetheme{coloque aqui o nome da apresentação ou tema}`

que deve ser digitado no preâmbulo. Esses tipos de apresentações são chamadas de temas. Seu formato, sua cor e outros aspectos já estão estabelecidos.

Dentro desse último comando é necessário colocar o nome dos temas. Os temas mais comuns são chamados por nomes de cidades:

*Antibes, Bergen, Berkeley, Berlin Boadilla, Copenhagen, Darmstadt, Dresden, Frankfurt, Goettingen, Hannover, Ilmenau, Juanlespins, Madrid, Malmoe, Montpellier, Pittsburgh, Rochester e Singapore.*

Existe um comando que permite modificar a cor de um tema:

```
\usecolortheme{escreva aqui o nome da cor}
```

Os nomes de temas de cores mais comuns são:

*albatross, crane, beetle, dove, fly, seagull, wolverine e beaver.*

Observe agora o tema Madrid. No fim de cada slide aparece um local para colocar outras informações como nome de autor, data, título, instituto.

Os comandos necessários para definir essas informacoes devem ser colocados no preâmbulo. Os comandos são os seguintes:

```
\title {escreva aqui seu título}  
\author {escreva aqui seu nome}  
\institute{escreva aqui o nome da sua faculdade}
```

Em outros temas a data não aparece no fim do slide, ela so aparece na folha de rosto dos slides. E a folha de rosto so aparece dentro do slide (dentro do frame) que possui o comando

```
\titlepage.
```

Se você quiser colocar um título em apenas um slide, você pode usar o comando

```
\frametitle{título do slide}
```

dentro de cada slide.

E possível modificar as cores do título, da data, do instituto, do título do slide com os seguintes comandos:

```
\setbeamercolor{normal text}{fg=black}  
\setbeamercolor{alerted text}{fg=red}  
\setbeamercolor{author}{fg=blue}  
\setbeamercolor{institute}{fg=gray}  
\setbeamercolor{date}{fg=gray}  
\setbeamercolor{frametitle}{fg=red}
```

Criação de blocos:

E possível dentro de cada slide(cada frame) criar blocos que destaquem parte do texto, por exemplo o enunciado de um teorema.

Devemos utilizar o seguinte comando:

```
\begin{block}{Título opcional}  
Conteúdo do bloco.  
\end{block}
```

**Enumeração:**

Se desejarmos enumerar certos itens dentro do nosso slide, devemos proceder como faríamos no latex:

```
\begin{enumerate}[i]
\item Primeiro tópico
\item Segundo tópico
\item Terceiro tópico
\end{enumerate}
```

```
\begin{document}
\begin{frame}
\includegraphics[scale=x]{imagem.jpg}
\end{frame}
```

```
\begin{frame}
\begin{itemize}
\item Item antes da pausa
15
\pause
\item Item depois da pausa
\end{itemize}
\end{frame}
```

```
\begin{itemize}
\item<1->Classe
\item<2->Sobreposições
\item<3->Transições
\end{itemize}
```

**Exemplo 6.1.** *Vamos criar uma estrutura básica:*

```
\documentclass{beamer}
\begin{document}
\begin{frame}
Texto do slide 1.
\end{frame}
\begin{frame}
Texto do slide 2.
\end{frame}
```

```
\end{document}
```

Agora que a construção de um slide básico já foi demonstrada, a estrutura de seções e subseções será mostrada, para que você organize melhor a sua apresentação.

**Exemplo 6.2.** `\documentclass{beamer}`

```
\begin{document}
\section{}
\begin{frame}
Slide 1.
\end{frame}
\begin{frame}
Slide 2.
\end{frame}
\section{Título opcional}
\begin{frame}
Slide 3.
\end{frame}
\end{document}
```

Existem uma série de opções e comandos que podem ser utilizados mas que estão além dos nossos objetivos. Mais informações podem ser obtidas nas referências que citamos no final desta apostila.

Vamos destacar os comandos:

```
\setbeamercolor{palette primary}{fg=cor1,bg=cor2}
\logo{\includegraphics[scale=x]{imagem.png}}
\setbeamertemplate{navigation symbols}{}
\setbeamertemplate{footline}[page number]
```

que tem as seguintes funções respectivamente:

```
%muda a cor da caixa com o titulo na abertura
%inclui uma imagem no canto inferior direito da apresentacao
% retira os simbolos de navegacao dos slides
%insere no canto inferior direito um contador de slides
```

**Teste 6.0.1.** *Produza uma apresentação sobre um tema de sua escolha (mínimo de 8 slides) utilizando o arquivo **slide.tex** disponível em **www.germano.prof.ufu.br***



---

---

# CAPÍTULO 7

---

## CRIANDO COMANDOS E AMBIENTES

---

---

### 7.1 Criando Comandos

---

A sintaxe para a criação de um novo comando é:

```
\newcommand{comando}[num parametros][valor opcional]{definicao}
```

No nome do comando, não podemos usar números, delimitadores, espaços ou caracteres especiais, e devemos sempre lembrar que o  $\text{\LaTeX}$  distingue letras maiúsculas de minúsculas.

#### 7.1.1 Comandos sem Parâmetros

São os tipos de comandos mais simples, como nomes de funções. Sabemos que existem várias funções pré-definidas no  $\text{\LaTeX}$ , mas as siglas das mesmas são em inglês, que nem sempre coincidem com as siglas em português. Um exemplo clássico disto é a função seno, cuja abreviatura em inglês é *sin*. Se quisermos inserir a função seno em uma equação, por exemplo, podemos digitar  $\text{\textbackslashmathrm{sen}}$  ou criar um comando para a mesma. Assim, se inserirmos no nosso texto

```
\newcommand{\sen}{\mathrm{sen}},
```

toda vez que digitarmos  $\text{\textbackslashsen}$ , o programa entenderá  $\text{\mathrm{sen}}$ .

Como no exemplo acima, sempre que quisermos criar um comando para nomes provenientes de abreviaturas, devemos usar a fonte romano reto.

**Exemplo 7.1.1.** *Abra um arquivo, digite os comandos abaixo e compile:*

```
\newcommand{\tg}{\mathrm{tg}}
\newcommand{\e}{\mathrm{e}}
```

```


$$\sec^2(x) - \tan^2(x) = 1$$


$$e^{x+y} = e^x e^y$$


```

**Observação 7.1.1.** Quando queremos definir um comando que envolve fórmulas matemáticas, podemos colocar o cifrão já na definição do comando ou só quando formos usá-lo no texto. Por exemplo, se definirmos `\newcommand{\soma}{\mathrm{x + y}}`, quando quisermos usar este comando não precisamos colocá-lo entre cifrões. Basta digitar `\soma` ao invés de  `$\soma$`.

### 7.1.2 Comandos com Parâmetros

Esses tipos de comandos têm uma sintaxe mais sofisticada que o tipo anterior. Devemos indicar qual o número de parâmetros do comando e especificá-los pelo símbolo # seguido da sua posição. Assim, #1, #2, #3 especificam o primeiro, segundo e terceiro parâmetros. Vejamos o exemplo:

**Exemplo 7.1.2.** Criaremos um comando para soma com 01 parâmetro. Digite o código-fonte abaixo e compile:

```

\newcommand{\norma}[1]{\|{#1}\|}

$$\|x + y\| \leq \|x\| + \|y\|$$


```

**Exemplo 7.1.3.** Vamos criar agora um comando com dois parâmetros para digitarmos n-uplas. Analisemos a sintaxe abaixo:

```

\newcommand{\nupla}[2]{({#1}_{#1}, \ldots, {#1}_{#2})}

```

Assim, o comando `\nupla` terá dois parâmetros: o primeiro é o “nome” do elemento da lista e o segundo parâmetro é o último índice que aparecerá na n-upla. Defina o comando digitando a linha acima e depois digite `\nupla{x}{j}` para criar  $(x_1, \dots, x_j)$

**Exemplo 7.1.4.** O comando que criaremos a seguir (com quatro parâmetros) nos permitirá escrever subtração de frações de maneira mais curta.

```

\newcommand{\fr}[4]{\frac{#1}{#2} - \frac{#3}{#4}}

$$\frac{3}{2} - \frac{7}{5}$$


```

**Observação 7.1.2.** Alguns comandos têm parâmetros opcionais, como por exemplo o comando `\sqrt`, que pode ser `\sqrt{x}` ou `\sqrt[n]{x}`. O que vem entre [ ] é parâmetro opcional. Um comando pode ter até 9 parâmetros, mas somente um - o primeiro deles - pode ser definido como opcional.

### 7.1.3 Comandos Duais

Comandos duais são aqueles que funcionam tanto no modo matemático quanto no modo texto. Comandos desse tipo são conseguidos utilizando-se o `\ensuremath`.

**Exemplo 7.1.5.** Digite o código-fonte abaixo e compile:

```

\newcommand{\Rea}{\ensuremath{\mathbf{R}}}
\newcommand{\Rac}{\ensuremath{\mathbf{Q}}}

$$\Rea, \Rac$$


```

### 7.1.4 Redefinindo Comandos

Um comando já existente pode ser redefinido pelo `\renewcommand` que tem a mesma sintaxe do `\newcommand`. Por exemplo, `\newcommand{\sin}{\mathrm{sen}}` redefine o `\sin` para produzir `sen` ao invés de `sin`. Entretanto, não é aconselhável fazer isso, já que com essa ação destruimos o comando em inglês e algumas vezes precisamos misturar inglês com português, o que requer o uso tanto do `sin` como `sen`.

Assim, a melhor coisa a fazer é criar o `\sen` em vez de redefinir o `\sin`.

O comando `\providecommand`, que tem a mesma sintaxe do `newcommand` e do `renewcommand`, é usado para criar comandos apenas se os mesmos não existirem, ou seja, ele é ignorado caso o comando que estamos definindo já exista. Assim, `\providecommand{\sen}{\mathrm{sen}}` funciona da seguinte forma: caso não exista o `\sen`, define-o como `\mathrm{sen}`. Caso exista, mantém como está.

### 7.1.5 O Comando *def*

Também podemos criar novos comandos através do `\def`, que é mais curto do que `\newcommand` e `\renewcommand`, mas o `\def` não checa se o comando que queremos criar já existe ou não. Por exemplo, se tentarmos criar comandos para produzir conjuntos de números e os nomearmos de real, racional, inteiro, natural, etc, o `\newcommand` avisa que o `\natural` já existe, evitando que tal comando seja destruído. Caso estejamos cientes do que o comando já existe, e desejemos redefiní-lo, usaremos `\renewcommand`, que também avisará caso tal comando ainda não exista.

Como `\def` não oferece este tipo de recurso, é aconselhável evitá-lo, a não ser que o tenhamos um bom domínio de  $\text{\LaTeX}$ .

O `\def` é útil quando precisamos definir o comando, independente de existir ou não. Vejamos, no exemplo abaixo, como usar o `\def`.

**Exemplo 7.1.6.** *Digite o código-fonte abaixo e compile:*

```
\def\arctg{\mathrm{arctg}}
\def\dys{\displaystyle}
$\arctg{x}$
$\int_{a}^{b} \ln(x)dx$
$\dys{\int_{a}^{b} \ln(x)dx}$
```

**Teste 7.1.1.** *Crie um comando (chame-o de *baskara*) que permita-nos escrever as raízes de uma equação do segundo grau de maneira mais curta, isto é, um comando para que quando digitemos*

`$$\baskara{a}{b}{c}$$`

*tenhamos como saída*

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

---

## 7.2 Criando Ambientes

---

Uma grande parte dos textos em L<sup>A</sup>T<sub>E</sub>X é constituída de ambientes, e já estamos acostumados a trabalhar com vários deles: *array* (criar matrizes), *center* (centralizar o texto), *equation* (criar equações numeradas), *enumerate* (criar uma lista enumerada), *tabular* (criar tabelas), só para citar alguns exemplos.

Um ambiente é uma configuração local, iniciado pelo `\begin{<nome do ambiente>}` e encerrado por `\end{<nome do ambiente>}`, ou seja, tudo que é definido pelo ambiente vale para o texto que está entre `\begin{<nome do ambiente>}` e `\end{<nome do ambiente>}`.

A sintaxe do comando para a criação de novos ambientes é:

```
\newenvironment{ambiente}[número de parâmetros][valor do opcional]
{início do ambiente}{fim do ambiente}.
```

Vamos entender melhor como criar novos ambientes. Só para constar, *environment* é ambiente em inglês.

### 7.2.1 Ambientes sem Argumentos

São os tipos de ambientes mais simples. Como exemplo, vamos criar um ambiente que alinhe o texto à direita e o deixe em negrito.

**Exemplo 7.2.1.** *Vamos digitar e compilar o seguinte código:*

```
\newenvironment{zezinho}{\begin{flushright}\bf}{\end{flushright}}
\begin{zezinho} Em um triângulo isósceles os ângulos da base são
congruentes.
\end{zezinho}
```

**Exemplo 7.2.2.** *Criemos agora um ambiente que centraliza e deixa o texto com letra grande.*

```
\newenvironment{joaozinho}{\begin{center}\Huge}{\end{center}}
\begin{joaozinho} Em um triângulo isósceles os ângulos da base são
congruentes.
\end{joaozinho}
```

**Exercício 7.2.1.** *Crie um novo ambiente, com o seu nome, que deixe o texto alinhado à esquerda e com letra pequena. (use o comando `\tiny` para a letra pequena.)*

### 7.2.2 Ambientes com Argumentos

Assim como os comandos, os novos ambientes podem ter até 9 parâmetros ou argumentos, sendo que o primeiro deles pode ser opcional.

Vamos criar um ambiente para escrever demonstrações, que coloca automaticamente um quadradinho preto no final. Para tal, usaremos o comando `\rule{largura}{altura}`, que produz

um retângulo preto com a largura e altura desejadas, que devem ser acompanhadas da unidade. Aqui podemos usar *cm* ou a unidade *ex*, que é aproximadamente o tamanho de um *x* na fonte usada.

**Exemplo 7.2.3.** *Digite e compile:*

```
\newenvironment{dem}[1][Demonstração]{\textbf{\textit{\#1:}\ }}{\rule{1ex}{1ex}}
\begin{dem} Sejam $ABC$ e $DEF$ triângulos tais que ...
\end{dem}
```

**Exemplo 7.2.4.** *Voltando ao exemplo anterior, se quisermos que o quadradinho fique no final da linha, usamos o comando \hfill.*

```
\newenvironment{demo}[1][Demonstração]{\textbf{\textit{\#1:}\ }}{\hfill\rule{0.2cm}{0.2cm}}
\begin{demo} Sejam $ABC$ e $DEF$ triângulos tais que ...
\end{demo}
```

Nos dois exemplos acima, o parâmetro é opcional. Caso não coloquemos nada (como fizemos), virá a palavra Demonstração. Podemos mudar isso.

**Exemplo 7.2.5.** *Digite:*

```
\begin{dem}[Prova] Sejam $ABC$ e $DEF$ triângulos tais que ...
\end{dem}
```

Vamos criar agora um ambiente com dois parâmetros, que são textos que ficarão dentro de molduras, centralizadas e com espaço horizontal de 5cm entre elas.

**Exemplo 7.2.6.** *Digite e compile:*

```
\newenvironment{caixinhas}[2]{\begin{center}\fbox{\#1}\hspace{5cm}\fbox{\#2}}
{\end{center}}
\begin{caixinhas}{Estudar...}{Estudar Muito!}
\end{caixinhas}
```

### 7.2.3 Redefinindo Ambientes

Podemos redefinir um ambiente usando o comando `\renewenvironment`, que tem a mesma sintaxe do `\newenvironment`. Como exemplo, vamos redefinir o ambiente `zezinho`, que criamos no começo da aula, para que ele fique em itálico e não em negrito.

**Exemplo 7.2.7.** *Vamos mudar o zezinho:*

```
\renewenvironment{zezinho}{\begin{flushright}\it}{\end{flushright}}
\begin{zezinho} Em um triângulo isósceles os ângulos da base são
congruentes.
\end{zezinho}
```

**Teste 7.2.1.** *Crie um ambiente (com dois parâmetros) que deixe o texto em negrito, a primeira parte alinhada à esquerda e a segunda alinhada à direita, com um espaço vertical de 3cm entre elas.*

---

# CAPÍTULO 8

---

## DIVIDINDO O DOCUMENTO

---

Para a elaboração de um documento muito extenso, é interessante dividi-lo em vários arquivos menores, para facilitar a sua manipulação. Vejamos como fazê-lo.

---

### 8.1 Arquivo Mestre

---

Quando dividimos o arquivo fonte em vários arquivos, temos um arquivo principal e as outras partes do documento. O arquivo principal ou arquivo mestre contém as informações que colocamos no preâmbulo dos documentos e a sequência de inclusão dos arquivos que contêm as partes do documentos.

Para incluir um arquivo podemos usar o comando `\input{nomedoarquivo.tex}` ou `\include{nomedoarquivo.tex}`. Uma das diferenças entre esses dois comandos é que o `\include` muda de página entre um arquivo e outro, enquanto o `\input` não muda. Mais à frente destacaremos outras diferenças. Quando usamos um desses comando, é como se copiássemos e colássemos naquela posição o conteúdo do arquivo incluso. Um exemplo de arquivo principal é dado abaixo:

**Exemplo 8.1.1.** *Um arquivo principal.*

```
\documentclass[12pt,a4paper,oneside,titlepage]{book}
\usepackage[latin1]{inputenc}
\usepackage[brazil]{babel}
\usepackage{amssymb,amsmath}
\setlength{\textwidth}{15cm} % LARGURA DO TEXTO
\setlength{\textheight}{22cm} % ALTURA DO TEXTO

\begin{document}

\tableofcontents % Comando para fabricar o índice
```

```
\include{cap1}  
\include{cap2}  
\include{cap3}  
\include{cap4}  
  
\end{document}
```

Isso significa que temos quatro arquivos salvos como cap1, cap2, cap3 e cap4, todos com extensão .tex. Nosso arquivo de saída conterá os quatro. Para produzir o documento, salvamos os arquivos inclusos e compilamos apenas o arquivo mestre.

Os arquivos que compõem as partes do documento (cap1, cap2, cap3 e cap4 no exemplo acima) não contêm preâmbulo, `\begin{document}`, etc. Neles se inclui apenas o corpo do documento-texto, fórmulas - tudo que digitamos entre `\begin{document}` e `\end{document}`.

Em geral, quando escrevemos livros, relatórios, apostilas, dissertações, criamos um arquivo para cada capítulo. Neste caso, todos os arquivos devem começar com o comando `\chapter`.

**Exercício 8.1.1.** *Crie um “livro” de integrais com 3 capítulos em arquivos separados. Capítulo 1 : Integrais Simples, Capítulo 2 : Integrais Duplas e Capítulo 3: Integrais Triplas.*

---

## 8.2 Restringindo Arquivos na Saída

---

À medida que formos incluindo outros arquivos no nosso arquivo principal, pode ser interessante excluirmos alguns arquivos da saída, para que a compilação seja mais rápida. Por exemplo, suponhamos que temos prontos os capítulos 1 e 2 de um livro e estamos digitando o capítulo 3. Quando compilarmos o arquivo mestre, não é necessário compilar os capítulos 1 e 2, pois já os temos concluídos.

Para restringir a saída, temos duas opções: a primeira é simplesmente comentar (com o símbolo de porcentagem) o arquivo que não queremos compilar no momento. Depois, para voltar a compilar esse arquivo, basta tirarmos o símbolo de porcentagem. Vejamos o exemplo:

**Exemplo 8.2.1.** *No exemplo abaixo, quando rodarmos o arquivo principal, não serão levados em consideração os capítulos 1 e 3, apenas o 2 e o 4.*

```
\documentclass[12pt,a4paper,oneside,titlepage]{book}  
\usepackage[latin1]{inputenc}  
\usepackage[brazil]{babel}  
\usepackage{amssymb,amsmath}  
\setlength{\textwidth}{15cm} % LARGURA DO TEXTO  
\setlength{\textheight}{22cm} % ALTURA DO TEXTO  
  
\begin{document}
```

```
\tableofcontents % Comando para fabricar o índice
%\include{cap1}
\include{cap2}
%\include{cap3}
\include{cap4}

\end{document}
```

**Exercício 8.2.1.** *Volte ao livro que você criou e exclua da compilação o capítulo de integrais duplas.*

A desvantagem de usar o método acima para restringir arquivos na saída é que perdemos informações dos mesmos, como sumário, referências, etc. Por exemplo, no ex. (8.2.1), o cap2 será numerado como capítulo e 1 e o cap4 será o capítulo 2. Deve ter acontecido algo parecido no seu último exercício. Para evitar isso, podemos colocar no préambulo do arquivo principal o seguinte comando `\includeonly{arquivos}`, sendo que os nomes dos arquivos devem ser separados por vírgula e não devem conter a extensão `.tex`.

**Observação 8.2.1.** *Quando usamos o comando `\input` ao invés do `\include`, não é possível fazer o procedimento acima. Essa é uma diferença entre esses dois comandos. Outra diferença é que o arquivo dentro do `\include` não pode conter outra inclusão, e o de dentro do `\input` pode. Além disso, o `\include` ignora quando não encontra o arquivo citado(referenciado), enquanto o `\input` acusa erro.*

**Exemplo 8.2.2.** *Podemos suprimir os capítulos 1 e 3 sem mudar a numeração dos mesmos.*

```
\documentclass[12pt,a4paper,oneside,titlepage]{book}
\usepackage[latin1]{inputenc}
\usepackage[brazil]{babel}
\usepackage{amssymb,amsmath}
\setlength{\textwidth}{15cm} % LARGURA DO TEXTO
\setlength{\textheight}{22cm} % ALTURA DO TEXTO

\includeonly{cap2,cap4}

\begin{document}

\tableofcontents % Comando para fabricar o índice
\include{cap1}
\include{cap2}
\include{cap3}
\include{cap4}

\end{document}
```



**Exemplo 8.2.3.** *No livro que você criou, exclua o capítulo 2 sem mudar a numeração dos demais.*

**Teste 8.2.1.** *Crie um livro à sua escolha com 5 capítulos em artigos separados.*

## 8.3 Dividindo um Capítulo

Já sabemos como dividir um texto em vários capítulos. Agora suponhamos que um capítulo seja longo e que desejamos dividir tal capítulo em dois ou mais arquivos. Neste caso, devemos lembrar que o comando `\include` efetua a mudança de página, o que poderá não ser interessante quando queremos dividir um mesmo capítulo em vários arquivos diferentes. Assim, se queremos dividir um capítulo em pedaços, devemos colocar cada seção do capítulo em um arquivo e usar o comando `\input`. O arquivo que não é a parte inicial do capítulo deve começar com `\section` e não com `\chapter`.

**Exemplo 8.3.1.** *Salve os arquivos `cap1`, `cap1a`, `cap1b`, `cap2`, e `livro` abaixo e a seguir compile o arquivo `livro`.*

### Arquivo cap1

```
\chapter{Equações Diferenciais Ordinárias}
```

Uma equação diferencial ordinária [EDO] é uma equação do tipo  $f(x, y, y', y'', y''', \dots, y^{(n)}) = 0$ , onde  $y$  é uma função desconhecida da variável independente  $x$ .

### Arquivo cap1a

```
\section{EDO de 1ª Ordem}
```

Aqui estudaremos problemas do tipo  $y' = f(x, y)$ ,  $y(x_0) = y_0$ .

### Arquivo cap1b

```
\section{EDO de 2ª Ordem}
```

Aqui estudaremos problemas do tipo  $y'' + p(t)y' + q(t)y = g(t)$ , onde  $p, q$  e  $g$  são contínuas em um intervalo  $I$ .

### Arquivo cap2

```
\chapter{Equações Diferenciais Parciais}
```

Uma equação diferencial parcial [EDP] é uma equação que envolve uma função de duas ou mais variáveis e pelo menos uma de suas derivadas parciais.

### Arquivo livro

```
\documentclass[12pt,a4paper,oneside,titlepage]{book}
\usepackage[latin1]{inputenc}
\usepackage[brazil]{babel}
\usepackage{amssymb,amsmath}
\setlength{\textwidth}{15cm} % LARGURA DO TEXTO
\setlength{\textheight}{22cm} % ALTURA DO TEXTO

\begin{document}

\tableofcontents % Comando para fabricar o índice
\include{cap1}
\include{cap1a}
\include{cap1b}
\include{cap2}

\end{document}
```

**Exemplo 8.3.2.** *Volte ao exemplo anterior, troque os comandos `\include` por `\input` e observe a diferença no arquivo de saída.*

---

## 8.4 Capítulos ou Seções com Título Longo

---

Quando o título do capítulo ou da seção é muito longo, podemos usar um título abreviado, colocado entre colchetes antes do título completo. Este título abreviado será usado no sumário e nos cabeçalhos.

**Exemplo 8.4.1.** *No exemplo anterior, volte ao arquivo **cap1** e troque a primeira linha por `\chapter[EDO]{Equações Diferenciais Ordinárias}`. Veja o que acontece no sumário e no cabeçalho.*

**Observação 8.4.1.** *No exercício acima, no arquivo mestre deveremos usar o comando `\include` e não `\input` para que mudemos de página a cada seção e apareçam os cabeçalhos.*

---

# CAPÍTULO 9

---

## BIBLIOGRAFIA E INDICE REMISSIVO

---

---

### 9.1 Referências Bibliográficas

---

Existem duas maneiras de produzirmos referências bibliográficas: através do BibTeX ou por meio do ambiente `thebibliography`. A primeira maneira consiste em criar um banco de dados com todas as referências bibliográficas que forem sendo usadas. Depois, é só citar qualquer uma das referências que faça parte do banco de dados. Mas isso não é uma coisa trivial de ser feita. Aqui aprenderemos apenas a usar o `thebibliography`.

#### 9.1.1 O Ambiente `thebibliography`

Quando usamos o ambiente `thebibliography` para criarmos as referências bibliográficas, devemos fazer manualmente a verificação das citações, assim como a ordenação dos itens, que deve ser feita em ordem alfabética dos nomes dos autores.

Em geral, listamos nas referências apenas os itens citados no documento, mas nos casos de materiais sem rigor científico, poderemos incluir referências não citadas.

O argumento do ambiente é opcional e significa o maior rótulo para calcular alinhamento. Assim, se escrevermos `thebibliography{00}`, estamos dizendo para o T<sub>E</sub>X que não teremos mais do que 99 referências bibliográficas, ou seja, nenhuma referência será numerada com três dígitos..

Cada item da bibliografia é especificado pelo `\bibitem`. O primeiro argumento (opcional) é o rótulo e o segundo (obrigatório) é a chave para a referência ser citada depois. O comando para citarmos uma referência é o `\cite{chave}`. Por exemplo, se a bibliografia contém o item

```
\bibitem{medeiros} \textrm{Silva, Sebastião Medeiros},  
\textit{Matemática: Para os Cursos de Economia, Administração e  
Ciências Contábeis - Volume 1}. Editora Atlas.
```

significa que para citarmos esta referência, devemos escrever:

De acordo com `\cite{medeiros}`

...

**Exemplo 9.1.1.** *Inclua, em um documento de sua autoria, os comandos abaixo, compile e observe a saída.*

```
\begin{thebibliography}{99}
\bibitem{guidorizzi} \textrm{Guidorizzi, Hamilton L.},
\textit{Matemática para Administração}. Editora LTC.

\bibitem{leithold} \textrm{Leithold, L.},
\textit{Matemática Aplicada à Economia e Administração}. Editora
Harbra.

\bibitem{murolo} \textrm{Murolo, A. ; Bonetto, G.},
\textit{Matemática aplicada à: Administração, Economia,
Contabilidade}. Pioneira Thompson Learning.

\bibitem{thomas} \textrm{Thomas, G. B.},
\textit{Cálculo - Volume 1}. Editora Pearson.

\bibitem{weber} \textrm{Weber, Jean E.},
\textit{Matemática para Economia e Administração}. Editora Harbra
Ltda.

\end{thebibliography}
```

**Observação 9.1.1.** *A formatação da referência bibliográfica no ambiente `thebibliography` deverá ser feita manualmente. Como observado no exemplo acima, o nome da obra em negrito é feito através do comando `\textit{nome}`, por exemplo.*

**Teste 9.1.1.** *No documento onde você incluiu a bibliografia acima, cite, em alguma parte do texto, o livro do Leithold.*

---

## 9.2 Criando o Índice Remissivo

---

Para criarmos o índice remissivo em um documento, devemos colocar no preâmbulo os comandos `\usepackage{makeidx}` e `\makeindex`. No local onde queremos que o índice remissivo fique, que em geral é no final do documento, inserimos o comando `\printindex`.

Para acrescentarmos itens ao índice remissivo, usamos o comando `\index{entrada}`, onde *entrada* é a palavra ou expressão que constará no índice. Não devemos colocar espaços entre a palavra referida e o comando.

**Exemplo 9.2.1.** *Digite o código-fonte abaixo, compile duas vezes e observe a saída.*

```
\documentclass[a4paper,twoside]{article}

%*****
%INSTRUÇÕES SOBRE USO DE PACOTES
%*****

\usepackage[latin1]{inputenc} % Permite usar acentuação direto do teclado
\usepackage{amssymb,amsmath}
\usepackage{makeidx}

%*****
%FORMATAÇÃO DAS PÁGINAS
%*****

\setlength{\textwidth}{15cm} %LARGURA DO TEXTO
\setlength{\textheight}{22cm} %ALTURA DO TEXTO


\makeindex % Permite criar o índice remissivo


\begin{document}

\begin{center}
\textbf{Definição da Integral Definida}\index{Integral Definida}}
\end{center}
Se  $f$  for uma função definida no intervalo fechado  $[a,b]$ , então
a  $\textbf{integral definida}$  de  $f$  de  $a$  até  $b$ , denotada por  $\int_a^b f(x)dx$ ,
será dada por:

$$\int_a^b f(x)dx = \lim_{||\Delta|| \rightarrow 0} \sum_{i=1}^n f(c_i)\Delta x_i,$$

se esse limite existir.

\newpage

\begin{center}
\textbf{Teorema Fundamental do Cálculo}\index{TFC}}
```

```
\end{center}
```

Seja  $f$  uma função contínua no intervalo fechado  $[a,b]$ , e seja  $x \in [a,b]$ .

Se  $F$  for a função definida por  $F(x) = \int_a^x f(t)dt$  então

$F'(x) = f(x)$ .

```
\printindex
```

```
\end{document}
```

**Observação 9.2.1.** *Note que o título do índice remissivo ficou em inglês(*Index*). Para mudá-lo, inserimos o comando `\renewcommand{\indexname}{Índice Remissivo}` logo acima do comando `\printindex`. Faça-o!*

### 9.2.1 Criando Sub-entradas

Podemos criar uma ou mais sub-entradas para uma entrada do índice remissivo. Para tal, usamos o comando `\index{entrada!sub-entrada}`, com “!” separando a entrada da sub-entrada e sem espaços.

**Exemplo 9.2.2.** *Volte ao exemplo anterior, substitua o comando `\index{TFC}` por `\index{TFC!Primeira Forma}`, compile duas vezes e observe a saída.*

---

# CAPÍTULO 10

---

## DIAGRAMAS, TABELAS, CAIXAS GRÁFICAS

---

### 10.1 Diagramas

---

Aprenderemos como usar o pacote *Xy-pic* para construir diagramas. Tal pacote tem vários outros usos, que não exploraremos neste curso. Para usar o pacote citado, devemos colocar no preâmbulo do documento o comando `\usepackage[all]{xy}`.

#### 10.1.1 Diagramas Simples

O comando que usaremos para criar diagramas é o `xymatrix`, que tem sintaxe bem parecida com a do ambiente `array`, que já conhecemos. A diferença é que o `xymatrix` permite que coloquemos setas, molduras, etc. No caso de setas, podemos usar em vários sentidos e direções. A posição da seta será dada pela inicial da palavra em inglês para a posição (ou uma combinação de iniciais). Por esse motivo, lembramos abaixo algumas palavras em inglês que serão importantes neste capítulo.

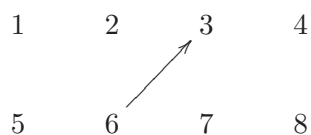
Arrow: Seta	
Left : Esquerda	Down : Para baixo
Right : Direita	Up : Para cima

Para construirmos uma seta, usamos o comando `\ar[direção]`. Por exemplo, se digitarmos como uma das entradas do `xymatrix` o comando `5\ar[lu]`, teremos como saída, após o 5, uma seta à esquerda e para cima. Abaixo um exemplo de código fonte que gera um diagrama simples.

**Exemplo 10.1.1.** *Digite e compile:*

```
\xymatrix{1 & 2 & 3 & 4 \\
5 & 6 \ar[ru] & 7 & 8 \\}
```

A saída será:

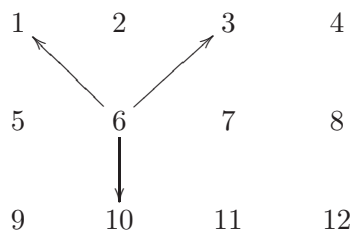


Podemos ter várias setas partindo de uma mesma entrada. Só devemos observar se a seta faz sentido. Por exemplo, não podemos colocar na primeira entrada da matriz uma seta à esquerda. Veja o exemplo abaixo.

**Exemplo 10.1.2.** *Digite e compile:*

```
\xymatrix{1 & 2 & 3 & 4 \\
5 & 6 \ar[ru] \ar[lu] \ar[d] & 7 & 8 \\
9 & 10 & 11 & 12 }
```

A saída será:

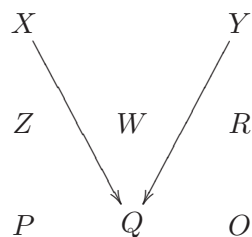


Podemos deixar, propositalmente, uma(ou mais) entrada(s) da matriz em branco, dependendo do nosso objetivo. Só não podemos esquecer dos símbolos & e \\.

**Exemplo 10.1.3.** *Digite e compile:*

```
\xymatrix{X \ar[ddr] & & Y\ar[ddl] \\
Z & W & R \\
P & Q & O }
```

A saída será:




---

## 10.2 Símbolos e Fórmulas

---

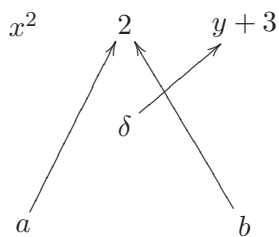
Podemos ter fórmulas matemáticas e símbolos como entradas, sem necessidade de digitar \$.



**Exemplo 10.2.1.** *Digite e compile:*

```
\xymatrix{x^2 & 2 & y + 3 \\
& \delta \ar[ru] & \\
a \ar[ruu]& & b\ar[luu] \\ }
```

A saída será:

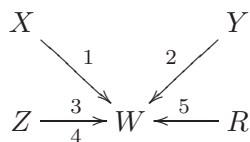


## 10.3 Índices

As setas dos diagramas podem vir acompanhadas de índices, com o uso dos símbolos  $\wedge$  e  $\_$ . Segue um exemplo.

```
\xymatrix{X \ar[dr]^1 & & Y\ar[dl]_2 \\
Z \ar[r]^3_4 & W & R\ar[l]_5 \\ }
```

Cuja saída é:



## 10.4 Tabelas Longas. Listas de Figuras e Tabelas.

Aprenderemos como fazer tabelas longas e como incluir no documento listas de figuras e tabelas utilizadas.

### 10.4.1 Tabelas Longas

Para elaborar uma tabela grande, que seja quebrada em mais de uma página, usamos o pacote `longtable`. O ambiente `longtable` permite criar tabelas longas que podem ser quebradas em várias páginas.

O `longtable` não pode ser colocado dentro do `table` e o `caption` tem a versão com “\*”, na qual a tabela em questão não aparece na lista de tabelas, quando esta é gerada.

Os parâmetros do `longtable` são os mesmos do `tabular`, suportando inclusive o `hhline`.

**Exemplo 10.4.1.** *Digite e compile:*

[illegible]



Nas listas constarão os nomes das figuras e tabelas e as respectivas páginas em que estão localizadas.

**Exemplo 10.4.2.** *Digite e compile:*

```
\begin{document}

\listoffigures

\listoftables

\newpage

\begin{figure}[h]
\begin{center}
{\includegraphics[scale=1.2]{triangulo.eps}}
\caption{Triângulo}\end{center}
\end{figure}

\begin{figure}[h]
\begin{center}
{\includegraphics[scale=0.8]{pentagono.eps}}
\caption{Pentágono}\end{center}
\end{figure}

\begin{figure}[h]
\begin{center}
{\includegraphics[scale=1]{circunferencia.eps}}
\caption{Circunferência}\end{center}
\end{figure}

\begin{table}[h]
\begin{tabular}{|l|c|r|p{5cm}|}
\hline
1 & 2 & 3 & 4\\ \hline
11 & 12 & 13 & 14\\ \hline
111 & 112 & 113 & 114 \\ \hline
\end{tabular}
\caption{Exemplo}
\end{table}

\begin{table}[h]
\begin{tabular}{|c|c|c|c|c|}
\hline
\end{tabular}
\end{table}
```

```

\hline
\multicolumn{5}{|c|}{Horário da Monitoria} \\ \hline
SEG & TER & QUA & QUI & SEX \\ \hline
12:00 & 14:00 & 11:30 & 17:30 & 18:00 \\ \hline
\end{tabular}
\caption{Monitoria}
\end{table}

\end{document}

```

---

## 10.5 Caixas Gráficas

---

Com o pacote `graphicx` (não esqueça de colocar no preâmbulo), podemos utilizar caixas gráficas para alterar o tamanho de um texto, assim como para rotacioná-lo. Vejamos como fazer isto.

### 10.5.1 Tamanho do Texto

Como observação inicial, destacamos que o argumento da caixa gráfica não pode conter parágrafos. Caso necessitemos inserir um parágrafo, usamos o `minipage`. Também observamos que nem todo visualizador de DVI suporta as caixas gráficas. Neste caso, devemos converter para PS ou PDF.

O comando mais simples é o `scalebox`, que simplesmente altera o tamanho do texto, multiplicando pela escala desejada.

Por exemplo, `\scalebox{4}{Amplia 4 vezes}` produz

**Amplia 4 vezes.**

Já `\scalebox{0.5}{metade do tamanho}` produz

metade do tamanho.

Podemos usar o `scalebox` para alterar também o tamanho vertical do texto, através de um argumento opcional.

Por exemplo, `\scalebox{2}[4]{Amplia 2x4 vezes}`, produz

**Amplia 2x4 vezes**, ou seja, amplia 2 vezes na horizontal e 4 na vertical.

O comando `resizebox` redefine o tamanho da caixa de texto, independente do tamanho original. Nele devem ser especificados o tamanho horizontal e o vertical, inclusive a unidade de medida, mas um deles pode ser substituído por “!”. Neste caso, o tamanho é ajustado de forma a manter a proporção.

**Exemplo 10.5.1.** *Digite os comandos abaixo, compile e observe a saída:*

```
\resizebox{8cm}{!}{10cm de comprimento}
```


`\resizebox{!}{0.5cm}{0.5cm de altura}`

`\resizebox{4cm}{2cm}{4cm de comprimento e 2cm de altura}`

**Observação 10.5.1.** *Como observado no exemplo acima, quando especificamos a altura e a largura do texto ao mesmo tempo, podemos deformá-lo. Devemos ter cuidado com isso!*

### 10.5.2 Rotação do Texto

Quando desejamos rotacionar o texto, usamos o comando `rotatebox`, que é bem simples de ser utilizado.

Por exemplo, `\rotatebox{30}{Trinta Graus}` produz .

Um pouco mais sofisticado é o uso do parâmetro opcional, com o qual podemos especificar o centro de rotação, usando combinações de l (left=esquerda), r(right=direita), t(top=parte de cima), b (bottom=parte de baixo) e c (center=centro).

Por exemplo, `\rotatebox[origin=rb]{-60}{Outros eixos}` produz



isto é, rotaciona o texto em menos 60° a partir da parte de baixo da direita.

Outro exemplo: `\rotatebox[origin=c]{30}{centro}` produz



ou seja, rotaciona o texto em 30° em torno do centro.

O último comando que aprenderemos é o `reflectbox`, que efetua espelhamento.

Por exemplo, `\reflectbox{espelho}` produz

.

**Observação 10.5.2.** *Os comandos apresentados neste capítulo podem ser combinados entre si.*