FMA 215

Aula 9: Documentos em LaTeX

Prof. Jorge L. deLyra

23 de Maio de 2002

1 Alguns Conceitos Relevantes

Processamento de texto: para o processamento de texto, começamos codificando tanto o conteúdo do texto quanto um conjunto de comandos de formatação em um arquivo-fonte ASCII, também chamado de "arquivo manuscrito". Este arquivo-fonte é lido por um compilador que interpreta os comandos e produz como resultado um outro arquivo, que não é necessariamente um arquivo ASCII, contendo o texto formatado. No caso do LaTeX este arquivo estará em uma linguagem padronizada chamada DVI (device-independent), que é independente do tipo de dispositivo no qual se pretende imprimir ou visualizar o texto formatado. Arquivos DVI podem ser visualizados diretamente na tela ou traduzidos para outros formatos.

Depois de executado o compilador LaTeX, outros programas podem ser usados para transformar o arquivo DVI que ele cria em outros tipos de arquivo, contendo o texto formatado em variadas linguagens específicas deste ou daquele dispositivo de impressão ou de visualização. Programas que fazem este tipo de tradução de formato de arquivos também são denominados de "filtros". Um exemplo de linguagem específica é a linguagem PCL (Printer Control Language) usada pelas impressoras HP (Hewlett-Packard) e seus clones, outro é a linguagem ESC/P da Epson. Uma linguagem muito popular, que segue um padrão aberto e que é entendida por muitos dispositivos, é a linguagem PS (PostScript), que é a que vamos usar aqui. Este tipo de arquivo também pode ser visualizado diretamente na tela do computador.

2 Tarefas

Vamos explorar aqui os fundamentos do processador de texto LaTeX, que é um programa feito para produzir resultados de alta qualidade tipográfica. É um sistema modular, extensível através de pacotes que podem ser escritos por qualquer um que tenha os conhecimentos técnicos necessários. Também é possível enriquecê-lo através da definição e inclusão de novos formatos e estilos de documentos. Como o número de comandos, estruturas e possibilidades da linguagem é grande e não poderá ser todo coberto aqui, recomendamos aos interessados a aquisição de um livro como, por exemplo, o da referência bibliográfica [1], que inclui um manual de referência completo. Existe também um velho tutorial, mais apropriado para uma versão mais antiga do LaTeX, mas que ainda é muito útil e contém algum material que complementa o que veremos aqui. Ele está disponível na rede como mostra a referência bibliográfica [2].

Uma das características básicas de uma linguagem como o LaTeX é que a formatação do arquivo manuscrito não é refletida no resultado final. Desta forma ficamos livres para formatar o arquivo manuscrito de qualquer forma conveniente para poder escrevê-lo e lê-lo confortavelmente no terminal, sem que isto afete o resultado final da formatação do documento. Como estaremos usando o `emacs` para editar os arquivos, vamos usar nestas notas um estilo de formatação da fonte que é conveniente para o uso daquele editor. Mas isto é, de fato, uma questão de escolha pessoal.

O compilador do LaTeX já vem com vários tipos diferentes de documento definidos. Nos exemplos vamos usar o tipo `article`, que é o que serve de forma mais geral para variados tipos de documentos, podendo incluir seções. Há também o formato `report`, que é mais extenso, podendo incluir capítulos além de seções. Além destes, existem também o tipo `letter` e o tipo `book`. Além disso é possível definir novos tipos de documentos, ou variações destes tipos básicos, para coisas como cartas timbradas, teses, formulários para requisições e assim por diante.

1. Um arquivo manuscrito (manuscript file), ou seja a fonte de um documento formatado em LaTeX, é um arquivo ASCII que você cria com um editor de texto. Ele tem duas partes básicas, uma parte inicial contendo declarações e uma segunda parte contendo o documento em si. Aqui está um exemplo minimalista do conteúdo de um arquivo deste tipo:

```
% JLdL 25Jun00.

\documentclass[12pt]{article}

\begin{document}

 Hello!

\end{document}
```

A parte contendo o documento é aquela circundada pelos comandos `\begin` e `\end`, com argumento `{document}`. No LaTeX todas as principais estruturas de formatação são definidas desta forma, em regiões do texto, através dos comandos `\begin` e `\end`. Argumentos obrigatórios de comandos são sempre denotados pelas chaves `{` e `}`.

Usando um editor de texto (o `emacs`, por exemplo) crie um arquivo chamado `hello.tex` com este conteúdo.

2. Para produzir o arquivo com o documento formatado, é necessário rodar o compilador da linguagem, que é o comando `latex`. Faça isto, executando `latex hello.tex`. Se houver algum erro, verifique cuidadosamente o seu arquivo manuscrito e corrija-o onde necessário. Em seguida use o comando `ls` para verificar que foi criado um arquivo `hello.dvi`. Execute o comando `file hello.*` para verificar que o sistema identifica corretamente o tipo de cada um dos arquivos.

3. Visualize na tela o arquivo em formato DVI executando o comando `xdvi hello.dvi`.

4. Para transformar o documento para o formato PS, execute o comando `dvips hello.dvi`. Em seguida repita os comandos `ls` e `file` como você os usou na tarefa do ítem 2. Observe que, em nossos sistemas, o comando `dvips` está modificado através de um alias. Por "default" ele envia o resultado para a impressora do sistema, não para um arquivo. Execute `alias dvips` para ver como estamos fazendo para que o resultado vá para um arquivo, usando a opção `-o`.

5. Visualize na tela o arquivo em formato PS executando o comando `gv hello.ps`.

6. Dentro do corpo do texto do documento podemos colocar outras estruturas de formatação. Denominamos estas estruturas de "environments" e elas são sempre iniciadas por um comando `\begin` e terminadas por um comando `\end`, com argumentos que determinam qual é o environment que vai estar sendo definido. Aqui está um exemplo de uso do environment `center`:

   ```
   % JLdL 25Jun00.

   \documentclass[12pt]{article}

   \begin{document}

    Hello!

   \begin{center}
    Hello center!
   \end{center}

    Hello again!

   \end{document}
   ```

   Digite estes novos elementos em seu arquivo, rode de novo o LaTeX e olhe o resultado com o `xdvi`.

7. Uma das muitas coisas que podemos fazer é mudar o tamanho e tipo das fontes, ou seja, o formato das letras. Para fazer isto usa-se uma "declaração", que é um comando que muda alguma coisa a partir do ponto onde é colocado. Em geral os efeitos de uma declaração são limitados ao que chamamos de *escopo* da declaração, em geral o environment dentro do qual ela se encontra. Por exemplo, a declaração `\bf` muda a fonte para o tipo "boldface" ou seja, negrito, enquanto `\it`

muda a fonte para itálico. Já a declaração `\large` aumenta o tamanho das letras. Aqui está um exemplo:

```
% JLdL 25Jun00.

\documentclass[12pt]{article}

\begin{document}

 {\large Hello!}

\begin{center}\bf
 Hello center!
\end{center}

 Hello again!

 {\it Here is some italic text.}

 Hello yet again!

\end{document}
```

Existem também as declarações `\small` e `\normalsize`, entre outras. Digite e rode mais este exemplo. Observe como as chaves também delimitam a validade das declarações. As chaves são os caracteres separadores básicos da linguagem.

8. Você terá notado que separamos algumas porções do texto de outras através de linhas em branco. Linhas em branco definem os limites de um parágrafo. Esta é uma das poucas características de formatação do arquivo manuscrito que têm alguma influência no resultado final. Aqui está um exemplo com um número menor de parágrafos, cada um deles maior do que os do caso anterior:

```
% JLdL 25Jun00.

\documentclass[12pt]{article}

\begin{document}

 Hello! You might have noticed that this is a
 paragraph. It is in fact a nice, short paragraph, like
 all paragraphs should be, specially when you have
 absolutelly nothing to say in them.

\begin{center}\bf
 Hello center!
\end{center}

 Hello again! {\it Here is some italic text.} Hello yet
 again! Let's keep saying hello over and over again
 until this thing becomes clearly and undoubtedly a
 paragraph, for we ran out of things to say.

\end{document}
```

Digite isto em seu arquivo, rode o LaTeX e veja o que acontece.

9. Você terá percebido que os exemplos até agora não estão escritos no vernáculo. Isto se deve ao fato de que, para habilitar a capacidade de acentuar as palavras, é necessário ativar uma outra propriedade do LaTeX, a capacidade de usar pacotes (packages). Há um comando, que podemos colocar nas declarações iniciais, que permite que usemos pacotes que estendem as capacidades do LaTeX. O nome do comando é `\usepackage`, com um argumento que é o nome do pacote. Aqui está um exemplo:

```
% JLdL 25Jun00.

\documentclass[12pt]{article}
\usepackage[latin1]{inputenc}

\begin{document}

 Olá!

\end{document}
```

Para digitar letras acentuadas no Emacs, basta digitar primeiro o acento e depois a letra. Por exemplo, `'a` produz `á`. Para a cedilha, use `~c`. Caso nada disto esteja funcionando, aperte uma única vez a tecla `[F6]` e tente de novo. Digite em seu arquivo o comando `\usepackage` como se vê acima, digite algum texto usando todos os acentos possíveis do Português, rode o LaTeX e verifique o resultado. Às vezes é conveniente usar também o pacote `babel` com a opção `brazil`, que implementa outras adaptações para o Português.

10. Outra coisa importante que devemos saber fazer é formatar símbolos e equações matemáticas, como por exemplo $\vec{f} = m\vec{a}$. Aqui está um exemplo:

```
% JLdL 25Jun00.

\documentclass[12pt]{article}
\usepackage[brazil]{babel}\usepackage[latin1]{inputenc}

\begin{document}

 Aqui está um pouco de matemática formatada com o
 \LaTeX. Esta é uma equação em modo ``display
 math'', ou seja, colocada em evidência, em um
```

```
 parágrafo todo dela:

\[
 \alpha x+\frac{\beta}{2}x^{2}+\frac{\gamma}{3}x^{3}=
 \int_{0}^{x}\left(\alpha+\beta\chi+\gamma\chi^{2}\right)
\]

 Também é possível colocar matemática no meio do
 texto de um parágrafo, por exemplo dizer que
 $\alpha$ é uma letra grega que representa um
 parâmetro constante, ou que $\beta\neq\gamma$.

\end{document}
```

Digite ou copie este texto em seu arquivo e rode o LaTeX para ver o que acontece. Calcule a derivada $\partial/\partial\sqrt{x}$ desta equação e formate a equação resultante. Observe o uso aqui do pacote `babel` com a opção `brazil`, que faz com que a quebra de palavras no final de linhas (hifenação) seja feita de acordo com as regras do Português.

11. Outra coisa importante que precisamos saber é que o LaTeX é capaz de numerar automaticamente variados elementos do texto. Por exemplo, é possível definir seções de um documento, cada uma com equações, de tal forma que tanto as seções quanto as equações sejam numeradas. Aqui está um exemplo:

```
% JLdL 25Jun00.

\documentclass[12pt]{article}
\usepackage[brazil]{babel}\usepackage[latin1]{inputenc}

\begin{document}

\section{Esta é a primeira seção}
```

```
 Nesta seção vamos mostrar que

\begin{equation}
 \alpha x=\frac{\partial y}{\partial t},
\end{equation}

\noindent
 onde $x$, $y$ e $t$ são reais e

\begin{equation}
 \beta y=\frac{\partial x}{\partial t}.
\end{equation}

\section{Esta é a segunda seção}

 Nesta seção vamos mostrar que, muito pelo contrário,

\begin{equation}
 \gamma t=\frac{\partial y}{\partial x},
\end{equation}

\noindent
 onde $\gamma$ é uma função real arbitrária,

\begin{equation}
 \gamma=\gamma(a,b,c,d,e,f),
\end{equation}

\noindent
 de onde segue que a quantidade de ``nonsense'' no
 universo é ilimitada.

\end{document}
```

Digite ou copie este texto dentro de seu arquivo e rode o LaTeX para verificar o que acontece com a numeração dos elementos.

12. Elementos numerados podem ser marcados com o comando `\label` de forma que se possa fazer referência a eles de forma estruturada, usando o "label" e não os números explicitamente. Aqui está um exemplo:

```
% JLdL 25Jun00.

\documentclass[12pt]{article}
\usepackage[brazil]{babel}\usepackage[latin1]{inputenc}

\begin{document}

 Nesta seção vamos mostrar que não é verdade que

\begin{equation}\label{obvia}
 \alpha x=\frac{\partial y}{\partial t},
\end{equation}

\noindent
 onde $x$, $y$ e $t$ são reais e

\begin{equation}
 \beta y=\frac{\partial x}{\partial t}.
\end{equation}

 Na realidade a equação~(\ref{obvia}) é de fato
 obviamente falsa e não é necessária nenhuma
 demonstração.

\end{document}
```

Os labels podem ser quaisquer combinações de letras e do hífen. Digite ou copie este texto dentro de seu arquivo e rode o LaTeX para verificar o que acontece com a numeração dos elementos e com a referência. Quando usamos referências num texto, em geral será necessário rodar o LaTeX duas vezes para que os números das referências saiam corretamente. Isto se deve ao fato de que os números das referências que queremos fazer ainda não são conhecidos quando rodamos o compilador pela primeira vez. Na primeira vez que é rodado o compilador numera os objetos que têm labels e escreve a informação sobre os labels existentes num arquivo auxiliar `<arquivo>.aux`, que é lido da segunda vez para definir os números das referências que são feitas aos labels.

13. Outra capacidade muito importante é a de incluir figuras dentro do texto. Existem vários pacotes que nos permitem fazer isto. O mais simples é o `epsfig`. Para usá-lo, primeiro rode o LaTeX no texto que tem em seu arquivo `hello.tex` neste momento e, em seguida, crie um arquivo em formato EPS (Encapsulated Post-Script) usando o comando

    ```
    dvips -E -o aula-09-imagem.eps hello.dvi
    ```

    que irá criar um arquivo `aula-09-imagem.eps` contendo a imagem do texto. Dê uma olhada neste arquivo usando o comando

    ```
    gv aula-09-imagem.eps
    ```

    Em seguida escreva um outro arquivo com o conteúdo baseado no exemplo que segue.

```
% JLdL 25Jun00.

\documentclass[12pt]{article}\pagestyle{empty}
\usepackage[brazil]{babel}\usepackage[latin1]{inputenc}
\usepackage{epsfig}

\begin{document}

 Este é um texto que inclui uma bela equação, que
 representa a ação de um modelo muito popular da
 teoria quântica de campos, o assim-chamado modelo
 polinomial $\lambda\varphi^{4}$,

\[
 S[\varphi]
 =\frac{1}{2}\sum_{\vec{n}}^{N^{d}}\sum_{\mu}^{d}
 \left(\Delta_{\mu}\varphi(\vec{n})\right)^{2}
```

```
 +\frac{\alpha}{2}
 \sum_{\vec{n}}^{N^{d}}\varphi^{2}(\vec{n})
 +\frac{\lambda}{4}
 \sum_{\vec{n}}^{N^{d}}\varphi^{4}(\vec{n}).
\]

\noindent
 Além disso ele contém uma cópia da imagem de uma
 página de um outro documento, incluída aqui em
 formato EPS, devidamente centrada nesta página

\begin{center}
\epsfig{file=aula-09-imagem.eps,scale=0.8}
\end{center}

\noindent
 Onde será que isto vai parar? Será que dá para
 colocar uma página dentro da mesma página dentro
 da mesma página dentro da mesma página dentro da
 mesma página\ldots?\ldots

\end{document}
```

Rode o compilador neste segundo arquivo e olhe o resultado com o comando `xdvi`. Produza um arquivo PS com o comando `dvips` e examine-o com o comando `gv` para verificar que a figura foi de fato incluída.

14. O título, autor e data do documento são formatados automaticamente. Para definir a informação que vai aparecer nestes elementos, há comandos de declaração que devem ser colocados na parte inicial do arquivo manuscrito. Aqui está um exemplo:

```
% JLdL 25Jun00.

\documentclass[12pt]{article}
\usepackage[brazil]{babel}\usepackage[latin1]{inputenc}

\title{Como Subir Escadas}

\author{Rolando Escadabaixo}

\begin{document}

% É este o comando que produz a página de título.
\maketitle

\section{Introdução}

 Olá, pessoal, vamos subir algumas escadas!

\end{document}
```

Observe o uso do comando `\maketitle`. É este comando que produz o resultado que se vê, não os comandos de declaração `\title` e `\author`. Observe que o comando `\maketitle` está, ao contrário dos comandos de declaração, dentro do corpo de texto do documento. Digite estes elementos em seu arquivo e rode o LaTeX para ver o que acontece. Em seguida comente o comando `\maketitle` e rode de novo o compilador. Para comentar uma linha, basta colocar um caracter `%` no início dela. Tudo que estiver à direita deste caracter será ignorado.

15. Outra estrutura muito útil é a formatação de listas itemizadas de texto, que podem ser numeradas ou não. Aqui está um exemplo de lista com elementos não-numerados.

```
% JLdL 25Jun00.

\documentclass[12pt]{article}
\usepackage[brazil]{babel}\usepackage[latin1]{inputenc}

\begin{document}

 Esta é uma lista não-numerada de coisas.

\begin{itemize}

 \item Esta é a primeira coisa.

 \item Esta é a segunda coisa.

 \item Esta é a terceira coisa.

 \item O que acontece quando o texto da coisa é
 mais longo? Como você vê, forma-se um parágrafo
 dentro da estrutura da lista.

 \item Mais uma coisa curtinha.

\end{itemize}

\end{document}
```

Digite este conteúdo em seu arquivo, rode o LaTeX e olhe o resultado. As estruturas de listas podem ser colocadas umas dentro das outras (podem ser "nested"). Basta colocar uma lista como ítem da outra. Construa um exemplo com duas listas, uma dentro da outra. Para fazer uma lista numerada basta trocar o argumento `itemize` por `enumerate`. Tente fazer isto também.

16. Também é possível formatar tabelas, com qualquer número de linhas e colunas. As tabelas podem ou não ter linhas verticais e horizontais

entre elementos e colunas. É possível formatar tabelas de complexidade arbitrariamente alta, aqui está um exemplo simples.

```
% JLdL 25Jun00.

\documentclass[12pt]{article}
\usepackage[brazil]{babel}\usepackage[latin1]{inputenc}

\begin{document}

 Esta é uma tabela simples, com três colunas e
 algumas linhas, centrada na página.

\begin{center}
\begin{tabular}{|lcr|}
\hline
 Esquerda  & Centro & Direita \\
\hline
 Elemento 1 1 & Elemento 1 2 & Elemento 1 3 \\
 Elemento 2 1 & Elemento 2 2 & Elemento 2 3 \\
 Elemento 3 1 & Elemento 3 2 & Elemento 3 3 \\
\hline
\end{tabular}
\end{center}

\end{document}
```

Observe como o comando `\\` marca o final das linhas e o caracter `&` as separações entre colunas. O comando `\\` é o comando universal de quebra de linha no LaTeX, ele também funciona em outros environments como, por exemplo, o `center`, mas ele não deve ser usado em parágrafos normais, onde a quebra de linhas é feita de forma automática pelo LaTeX. Observe também que cada coluna está alinhada de uma forma, à esquerda, centrada e à direita. Digite e rode este exemplo também.

17. Finalmente, é possível colocar figuras e tabelas em environments especiais chamados `figure` e `table`, que são chamados de "floating environments" ou "ambientes flutuantes". Tanto as figuras quanto as tabelas formatadas desta forma são colocadas, não onde você as coloca no texto, mas em posições apropriadas que dependem da paginação resultante do processamento do texto. As figuras e tabelas são numeradas independentemente, mas o processamento destes dois environments pelo compilador é o mesmo. As figuras e tabelas são enumeradas através de "captions", ou seja, legendas que podem conter uma descrição do conteúdo e que podem receber labels através dos quais você pode fazer referências a estes objetos flutuantes. Aqui está o exemplo anterior reformatado para usar o environment `table` e acrescido de uma figura usando o environment `figure`.

```
% JLdL 19May02.

\documentclass[12pt]{article}
\usepackage[brazil]{babel}\usepackage[latin1]{inputenc}
\usepackage{epsfig}

\begin{document}

 A tabela~\ref{tabsimp} é uma tabela simples, com três
 colunas e algumas linhas, centrada na página, colocada
 em um ambiente flutuante com uma legenda ou ``caption''.

\begin{table}
\centering
\begin{tabular}{||l|l|l||}
\hline
\hline
 Coluna 1  & Coluna 2 & Coluna 3 \\
\hline
\hline
 Elemento 1 1 & Elemento 1 2 & Elemento 1 3 \\
\hline
 Elemento 2 1 & Elemento 2 2 & Elemento 2 3 \\
```

```
\hline
 Elemento 3 1 & Elemento 3 2 & Elemento 3 3 \\
\hline
 Elemento 4 1 & Elemento 4 2 & Elemento 4 3 \\
\hline
 Elemento 5 1 & Elemento 5 2 & Elemento 5 3 \\
\hline
 Elemento 6 1 & Elemento 6 2 & Elemento 6 3 \\
\hline
\hline
\end{tabular}
\caption{Esta é uma tabela simples, como a que já vimos
antes, mas com mais linhas de separação.}\label{tabsimp}
\end{table}

 Já a figura~\ref{figtext} contém a cópia da imagem de
 uma página de um outro documento que já vimos antes,
 incluída aqui em formato EPS e centrada na página. Ela
 também está dentro de um environment flutuante e contém
 uma legenda ou caption, com um label associado.

\begin{figure}
\centering
\epsfig{file=aula-09-imagem.eps,scale=0.8}
\caption{Esta é uma página formatada que foi escrita em
\LaTeX\ e incluída aqui como uma figura.}\label{figtext}
\end{figure}

\end{document}
```

Observe o comando `\centering` que centra o conteúdo dentro dos environments `table` e `figure`. Digite e rode este último exemplo, lembrando de rodar o compilador `latex` duas vezes para obter a numeração correta nas referências.

3 Problemas e Dicas

1. Rode 10 vezes o comando `fortune`, redirecionando o output para um arquivo chamado `fortune.tex`. Em seguida edite o arquivo com o Emacs ou outro editor de texto, colocando os comandos do LaTeX que são necessários para formatar o texto de forma apropriada. Dê uma olhada nos environments `quote` e `verse`, eles poderão ser úteis.

2. Tente combinar todos os elementos mostrados no texto da aula de forma a construir o esqueleto de um relatório de laboratório de física básica. O esqueleto consiste da estrutura de formatação do documento, possivelmente incluindo a sugestão dos nomes de seções, mas sem o conteúdo específico de qualquer experimento em particular. Observe que você pode incluir notas de rodapé numa página usando o comando `\footnote{<texto>}`. A numeração destas notas é automática.

4 Uma Lista de Comandos Relevantes

`dvips [-E] <arquivo>.dvi` : é um filtro que traduz arquivos em formato DVI para arquivos em formato PS; com a opção `-E`, produz os arquivos de saída em formato EPS.

`fortune` : coloca na tela frases engraçadas (bom, pelo menos às vezes) de forma aleatória; as frases podem ser curtas ou consistir na realidade de textos mais longos, com no máximo cerca de 24 linhas.

`gv <arquivo>.ps` : é um aplicativo X11 para a visualização de arquivos no formato PS (PostScript) ou EPS (Encapsulated PostScript); é independente de qual tenha sido a origem do arquivo, interpreta qualquer arquivo neste tipo de formato.

`latex <arquivo>.tex` : é um compilador que lê e interpreta os comandos de formatação da linguagem LaTeXescrevendo arquivos em formato DVI contendo documentos formatados; trata-se de um poderoso programa de processamento de texto, ou programa de "typesetting", que produz output formatado de alta qualidade.

`xdvi <arquivo>.dvi` : é um aplicativo X11 para a visualização de arquivos no formato DVI produzido pelo LaTeX.

Referências

[1] "LaTeX, a document preparation system", by Leslie Lamport; second edition, 1994, Addison-Wesley.

[2] "Tutorial de LaTeX", por Jorge L. deLyra, circa 1990, pode ser encontrado on-line no endereço `http://latt.if.usp.br/~delyra/tutorial/`.

5 Arquivos de Configuração

Como é o caso com a maior parte dos compiladores, o comando `latex` não usa arquivos de configuração. Por outro lado, ao contrário do que acontece normalmente com os compiladores de linguagens de programação tais como C, Fortran e Pascal, o `latex` não usa opções de linha de comando. Todos os comandos para o compilador devem ser incluídos junto com o texto no arquivo a ser compilado por ele.