
INTRODUÇÃO AO L^AT_EX

Reginaldo J. Santos
Departamento de Matemática-ICEx
Universidade Federal de Minas Gerais
<http://www.mat.ufmg.br/~regi>

Abril de 2002
última atualização em
11 de setembro de 2009

Sumário

1	Introdução	5
2	Texto, Comandos e Ambientes	9
2.1	Um aviso	9
2.2	Estrutura Básica de um Arquivo Fonte \LaTeX	9
3	Diferentes Formas de Exibição de Texto	13
3.1	Mudando o Tipo das Letras	13
3.2	Texto em Cores	14
3.3	Centralizando e Indentando	14
3.4	Listas	15
3.4.1	Ambiente <code>itemize</code>	15
3.4.2	Ambiente <code>enumerate</code>	16
3.4.3	Ambiente <code>list</code>	16
3.5	Teoremas, Proposições, etc	17
4	Fórmulas Matemáticas	20
4.1	Principais Elementos do Modo Matemático	20
4.2	Expoentes e Índices	21
4.3	Frações	22
4.4	Raízes	22
4.5	Somatórios e Integrais	22
4.6	Coefficientes Binomiais	23
4.7	Pontos, Espaços e Texto no Modo Matemático	24

4.8	Símbolos Matemáticos	24
4.9	Funções Matemáticas	28
4.10	Fórmulas com Til, Barra, Chapéu, etc	29
4.11	Tamanho automático de parênteses e similares	29
4.12	Matrizes	30
4.13	Equações ou Inequações Multi-linhas	31
4.14	Fórmulas dentro de uma Moldura	34
4.15	Mudança dos Tipos de Letras no Modo Matemático	35
4.16	Espaços no Modo Matemático	35
4.17	Usando o MATLAB® para Gerar Fórmulas	36
5	Figuras	37
5.1	Ambiente picture	37
5.2	Figuras Produzidas por Outros Programas	40
5.3	Ambiente figure	43
6	Partes do Documento	45
6.1	Título	45
6.2	Seções e Capítulos	45
6.3	Conteúdo	46
6.4	Bibliografia	46
6.5	Índice Alfabético	49
6.6	Apêndice	50
7	Outros Recursos	51
7.1	Caracteres Especiais	51

7.2	Referências Cruzadas	51
7.3	Espaçamento	53
7.3.1	Espaçamento Horizontal	53
7.3.2	Espaçamento Vertical	54
7.4	Linha Horizontal	55
7.5	Retângulo Sólido	55
7.6	Texto Emoldurado	56
7.7	Dividindo o Arquivo Fonte	56
7.8	Tabelas	58
7.9	Duas Colunas	64
7.10	Traduzindo para PS e PDF	64
7.11	Mais Informações na Internet	66
	Referências	67

1 Introdução

O \LaTeX é um pacote feito para a preparação de textos impressos de alta qualidade, especialmente para textos matemáticos. Ele foi desenvolvido por Leslie Lamport a partir do programa \TeX criado por Donald Knuth.

Podemos dividir os programas de processamento de texto em duas classes. Com os chamados *processadores de texto*, existe um menu na tela apresentando os recursos, que podem ser usados no processamento do texto, que por sua vez podem ser selecionados com o uso do mouse. Depois de selecionado um recurso, o texto é digitado e aparece na tela exatamente como vai ser impresso no papel. O usuário pode ver logo no estágio de entrada do texto, se o texto será impresso como esperado. Este método é chamado “what-you-see-is-what-you-get” ou simplesmente WYSWYG.

A segunda classe, que é a que pertence o \LaTeX , o processamento do texto é feito em duas etapas distintas. O texto a ser impresso e os comandos de formatação são escritos em um **arquivo fonte** com o uso de um *editor de textos*, isto é, um programa que escreve textos em meio magnético. Em seguida o arquivo fonte é submetido a um programa formatador de textos, no nosso caso o \LaTeX , que gera um arquivo de saída, que pode ser impresso ou visualizado na tela. Programas deste tipo podem parecer inicialmente mais complicados do que os do outro tipo, mas apresentam uma série de vantagens em relação aos processadores de texto, como por exemplo:

- Mudanças na formatação do texto inteiro com apenas a mudança de alguns comandos.
- Escrita de fórmulas complexas usando apenas comandos, por exemplo, $\int_0^a e^{-x^2} dx$, é impressa com o comando `$\int_0^a e^{-x^2} dx$` ;

- Numeração automática de fórmulas, seções, definições, exemplos e teoremas, o que permite que você faça mudanças na ordem do texto sem que seja necessário trocar os números dos itens.
- As citações a fórmulas, seções, definições, exemplos, teoremas além de citações bibliográficas também podem ser automatizadas, de forma que mudanças no texto não produzem erros nas citações.

O \LaTeX é um programa de código aberto, por isso existem várias implementações. Usamos uma implementação chamada \MikTeX , uma interface gráfica junto com um editor de textos chamada, WinShell, e um visualizador, YAP. Todos estes programas são “freeware”, isto é, são gratuitos e podem trabalhar juntos como se fossem um só. Para facilitar, disponibilizamos as instruções para a instalação dos programas em <http://www.mat.ufmg.br/~regi/latex/instlat2.html>.

Depois dos programas devidamente instalados, para começar a usar o \LaTeX você deve clicar em **Iniciar**, depois em **Programas**, depois em **WinShell** e finalmente clicar em **WinShell**. Depois de inicializado o WinShell, a primeira coisa a fazer é abrir um arquivo no editor do WinShell. Para isto, no menu, escolha **Open...** e o nome do arquivo existente ou **New** e ele inicia um novo arquivo.

Depois você pode processar o arquivo fonte através do \LaTeX clicando no botão **TEX** criando assim um arquivo com o mesmo nome mas com terminação **.dvi**. Para visualizar ou imprimir este arquivo clique no botão **DVI**.

A seguir está um pequeno arquivo exemplo, que você pode usar como modelo para os seus arquivos. Para isto marque o seu conteúdo com o mouse, copie-o para a área de transferência e cole-o no \TeX Shell.

```
% Este é um pequeno arquivo fonte para o LaTeX
```

```
% Use este arquivo como modelo para fazer seus próprios arquivos LaTeX.
% Tudo que está à direita de um % é um comentário e é ignorado pelo LaTeX.
%
\documentclass[a4paper,12pt]{article}% Seu arquivo fonte precisa conter
\usepackage[brazil]{babel}           % estas quatro linhas
\usepackage[latin1]{inputenc}        % além do comando \end{document}
\begin{document}                     % no fim.
```

```
\section{Texto, Comandos e Ambientes} % Este comando faz o título da seção.
```

Um arquivo fonte do `\LaTeX` contém além do texto a ser processado, comandos que indicam como o texto deve ser processado. Palavras são separadas por um ou mais espaços. Parágrafos são separados por uma ou mais linhas em branco. A saída não é afetada por espaços extras ou por linhas em branco extras. A maioria dos comandos do `\LaTeX` são iniciados com o caracter `\backslash`. Uma `\backslash` sozinha produz um espaço. Um ambiente é uma região do texto que tem um tratamento especial. Um ambiente é iniciado com `\texttt{\backslash$begin\{nome do ambiente\}}` e terminado por `\texttt{\backslash$end\{nome do ambiente\}}`.

%Aspas são digitadas assim:
‘‘Texto entre aspas’’.

%Texto em itálico deve ser digitado como:
`\textit{Isto está em itálico}`.

%Texto em negrito deve ser digitado como:
`\textbf{Isto está em negrito}`.

`\subsection{Um aviso}` % Este comando faz o título da subseção.

Lembre-se de não digitar nenhum dos 10 caracteres especiais

% & \$ # % _ { } ^ ~ \
`\& \$ \# \% _ \{ \} \^{} \~{} \\backslash` exceto como um
comando!

`\end{document}` % O arquivo fonte termina com este comando.

O texto anterior foi conseguido com o uso do ambiente `verbatim` (significa ao pé da letra).

`\begin{verbatim}`

Tudo que estiver dentro deste
ambiente é impresso exatamente como é digitado.

`\end{verbatim}`

O ambiente `verbatim` cria um parágrafo para o texto. O mesmo efeito dentro de uma linha é conseguido com o uso do comando `\verb!texto!`. Se no texto aparecer o caracter `!` ele pode ser substituído por outro diferente de `*` que não apareça no texto.

2 Texto, Comandos e Ambientes

Um arquivo fonte do \LaTeX contém além do texto a ser processado, comandos que indicam como o texto deve ser processado. Palavras são separadas por um ou mais espaços. Parágrafos são separados por uma ou mais linhas em branco. A saída não é afetada por espaços extras ou por linhas em branco extras. A maioria dos comandos do \LaTeX são iniciados com o caracter `\`. Uma `\` sozinha produz um espaço. Um ambiente é uma região do texto que tem um tratamento especial. Um ambiente é iniciado com

`\begin{nome do ambiente}` e terminado por `\end{nome do ambiente}`.

“Texto entre aspas”.

Isto está em itálico.

Isto está em negrito.

2.1 Um aviso

Lembre-se de não digitar nenhum dos 10 caracteres especiais `& $ # % _ { } ^ ~ \` exceto como um comando! Eles são impressos com os comandos

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

2.2 Estrutura Básica de um Arquivo Fonte \LaTeX

Todo arquivo fonte \LaTeX , que normalmente tem a extensão `.tex` tem um preâmbulo e um corpo. No preâmbulo estão comandos que especificam parâmetros globais para o processamento do texto, tais como tipo de documento, formato do papel, altura e largura do texto, a forma de saída das páginas com a sua paginação e cabeçalhos automáticos. O preâmbulo deve conter

pelo menos o comando `\documentclass`. Se não existem outros comandos no preâmbulo, então \LaTeX seleciona valores standard para as várias variáveis que determinam o formato global do texto.

O preâmbulo termina com o comando `\begin{document}`. Tudo que fica abaixo deste comando é interpretado como corpo. No corpo fica o texto propriamente dito junto com comandos adicionais, que só têm efeito local.

O corpo termina com o comando `\end{document}`. Isto também é normalmente o fim do arquivo fonte.

Esta é estrutura de um arquivo \LaTeX .

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

Comandos globais

```
\begin{document}
```

Texto e comandos de efeito local

```
\end{document}
```

No comando `\documentclass[opções]{estilo}` o estilo pode ser: `article`, `report`, `book` ou `letter`. Algumas das opções são: `10pt`, `11pt` ou `12pt` para o tamanho base das letras usadas no texto, `a4paper` se o papel for A4 (para o papel letter, não é necessário nenhuma indicação), `landscape` para a impressão no modo paisagem, `twocolumn` para a impressão em duas colunas, `twoside` para a impressão nos dois lados do papel, `titlepage` para que no estilo `article` seja gerada uma página separada com o título (para os outros tipos não é necessária esta opção).

Alguns exemplos de Comandos globais:

`\usepackage[brazil]{babel}` indica para usar o pacote `babel`, com a opção de língua `brazil`, faz com que ele gere datas e nomes como Capítulo, Bibliografia em português com estilo brasileiro.

`\usepackage{graphicx,color}` indica para usar os pacotes `graphicx` e `color`, que permitem incluir figuras e colorir o texto.

`\usepackage[latin1]{inputenc}` indica para usar o pacote `inputenc` com a opção `latin1`, que define uma codificação para os caracteres em que os acentos são digitados diretamente pelo teclado.

`\usepackage{amsthm,amsfonts}` indica para usar os pacotes da American Mathematical Society `amsthm` e `amsfonts`. O primeiro, entre outras coisas, define um estilo para a escrita dos teoremas e o segundo adiciona alguns estilos de letras, por exemplo \mathbb{R} , \mathbb{C} e \mathbb{N} foram geradas com `$$\mathbb{R}$$`, `$$\mathbb{C}$$` e `$$\mathbb{N}$$` por causa da adição deste pacote.

`\setlength{\textwidth}{16 cm}` indica que a largura do texto é de 16 cm.

`\setlength{\textheight}{20 cm}` indica que a altura do texto é de 20 cm.

`\evensidemargin 0 cm` indica que a margem esquerda das páginas pares é zero. O tamanho real da margem esquerda das páginas pares é a soma da variável `\hoffset` com a variável `\evensidemargin`.

`\oddsidemargin 0 cm` indica que a margem esquerda das páginas ímpares é zero. O tamanho real da margem esquerda das páginas ímpares é a soma da variável `\hoffset` com a variável `\oddsidemargin`.

`\topmargin 0 cm` indica que a margem superior é zero. O tamanho real da margem superior é a soma da variável `\voffset` com a variável `\topmargin`.

`\baselineskip 65 mm` define a distância entre as linhas como sendo de 65 mm.

`\sloppy` reduz o número de divisões nas palavras que são impressas nos finais das linhas.

Além destes, um comando que pode ser útil é o `\newcommand`. Com ele você pode definir nomes mais simples para comandos que têm nomes grandes. Por exemplo:

`\newcommand{\binv}{\backslash}` dá um novo nome para o comando que imprime `\`.

`\newcommand{\pot}{\^{} }` cria um novo nome para o comando que imprime `^`.

`\newcommand{\til}{\~{} }` define um novo nome para o comando que imprime `~`.

3 Diferentes Formas de Exibição de Texto

O \LaTeX permite uma variedade de formas de exibição de texto: pode-se mudar o tamanho e o tipo das letras, pode-se centralizar, criar diferentes tipos de listas, etc.

3.1 Mudando o Tipo das Letras

1. Mudando o estilo:

```
\textit{itálico} produz itálico;  
\textbf{negrato} produz negrato;  
\textrm{romano} produz romano;  
\textsf{sans serif} produz sans serif;  
\texttt{máquina de escrever} produz máquina de escrever;  
\textsc{caixa alta} produz CAIXA ALTA.
```

2. Mudando o tamanho:

```
{\tiny o menor} produz o menor;  
\scriptsize muito pequeno} produz muito pequeno;  
\footnotesize menor} produz menor;  
\small pequeno} produz pequeno;  
\large grande} produz grande;  
\Large maior} produz maior;  
\LARGE maior ainda} produz maior ainda;  
\huge ainda maior} produz ainda maior;  
\Huge o maior de todos} produz o maior de todos.
```

3.2 Texto em Cores

Para escrever em cores é preciso colocar no preâmbulo o comando

```
\usepackage{graphicx,color}
```

Depois podemos usar o comando `\textcolor{cor}{texto}` para colorir o texto como em `texto em azul`, `texto em vermelho` e `texto em verde` que foram conseguidos com `\textcolor{blue}{texto em azul}`, `\textcolor{red}{texto em vermelho}` e `\textcolor{green}{texto em verde}`.

Também o comando `{\color{cor1}texto}` pode ser usado. Por exemplo `texto em azul`, `texto em vermelho` e `texto em verde` foram conseguidos com `{\color{blue}texto em azul}`, `{\color{red}texto em vermelho}` e `{\color{green}texto em verde}`.

3.3 Centralizando e Indentando

1. Centralizando:

```
\begin{center}
linha 1 \\ linha 2\\ \ldots \\ linha n
\end{center}
```

produz

```
linha 1
linha 2
...
linha n
```

2. Indentando:

```
\begin{quote}  
Texto a ser indentado.  
\end{quote}
```

produz

 Texto a ser indentado.

3.4 Listas

As listas são construídas com ambientes:

3.4.1 Ambiente `itemize`

```
\begin{itemize}  
\item Os itens são precedidos por  $\bullet$ ;  
\item Os itens são separados por um espaço adicional.  
\end{itemize}
```

produz

- Os itens são precedidos por •;
- Os itens são separados por um espaço adicional.

3.4.2 Ambiente `enumerate`

```
\begin{enumerate}
\item Os itens são numerados com algarismos arábicos, no primeiro nível,
  \begin{enumerate}
    \item são numerados com letras no segundo nível e
      \begin{enumerate}
        \item são numerados com algarismos romanos no terceiro nível.
      \end{enumerate}
    \end{enumerate}
  \end{enumerate}
\end{enumerate}
```

produz

1. Os itens são numerados com algarismos arábicos, no primeiro nível,
 - (a) são numerados com letras no segundo nível e
 - i. são numerados com algarismos romanos no terceiro nível.

3.4.3 Ambiente `list`

```
\newcounter{quest}
\begin{list}{\textbf{Questão \arabic{quest}.}}{\usecounter{quest}
\setlength{\labelwidth}{-2mm} \setlength{\parsep}{0mm}
\setlength{\leftmargin}{0mm}}
\renewcommand{\labelenumi}{(\alph{enumi})}
```



```
\item Esta é a primeira questão  
com alguns itens:  
\begin{enumerate}  
  \item Este é o primeiro item  
  \item Segundo item  
\end{enumerate}  
\item Esta é a segunda questão.  
\end{list}
```

produz

Questão 1. Esta é a primeira questão com alguns itens:

- (a) Este é o primeiro item
- (b) Segundo item

Questão 2. Esta é a segunda questão.

3.5 Teoremas, Proposições, etc

O comando `\newtheorem{ambiente}{título}` define um ambiente para a escrita de teoremas, proposições, etc, onde `ambiente` é um apelido para o ambiente e `título` é o título que será impresso no início do texto, como **Teorema**, **Lema**, **Corolário**, etc. Por exemplo:

```
\newtheorem{teo}{Teorema}[section]  
\newtheorem{lema}[teo]{Lema}  
\newtheorem{cor}[teo]{Corolário}  
\newtheorem{prop}[teo]{Proposição}
```

define quatro ambientes com apelidos `teo`, `lema`, `cor` e `prop`. A numeração é automática e o argumento `[teo]` faz com que os outros ambientes sigam a numeração do ambiente `teo`. O argumento `[section]` faz com que apareça o número da seção junto ao número do teorema.

Uma vez definidos os ambientes no corpo do arquivo fonte, eles podem ser chamados em qualquer ponto após a definição dos ambientes, como no exemplo seguinte:

```
\begin{teo}[Pitágoras]
```

```
Em todo triângulo retângulo o quadrado do comprimento da  
hipotenusa é igual a soma dos quadrados dos comprimentos dos  
catetos.
```

```
\end{teo}
```

produz

Theorem 3.1 (Pitágoras). *Em todo triângulo retângulo o quadrado do comprimento da hipotenusa é igual a soma dos quadrados dos comprimentos dos catetos.*

Outros recursos que são necessários para se escrever um texto matemático com qualidade como ambientes para as demonstrações e para definições e exemplos estão no pacote da American Mathematical Society `amsthm`. Os recursos que descreveremos a seguir nesta seção só funcionam se for colocado no preâmbulo o comando

```
\usepackage{amsthm}
```

Para as demonstrações existe o ambiente `proof`. Que é usado na forma

```
\begin{proof}
```

```
. . .
```

```
\end{proof}
```

Exemplo

Demonstração. Para demonstrar o Teorema de Pitágoras ...

□

Os ambientes para definições e exemplos podem ser criados com os comandos

```
\theoremstyle{definition}  
\newtheorem{defi}{Definição}  
\newtheorem{exem}{Exemplo}
```

Depois podem ser usados da seguinte forma

Exemplo 1. Este é um exemplo do uso do ambiente `exem` definido acima.

que foi obtido com

```
\begin{exem}  
Este é um exemplo do uso do ambiente \texttt{exem} definido acima.  
\end{exem}
```

Definição 1. Definimos o produto de ...

que foi conseguido com

```
\begin{defi}  
Definimos o produto de ...  
\end{defi}
```

4 Fórmulas Matemáticas

4.1 Principais Elementos do Modo Matemático

Fórmulas matemáticas são produzidas digitando no arquivo fonte texto descrevendo-as. Isto significa que o L^AT_EX deve ser informado que o texto que vem a seguir é uma fórmula e também quando ela termina e o texto normal recomeça. As fórmulas podem ocorrer em uma linha de texto como $ax^2 + bx + c = 0$, ou destacada do texto principal como

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

No meio do texto o modo matemático é iniciado e terminado com o sinal `$`. A primeira fórmula acima foi produzida com `$ax^2+bx+c=0$`. Quando a fórmula é destacada, então o modo matemático é iniciado e terminado com `$$`. A segunda fórmula acima foi produzida com `$$`

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

As fórmulas destacadas podem ser numeradas usando o ambiente `\begin{equation}` *equação* `\end{equation}`. Por exemplo

```
\begin{equation}
x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}.
\end{equation}
```

produz

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \quad (1)$$

Os espaços digitados nas fórmulas são ignorados pelo \LaTeX . Os símbolos matemáticos:

+ - = < > / : ! ' | [] ()

podem ser digitados diretamente do teclado. Por exemplo:

$|x| < a$ se, e somente se, $-a < x < a$

foi produzido com

`$|x| < a$ se, e somente se, $-a < x < a$`

As chaves `{ }` servem para agrupar logicamente partes da fórmula e não são impressas diretamente. Para incluir chaves em uma fórmula tem que ser usados `\{` e `\}`.

4.2 Expoentes e Índices

O \LaTeX permite, de maneira simples, a produção de qualquer combinação de expoentes e índices com o tamanho correto. O caracter `^` indica que o próximo caracter é um expoente e `_` indica que o próximo caracter é um índice. Por exemplo:

`$$x^2, a_n, x_i^2, x^{2n}, x^{y^2}, x^{y_1}$$`

produz

$$x^2, a_n, x_i^2, x^{2n}, x^{y^2}, x^{y_1}$$

Observe que quando índices e expoentes ocorrem juntos a ordem é indiferente. Observe também que quando o índice ou expoente tem mais de um caracter, eles devem estar entre chaves.

4.3 Frações

Pequenas frações no meio do texto podem ser escritas usando o caracter `/`, como `$(a+b)/2$`, que produz $(a + b)/2$. Para frações mais complexas o comando

`\frac{numerador}{denominador}`

é empregado para escrever o numerador em cima e o denominador em baixo com um traço de fração entre eles. Por exemplo,

`$(\frac{a+b}{2})$` produz $\frac{a+b}{2}$.

Para que uma fórmula que aparece no meio do texto apareça maior usamos o comando `\displaystyle` no início da fórmula. Por exemplo,

`$(\displaystyle\frac{a+b}{2})$` produz $\frac{a + b}{2}$.

O efeito inverso, ou seja, se uma fração que aparece numa fórmula destacada está com o tamanho muito grande podemos diminuí-la com os comandos

`\textstyle` ou `\scriptstyle` ou ainda `\scriptscriptstyle`

4.4 Raízes

As raízes são produzidas com o comando

`\sqrt[n]{radicando}`

Por exemplo, `$(\sqrt[3]{8})=2$` produz $\sqrt[3]{8} = 2$. Se o argumento opcional `[n]` for omitido, então a raiz quadrada é gerada. Por exemplo, `$(\sqrt{4})=2$` produz $\sqrt{4} = 2$. O tamanho e o comprimento do radical são automaticamente ajustados ao tamanho do radicando.

4.5 Somatórios e Integrais

Somatórios e integrais são obtidos com os comandos `\sum` e `\int`, respectivamente. Eles podem aparecer em diferentes estilos, dependendo se aparecem no meio do texto ou destacado e se é usado o comando `\limits`. Por exemplo, `\sum_{i=1}^n a_i` e `\int_a^b f(x)dx` produzem $\sum_{i=1}^n a_i$ e $\int_a^b f(x)dx$ respectivamente. Usando o comando `\limits`, `\sum\limits_{i=1}^n a_i` e `\int\limits_a^b f(x)dx` produzem $\sum_{i=1}^n a_i$ e $\int_a^b f(x)dx$ respectivamente.

Destacado do texto

\$\$

`\sum_{i=1}^n a_i \quad \quad \int_a^b f(x)dx`

\$\$

produzem

$$\sum_{i=1}^n a_i \quad \text{e} \quad \int_a^b f(x)dx$$

respectivamente.

4.6 Coeficientes Binomiais

Os **coeficientes binomiais** são obtidos com o comando `\choose`. Por exemplo

$$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1}$$

foi obtido com

\$\$

$\{n+1 \choose k\} = \{n \choose k\} + \{n \choose k-1\}$

\$\$

4.7 Pontos, Espaços e Texto no Modo Matemático

Os comandos `\ldots` e `\cdots` produzem três pontos, como em $a_1 + \cdots + a_n$, que foi produzida com `$a_1+\cdots + a_n$`. O comando `\vdots` produz três pontos na vertical. Um `~` (til) entre os elementos da fórmula faz com que não haja quebra de linha neste local. Os comandos `\quad` e `\qquad` produzem espaços no modo matemático, sendo o último maior do que o primeiro. O comando `\mbox` deve ser usado para produzir texto no modo matemático. Por exemplo,

\$\$

$x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad \mbox{e} \quad$

$x_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}.$

\$\$

produz

$$x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad \text{e} \quad x_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}.$$

4.8 Símbolos Matemáticos

Além dos símbolos disponíveis através do teclado existem vários símbolos que podem ser impressos usando o \LaTeX . Eles são impressos com o nome do símbolo após uma barra invertida. Os seus nomes são semelhantes aos seus significados matemáticos.

Letras Gregas					
Símbolo	Comando	Símbolo	Comando	Símbolo	Comando
α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>
δ	<code>\delta</code>	ϵ	<code>\epsilon</code>	ε	<code>\varepsilon</code>
ζ	<code>\zeta</code>	η	<code>\eta</code>	θ	<code>\theta</code>
ϑ	<code>\vartheta</code>	ι	<code>\iota</code>	κ	<code>\kappa</code>
λ	<code>\lambda</code>	μ	<code>\mu</code>	ν	<code>\nu</code>
ξ	<code>\xi</code>	\omicron	<code>\omicron</code>	π	<code>\pi</code>
ϖ	<code>\varpi</code>	ρ	<code>\rho</code>	ϱ	<code>\varrho</code>
σ	<code>\sigma</code>	ς	<code>\varsigma</code>	τ	<code>\tau</code>
υ	<code>\upsilon</code>	ϕ	<code>\phi</code>	φ	<code>\varphi</code>
χ	<code>\chi</code>	ψ	<code>\psi</code>	ω	<code>\omega</code>
Γ	<code>\Gamma</code>	Δ	<code>\Delta</code>	Θ	<code>\Theta</code>
Λ	<code>\Lambda</code>	Ξ	<code>\Xi</code>	Π	<code>\Pi</code>
Σ	<code>\Sigma</code>	Υ	<code>\Upsilon</code>	Φ	<code>\Phi</code>
Ψ	<code>\Psi</code>	Ω	<code>\Omega</code>		

Operadores Binários					
Símbolo	Comando	Símbolo	Comando	Símbolo	Comando
\pm	<code>\pm</code>	\mp	<code>\mp</code>	\times	<code>\times</code>
\div	<code>\div</code>	\cdot	<code>\cdot</code>	$*$	<code>\ast</code>
\star	<code>\star</code>	\dagger	<code>\dagger</code>	\ddagger	<code>\ddagger</code>
\cap	<code>\cap</code>	\cup	<code>\cup</code>	\setminus	<code>\setminus</code>
\vee	<code>\vee</code>	\wedge	<code>\wedge</code>	\otimes	<code>\otimes</code>
\triangle	<code>\bigtriangleup</code>	∇	<code>\bigtriangledown</code>	\oplus	<code>\oplus</code>
\triangleleft	<code>\triangleleft</code>	\triangleright	<code>\triangleright</code>	\odot	<code>\odot</code>
\circ	<code>\circ</code>	\bigcirc	<code>\bigcirc</code>	\diamond	<code>\diamond</code>

Relações					
Símbolo	Comando	Símbolo	Comando	Símbolo	Comando
\leq	<code>\leq</code>	\geq	<code>\geq</code>	\sim	<code>\sim</code>
$\not<$	<code>\not<</code>	$\not>$	<code>\not></code>	\neq	<code>\neq</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\simeq	<code>\simeq</code>
\in	<code>\in</code>	\notin	<code>\notin</code>	\equiv	<code>\equiv</code>
\perp	<code>\perp</code>	\propto	<code>\propto</code>	\cong	<code>\cong</code>

Outros Símbolos					
Símbolo	Comando	Símbolo	Comando	Símbolo	Comando
\forall	<code>\forall</code>	\exists	<code>\exists</code>	∞	<code>\infty</code>
∇	<code>\nabla</code>	∂	<code>\partial</code>	\emptyset	<code>\emptyset</code>
\Re	<code>\Re</code>	\Im	<code>\Im</code>	\neg	<code>\neg</code>

Setas					
Símbolo	Comando	Símbolo	Comando	Símbolo	Comando
\leftarrow	<code>\gets</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\rightarrow	<code>\to</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Leftrightarrow	<code>\iff</code>	\Updownarrow	<code>\Updownarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>		

Símbolos com Dois Tamanhos					
Símbolo	Comando	Símbolo	Comando	Símbolo	Comando
$\Sigma \quad \sum$	<code>\sum</code>	$\Pi \quad \prod$	<code>\prod</code>	$\odot \quad \bigodot$	<code>\bigodot</code>
$\int \quad \int$	<code>\int</code>	$\oint \quad \oint$	<code>\oint</code>	$\otimes \quad \bigotimes$	<code>\bigotimes</code>
$\cup \quad \bigcup$	<code>\bigcup</code>	$\cap \quad \bigcap$	<code>\bigcap</code>	$\oplus \quad \bigoplus$	<code>\bigoplus</code>

Aos símbolos que existem em dois tamanhos podem ser acrescentados limites inferiores e superiores. Por exemplo,

$$\bigcup_{i=0}^n A_i$$

é conseguido com `$$$ \bigcup_{i=0}^n A_i $$$`. Enquanto, $\int_0^\infty f(t)dt$ é conseguido com `$$$ \int \limits_0^\infty f(t)dt $$$`. O comando `\limits` faz com que no modo texto os limites apareçam realmente embaixo e acima da integral.

Podemos redefinir a maneira como é impresso a parte real, a parte imaginária de um número complexo e o conjunto vazio redefinindo os comandos `\Re`, `\Im` e `\emptyset`. Podemos redefinirlos, por exemplo, como

```
\renewcommand{\Re}{\mathrm{Re}}
\renewcommand{\Im}{\mathrm{Im}}
\renewcommand{\emptyset}{\mbox{\large{o}}}
```

Desta forma `\Re(z)`, `\Im(z)` e `\emptyset` produzem $\operatorname{Re}(z)$, $\operatorname{Im}(z)$ e \emptyset .

4.9 Funções Matemáticas

São usadas tipos de letras diferentes para variáveis e para nomes de funções. Por exemplo, `\cos x` produz $\cos x$, enquanto `\cos x` produz $\cos x$. Muitos nomes de funções já estão definidos, bastando para chamá-los usar um `\` antes do nome (principalmente aquelas em que o nome em português é o mesmo que em inglês). Para outros nomes é necessário defini-los. Por exemplo,

```
\usepackage{amsmath}
\DeclareMathOperator{\sen}{sen}
\DeclareMathOperator{\dist}{dist}
```

no preâmbulo do arquivo fonte define o nome das funções `sen` e `dist`. Depois de definidas, podemos usar como em `\sen x` e `\sen(x)`, que foram produzidos com `\sen x` e `\sen(x)`, respectivamente. Alguns nomes de funções pré definidos são: `\arccos`, `\arctan`, `\cos`, `\cosh`, `\cot`, `\coth`, `\csc`, `\det`, `\exp`, `\lim`, `\ln`, `\log`, `\max`, `\min`, `\sec`, `\tan`, `\tanh`. Por exemplo, para se conseguir

$$\lim_{x \rightarrow 0} \frac{\operatorname{sen} x}{x},$$

deve-se escrever $\lim_{x \rightarrow 0} \frac{\sin x}{x}$. Esta fórmula no meio do texto, $\lim_{x \rightarrow 0} \frac{\sin x}{x}$, é conseguida com $\lim_{x \rightarrow 0} \frac{\sin x}{x}$, ou $\lim_{x \rightarrow 0} \frac{\sin x}{x}$ é conseguida com $\lim\limits_{x \rightarrow 0} \frac{\sin x}{x}$.

4.10 Fórmulas com Til, Barra, Chapéu, etc

Os comandos `\overline{fórmula}` e `\underline{fórmula}` servem para colocar barras em cima e em baixo de uma letra ou uma fórmula. Por exemplo, $\overline{a^2} + \underline{bc}$ foi conseguido com `\overline{a}^2+\underline{bc}`. Pode-se colocar também chaves em cima e em baixo de fórmulas com os comandos `\overbrace{fórmula}` e `\underbrace{fórmula}`. Por exemplo, $\overbrace{x_1 + x_2 + \dots + x_{n-1}}^n + x_n$ foi obtida com `\overbrace{x_1+\underbrace{x_2+\ldots+x_{n-1}}_{n-2}+x_n}^n`. Setas em cima de letras são conseguidas com o comando `\vec{letra}`, como em \vec{v} , que foi obtido com `\vec{v}`. Também $V = \overrightarrow{AB}$ foi conseguido com `V=\stackrel{\rightarrow}{AB}`. Chapéu e til em cima de letras são conseguidas com os comandos `\hat{}`, `\widehat{}`, `\tilde{}` e `\widetilde{}`. Por exemplo, \hat{A} e \widehat{ABC} foram conseguidos com `\hat{A}` e `\widehat{ABC}`.

4.11 Tamanho automático de parênteses e similares

Os comandos `\left(` e `\right)` produzem parênteses com tamanho ajustado na altura para conter a fórmula que é englobada por eles. Observe a diferença entre $\cos(\frac{\pi}{2}x^2)$ e $\cos(\frac{\pi}{2}x^2)$ que foram obtidos com `\cos(\frac{\pi}{2}x^2)` e `\cos\left(\frac{\pi}{2}x^2\right)` respectivamente. O mesmo acontece com colchetes e chaves, usando os comandos `\left[`,

`\right]`, `\left\{` e `\right\}`. Os comandos `\left` e `\right` devem aparecer sempre aos pares. Uma exceção ao uso de tamanhos automáticos é o caso

$$\int_0^{\infty} e^{-st} dt = \frac{e^{-st}}{-s} \Big|_0^{\infty}$$

que foi conseguido com

```
$$\int_0^{\infty} e^{-st} dt = \frac{e^{-st}}{-s} \Big|_0^{\infty}$$
```

4.12 Matrizes

As matrizes são produzidas com o uso do ambiente `array`. Os elementos de uma mesma linha são separados pelo carácter `&` e as linhas são separadas por `\\`. É necessário passar para o `LATEX` como as colunas devem ser alinhadas. Isto é feito em seguida ao comando que inicia o ambiente. Por exemplo, `\begin{array}{ccrll}` diz ao `LATEX` que a matriz tem 5 colunas e que as duas primeiras devem ser alinhadas ao centro, que a do meio deve ser alinhada à direita e que as duas últimas devem ser alinhadas à esquerda. Por exemplo, as matrizes

$$A = \begin{bmatrix} 1 & 3 & 0 \\ 2 & 4 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 3 & -2 \end{bmatrix}, \quad \text{e} \quad C = \begin{bmatrix} 1 \\ 4 \\ -3 \end{bmatrix}$$

foram conseguidas com

```
$$
```

```
A=\left[\begin{array}{rrr}
```

```
1&3&0\\
```

```

2&4&-2
\end{array}\right],\quad
B=\left[\begin{array}{ccc}
1&3&-2
\end{array}\right],
\quad \mbox{e}\quad
C=\left[\begin{array}{r}
1\\4\\-3
\end{array}\right]

```

Um outro exemplo do uso do ambiente array é

$$|x| = \begin{cases} -x & \text{se } x \leq 0 \\ x & \text{se } x > 0 \end{cases}$$

```

$$|x|=\left\{\begin{array}{rc}
-x&\mbox{se } x\leq 0\\
x &\mbox{se } x>0
\end{array}\right.

```

Observe o uso do comando `\right.` para fechar o comando `\left{`. Para acrescentar mais espaço entre as linhas podem ser usados os comandos `\noalign{\smallskip}`, `\noalign{\medskip}` ou `\noalign{\bigskip}` depois de cada `\\`.

4.13 Equações ou Inequações Multi-linhas

Para a impressão de equações ou inequações em várias linhas pode-se usar os seguintes ambientes:

```
\begin{eqnarray}
```

```
linha 1\\
```

```
...
```

```
linha n
```

```
\end{eqnarray}
```

e

```
\begin{eqnarray*}
```

```
linha 1\\
```

```
...
```

```
linha n
```

```
\end{eqnarray*}
```

As linhas são separadas umas das outras por `\\` e cada linha tem a forma

fórmula da esquerda & fórmula do centro & fórmula da direita

Normalmente a fórmula do centro é um único caracter, o sinal da relação (por exemplo, $=$ ou \leq). A diferença entre `eqnarray` e `eqnarray*` é que no primeiro são acrescentadas às linhas números das equações. Exemplo:

$$\begin{aligned}
 (A + B)(A - B) &= (A + B)A + (A + B)(-B) \\
 &= AA + BA - AB - BB \\
 &= A^2 + BA - AB - B^2
 \end{aligned} \tag{2}$$

foi obtido com


```
\begin{eqnarray}
(A+B)(A-B) &=& (A+B)A + (A+B)(-B) \nonumber \\
&=& AA + BA - AB - BB \nonumber \\
&=& A^2 + BA - AB - B^2 \\
\end{eqnarray}
```

Observe o uso de `\nonumber` nas linhas que não devem ser numeradas.

Uma fórmula grande pode ser quebrada da seguinte maneira:

$$(a + b + c)^3 = \\ a^3 + 3a^2b + 3ab^2 + b^3 + 3ac^2 + \\ 3bc^2 + 3a^2c + 6abc + 3b^2c + c^3$$

que foi obtida com

```
\begin{eqnarray*}
\lefteqn{(a+b+c)^3=} \\
& & a^3 + 3a^2b + 3ab^2 + b^3 + 3ac^2 + \\
& & 3bc^2 + 3a^2c + 6abc + 3b^2c + c^3 \\
\end{eqnarray*}
```

Usando o pacote `amsmath` fórmulas muito grandes podem ser quebradas usando o ambiente `multline`. É necessário colocar no preâmbulo

```
\usepackage{amsmath}
```

Por exemplo

```
\begin{multline*}
(a+b+c+d)^3=\backslash\backslash
6acd+6abd+6bcd+6abc+3c^2d+3cd^2+3ad^2+3b^2c+3b^2d+3bc^2\backslash\backslash
+3bd^2+3a^2b+3a^2c+3a^2d+3ab^2+3ac^2+a^3+b^3+c^3+d^3.
\end{multline*}
```

produz

$$(a + b + c + d)^3 = \\ 6acd + 6abd + 6bcd + 6abc + 3c^2d + 3cd^2 + 3ad^2 + 3b^2c + 3b^2d + 3bc^2 \\ + 3bd^2 + 3a^2b + 3a^2c + 3a^2d + 3ab^2 + 3ac^2 + a^3 + b^3 + c^3 + d^3.$$

A trocando-se `multline*` por `multline` a equação fica numerada.

4.14 Fórmulas dentro de uma Moldura

As fórmulas podem ser colocadas dentro de uma moldura com o comando `\fbox`. Exemplo:

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

foi conseguido com

```
\begin{center}
\fbbox{$\displaystyle x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$}
\end{center}
```

4.15 Mudança dos Tipos de Letras no Modo Matemático

Uma mudança explícita nos tipos de letras dentro do modo matemático pode ser feito usando os comandos: `\mathrm`, `\mathcal`, `\mathbf`, `\mathsf`, `\mathtt`, `\mathit` e `\mathbb`. Por exemplo:

`\mathrm{AaBbCc}` produz $AaBbCc$, `\mathcal{ABC}` produz \mathcal{ABC} , `\mathbf{AaBbCc}` produz \mathbf{AaBbCc} , `\mathsf{aAbBcC}` produz aAbBcC , `\mathtt{aAbBcC}` produz \mathtt{aAbBcC} , `\mathit{aAbBcC}` produz aAbBcC e `\mathbb{ABCNR}` produz \mathbb{ABCNR} (este último necessita o comando `\usepackage{amsfonts}` no preâmbulo).

4.16 Espaços no Modo Matemático

Além dos comandos `\quad` e `\qquad` que produzem espaços horizontais no modo matemático, existem comandos para produzirem espaços menores ou até negativos.

`\,` produz um pequeno espaço, igual a $3/18$ de um `\quad`.

`\:` produz um espaço médio, igual a $4/18$ de um `\quad`.

`\;` produz um espaço maior, igual a $5/18$ de um `\quad`.

`\!` produz um espaço negativo de $-3/18$ de um `\quad`.

Por exemplo, $x^2/2$, $x^2/2$, $\int \int_D f(x)dx$ e $\iint f(x)dx$ foram produzidos com `\$x^2/2\$`, `\$x^2\!/2\$`, `\$ \int \int \limits_D f(x)dx \$` e `\$ \int \!\!\!\!\!\! \int \!\!\!\!\!\! \int \limits_{\!\!\!\!\!\! D} f(x) \, dx \$` respectivamente.

Com relação ao espaços verticais os comandos `\noalign{\smallskip}`, `\noalign{\medskip}` e `\noalign{\bigskip}` produzem espaços entre duas linhas de uma matriz.

4.17 Usando o MATLAB[®] para Gerar Fórmulas

No MATLAB existe um comando `latex(expr)` que gera uma representação em \LaTeX da expressão simbólica `expr`. Exemplo:

```
>> syms x, expr1=sin(x)^2+cos(x)^2;
>> latex(expr1)
ans = \left(\sin(x)\right)^2+\left(\cos(x)\right)^2
>> A=[1,2;3,4];
>> A=sym(A);
>> latex(A)
ans = \left[\begin{array}{cc}1&2\\\noalign{\medskip}3&4\end{array}\right]
```

Depois, basta marcar e copiar para a área de transferência, a representação em \LaTeX obtida e finalmente colar no arquivo fonte de \LaTeX que você está trabalhando.

5 Figuras

Podem ser construídos desenhos simples e diagramas com o \LaTeX . Além disso, podem ser inseridas figuras ou imagens geradas por outros programas.

5.1 Ambiente `picture`

As figuras criadas com o \LaTeX são construídas dentro do ambiente `picture`, cuja sintaxe é a seguinte:

```
\setlength{\unitlength}{unidade de comprimento}  
\begin{picture}(tam_x,tam_y)  
...  
\end{picture}
```

por exemplo,

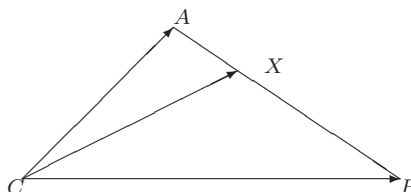
```
\setlength{\unitlength}{1.5 cm}  
\begin{picture}(3,4)  
...  
\end{picture}
```

cria uma figura de $3 \times 1.5 \text{ cm} = 4.5 \text{ cm}$ de largura por $4 \times 1.5 \text{ cm} = 6 \text{ cm}$ de altura. Os elementos da figura são posicionados usando-se o comando `\put(x_coord,y_coord){elemento}`. Os elementos podem ser

1. `\circle{diâmetro}` para uma circunferência e `\circle*{diâmetro}` para um círculo (cheio).

2. `\line(Δx,Δy){comprimento da projeção}` para um segmento de reta, onde Δx e Δy são números inteiros entre -6 e 6, primos entre si (isto é, não possuem divisores comuns) e **comprimento da projeção** é o comprimento da projeção do segmento no eixo x se este for diferente de zero ou o comprimento da projeção do segmento no eixo y , caso contrário.
3. `\vector(Δx,Δy){comprimento da projeção}` para um vetor, onde Δx , Δy e **comprimento da projeção** são como no item anterior.
4. Texto.

Exemplo:



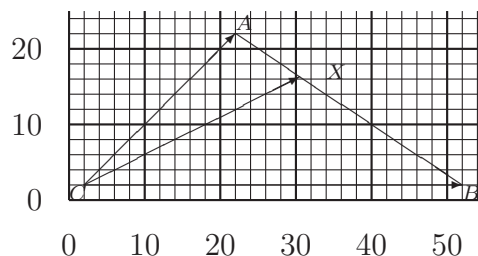
foi produzido com

```
\begin{center}
\setlength{\unitlength}{1mm}
\begin{picture}(55,25)
\put(0,0){\scriptsize$C$}
\put(2,2){\circle*{0.7}}
\put(2,2){\vector(1,0){50}} %segmento CB
```

```
\put(52,2){\circle*{0.7}}
\put(52,0){\scriptsize$B$}
\put(2,2){\vector(1,1){20}} %segmento CA
\put(22,22){\circle*{0.7}}
\put(22,22.5){\scriptsize$A$}
\put(2,2){\vector(2,1){28.5}} %segmento CX
\put(34,16){\scriptsize$X$}
\put(30.5,16.25){\circle*{0.7}}
\put(22,22){\line(3,-2){30}} %segmento AB
\end{picture}
\end{center}
```

A espessura das linhas pode ser especificada com o comando `\linethickness{espessura}`. Por exemplo, o comando `\linethickness{1.5mm}` faz com que todas as linhas desenhadas depois deste comando tenham a espessura de 1.5 mm. Também existem os comandos com espessuras pré-definidas: `\thicklines` para linhas grossas e `\thinlines` para voltar às linhas finas.

Para fazer o desenho você pode contar com a ajuda de um papel milimetrado colocado no fundo da figura durante o processo de construção do desenho e que depois pode ser retirado. Para isso você tem que colocar no preâmbulo do documento `\usepackage{graphpap}` e dentro do ambiente `picture` o comando `\graphpaper[tamanho](x0,y0)(tam_x,tam_y)`. Por exemplo



foi produzido com

```
\begin{center}
\setlength{\unitlength}{1mm}
\begin{picture}(55,25)
\graphpaper[2](0,0)(55,25)
. . .
\end{picture}
\end{center}
```

5.2 Figuras Produzidas por Outros Programas

Podem ser inseridas figuras geradas por outros programas. É preciso que no preâmbulo esteja o comando

```
\usepackage{graphicx}
```

As figuras devem estar nos formatos **ps** ou **eps**. Caso contrário deve-se usar algum programa que converta do formato original para **ps** ou **eps**. Um exemplo de tal programa é o **jpeg2ps** que é grátis e converte de **jpeg** para **ps** sem descomprimir a figura.

O comando para colocar a figura é


```
\includegraphics[height=altura]{nome do arquivo}
```

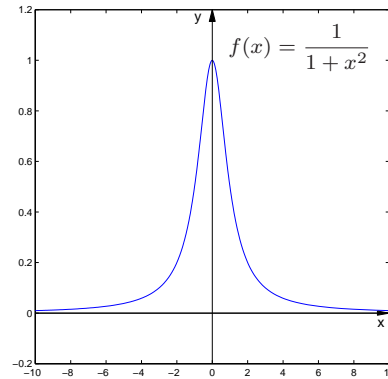


Figura 1: Figura proveniente do arquivo `exemgraf.ps`

Por exemplo: a Figura 1 foi inserida com

```
\includegraphics[height=5cm]{exemgraf.ps}
```

No MATLAB[®] foram usados os comandos seguintes para gerar esta figura:

```
>> syms x  
>> f=1/(1+x^2)  
f = 1/(1+x^2)  
>> plotf1(f, [-10,10])
```

```
>> eixos
>> print -dpsc d:\users\regi\ps\exemgraf.ps
```

Os comandos `plotf1` e `eixos` são comandos do pacote GAAL que está disponível na web na página do autor deste texto. Você pode colocar texto escrito no L^AT_EX por cima da figura colocando um ambiente `picture` de tamanho zero antes do comando `\includegraphics`. Na figura 1 colocamos $f(x) = \frac{1}{1+x^2}$ com os comandos

```
\begin{center}
\setlength{\unitlength}{1mm}
\begin{picture}(0,0)
%\graphpaper[2](0,0)(55,55)
\put(30,43){\scriptsize $f(x)=\displaystyle\frac{1}{1+x^2}$}
\end{picture}
\includegraphics[height=5cm]{exemgraf.ps}
\end{center}
```

foi usado o papel quadriculado que introduzimos anteriormente e depois da figura pronta o comando foi comentado.

Apenas um problema. As figuras geradas pelo MATLAB[®] não são transparentes. Para torná-las transparentes você deve abrir o arquivo da figura (no nosso caso `exemgraf.ps`) num editor de textos (por exemplo no T_EXShell). Procure por uma linha que termina com `PR` e comente-a acrescentando `%` no início. Se numa linha próxima tiver uma linha terminando com `MP` também comente-a. Finalmente salve o arquivo.

5.3 Ambiente figure

Este ambiente oferece a possibilidade da figura, que é produzida com o ambiente `picture` ou importada com o comando `\includegraphics`, ter uma legenda e além disso, flutuar no texto se colocando num local em que a perda de espaço seja mínima. A sintaxe é a seguinte:

```
\begin{figure}[onde]
figura
\caption{Texto da figura}
\end{figure}
```

o argumento `onde` especifica o local, onde é permitido a colocação da figura, ou seja, consiste de zero a quatro letras com os seguintes significados:

`h` aqui.

`t` em cima.

`b` em baixo.

`p` em uma página separada.

As letras devem ser colocados na ordem de prioridade. Se nenhum argumento for dado, `LATEX` assume a combinação `tbp`. Por exemplo a Figura 1 foi obtida com

```
\begin{figure}[htb]
\begin{center}
\setlength{\unitlength}{1mm}
\begin{picture}(0,0)
```

```
%\graphpaper[2](0,0)(55,55)
\put(30,43){\scriptsize $f(x)=\displaystyle\frac{1}{1+x^2}$}
\end{picture}
\includegraphics[height=5cm]{exemgraf.ps}
\end{center}
\caption{Figura proveniente do arquivo \texttt{exemgraf.ps}}
\end{figure}
```

O \LaTeX coloca a figura seguindo certos critérios. O comando `\clearpage` força a que todas as figuras pendentes (isto é, as que não foram impressas até o aparecimento do comando) sejam impressas.

6 Partes do Documento

6.1 Título

O título do documento é gerado com os comandos:

```
\title{texto do título}  
\author{autor1\\endereço1 \and autor2\\endereço2}  
\maketitle
```

Com estes comandos também a data atual é impressa. O comando `\date{texto}` coloca o `texto` no lugar da data atual. O comando `\thanks{texto de rodapé}` pode ser colocado no texto do título, do autor ou da data, produzindo uma marca onde o comando aparece e escreve `texto de rodapé` como texto de rodapé. Se o tipo de documento for `book`, então os comandos dados acima geram uma página separada para o título do documento.

6.2 Seções e Capítulos

O L^AT_EX dispõe dos comandos seguintes para produzir um seccionamento automático e seqüencial:

```
\part{título} \chapter{título} \section{título}  
\subsection{título} \subsubsection{título}
```

O comando `\part` está disponível apenas para o tipo de documento `book`. O comando `\chapter` está disponível apenas para os tipos de documento `book` e `report`. Os outros comandos de seccionamento estão disponíveis para todos os tipos de documento. Estes comandos formam um hierarquia. Os capítulos são produzidos com `\chapter`, que são divididos em seções com `\section`, que são divididas em subseções com `\subsection` e assim por diante. Se o tipo

de documento for `book`, então o comando `\chapter` sempre começa o capítulo em uma nova página e normalmente, nas páginas de números ímpares, gerando uma página em branco se necessário. Para se evitar que ele gere uma página em branco, inicie o documento com o comando `\documentclass[... ,oneside]{book}` ou com `\documentclass[... ,openany]{book}`. No primeiro caso o \LaTeX entende que o documento será impresso em somente um lado do papel e no segundo, que os capítulos podem ser iniciados em páginas pares também.

As páginas que vierem depois do comando `\frontmatter` e antes do comando `\mainmatter` serão numeradas com algarismos romanos. As páginas que vierem depois do comando `\mainmatter` serão numeradas em arábicos e recomeçam da página 1. Os capítulos que vierem depois do comando `\backmatter` não serão numerados, assim como os que vierem entre `\frontmatter` e `\mainmatter`.

6.3 Conteúdo

A tabela de conteúdo do documento (ou índice ou sumário) é gerado automaticamente com o comando `\tableofcontents`. Para que o conteúdo seja gerado de forma correta é necessário que o documento seja processado pelo \LaTeX pelo menos três vezes. Se no preâmbulo do seu documento contiver `\usepackage[brazil]{babel}`, então o título do conteúdo será “Sumário”. Se você quiser mudar este nome basta colocar o comando `\renewcommand{\contentsname}{novo nome}` antes de `\tableofcontents`.

6.4 Bibliografia

Para se gerar uma bibliografia em um documento, é necessário criar um arquivo na pasta onde está o arquivo fonte com nome terminado em `.bib` (por exemplo, `bibli.bib`) contendo um banco de dados bibliográficos. As entradas deste banco de dados devem ter a forma:

```
@TIPO{apelido,  
  AUTHOR = "Fulano de Tal",  
  TITLE = "O Título Tal",  
  PUBLISHER = "A Editora",  
  ADDRESS = "Cidade Tal",  
  . . .  
  YEAR = "1998" }
```

onde o TIPO pode ser BOOK (para livros), ARTICLE (para artigos), INCOLLECTION (para partes de um livro), PROCEEDINGS (para artigo em atas de congressos) ou TECHREPORT (para relatórios técnicos). A publicação vai ser citada no documento com o comando `\cite{apelido}`. Cada TIPO de entrada tem um conjunto de campos obrigatório e outro conjunto de campos opcional. Além do banco de dados bibliográficos, é necessário colocar no preâmbulo do arquivo fonte o comando `\bibliographystyle{plain}` e no local do documento onde a bibliografia deve aparecer deve-se colocar o comando `\bibliography{arquivo}`, onde `arquivo` é o nome do arquivo de banco de dados bibliográficos sem a terminação `.bib`. Nem todas as entradas do banco de dados vão aparecer na bibliografia do documento. Somente aquelas que tenham sido referenciadas no texto com o comando `\cite{apelido}` e as que não foram, mas que em qualquer local no texto (antes do comando `\bibliography{arquivo}`) apareceu o comando `\nocite{apelido1,apelido2,...}`.

No T_EXShell, depois de processar com o L^AT_EX o seu arquivo principal, você deve ir no menu em **TeX** e clicar em **BiBTeX**. Com isto um programa chamado BiB_TE_X lê o seu banco de dados, retira as entradas que foram referenciadas no texto e cria um arquivo com o mesmo nome do arquivo principal, mas com terminação `.bbl`. Este arquivo contém as publicações citadas com o comando `\cite{apelido}` e as que apareceram com o comando `\nocite{apelido1,apelido2,...}`, ordenadas e formatadas, para que sejam automaticamente

incluídas no seu documento no local, onde estiver o comando `\bibliography{arquivo}`, através do processamento do \LaTeX sobre o arquivo principal. Depois, é preciso que o arquivo principal seja processado pelo \LaTeX pelo menos duas vezes para que a bibliografia apareça de forma correta no seu documento.

Por exemplo, as entradas

```
@BOOK{kopka-daly,  
AUTHOR = "Helmut Kopka and Patrick W. Daly",  
TITLE = "A Guide to \LaTeX",  
PUBLISHER = "Addison-Wesley",  
ADDRESS = "Wokinghan-England", YEAR = "1994"}  
@ARTICLE{golub79,  
AUTHOR = "G. H. Golub and M. T. Heath and G. Wahba",  
TITLE = "Generalized cross-validation as a method for choosing a good ridge  
parameter",  
JOURNAL = "Technometrics",  
VOLUME = "21", PAGES = "215-223", YEAR = "1979"}
```

no arquivo de banco de dados bibliográficos e a colocação do comando `\cite{kopka-daly}` produz [2]. E o comando

```
\bibliography{bibli}
```

vai gerar algo semelhante a

Referências

[2] Helmut Kopka and Patrick W. Daly. *A Guide to \LaTeX* . Addison-Wesley, Wokinghan-England, 1994.

Se você não gostar do título, pode mudar colocando antes do comando `\bibliography{bibli}` o comando `\renewcommand{\bibname}{novo nome}` no estilo `book` e `\renewcommand{\refname}{novo nome}` nos estilos `article` e `report`

Para que a bibliografia apareça na tabela de conteúdo (ou sumário ou índice) é necessário que se coloque o comando `\addcontentsline{toc}{section}{Sumário}` ou `\addcontentsline{toc}{chapter}{Sumário}` imediatamente antes de `\bibliography{bibli}`.

6.5 Índice Alfabético

A criação de um índice alfabético é semelhante a de criação da bibliografia. As palavras que devem aparecer no índice devem ser indicadas com o comando `\index` que deve ter uma das três formas:

```
\index{entrada principal}
\index{entrada principal!subentrada}
\index{entrada principal!subentrada!subsubentrada}
```

As entradas não podem conter os caracteres `!`, `@` e `|`. Se uma entrada contém um comando `LATEX` ou uma palavra acentuada, então ela deve aparecer na forma `entrada alfabética@entrada impressa`. Exemplo:

```
\index{Angulo@Ângulo!entre retas}
\index{Conjunto R@Conjunto  $\mathbb{R}$ }.
```

Para que os comandos `\index` sejam considerados é necessário que no preâmbulo do arquivo principal, esteja o comando `\makeindex`. No local onde o índice vai ser impresso em seu documento deve conter o comando `\printindex`.

Se você não gostar do título do Índice, pode mudar colocando antes do comando `\printindex` o comando `\renewcommand\indexname{novο nome}`.

No `TeXShell`, depois de processar o seu arquivo principal com o `LaTeX` você deve ir no menu em **TeX** clicar em **MakeIndex** para criar um arquivo com o mesmo nome do arquivo principal, mas com terminação `.ind`, que vai ser incluído automaticamente, onde for colocado o comando `\printindex`, através do processamento do `LaTeX` sobre o arquivo principal. É necessário que o arquivo principal seja processado pelo menos duas vezes pelo `LaTeX` para que o índice apareça de forma correta no seu documento.

Para que o índice alfabético apareça na tabela de conteúdo (ou sumário ou índice) é necessário que se coloque o comando `\addcontentsline{toc}{section}{Índice}` ou `\addcontentsline{toc}{chapter}{Índice}` no arquivo com o mesmo nome do arquivo principal, mas com terminação `.ind` que foi criado.

6.6 Apêndice

O comando `\chapter*{Apêndice}` cria um capítulo que não é numerado, entretanto também não aparece no índice. Para que apareça no índice depois do comando anterior deve-se usar o comando `\addcontentsline{toc}{chapter}{Apêndice}`.

7 Outros Recursos

7.1 Caracteres Especiais

Aspas

São usados comandos diferentes para as aspas do início e as do final. Para as aspas do início usa-se duas crases ‘ ‘ e para as aspas do final, dois apóstrofes ’ ’. Exemplo: “aspas” é conseguida com ‘ ‘aspas’ ’.

(Símbolo de) Euro

Coloque no preâmbulo

```
\usepackage[official]{eurosym}
```

e use o comando `\euro` para imprimir €.

(Símbolo de) Graus

Use o comando `$^\circ` como em 30°C que foi obtido com `$30^\circ\mathrm{C}`.

Números Ordinais

Use o comando `\do` como em 1º que foi obtido com `1\do`.

7.2 Referências Cruzadas

O comando `\label{marca}` coloca uma marca naquele ponto do texto, onde ele aparece e pode ser usado para se referir a ele em outra parte do texto com o comando `\ref{marca}`.

O número da página do ponto onde a `marca` foi colocada pode ser impresso com o comando `\pageref{marca}`.

O identificador da marca `marca` pode ser qualquer conjunto de caracteres, que não inclua os caracteres especiais, que o faça lembrar a que ela se refere.

Se o comando `\label{marca}` é colocado dentro de um ambiente como `equation`, `eqnarray` ou `newtheorem`, então o comando `\ref{marca}` correspondente vai imprimir o número do ambiente em que a `\label{marca}` foi colocado. Para o ambiente `enumerate`, o comando `\ref{marca}` imprime o número do `\item`, onde `\label{marca}` foi colocado. Para o ambiente `figure` o comando `\label{marca}` deve ser colocado dentro do `\caption{...}`.

Por exemplo, no Teorema de Pitágoras escrevemos
`\begin{teo}[Pitágoras]\label{teo.pita} ... \end{teo}`
 e para nos referirmos a ele escrevemos
`...pelo Teorema \ref{teo.pita} na página \pageref{teo.pita} ...`
 e obtemos
`... pelo Teorema 3.1 na página 18 ...`

Para se referir à Figura 1 que está na página 41 foi necessário ter escrito dentro do ambiente `figure` da seguinte forma `\caption{\label{exemgraf}Figura proveniente do ...}`.

Para se referir a uma seção ou subseção é necessário a colocação do comando `\label{marca}` junto do nome da seção ou da subseção da seguinte forma
`\section{nome da seção\label{marca}}` ou
`\subsection{nome da subseção\label{marca}}`.

Por exemplo, no início desta subseção escrevemos
`\subsection{Referências Cruzadas\label{subsecrefcruz}}` e para nos referirmos a ela escrevemos
 A subseção `\ref{subsecrefcruz}` começou na página `\pageref{subsecrefcruz}`

e obtemos

A subseção 7.2 começou na página 51.

7.3 Espaçamento

7.3.1 Espaçamento Horizontal

Para produzir espaçamento horizontal usamos os comandos

```
\hspace{tamanho}
```

```
\hspace*{tamanho}
```

onde o `tamanho` é o comprimento do espaçamento, por exemplo `1.5cm` ou `3em` (aqui `em` é a unidade que corresponde a largura da letra M do tipo de letra atual). A forma sem `*` suprime o espaço se ele ocorre entre duas linhas ou no início de uma linha. O `tamanho` pode ser negativo, neste caso o comando funciona como um retrocesso.

Para retirar o recuo que é dado no início dos parágrafos pode ser usado o comando `\noindent` no início do parágrafo. O tamanho do recuo pode ser alterado com o comando `\setlength{\parindent}{tamanho}` onde `tamanho` é o seu comprimento, por exemplo `\setlength{\parindent}{0.5cm}` faz com que os próximos parágrafos tenham um recuo de 0.5 cm.

O comando `\hfill` é uma abreviação de `\hspace{\fill}`, que introduz espaço suficiente para que o que estiver à esquerda do comando fique alinhado à esquerda e o que estiver à direita do comando seja alinhado à direita. Por exemplo, com `Esquerda\hfill Direita` obtemos

EsquerdaDireita.

Várias ocorrências de `\hfill` dentro de uma linha faz com que o mesmo espaçamento seja introduzido de forma que a linha fique justificada à esquerda e à direita. Por exemplo, com

Esquerda\hfill Centro \hfill Direita obtemos

Esquerda Centro Direita

Se `\hfill` ocorre no início da linha, então o espaçamento é suprimido em concordância com o comportamento de `\hspace`. Neste caso, deve ser usado `\hspace*{\fill}`.

7.3.2 Espaçamento Vertical

Uma nova linha ou uma quebra de linha com ou sem espaço adicional pode ser conseguida com o comando

`\\[tamanho]`

O argumento opcional `tamanho` é o comprimento que especifica qual o espaço adicional que deve ser colocado entre as duas linhas. Se for desejado a diminuição do espaço o comando pode ser usado com um `tamanho` negativo. Por exemplo, com o comando `\\[10cm]`, a linha atual é terminada e um espaço vertical de 10 cm é inserido antes da nova linha. Enquanto, `\\[-0.38cm]`, quebra a linha atual e a nova linha ocorre a -0.38 cm de onde ela normalmente começaria.

Comandos para aumentar o espaçamento vertical entre parágrafos são:

`\bigskip` `\medskip` `\smallskip`

que adicionam espaço vertical dependendo do tamanho do tipo de letras declarado no comando `\documentclass`.

É possível, também, adicionar espaço extra entre parágrafos com o comando

`\vspace{tamanho}`

Se este comando ocorre dentro de um parágrafo, então o espaço adicional é introduzido após a linha atual. O parâmetro `tamanho` tem que conter uma unidade de comprimento e pode ser negativo, para mover o texto que vem a seguir para cima.

O comando `\vfill` é uma abreviação de `\vspace{\fill}`. Este é análogo a `\hfill` (introduzido acima), para espaçamento vertical, introduzindo espaçamento vertical suficiente para que a parte superior e a inferior do texto casem com as margens superior e inferior da página respectivamente.

O comando `\baselineskip tamanho` depois do `\begin{document}` estabelece o tamanho do espaçamento entre as linhas. O parâmetro `tamanho` tem que conter uma unidade de comprimento.

O comando `\newpage` pode ser usado para forçar uma mudança de página.

7.4 Linha Horizontal

Dois comandos que funcionam exatamente como `\hfill` são `\dotfill` e `\hrulefill`, que preenchem o espaço com pontos e linha respectivamente. Por exemplo, com `Esquerda\dotfill Centro \hrulefill Direita` obtemos

EsquerdaCentro _____Direita

O comando `\noindent\hrulefill` produz uma linha horizontal que ocupa toda a extensão do texto.

7.5 Retângulo Sólido

O comando

```
\rule[deslocv]{largura}{altura}
```

produz um retângulo preto. O deslocamento vertical `deslocv` é opcional e pode ser usado para deslocar o retângulo para cima se positivo e para baixo se negativo. Por exemplo,

```
\rule[1mm]{2cm}{1mm} \rule[-1mm]{2cm}{1mm} \rule{2cm}{1mm}
```

produz



7.6 Texto Emoldurado

Aqui temos uma caixa de parágrafo de 10 cm dentro de uma moldura dentro de uma segunda moldura.

O texto emoldurado acima foi obtido com

```
\begin{center}
\fbbox{\fbbox{
\parbox{10cm}{
  Aqui temos uma caixa de parágrafo
de 10 cm dentro de uma moldura dentro de uma segunda moldura. } } }
\end{center}
```

Um `\parbox` dentro de um `\fbbox` tem o efeito de criar um parágrafo que é emoldurado.

7.7 Dividindo o Arquivo Fonte

Para a produção de textos maiores é essencial, que se divida o arquivo fonte em arquivos menores e que possam ser processados em separado para diminuir o tempo de processamento. Para isso, pode ser empregado o comando


```
\include{arquivo}
```

junto com o comando

```
\includeonly{lista de arquivos}
```

no preâmbulo, contendo uma **lista de arquivos** que devem ser lidos pelo L^AT_EX. Os nomes dos arquivos na lista de arquivos devem ser separados por vírgula e a terminação `.tex` deve ser suprimida. Se o comando `\includeonly` for suprimido, então todos os arquivos são lidos e processados.

Os capítulos podem ser escritos em arquivos separados com nomes `cap1.tex`, `cap2.tex`, `cap3.tex`, O arquivo principal contém o texto

```
\documentclass[a4paper,12pt]{book}
```

```
. . . . .
```

```
\includeonly{. . .}
```

```
. . . . .
```

```
\begin{document}
```

```
\maketitle
```

```
\tableofcontents
```

```
\include{cap1}
```

```
\include{cap2}
```

```
\include{cap3}
```

```
. . . . .
```

```
\end{document}
```

Colocando uma entrada apropriada no comando `\includeonly` é possível o processamento de alguns capítulos seletivamente. Por exemplo, `\includeonly{cap1,cap2}` faz com que o L^AT_EX processe o título, o conteúdo e os capítulos contidos nos arquivos `cap1` e `cap2`.

7.8 Tabelas

Para construir tabelas usamos o ambiente `tabular`, cuja sintaxe é

```
\begin{tabular}{formato das colunas}  
... linhas ...  
\end{tabular}
```

Aqui, o `formato das colunas` deve ser uma entrada para cada coluna além, opcionalmente, de entradas extras para as bordas esquerda e direita e para a separação das colunas. Os símbolos possíveis para o `formato das colunas` são:

`l` a coluna é alinhada à esquerda;

`r` a coluna é alinhada à direita;

`c` a coluna é centralizada.

`p{tamanho}` o texto desta coluna é escrito em linhas de comprimento igual a `tamanho`. O `tamanho` deve conter a unidade, por exemplo `5cm`.

e para as bordas e a separação das colunas:

`|` desenha uma linha vertical;

`||` desenha duas linhas verticais;

`@{texto}` insere o `texto` em todas as linhas, entre as duas colunas onde ele aparece.

As `linhas` devem conter as entradas de cada linha da tabela separadas por `&` e terminadas por `\\`. Além disso, elas podem conter os comandos:

`\hline` - este comando deve somente aparecer antes da primeira linha ou imediatamente depois de `\\`. Ele faz desenhar uma linha horizontal de comprimento igual ao da tabela. Dois comandos `\hline` juntos produzem duas linha horizontais com um pequeno espaço entre elas.

`\multicolumn{num}{formato}{texto}` - este comando faz concatenar `num` colunas em uma só. O `formato` deve conter um dos caracteres `l`, `r` ou `c` antecedido e seguido por `|` ou `||`.

A tabela

Horário de Tópicos em Matemática - MAT 037/033					
Horário	Seg	Ter	Qua	Qui	Sex
13:00-14:40					
14:55-16:35					TURMA N
16:35-18:15	TURMA N		TURMA N		
18:15-19:00					
19:00-20:40			TURMA M		
20:55-22:35	TURMA M				

foi produzida com

```
\begin{center}
\begin{tabular}{|l||c|c|c|c|c|}
\hline
\multicolumn{6}{|c|}{\textbf{Horário de Tópicos em Matemática - MAT 037/033}}\\
\hline
```

```

Horário      &Seg  &Ter  &Qua  &Qui  &Sex\\
\hline\hline
13:00-14:40&      &      &      &      &      \\
\hline
14:55-16:35&      &      &      &TURMA N      \\
\hline
16:35-18:15&TURMA N      &      &TURMA N      &      &      \\
\hline
18:15-19:00&      &      &      &      &      \\
\hline
19:00-20:40&      &      &TURMA M      &      &      \\
\hline
20:55-22:35&TURMA M      &      &      &      &      \\
\hline
\end{tabular}
\end{center}

```

Se desejar que em alguma coluna cada linha da tabela tenha mais de uma linha de texto, basta usar no formato das colunas o formato `p{tamanho}` com o tamanho pequeno o suficiente para forçar o \LaTeX a mudar de linha.

Para que a tabela apareça no formato paisagem coloque no preâmbulo

```
\usepackage{lscape}
```

e use os comandos a seguir

```
\begin{landscape}
```

```
\begin{table}
.
.
.
\end{table}
\end{landscape}
```

Por exemplo

```
\begin{landscape}
\begin{table}
\begin{center}
\begin{tabular}{|l||c|c|c|c|c|}
\hline
\multicolumn{6}{|c|}{\textbf{Horário de Tópicos em Matemática - MAT 037/033}}\\
\hline
Horário & Seg & Ter & Qua & Qui & Sex\\
\hline\hline
13:00-14:40 & & & & & \\
\hline
14:55-16:35 & & & & TURMA N & \\
\hline
16:35-18:15 & TURMA N & & TURMA N & & \\
\hline
18:15-19:00 & & & & & \\
\hline
```

```

19:00-20:40&      &      &TURMA M      &      &      \\
\hline
\end{tabular}
\end{center}
\end{table}
\end{landscape}

```

produz a seguinte página

Horário de Tópicos em Matemática - MAT 037/033					
Horário	Seg	Ter	Qua	Qui	Sex
13:00-14:40					
14:55-16:35					TURMA N
16:35-18:15	TURMA N		TURMA N		
18:15-19:00					
19:00-20:40			TURMA M		

7.9 Duas Colunas

O comando `\twocolumn[título]` termina a página atual e inicia uma nova página com duas colunas por página. O argumento opcional `título` é escrito no início da página em uma coluna com a largura da página. O comando `\onecolumn` termina o modo de duas colunas.

Para se escrever somente dois parágrafos um do lado do outro pode ser usado o ambiente `minipage`. Por exemplo para se produzir os parágrafos seguintes:

Parágrafo do lado esquerdo.	Parágrafo do lado direito.
-----------------------------	----------------------------

foram usados os comandos

```
\begin{minipage}{7cm}
```

Parágrafo do lado esquerdo.

```
\end{minipage}\hfill
```

Parágrafo do lado direito.

```
\end{minipage}
```

7.10 Traduzindo para PS e PDF

Quando você processa o seu arquivo `.tex` com o \LaTeX é gerado um arquivo `.dvi` que pode ser impresso ou visualizado. Você pode querer gerar um arquivo `.ps` que pode ser visualizado usando o **Ghostview** ou então um arquivo `.pdf` que pode ser visualizado usando o **Acrobat Reader**.

No \TeX Shell para passar para PS basta clicar no botão **DVIPS**. Depois você pode visualizar o arquivo `.ps` clicando no botão **GhostView**. Neste momento você pode passar para PDF clicando no botão **PS2PDF**. Se o Acrobat Reader estiver aberto com um arquivo de mesmo nome que o seu arquivo principal ele deverá ser fechado, antes de rodar o **PS2PDF**.

Você pode adicionar recursos de hipertextos ao seu documento usando o pacote `hyperref` de Sebastian Rahtz. Para isso coloque no preâmbulo do seu documento

```
\usepackage[dvips,colorlinks,linkcolor=blue,hyperindex]{hyperref}
```

Este comando tem o efeito colateral das figuras .ps às vezes não aparecerem quando se visualiza usando o YAP. Para que as figuras apareçam no YAP, mas os links funcionem apenas no arquivo .dvi você pode usar o comando

```
\usepackage[colorlinks,linkcolor=blue,hyperindex]{hyperref}
```

Além dos comandos usuais do \LaTeX o pacote `hyperref` oferece outros comandos. Um deles que é muito útil é `\href{endereço url}{texto}` que coloca hyperlinks a endereços da web. Por exemplo,

<http://www.mat.ufmg.br/~regi>
foi produzido com

```
\href{http://www.mat.ufmg.br/~regi}{\texttt{http://www.mat.ufmg.br/~{}regi}}
```

Você pode encontrar mais informações no manual do pacote `hyperref` que está no seu computador. Se você está usando **MikTeX**, ele está em

C:\Program Files\MikTeX x.y\doc\latex\hyperref

Aqui x.y é o número da versão do **MikTeX**.

7.11 Mais Informações na Internet

A fonte de informações sobre \LaTeX na internet é o site do CTAN (Comprehensive \TeX Archive Network) <http://www.ctan.org>. Por exemplo, uma introdução não tão curta ao \LaTeX , pode ser encontrada em

<http://www.tug.org/texlive/Contents/live/texmf-doc/doc/portuguese/lshort-portuguese/ptlshort.pdf>.

Para quem tem que seguir as normas da ABNT na formatação do texto a página do grupo ABN \TeX em <http://abntex.codigolivre.org.br> pode ser muito útil.

Referências

- [1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L^AT_EX Companion*. Addison-Wesley Publishing Company, New York, 1993.
- [2] Helmut Kopka and Patrick W. Daly. *A Guide to L^AT_EX*. Addison-Wesley, Wokingham-England, 3a. edição, 1999.
- [3] Leslie Lamport. *L^AT_EX, A Document Preparation System*. Addison-Wesley, Reading, 2a. edição, 1994.