Mini-curso de LaTeX

Diogo Leite Rebouças
Luiz Paulo de Souza Medeiros
{diogolr,luizmedeiros}@dca.ufrn.br

26 a 29 de Julho de 2010

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 TEX

Em 1977, tentando explorar o potencial dos equipamentos digitais de impressão que começavam a surgir e também, com o desejo de reverter o processo de deteriorização da qualidade tipográfica que afetava seus próprios livros, Donald E. Knuth começou a escrever um programa de computador para processamento de textos e fórmulas matemáticas. Esse programa, conhecido como TEX(pronuncia-se téc), teve sua principal versão lançada em 1982, mas só em 1989, ao serem adicionados alguns recursos, o TEX passou a suportar melhor os caracteres de 8 bits.

O TEX é conhecido por ser extremamente estável, por funcionar em muitos tipos diferentes de computadores e por ser "livre de erros".

1.2 LATEX

LATEX (pronuncia-se latéc) é um pacote de macros que permite aos autores processar e imprimir seus trabalhos com a mais alta qualidade tipográfica, usando um layout profissional previamente definido. O LATEX foi escrito originalmente por Leslie Lamport e usa o formatador TEX como seu mecanismo de processamento.

Em 1994 Frank Mittelbach liderou uma equipe, conhecida por LATEX3, que melhorou o LATEX desenvolvido por Lamport, corrigindo bugs e incrementando macros das versões anteriores. Para distinguir a nova versão da anterior ela é chamada de LATEX 2ε.

Capítulo 2

Noções Básicas

Em um ambiente LaTeX, o LaTeX toma o lugar do diagramador e usa o TeX para a edição gráfica. Mas o LaTeX é apenas um programa e, desta forma, precisa receber mais informações. Ou seja, o autor deve fornecer informações adicionais que descrevam a lógica de seu trabalho. Todas essas informações são escritas no texto como comandos LaTeX.

Podemos dividir os programas de processamento de texto em duas classes. A primeira é a dos processadores de texto, nos quais existe um menu na tela apresentando os recursos que podem ser usados quando selecionados com o mouse. Depois de selecionado um recurso, o texto digitado na tela tem sua estrutura "visual" modificada e pode ser visualizado imediatamente pelo usuário, o que permite que este tenha certeza que seu texto será impresso corretamente. Esse método é chamado "what you see is what you get", ou simplesmente WYSIWYG e um exemplo clássico deste tipo de processadores de texto é o M$ Word.

Na segunda classe, à qual pertence o LaTeX, o processamento de texto é feito em duas etapas distintas. O texto a ser processado e os comandos de formatação são escritos em um arquivo fonte com o uso de um editor de textos simples (vim, gedit, kate etc.). Em seguida o arquivo fonte é submetido a um compilador, no nosso caso, o LaTeX, que gera um arquivo de saída que pode ser impresso ou visualizado na tela. Esses comandos definem o tipo de letra, a formatação do texto, símbolos especiais etc. Todos os comandos inicial com uma contra-barra (\).

2.1 Vantagens e desvantagens

Quando os adeptos do WYSIWYG encontram-se com os adeptos do LaTeX, geralmente uma discussão em defesa de cada um destes tipos de processadores é iniciada. A melhor coisa a se fazer nesse caso, por incrível que pareça, é não argumentar muito. Mas como ninguém é de ferro, eis uma munição para suas discussões a favor do LaTeX:

- Vários *layouts* criados por profissionais estão disponíveis na rede para serem baixados. Estes realmente deixam os textos com "cara de uma gráfica".
- O processamento de fórmulas matemáticas é suportado de uma maneira extremamente conveniente. Fórmulas do tipo $\int_0^t e^{-x^2} dx$ podem ser escritas facilmente como `\int_0^t e^{-x^2}dx`.
- Os usuários precisam aprender apenas poucos comandos de fácil compreensão, porém nunca se preocupam com o *layout* do documento.
- Até mesmo estruturas complexas como: notas de rodapé, referências, índices, lista de figuras, lista de tabelas e bibliografias são criadas com poucos comandos.

- Existem vários pacotes de atualização para muitas tarefas ainda não suportadas pelo LaTeX básico. Por exemplo, pacotes para inclusão de gráficos POSTSCRIPT ou para criar bibliografias de acordo com uma norma ou padrão.
- O LaTeX encoraja os autores a escreverem textos bem estruturados, pois é assim que ele funciona – por especificação de estrutura.
- O TeX, o mecanismo de formatação do LaTeX 2ε, é extremamente portável e gratuito. Consequentemente o sistema funciona em quase todos os modelos de hardware disponíveis.

O LaTeX também possui algumas desvantagens, mas é realmente muito complicado achar alguma que seja realmente importante. Com certeza outras pessoas poderão citar algumas delas `;)`.

- O LaTeX não funciona muito bem com pessoas que venderam sua alma...
- Embora alguns parâmetros possam ser ajustados em um *layout*, o desenvolvimento de um *layout* inteiro é difícil e demanda muito tempo[1].
- É muito difícil escrever documentos desestruturados e/ou desorganizados.
- Seu hamster pode, apesar de alguns passos iniciais encorajadores, nunca ser capaz de entender o conceito de "Marcadores Lógicos"

2.2 Arquivos de entrada do LaTeX

A entrada para o LaTeX é um arquivo de texto ASCII que pode ser criado por qualquer editor de texto simples, como dito anteriormente. Ele deverá conter o texto e os comandos que irão dizer ao LaTeX como processá-lo.

2.2.1 Espaços

Caracteres de espaçamento (espaço/tabulação/quebra de linha) são tradados uniformemente como "espaço" pelo LaTeX. Ou seja, muitos caracteres consecutivos de espaço ou tabulação são considerados como apenas um único espaço, bem como o uso de apenas uma quebra de linha. No entanto, quando usamos várias quebras de linha consecutivas, estamos definindo um fim de parágrafo. No modo matemático, é possível diminuir os espaços em branco usando `\!`.

Existem ainda dois outros comandos de espaçamentos. São eles: `\vspace{...}` e o `\hspace{...}`. Estes comandos produzem espaços verticais e horizontais, respectivamente, de acordo com seu argumento (normalmente dado em centímetros). Por exemplo: `\vspace{0.25cm}`.

2.2.2 Caracteres especiais

Os seguintes caracteres possuem uma característica diferente para o LaTeX:

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

[1] De acordo com rumores, a versão LaTeX3 facilitará um pouco em relação a isto

Quando inseridos em seu texto, ao invés de serem "escritos" como deveriam, farão com que o LaTeX faça coisas que não deseja. Para usar esses caracteres em seu texto, uma contra-barra (\) deverá ser adicionada antes de cada um deles. Por exemplo:

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

A única excessão é a própria contra-barra que quando usada dessa forma (\\) produz uma quebra de linha no texto a ser impresso. Portanto, para usar uma contra-barra em seu texto você deverá fazer:

```
\verb|\| ou $\backslash$
```

2.2.3 Comandos

Os comandos no LaTeX diferem maísculas de minúsculas (*case sensitive*) e seguem um dos dois formatos:

- Começam com uma contra-barra (\) e possuem um nome que consiste apenas em letras. Os comandos são terminados por um espaço, um número ou qualquer outro caractere "que não seja letra".
- Consistem de uma contra-barra e não mais do que UM caractere especial

Os comandos em LaTeX "consomem" os espaços os seguem. Se você deseja obter um espaço após um comando você precisa digitar {} ou \ após este. Desta forma, ficará explícito para o LaTeX que deverá existir um espaço após determinado comando.

2.2.4 Comentários

Quando o LaTeXencontra um %, ele ignora o restante da linha (considera um comentário). Assim sendo, você poderá estar usando este caractere especial para escrever notas ou dividir as seções de seu texto enquanto estiver digitando ou ainda para quebrar uma sequência de linhas em que não pode haver um parágrafo.

Para inserir comentários longos você pode usar o ambiente `comment`, definido no pacote verbatim (`\usepackage{verbatim}`):

```
\begin{comment}
...
\end{comment}
```

2.3 Estrutura do arquivo de entrada

Quando o LaTeX 2ε processa um arquivo de entrada, ele espera seguir uma certa estrutura. Dessa forma, temos uma sequência de comandos a obedecer.

Todo arquivo de entrada precisa começar com o comando

```
\documentclass
```

Este comando irá especificar que tipo de documento você pretende escrever. Após isso, você pode incluir comandos que influenciam o estilo de todo o documento ou carregar os pacotes que adicionam novos recursos ao sistema LaTeX. Para carregar os pacotes, você deve utilizar o comando

```
\usepackage
```

Após ter concluído a configuração de todo o documento, você deve iniciar o corpo do texto com o comando

```
\begin{document}
```

Assim como em qualquer outra linguagem de programação, todo comando `begin`, chave, parêntese ou ainda colchete aberto deve ser fechado. Portanto é altamente recomendado que você "feche" esse comando o quanto antes. O comando `\begin{document}` é "fechado" com um `\end{document}`. O corpo do texto deve estar contido entre esses dois últioms comandos. Qualquer coisa que siga o comando `\end{document}` será ignorada.

2.3.1 O preâmbulo

É considerado preâmbulo de um arquivo TeX tudo que estiver contido entre o comando `\documentclass` e o comando `\begin{document}`. A sintaxe do comando `\documentclass` é:

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

onde `opções` é um parâmetro não obrigatório, que pode conter informações sobre o tamanho da fonte, tipo de papel etc., e o parâmetro `classe` é obrigatório e define o tipo de texto (artigo, tese, livro, relatório...).

As classes mais comuns são: `article` (artigo), `report` (relatório ou tese), `book` (livro), `slides` (transparências) e `letter` (carta) e as opções possíveis são `a4paper` (papel de tamanho A4), `letterpaper` (papel de tamanho carta), `10pt` (fonte de tamanho 10 pontos - *default*), `twocolumn` (texto em duas colunas), `twoside` (impressão nos dois lados do papel) etc.

Por exemplo, um comando como:

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

define a classe artigo, em um papel de tamanho A4 com fonte de tamanho 12 (utilizado nesta apostila).

Um outro comando muito utilizado no preâmbulo de um documento LaTeX é o `\usepackage`. Ao carregar novos pacotes, a capacidade de formatação do LaTeX aumenta signficativamente. Por exemplo, o comando `\usepackage{graphicx}` permite que sejam inseridas figuras ou gráficos no texto, `\usepackage[latin1]{inputenc}` permite que sejam colocados acentos nas palavras do arquivo de entrada (dependendo do caso, `latin1` deve ser substituído por `utf8` ou outra forma de codificação de caracteres). Já o comando `\usepackage[brazil]{babel}` faz com que o LaTeX "fale português".

Capítulo 3

Formatando o texto

O principal ponto na escrita de um texto é trazer idéias, informação ou conhecimento para o leitor. O leitor, com certeza, entenderá melhor o texto se as idéias estiverem bem estruturadas.

Diferentemente dos outros sistemas de editoração, no LaTeX você informa qual deve ser a estrutura lógica do texto. A qualidade tipográfica promovida pelo LaTeX pode ser configurada de acordo com "regras" escritas nos arquivos de definição de classe do documento e em vários arquivos de estilo.

O LaTeX sempre tenta produzir as melhores quebras de linha possíveis. Se ele não consegue encontrar uma forma de quebrar as linhas de modo que continue seguindo os padrões ele permite que a linha ultrapasse a margem direita do parágrafo e avisa com uma mensagem (*overfull hbox*) enquanto compila o arquivo de entrada. Isso acontece muitas vezes quando o LaTeX não consegue encontrar um bom lugar para hifenizar uma palavra[1]. Para evitar isso, você pode instruir o LaTeX para diminuir seus padrões de espaçamento com o comando `\sloppy`. Ele evita que em linhas muito longas o espaço entre as palavras seja aumentado – mesmo que o resultado final não seja otimizado. Neste caso, o aviso *undefull hbox* é dado ao usuário. Em muitos casos o resultado não fica muito bom. O comando `\fussy` retorna o LaTeX a seu padrão.

3.1 Tipos e tamanhos das letras

Para alterar o tipo de escrita das letras, podem ser utilizados os comandos da tabela 3.1.

Comando	Resultado
`{\rm Romano}`	Romano
`{\bf Negrito}`	**Negrito**
`{\sl Inclinado}`	*Inclinado*
`{\sf Sans Serif}`	Sans Serif
`{\it Italico}`	*Itálico*
`{\sc Caixa Alta}`	CAIXA ALTA
`{\tt Monospace}`	`Monospace`

Tabela 3.1: Tabela de tipos de fonte.

[1] Embora o LaTeX dê um aviso quando isso ocorre (*overfull hbox*) e mostre a linha onde está o problema, estas linhas não são muito fáceis de achar. Se você usar a opção *draft* no comando *documentclass*, estas linhas serão marcadas com uma pequena linha preta na margem direita

Também poderiam ter sido utilizados os comandos `\textrm{...}`, `\textbf{...}`, `\textsl{...}`, etc. Esses comandos são equivalentes aos anteriores, entretanto permitem encadeamento de comandos (negrito e itálico).

Ao contrário dos outros processadores de texto, em que o tamanho da fonte é definido por um número, o LaTeX utiliza nomes para definir o tamanho das fonte. Os tamanhos existentes podem ser vistos na tabela 3.2.

Comando	Resultado
`{\tiny Tamanho}`	Tamanho
`{\scriptsize Tamanho}`	Tamanho
`{\footnotesize Tamanho}`	Tamanho
`{\small Tamanho}`	Tamanho
`{\normalsize Tamanho}`	Tamanho
`{\large Tamanho}`	Tamanho
`{\Large Tamanho}`	Tamanho
`{\LARGE Tamanho}`	Tamanho
`{\huge Tamanho}`	Tamanho
`{\Huge Tamanho}`	Tamanho

Tabela 3.2: Tabela de tamanhos de fonte

3.2 Texto sublinhado

Os comandos para sublinhar os textos não são nativos do LaTeX. Portanto deve-se utilizar o pacote `ulem.sty`[2] para que seja possível sublinhar seus textos. Para utilizar o pacote, devemos usar o seguinte comando

```
\usepackage[normalem]{ulem}
```

no preâmbulo do código e para sublinhar o texto, deve-se usar os comandos mostrados na tabela 3.3.

Comando	Resultado
`\uline{Sublinhado}`	Sublinhado
`\uuline{Duplo sublinhado}`	Duplo sublinhado
`\uwave{Sublinhado curvo}`	Sublinhado curvo
`\sout{Riscado}`	~~Riscado~~
`\xout{Muito Riscado}`	~~Muito Riscado~~

Tabela 3.3: Tipos de texto sublinhado

[2]Todo pacote em LaTeX é definido por um arquivo de estilo com extensão *sty*.

3.3 Acentuação

Originalmente, a acentuação em LaTeX era feita colocando-se uma contra-barra seguida do acento e da letra. Com exceção do cedilha, que era construído com um `\c` seguido de um *c* minúsculo/maiúsculo.

Como a língua portuguesa utiliza muito a acentuação, seria muito trabalhoso digitar um texto dessa maneira. Para resolver este problema, utiliza-se o pacote `inputenc`, permitindo que a acentuação seja feita de forma normal (no próprio corpo do texto).

Para utilizar esse pacote, o seguinte comando deve ser digitado no preâmbulo:

```
\usepackage[latin1]{inputenc}
```

3.4 Ambientes

Define-se por ambiente, todo comando do tipo

```
\begin{ambiente}
...
\end{ambiente}
```

onde `ambiente` é o nome do ambiente. Os ambientes podem ser chamados várias vezes um dentro do outro desde que a ordem de chamada seja mantida.

```
\begin{aaa} ... \begin{bbb} ... \end{bbb} ... \end{aaa}
```

Nas seguintes subseções serão mostrados os principais ambientes.

3.4.1 Itemizar, Enumerar e Descrever

O ambiente `itemize` é usado para criar listas simples, o ambiente `enumerate`, listas enumeradas e o ambiente `description`, descrições.

O exemplo abaixo mostra rapidamente como utilizar cada um desses ambientes:

```
\begin{itemize}
    \item Aqui jaz uma linha
    \item [-] Será que agora vai com hífen?
    \begin{enumerate}
        \item Acho que sim
        \item $A \sin{(\frac{\beta t + \varphi}{1})}$
        \begin{description}
            \item [Teste a:] Também sem descrição
            \item [Teste b:] Também sem descrição 2
        \end{description}
    \end{enumerate}
\end{itemize}
```

- Aqui jaz uma linha

- Será que agora vai com hífen?

  1. Acho que sim
  2. $A \sin\left(\frac{\beta t+\varphi}{1}\right)$

     Teste a: Também sem descrição

     Teste b: Também sem descrição 2

3.4.2 Alinhamento do texto

Assim como os editores de texto, o LaTeX também permite que o usuário altere o alinhamento do texto. Isto pode ser feito através dos ambientes `center`, para centralizar o texto, `flushleft`, para alinhar o texto à esquerda, e `flushright`, para alinhar o texto à direita.

Caso o texto não seja escrito dentro de nenhum desses ambientes ele será automaticamente alinhado a esquerda.

3.4.3 Citações e Versos

O ambiente `quote` é usado para citações, frases retiradas de outros textos ou exemplos. Veremos um exemplo de citação abaixo.

Uma curiosidade sobre o LaTeX é:

> O motivo pelo qual as páginas do LaTeX possuem as bordas tão grandes é devido a uma regra de tipografia que diz que, em média, nenhuma linha deve ter mais de 66 caracteres. Esse é o mesmo motivo pelo qual os jornais usam impressão em colunas.

Existem dois ambientes similares, o `quotation` e o `verse`. O ambiente `quotation` é indicado para citações que se estendem por parágrafos, já que realiza a auto-identação. O ambiente `verse` é usado para poemas em que a quebra de linha é importante. As linhas podem ser separadas por `\\` ou por uma linha em branco no fim de cada verso.

3.4.4 Códigos

Alguns blocos de texto não devem conter nenhum tipo formatação, para que possa ficar mais claro o seu conteúdo, um exemplo clássico são os blocos de código-fonte.

Para a exibição de blocos sem formatação, devemos utilizar o ambiente `verbatim`. Este ambiente pode ser utilizado de duas formas. Uma através do do comando `\verb|...|`, em que o caractere delimitador `|` pode ser substituído por qualquer outro caractere especial que não faça parte do bloco de texto. E o outro modo é através do uso do ambiente `verbatim`, como mostrado abaixo:

```
\begin{verbatim} ... \end{verbatim}
```

Existe também uma variação do `verbatim`, o `verbatim*`, que é utilizado para enfatizar o número de espaços em branco no texto.

3.4.5 Tabelas

O ambiente `tabular` pode ser usado para criar tabelas com linhas horizontais e verticais opcionais. O LaTeX determina automaticamente a largura das colunas. O argumento `especificação` do comando:

```
\begin{tabular}{especificação}
```

define o formato da tabela. As letras `l`, `r` e `c`, definem colunas com alinhamento, à esquerda, à direita e centralizado, respectivamente; `p{largura}` para uma coluna contendo texto justificado com quebras de linha, e `|` para linha vertical.

Dentro de um ambiente `tabular`, `&` pula para a próxima coluna, `\\` inicia uma nova linha e `\hline` insere uma linha horizontal. Você pode adicionar linhas parciais usando `\cline{j-i}` onde j e i são os números das colunas por onde a linha se estenderá. Por exemplo:

```
\begin{tabular}{|r|l|}
\hline
7C0 & hexadecimal \\
3700 & octal \\ \cline{2-2}
11111000000 & binário \\
\hline \hline
1984 & decimal \\
\hline
\end{tabular}
```

7C0	hexadecimal
3700	octal
11111000000	binário
1984	decimal

```
\begin{tabular}{|p{4.7cm}|}
    \hline
    Essa é uma coluna com margem limitada e conteúdo justificado.\\
    \hline
\end{tabular}
```

Essa é uma coluna com margem limitada e conteúdo justificado.

O separador de coluna pode ser especificado com a construção `@{...}`. Este comando elimina os espaços entre as colunas e os substitui pelo que está entre as chaves. Uma aplicação é a eliminação de espaços em uma tabela. Um outro uso comum para este comando é explicado abaixo no problema do alinhamento decimal, como não existe um comando para alinhar os pontos decimais das colunas, é possível "trapacear" e fazer isso usando duas colunas: umas alinhada à direita, para a parte inteira, e outra alinhada à esquerda para as casas decimais. O comando `@{.}` na linha `\begin{tabular}` substitui o espaçamento entre as colunas pelo ".", dando a aparência de uma coluna alinhada pelo ponto decimal. Não esqueça de substituir o ponto decimal em seus números pelo divisor de colunas (&)! Uma coluna de identificação pode ser colocada usando o comando `\multicolumn`.

```
\begin{tabular}{c r @{.} l}
    Expressão do Pi &
    \multicolumn{2}{c}{Valor} \\
    \hline
    $\pi$               & 3&1416 \\
    $\pi^{\pi}$         & 36&46   \\
    $(\pi^{\pi})^{\pi}$ & 80662&7 \\
\end{tabular}
```

Expressão do Pi	Valor
π	3.1416
π^{π}	36.46
$(\pi^{\pi})^{\pi}$	80662.7

Os materias produzidos em um ambiente `tabular` sempre ficam juntos em uma página. Se você quiser produzir tabelas grandes você pode precisar dos ambientes `supertabular` ou `longtabular`.

Capítulo 4

Fórmulas Matemáticas com LaTeX

Neste capítulo iremos abordar, **superficialmente**, o principal recurso do TeX: edição de textos matemáticos. Embora essas explicações sejam suficientes para muitas pessoas, não se desespere se você não puder encontrar uma solução para suas necessidades de edição de textos matemáticos. É muito provável que seu problema esteja resolvido com a AMS-LaTeX ou em algum outro pacote.

4.1 Modo matemático

O LaTeX tem um modo especial para edição de textos matemáticos. Textos matemáticos dentro de um parágrafo é digitado entre `\(` e `\)`, entre $ e $ ou entre `\begin{math}` e `\end{math}`. Por exemplo:

```
Adicione o quadrado de $a$ ao quadrado de $b$ para obter o quadrado de
$c$. Ou seja, $c^{2}=a^{2}+b^{2}$.
```

Adicione o quadrado de a ao quadrado de b para obter o quadrado de c. Ou seja, $c^2 = a^2 + b^2$.

É preferível *agrupar* as grandes equações ou fórmulas matemáticas ao invés de editá-las em linhas separadas. Para isso você deve incluí-las entre `\[ ... \]` ou usando o ambiente `displaymath`.Ambos produzem fórmulas não numeradas. Se você deseja que o LaTeX enumere as equações utilize o ambiente `equation`.

```
Adicione o quadrado de $a$ ao quadrado de $b$ para obter o quadrado de
$c$. Ou seja:
\begin{displaymath}
c^{2}=a^{2}+b^{2}$
\end{displaymath}
```

Adicione o quadrado de a ao quadrado de b para obter o quadrado de c. Ou seja:

$$c^2 = a^2 + b^2$$

Note que as expressões serão exibidas em um estilo diferente usando o ambiente `displaymath`:

```
$\lim_{n \to \infty}
 \sum_{k=1}^{n} \frac{1}{k^2}
 = \frac{\pi^2}{6}$
```

$\lim_{n \to \infty} \sum_{k=1}^{n} \frac{1}{k^2} = \frac{\pi^2}{6}$

```
\begin{displaymath}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{displaymath}
```

$$\lim_{n \to \infty} \sum_{k=1}^{n} \frac{1}{k^2} = \frac{\pi^2}{6}$$

Utilizando o ambiente `equation` é possível referenciar uma equação com os comandos `\label` e `\ref`

```
\begin{equation}\label{eqEPS}
\epsilon > 0
\end{equation}
De (\ref{eqEPS}), temos \ldots
```

$$\epsilon > 0 \tag{4.1}$$

De (4.1), temos ...

Existem diferenças entre o *modo matemático* e o *modo texto*. Por exemplo, no *modo matemático*:

1. A maioria dos espaços e quebras de linha não possuem nenhum significado, pois todos os espaços são criados logicamente a partir das expressões matemáticas ou a partir de comandos especiais como `\,`, `\quad`, `\quad`.
2. Linhas vazias não são permitidas. Apenas um parágrafo por fórmula.
3. Cada letra é considerada como sendo o nome de de uma variável e será processada como tal. Se você quiser processar texto normal dentro de uma fórmula (ou seja, fonte e espaçamento normal) então você deve usar o comando `\textrm{...}`.

A maioria dos comando matemáticos atuam apenas no próximo caractere. Quando se deseja "afetar" vários caracteres, estes devem ser colocados entre {}.

4.2 Potências e índices

Potências e índices podem ser especificados utilizando, os caracteres ^ e _, respectivamente.

```
$e^{-\alpha t}$ \quad $e^{x^2} \neq {e^x}^2$
```

$e^{-\alpha t} \quad e^{x^2} \neq e^{x2}$

α	\alpha	θ	\theta	o	o	υ	\upsilon
β	\beta	ϑ	\vartheta	π	\pi	ϕ	\phi
γ	\gamma	ι	\iota	ϖ	\varpi	φ	\varphi
δ	\delta	κ	\kapaa	ρ	\rho	χ	\chi
ε	\epsilon	λ	\lambda	ϱ	\varrho	ψ	\psi
ε	\varepsilon	μ	\mu	σ	\sigma	ω	\omega
ζ	\zeta	ν	\nu	ς	\varsigma		
η	\eta	χ	\chi	τ	\tau		

Tabela 4.1: Letras gregas minúsculas

Γ	\Gamma	Λ	\Lambda	Σ	\Sigma	Ψ	\Psi
Δ	\Delta	Ξ	\Xi	Υ	\Upsilon	Ω	\Omega
Θ	\Theta	Π	\Pi	Φ	\Phi		

Tabela 4.2: Letras gregas maiúsculas

4.3 Letras gregas

O LaTeX incorpora letras gregas para serem utilizadas no ambiente matemático, elas podem ser vistas nas tabela 4.1 e 4.2.

4.4 Frações e raízes

Uma fração é criada com o comando `\frac{...}{...}`. Em alguns casos a forma 1/2 é preferível, principalmente para "frações pequenas" ou quando usamos $...$ ao invés dos ambientes `displaymath` ou `equation`.

```
$1 \frac{1}{2}$~horas
\begin{displaymath}
\frac{ x^{2} }{ k+1 }\qquad
x^{ \frac{2}{k+1} }\qquad
x^{ 1/2 }
\end{displaymath]
```

$1\frac{1}{2}$ horas

$$\frac{x^2}{k+1} \qquad x^{\frac{2}{k+1}} \qquad x^{1/2}$$

A raiz quadrada é produzida com `\sqrt`, a n-ésima raiz é produzida com `sqrt[n]` O tamando do sinal de radiciação é definido automaticamente pelo LaTeXSe você quer apenas o sinal use `\surd`.

```
$\sqrt{x}$ \qquad
$\sqrt{ x^{2}+\sqrt{y} }$
\qquad $\sqrt[3]{2}$
$\qquad \surd[x^2 + y^2]$
```

$$\sqrt{x} \qquad \sqrt{x^2+\sqrt{y}} \qquad \sqrt[3]{2}\,\sqrt{[x^2+y^2]}$$

4.5 Funções

Os nome de funções são geralmente criados em fonte normal e não em itálico como as variáveis. Conseqüêntemente o LATEX possui comandos para criar os nomes das mais importantes funções. Estes comandos podem ser vistos na tabela 4.3.

```
\arccos  \cos   \csc  \exp   \ker     \limsup  \min
\arcsin  \cosh  \deg  \gcd   \lg      \ln      \Pr
\arctan  \cot   \det  \hom   \lim     \log     \sec
\arg     \coth  \dim  \inf   \liminf  \max     \sin
\sinh    \sup   \tan  \tanh
```

Tabela 4.3: Nomes de funções

4.6 Parênteses, colchetes e chaves

Existem duas formas de se trabalhar com parênteses, colchetes e chaves em LATEX. Uma delas é usando comandos que ajustam o tamanho do delimitador de acordo com a fórmula a ser escrita. Obviamente, a outra é usando delimitadores com tamanho fixo. Os delimitadores de tamanho variável devem sempre ser usados aos pares, enquanto que os de tamanho fixo podem ser usados sozinhos.

Os delimitadores de tamanho variável pode ser usados da seguinte forma:

```
\left( ... \right)   → parênteses
\left[ ... \right]   → colchetes
\left{ ... \right}   → chaves
```

Já os de tamanho fixo:

```
\bigl(,  \biggl(,   \Bigl(,   \Biggl(,
\bigr),  \biggr),   \Bigr),   \Biggr),
\bigr],  \biggr],   \Bigr],   \Biggr],
\bigl{,  \biggl{,   \Bigl{,   \Biggl{,
```

Vejamos alguns exemplos:

```
\[
\biggl(\frac{10+x}{x+10}) \quad \quad
\left[\alpha^{-1}\beta\left(\frac{10\sqrt[n]{2}}{3}\right)\right]
\]
```

$$\Big(\frac{10+x}{x+10}\Big) \qquad \left[\alpha^{-1}\beta\left(\frac{10\sqrt[n]{2}}{3}\right)\right]$$

```
\[
\Biggl[-\frac{b^2-a^2}{w} + w \Biggr]_{b-a}^{b+a}
\]
```

$$\left[-\frac{b^2-a^2}{w}+w\right]_{b-a}^{b+a}$$

Algumas vezes também precisamos usar chaves em expressões. Para isso usamos os comandos `\underbrace` e `overbrace`:

```
\[
x = \overbrace{a+b}^m+\underbrace{c+d}_n = m+n
\]
```

$$x = \overbrace{a+b}^{m}+\underbrace{c+d}_{n} = m+n$$

4.7 Limites, derivadas, somatórios, produtórios e integrais

Trabalhar com limites, derivadas, somatórios, produtórios e integrais em LaTeX é mais simples do que você pensa. Vejamos alguns exemplos:

Para obtermos expressões como

$$\lim_{var \to valor} func$$

digitamos `$$\lim_{var \to valor} func$$`.

Outros exemplos com limites:

`$\lim_{x \to a} f(x) = f(a)$`	$\lim_{x \to a} f(x) = f(a)$
`$\lim_{x \to \infty} (1+\frac{1}{x})^x = e$`	$\lim_{x \to \infty} (1+\frac{1}{x})^x = e$
`$\lim_{x \to 0} \frac{\sin(\pi x)}{\pi x} = 1$`	$\lim_{x \to 0} \frac{\sin(\pi x)}{\pi x} = 1$

Já com derivadas ...

A equação do calor de Fourrier, mostrada na equação

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}$$

pode ser obtida com:

```
\[
\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}
                              + \frac{\partial^2 u}{\partial y^2}
                              + \frac{\partial^2 u}{\partial z^2}
\]
```

Para obter um somatório, como

$$\sum_{i=1}^{2n}$$

devemos digitar:

```
\[
\sum_{i=1}^{2n}
\]
```

O somatório

$$\sum_{k=1}^{n} k^2 = \frac{1}{2} n (n+1)$$

é obtido com:

```
\[
\sum_{k=1}^n k^2 = \frac{1}{2} n (n+1)
\]
```

De maneira análoga, trabalhamos com produtórios. O produtório

$$\prod_{k=1}^{n} k^2 \neq \frac{1}{2} n (n+1)$$

pode ser obtido a partir de

```
\[
\prod_{k=1}^n k^2 \neq \frac{1}{2} n (n+1)
\]
```

Agora, falando sobre integrais ...

O símbolo de integral $\int$ é obtido através do comando `\int` e os limites de integração são tratados como índices e potências. Nos exemplos abaixo você perceberá que antes da variável de integração utilizamos sempre um '`\ `'. Isso serve para que o 'd' não fique tão junto da função a ser integrada.

```
$$\int_0^{+\infty} x^n e^{-x} \ dx = n!$$

$$\int_{x^2 + y^2 \leq R^2} f(x,y)\ dx\ dy
   = \int_{\theta=0}^{2\pi} \int_{r=0}^R
      f(r\cos\theta,r\sin\theta) r\ dr\ d\theta$$

$$\int_0^R \frac{2x\ dx}{1+x^2} = \log(1+R^2)$$
```

$$\int_0^{+\infty} x^n e^{-x}\,dx = n!$$

$$\int_{x^2+y^2\leq R^2} f(x,y)\,dx\,dy = \int_{\theta=0}^{2\pi}\int_{r=0}^{R} f(r\cos\theta, r\sin\theta) r\,dr\,d\theta$$

$$\int_0^R \frac{2x\,dx}{1+x^2} = \log(1+R^2)$$

Em algumas integrais múltiplas, o LaTeX coloca muito espaço entre os símbolos da integral. Para melhorar a aparência da equação, podemos usar `\!` para diminuir o espaço em excesso. Por exemplo a integral múltipla

$$\int_0^1\int_0^1 x^2y^2\,dx\,dy$$

é obtida digitando-se:

```
\[
\int_0^1 \! \! \! \int_0^1 x^2 y^2\ dx\ dy
\]
```

Se tivéssemos digitado

```
\[
int_0^1 \int_0^1 x^2 y^2\ dx\ dy
\]
```

teríamos obtido:

$$\int_0^1 \int_0^1 x^2y^2\,dx\,dy$$

Quando quisermos utilizar integral de linha, devemos digitar `\oint`, como mostrado no exemplo abaixo:

$$\oint_C u(x,y)\,dx + v(x,y)\,dy$$

que pode ser obtido da seguinte maneira:

```
\[
\oint_C u(x,y)\ dx + v(x,y)\ dy
\]
```

4.8 Vetores e conjugados

Podemos criar vetores através dos comandos `\vec` ou `overrightarrow{expressão}`. O comando `\vec` só funciona para a letra que o segue. Por exemplo:

```
\[
\overrightarrow{P_1P_2} = (x_2\vec i + y_2 \vec j + z_2 \vec k)
                        - (x_1\vec i + y_1 \vec j + z_1 \vec k)
\]
```

$$\overrightarrow{P_1P_2} = (x_2\vec{i} + y_2\vec{j} + z_2\vec{k}) - (x_1\vec{i} + y_1\vec{j} + z_1\vec{k})$$

Já para conjugados, podemos utilizar o comando `\bar` ou `\overline{expressão}`. Assim como no `\vec` o `\bar` só funciona para a letra que o segue.

```
\[
z = a+bi \Rightarrow \bar z = a - bi
\]
```

$$z = a + bi \Rightarrow \bar{z} = a - bi$$

```
\[
m(\overline{AC})^2 = m(\overline{AB})^2 + m(\overline{BC})^2
\]
```

$$m(\overline{AC})^2 = m(\overline{AB})^2 + m(\overline{BC})^2$$

4.9 Matrizes

Matrizes podem ser construídas com o ambiente `array` da seguinte forma:

```
\begin{array}{especificação da matriz}
    % definição de cada linha com \\ no final
\end{array}
```

Observando-se o seguinte:

- Assim como nas tabelas, no campo especificação determina-se o número de colunas e seu alinhamento (c, l ou r). Por exemplo, a especificação `{cccc}` indica que a matriz terá quatro colunas com todos os elementos centralizados.
- Tambémm como nas tabelas os elementos das linhas são separados por '&' e no final de cada linha deve-se usar '`\\`'.
- Os parênteses, colchetes ou chaves devem ser definidos antes do `\begin{array}` e depois do `\end{array}`. Para isso podemos usar os delimitadores de tamanho variável (olhar tópico 4.6).

```
\[
A = \left[
\begin{array}{cccc}
    a_{11} & a_{12} & \cdots & a_{1m}\\
    a_{21} & a_{22} & \cdots & a_{2m}\\
    \vdots & \vdots & \ddots & \vdots\\
    a_{n1} & a_{n2} & \cdots & a_{nm}\\
\end{array}\right]
\]
```

$$A = \left[\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{array} \right]$$

Capítulo 5

Imagens e gráficos

5.1 Introdução

Inicialmente o LaTeX não possuía um método simples para inserção de imagens em documentos. Apenas com o lançamento do LaTeX 2ε comandos mais robustos e eficientes foram sendo introduzidos, facilitando a nossa vida. Na verdade existem basicamente dois pacotes com funcionalidades semelhantes que dão suporte para a manipulação de figuras: o `graphics` e o `graphicx`.Por apresentar uma sintaxe mais amigável, utilizaremos o pacote `graphicx`. Deste modo, devemos inserir no preâmbulo de todo documento em que iremos acrescentar figuras a seguinte linha:

```
\usepackage{graphicx}
```

Como os arquivos de saída DVI são geralmente convertidos para PostScript (`*.ps`), o formato mais utilizado em imagens inseridas em documentos LaTeX é o EPS (*Encapsulated PostScript*).

O LaTeX ainda suporta outros formatos de imagens, tais como JPEG e GIF. No entanto, a maioria dos conversores de DVI possibilita apenas a inclusão de figuras no formato EPS. A utilização de imagens JPEG é possível ao se utilizar o `pdflatex` para gerar o documento. Entretanto, o `pdflatex` é mais lento que o `latex` e normalmente só é usado para fazer a compilação final.

5.2 Convertendo imagens para o formato EPS

Existem diversos programas para a conversão de imagens em formato EPS. O ImageMagick, disponível para diversas plataformas (incluindo RWindow$ e Linux), pode ser encontrado no endereço a seguir.

```
ftp://ftp.imagemagick.org/pub/ImageMagick/binaries/
```

A utilização deste utilitário é a mais óbvia possível:

```
$ convert imagem.jpg imagem.eps
```

5.3 Inserindo uma imagem

Para inserirmos uma figura, apenas o comando `\includegraphics{}` é necessário. Este comando possui a seguinte sintaxe:

```
\includegraphics[opções]{nome_do_arquivo.eps}
```

Algumas das `opções` possíveis estão listadas na tabela 5.1

`height`	Altura (Ex.: `height=40pt`)
`width`	Largura (Ex.: `width=0.2cm`)
`scale`	Redimensionamento (Ex.: `scale=0.3`)
`rotate`	Rotação em graus (Ex.: `rotate=90`)

Tabela 5.1: Opções de inserção de figuras

O trecho de código abaixo serve para inserir uma imagem no início de uma linha, como mostrado a seguir:

```
\includegraphics[width=2cm]{morro}
```

Perceba que identificamos a imagem sem determinar sua extensão. Dependendo do modo de compilação (`pdflatex` ou `latex`) é que isso vai ser determinado (se as imagens EPS ou JPEG serão carregadas).

As imagens também podem ser inseridas em uma ambiente `figure`. Desta forma podemos definir opções extras para as imagens. Utilizando este ambiente, podemos, por exemplo, adicionar legendas e definir rótulos.

Segue um trecho de código que insere as imagens mostradas na figura 5.1.

```
\begin{figure}[htb]
\centering
\includegraphics[width=2cm]{morro}
\includegraphics[width=3cm]{morro}
\includegraphics[width=4cm]{morro}
\caption{Morro do careca com 2, 3 e 4 cm respectivamente.}
\label{figExemplo}
```

Figura 5.1: Morro do careca com 2, 3 e 4 cm respectivamente.

Você deve ter notado que foram passados parâmetros `[htp]` após a declaração do ambiente `figure`. Esses parâmetros especificam o alinhamento da imagem e cada letra possui um "valor" específico:

h	Here – Insere a imagem “aqui” onde o comando ocorreu.
t	Top – Insere a imagem na parte superior da página.
b	Bottom – Insere a imagem na parte inferior da página.
p	Page – Permite que uma nova página seja criada para a inserção.

Tabela 5.2: Opções de alinhamento

5.4 Imagens lado a lado

Existem algumas maneiras de se inserir imagens lado a lado. O objetivo principal de se fazer isso é de se ter legendas e rótulos individuais para cada uma delas.

Quando se quer inserir duas ou mais imagens lado a lado em um texto, usa-se um mesmo ambiente `figure`, especificando um ambiente `minipage` para cada figura a ser inserida. Dentro do ambiente `minipage` podemos especificar os atributos de cada uma das figuras a serem inseridas. Para cada ambiente pode ser especificado um alinhamento diferente.

Segue um exemplo de utilização:

Figura 5.2: Exemplo 1

Figura 5.3: Exemplo 2

Figura 5.4: Exemplo `minipage`.

```
\begin{figure}[htb]
\centering
    \begin{minipage}[b]{0.30\linewidth}
    \includegraphics[width=4cm]{imgs/morro}
    \caption{Exemplo 1}
    \end{minipage}
\hspace{0.03\textwidth}
    \begin{minipage}[b]{0.30\linewidth}
    \includegraphics[width=4cm]{imgs/morro}
    \caption{Exemplo 2}
    \end{minipage}
\caption{Exemplo {\tt minipage}.}
\label{figExemplo2}
\end{figure}
```

Capítulo 6

Apresentações em LaTeX – Beamer
SlideShow

A classe `beamer` foi desenvolvida por Till Tantau e tem por finalidade a criação de apresentações (*SlideShow*). As apresentações são criadas assim como qualquer outro documento LaTeX e no final um arquivo PDF pode ser gerado, o que as torna extremamente portáveis.

6.1 Criando uma apresentação

Criar uma apresentação usando o `Beamer` é algo bastante simples e segue uma "receita de bolo":

```
\documentclass{beamer}

\usepackage[latin1]{inputenc}
%\usepackage[utf8]{inputenc} % se a codificacao do programa
                             % for utf8
\usepackage[brazil]{babel}
\usepackage{graphicx}
\usepackage{beamerthemesplit}

\usecolortheme{beaver} % apenas um dos muitos temas disponíveis

\title[Título Breve]{Um título muito longo}
\author[Nome Sobrenome]{Nome Muito Longo Sobrenome}
\institute{Nome da Universidade, programa etc}
\date{\today}
\logo{} % aqui pode ser adicionada alguma logomarca
        % da(o) instituicao/projeto (com o includegraphics)

\begin{document}

\frame{\titlepage}

\frame
{
    \frametitle{Título do slide}
```

```
    \framesubtitle{Aqui vai um subtítulo}

    Inicio do texto
}
```

Assim como em qualquer outro documento LaTeX você pode adicionar fórmulas, citações, itens, enumerações... E além disso, não precisará se preocupar com a "organização" do *slide*. Basta que se preocupe com o conteúdo da apresentação.

6.2 Efeitos de Transição

Ao adicionar itens ou enumerações você poderá optar por adicionar efeitos de transição. Existem dois modos simples de se fazer isso.

6.2.1 Usando o `pause`

```
\begin{frame}
\begin{itemize}
\item
  O primeiro slide;
\pause
\item
  O segundo slide;
\pause
\item
  O último slide.
\end{itemize}
\end{frame}
```

6.2.2 Usando o `<+->`

```
\begin{itemize}
\item<1-> Um
\item<2-> Dois
\item<3-> Três
\end{itemize}
```

ou

```
\begin{itemize}<+->
\item Um
\item Dois
\item Três
\end{itemize}
```

6.2.3 Sequência de imagens – Usando a opção `<n>`

```
\includegraphics<1>{endfig1}
\includegraphics<2>{endfig2}
\includegraphics<3>{endfig3}
```

6.3 Destacando informações

6.3.1 Uso de blocos

```
\begin{block}{Características Técnicas}
\begin{enumerate}
\item portátil;
\item compacto;
\item vibrante.
\end{enumerate}
\end{block}
```

6.3.2 Uso de colunas

```
\begin{columns}
\column{0.5\textwidth}
Coluna da esquerda
\column{0.5\textwidth}
Coluna da direita
\end{columns}
```

Referências Bibliográficas

[do Vale Pereira and da Silveira, 2004] do Vale Pereira, D. R. and da Silveira, R. W. R. (2004). *Desenvolvendo textos científicos com LaTeX*.

[Mafra et al., 2006] Mafra, P., Kroeger, P., and Obelheiro, R. R. (2006). *TeX BR*. Site: http://www.tex-br.org; Lista de discussão: http://marc.info/?l=tex-br [Acessados em 02/09/2007].

[Oetiker et al., 2002] Oetiker, T., Partl, H., Hyna, I., and Schlegl, E. (2002). *Introdução ao LaTeX 2ε– LaTeXem 105 minutos*. Tradução por Démerson André Polli.

[Tantau, 2007] Tantau, T. (2007). *User Guide to the Beamer Class, Version 3.07*. http://latex-beamer.sourceforge.net. [Acessado em 02/09/2007].