

Introdução ao \LaTeX

Empresa Júnior Marie Curie

Gilberto M. Nakamura^a and Ana Carolina P. Monteiro^{a,b,1}

^aUniversidade de São Paulo; ^bEmpresa Júnior Marie Curier

A Empresa Júnior Marie Curie tem como um dos seus objetivos promover o crescimento técnico-científico do meio no qual está inserida. Este mini-curso é uma maneira de cumprir tal meta.

Sumário

1	Estrutura básica	2
2	Ambientes matemáticos	6
3	Figuras e ambientes flutuantes	7
4	Estruturas químicas	8
5	Citações e referências	10

Uma das dúvidas mais recorrentes sobre \LaTeX é a seguinte: qual a vantagem do \LaTeX sobre um processador de texto tradicional (por exemplo, *Word*)? Uma resposta simples seria o fato de ser gratuito, mas claramente isso **não** é verdade pois existem alternativas gratuitas para processadores de texto.

A vantagem repousa em duas diretrizes dos documentos \LaTeX . A primeira e mais importante é permitir a produção de documentos de alta qualidade por usuários *comuns*. Considere este documento, por exemplo. A atenta observação dos detalhes textuais – distribuição dos espaçamento entre os caracteres por linha, as fontes matemática, *etc* – permite inferir que estes variam de linha para linha, parágrafo por parágrafo, com intuito de facilitar a leitura. De quebra, a aparência textual é enriquecida.

A segunda vantagem repousa na padronização. Uma vez formatados, o resultado final é **independente** do tipo de computador, sistema operacional ou versão do *software*. Isto quer dizer que um arquivo \LaTeX feito na década de 80, se compilado hoje em qualquer sistema operacional, será idêntico à versão imaginada pelo seu autor. No caso de processadores de texto tradicionais, seus desenvolvedores vem se preocupando cada vez mais a oferecer suporte a versões descontinuadas, já que atualizações significam aumento de custos para empresas e usuários. Infelizmente, isso nem sempre é possível, seja por ausência de suporte ou descontinuação do *software*.

Naturalmente, tais diretrizes requerem um custo intelectual para autores. Para aqueles versados em linguagens de programação, a formatação \LaTeX é intuitiva e natural. De fato, um documento `.tex` pode ser tratado

como um código computacional. Porém, isso torna-se uma verdadeira barreira para a ampla adoção do \LaTeX por usuários comuns. Claro, toda montanha parece desafiadora à primeira vista; o mesmo acontece aqui. Contudo, uma vez entendidos os conceitos básicos sobre documentos \LaTeX , a barreira metafórica tornar-se-á, no máximo, um morrinho.

O \LaTeX possui código aberto e é distribuído gratuitamente para os sistemas operacionais mais relevantes. Usuários de Linux (baseado em sabores de Debian ou Ubuntu) devem instalar o programa por meio da linha de comando “`sudo apt-get install texlive-full`” e usuários OSX devem usar “`ports`”. Usuários de Windows devem realizar o download de uma distribuição, seguida da sua instalação. Existem duas versões populares:

- [MikTeX](#)
- [TeXlive](#)

Recomenda-se também a instalação do [TeXstudio](#) para facilitar a produção de documentos.

Lembrete

Durante a prática, a fim de evitar eventuais complicações com instalações em computadores pessoais, soluções em nuvem serão utilizadas. As duas plataformas gratuitas recomendadas são [Overleaf](#) e [SAGE](#). Em ambas, as marcações \LaTeX são atualizadas frequentemente e o documento final é exibido ao lado do documento texto puro.

Aspectos gerais

\LaTeX é recomendado para usuários que desejam publicar textos padronizados de alta qualidade, independente de sistema operacional ou versões de *software*. Essas características são compatíveis com publicações literárias e científicas. Fórmulas matemáticas recebem especial atenção e são amplamente utilizadas em outros formatos populares. Este manuscrito fornece contato inicial com \LaTeX e sua organização. Ao longo do texto, propõe-se exercícios de fixação. A duração total deste curso é de 4 horas.

GMN declara não existir conflito de interesse na produção deste material.

¹ E-mail: mariecuriejr.usp@gmail.br

Estrutura básica

A criação de um documento legível ocorre em duas etapas. Na primeira etapa, o autor formata o texto utilizando a *marcação* \LaTeX e seu conteúdo é salvo em um arquivo `.tex`. A estrutura de comunicação do texto segue as direções desejadas pelo autor e, por isso, é a fase que requer sua participação direta e criativa. O domínio dessa etapa requer tão somente o aprendizado da lógica por trás da marcação textual. A característica da marcação é o uso de *comandos* e *ambientes*. Devido à importância da marcação textual, será o principal objeto de estudo deste material.

Na etapa seguinte, o arquivo `.tex` é processado por um executável. Essa fase, conhecida por fase de compilação ou simplesmente compilação, não exige participação do usuário e transforma o arquivo com marcações \LaTeX em um arquivo legível como o `.pdf`, por exemplo. Existem diversos compiladores de \LaTeX . A escolha do compilador depende essencialmente das características desejadas do objeto final. Por exemplo, o compilador `latex` transforma o arquivo `.tex` em `.dvi`.¹ Atualmente, os compiladores mais relevantes são `pdflatex`, `xelatex` e `luatex`, pois o formato do arquivo final é `.pdf`.

Notação

Para facilitar o aprendizado da marcação, realçamos a sintaxe \LaTeX por meio de cores.

```
\comando{argumento}
\begin{ambiente}
  conteúdo do ambiente %comentário1
  % comentário2
\end{ambiente}
```

Comandos e ambientes. A sintaxe por trás do esquema de marcação de documentos \LaTeX emprega *comandos* e *ambientes* para gerenciar tanto a aparência e formato geral do documento – se é livro ou artigo, se folha A4 ou A5, etc – como cada ente textual – seção, capítulo, etc.

Comandos são os menores elementos de marcação presente em \LaTeX e são introduzidos pelo autor, a fim de modificar alguma propriedade do texto ou mesmo do documento. Eles apresentam uma característica recorrente em linguagens de programação: **comandos** são identificados pelo símbolo “\”, possuem **argumentos** e, eventualmente, **opções**, como ilustrado abaixo.

```
\comando[opção]{argumento}
```

Argumentos são identificados por chaves `{argumento}`, enquanto opções são identificadas por colchetes

¹ O formato DVI produz um documento binário, isto é, um arquivo apropriado para computadores e programas computacionais. Em geral, arquivos binários não são legíveis por humanos não treinados. O formato DVI apesar de não ser popular pode ser convertido em diversos formatos, inclusive pdf.

[*opções*]. Há ordenamento explícito: opções aparecem entre os nome do comando e seus argumentos. A definição de seções ou capítulos são bons exemplos desta regra.

```
\section{Nome da seção}
\chapter{Nome do capítulo}
```

Nestes casos, “Nome da seção” claramente é argumento do comando `\section`. O mesmo raciocínio aplica-se ao comando `\chapter`.

A presença de argumentos nem sempre é necessária. Existem comandos como

```
\maketitle
\newpage
```

que não precisam de argumentos. Aqui, `\newpage` encerra o uso da página atual e continua o texto na página seguinte, enquanto `\maketitle` produz a capa com título e autores do documento. Outra possibilidade é que o comando precise de mais de um argumento. Nesse caso, basta adicionar quantos argumentos forem necessários, sempre separados por chaves:

```
\comando{opção1}{opção2}{opção3}
```

Não convém listar, neste momento, todos os comandos disponíveis. O motivo é bem simples: a quantidade de comandos padrões é enorme. Na prática, utiliza-se frequentemente apenas alguns poucos comandos de texto. Por outro lado, existem inúmeros comandos matemáticos amplamente utilizados. A boa notícia é que eles geralmente descrevem a estrutura matemática desejada, o que facilita seu uso adequado. Esses serão objeto de estudo em seção separada.

Já os *ambientes* podem ser vistos como uma sequência ordenada de comandos, visando a padronização de textos, fórmulas matemáticas ou elementos flutuantes como figuras e tabelas. Assim como os comandos, ambientes são facilmente identificáveis. A diferença é que ao invés de um único símbolo, utiliza-se dois comandos: o início do ambiente usa o comando `\begin{nome do ambiente}` enquanto o comando `\end{nome do ambiente}` indica o término do ambiente. Tudo que se encontrar entre `\begin` e `\end` e considerado como elemento do ambiente e será processado de acordo com suas regras.

```
\begin{ambiente}
  texto contido no ambiente
\end{ambiente}
```

Veremos diversos ambientes em detalhes nas seções apropriadas mas uma prévia vale a pena. Considere o ambiente `equation`. Ele produz equações matemáticas formatadas de acordo com o estilo do documento. Por exemplo,

```
\begin{equation}
  \imath\hbar\frac{\partial\psi}{\partial t}=
  \mathcal{H}\psi.
\end{equation}
```

produz

$$i\hbar \frac{\partial \psi}{\partial t} = \mathcal{H}\psi. \quad (1)$$

Finalmente, o símbolo % é interpretado como delimitador de comentários. Tudo que estiver na linha após % será ignorado pelo compilador. Comentários são ferramentas muito úteis para documentar modificações e testes, como ilustra o exemplo abaixo.

```
% o comando \author{} serve para  
% identificar o autor do texto  
\author{meuNome}
```

Documento. Comandos e ambientes permitem mudanças e ajustes finos na estrutura do texto, de maneira a melhor adequá-lo às vontades do autor. Contudo, tais ações são pequenas dentro da classe e estilo do documento. Nesta seção, iremos apresentar a estrutura geral de um documento L^AT_EX; como alterar a classe e ajustar os aspectos mais relevantes do documento, tal qual linguagem e modificação. A Fig. 1 ilustra o esquema geral.

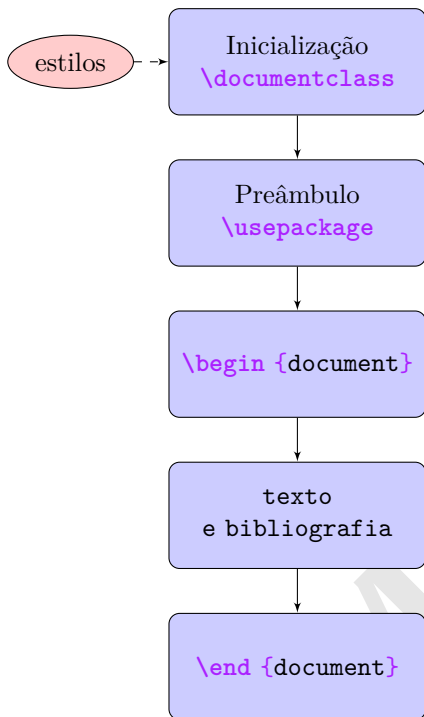


Fig. 1. Esquema geral para documentos L^AT_EX. A escolha da classe e o estilo ocorrem por meio do comando `\documentclass {classe}`. No preâmbulo, o autor define pacotes adicionais para personalizar e introduzir ferramentas adicionais ao documento através do comando `\usepackage {nomepacote}`. O texto a ser disponibilizado para leitura, devidamente formatado, é inserido no ambiente `document`.

Qualquer arquivo `.tex` é dividido em duas partes, a saber, o preâmbulo e o corpo. O preâmbulo contém as definições sobre o formato geral e o estilo do documento a ser gerado. O preâmbulo sempre é iniciado pelo comando `\documentclass`:

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

O argumento `classe` especifica qual o padrão geral de formatação. Dentre as classes pré-definidas, três muito relevantes:

- `article`;
- `book`;
- `report`.

Cada modelo apresenta um conjunto de características e normas. Não cabe ao autor se preocupar com as normas, isto é tarefa do L^AT_EX², mesmo porque normas variam de classe para classe.

Algumas dessas características são comuns a todos, outras nem sempre. É o caso de título e nome dos autores, tamanho do papel, tamanho da fonte, etc. No caso do `article`, o título (`\title`) é sucedido pelo nome dos autores (`\author`) e respectivas filiações. Em seguida, há opcionalmente uma área destinada ao resumo do artigo (`\abstract`) que precede a definição das palavras-chave (`\keywords`), identificador de objeto digital (`\doi`) e o corpo do texto. Apenas seções (`\section`), subseções (`\subsection`) e apêndices (`\appendix`) são permitidos. Sumários (`\tableofcontents`) são opcionais para manuscritos extensos. Já no modelo (`book`), a capa encontra-se numa folha separada, sucedida ou não por folha em branco e ficha catalográfica, e com corpo textual separado em partes (`\part`) e capítulos (`\chapter`).

Ainda existem as opções não-relacionadas às normas das classes como tamanho da fonte e do papel. Essas opções afetam, geralmente, o estilo da classe. Se nenhuma opção for passada ao comando `\documentclass`, assume-se as opções padrões. Abaixo encontra-se a lista com as opções padrões e suas variações mais importantes:

- `letterpaper` : padrão utiliza papel de tamanho $216\text{mm} \times 279\text{mm}$ (carta). Alternativamente, a opção `a4paper` especifica o papel A4;
- `10pt` : padrão de fonte 10. As opções `11pt` ou `12pt` incrementa a fonte para 11 ou 12, respectivamente;
- `onecolumn` : única coluna. A opção `twocolumn` especifica corpo em duas colunas. É possível obter múltiplas colunas no corpo do documento por meio do ambiente `multicolumn`.

Classes genéricas são úteis para teste e até mesmo produção final de arquivos. Além destas, existem outras classes que apresentam melhorias significativas na apresentação visual e organizacional do texto. Essas classes são, em geral, derivadas das versões genéricas, construídas e disponibilizadas gratuitamente por usuários e/ou editoras. Ao longo deste curso, duas classes derivadas para artigos científicos serão utilizadas:

²Essa afirmação segue o espírito das diretrizes básicas. Entretanto, alguns comandos exigem algum tipo de ordenamento. O comando `\maketitle` requer que tanto `\author` quanto `\title` tenham sido previamente definidos, não importando a posição relativa entre os mesmos.

- a) RevTeX (`revtex4-1`), utilizada pela *American Physical Society* (APS) e pelo *American Institute of Physics* (AIP);
- b) Achemso (`achemso`), utilizada pela *American Chemical Society* (ACM).

Existem diversos outros modelos disponíveis, seja na distribuição de L^AT_EX ou nos repositórios das editoras. A escolha do modelo mais apropriado cabe inteiramente ao autor.

É importante mencionar a modificação do modelo de classe altera *apenas* a formatação do conteúdo do documento, mantendo o texto intacto. Se quisermos mudar um texto da classe genérica `article` para a classe derivada `revtex4-1`, no estilo da AIP, compatível com modelo impresso, basta mudar de

```
\documentclass{article}
```

para

```
\documentclass[aip,reprint]{revtex4-1}
```

Classe RevTeX

Esse modelo é popular não só para a submissão de artigos científicos como também preparação de trabalhos e projetos científicos. Um dos motivos de sua popularidade é que com a opção `reprint`, o arquivo final assemelha-se com a versão final impressa. O conjunto de detalhes resulta num produto final com elevada qualidade. As opções mais comuns são

- `reprint` ou `preprint`: determina se o documento será semelhante à versão impressa final ou mais apropriada para correções;
- `aps` ou `aip`: determina se o documento seguirá o padrão da APS ou da AIP. No caso da versão APS também deve-se escolher o estilo, que depende da revista (`pri`, `pra`, etc);
- `letterpaper` ou `a4paper`;
- `10pt`, `11pt` ou `12pt`: altera o tamanho da fonte;
- `linenumbers`: enumera as linhas do documento;
- `showkeys` e `showpacs`: mostram, respectivamente, áreas reservadas para palavras-chave e código PACS.

O RevTeX requer os comandos `\autor` e `\title` dentro do ambiente `document`, enquanto o comando `\maketitle` é executado somente após o `abstract`. Essas três particularidades diferem do comportamento genérico da classe `article`.

Ainda no preâmbulo, o usuário pode definir novos comandos e referenciar pacotes contendo conjuntos de comandos com ações específicas, fontes adicionais, dicionários, etc. É esse conjunto de pacotes que dá ao usuário final do L^AT_EX um grande leque de ferramentas, habilitando a realização de tarefas muitas vezes complexas ou repetitivas.³ Para se introduzir um pacote, utiliza-se o comando

```
\usepackage[opção]{nomedopacote}
```

O usuário deve decidir quais pacotes utilizar dependendo unicamente de sua necessidade. Por exemplo, caso o usuário necessite de fontes matemáticas `amsmath`,⁴ basta escrever

```
\usepackage{amsmath}
```

Antes de finalizar a discussão sobre o preâmbulo, convém explicitar alguns detalhes técnicos necessários para a Língua Portuguesa, mais especificamente sobre acentos. Nas línguas latinas, a acentuação tem lugar cativo nas regras ortográficas, aparecendo em diversas palavras, sob os mais diversos símbolos. O mesmo não ocorre na língua Inglesa. Por motivos históricos – muito infelizes, diga-se de passagem – a codificação de caracteres de entrada, o que é digitado e transcrito para o computador, bem como as fontes de saída, não contemplavam acentos latinos. Esse provavelmente é uma das fontes de irritação e também que dificultam a adoção do L^AT_EX para autores na Língua Portuguesa. Todavia, esse problema é fácil de ser contornado através de três pacotes apropriados:

```
\usepackage[brazilian]{babel}
```

Especifica o Português Brasileiro como língua do corpo documental. Essa opção afeta principalmente a hifenação de palavras mas também dá conta de regras particulares à Língua selecionada. Outras alternativas para o pacote `babel` são `english` e `portuguese`, Inglês e Português Europeu, respectivamente.

```
\usepackage[utf8]{inputenc}
```

O pacote `inputenc` descreve a codificação dos caracteres de entrada, que nada mais é do que a associação de determinados símbolos com códigos binários. Parece complicado mas um exemplo deixará isso mais claro. Considere o símbolo “ç” (cedilha). Ele não consta na tabela ASCII, que nos primórdios da Computação era a tabela dominante. Isso significa que a palavra “criança” seria transcrita como “criana”. A introdução da opção `utf8` modifica a codificação

³ Os pacotes `ifthen` e o comando `foreach` permitem a execução de cláusulas condicionais e de laços de execução, facilitando a codificação de tarefas.

⁴ Autores de textos matemáticos frequentemente utilizam a trinca `amsmath`, `amsmath`, `amssymb` que definem comandos matemáticos, fontes e símbolos definidos pela *American Mathematical Society* (AMS).

para o padrão Unicode-8, amplamente utilizado nos dias atuais, e que contém acentos e cedilha.⁵

```
\usepackage [T1]{fontenc}
```

Especifica a codificação da fonte do arquivo final. A opção T1 contempla fontes latinas.

Uma vez terminado o preâmbulo, realiza-se a edição de conteúdo. Todo o texto que será exibido na versão final deve ser colocado dentro do ambiente `document`:

```
\begin{document}
```

Aqui entra o texto, organizado em capítulos, seções e subseções conforme desejado pelo autor.

```
\end{document}
```

As separações lógicas do texto são delegadas ao autor, isto é, quantas seções e subseções, se existirá apêndices ou sumário, etc. Ao \LaTeX cabe normatizar as divisões textuais, sumário, referências, cabeçalho e rodapé, hifenização, disposição de figuras e a renderização e numeração de equações matemáticas. Adicionalmente, o \LaTeX também normatiza espaçamentos no texto antes de seu processamento. Assim, se o autor escrever

O ovo é azul, acho

ou

O ovo é
azul
, acho

o resultado final será “O ovo é azul, acho”, que demonstra a insensibilidade com quebras simples de linha e espaços brancos. Quebras duplas de linha, isto é, uma linha em branco determina o início de um novo parágrafo.

Abaixo listamos os comandos de organização textual:

- linha vazia

Novo parágrafo

- `\section {nomeseção}`

Seção

- `\subsection {nomesubseção}`

Subseção

- `\chapter {nomecapítulo}`

Capítulo

- `\tableofcontents`

Sumário

- `\maketitle`

Construção do título seguido de autor.

Classes Article e Achemso

A classe genérica `article` serve de base para uma grande variedade de outros modelos como, por exemplo, `achemso`. Esta última é a classe pré-definida pela ACS, que recomenda sua adoção na submissão de manuscritos científicos para um conjunto formado por 50 revistas, entre elas *Chem. Rev.* e *Nano Lett.*

As diferenças mais relevantes entre as duas classes é o conjunto de opções iniciais. No caso da `achemso`, duas opções são alteram o produto final:

`layout=traditional` ou `twocolumn`

Determina se o formato será única coluna e duplo espaçamento ou duas colunas e espaçamento simples, respectivamente.

`journal=journal_code`

A opção `journal` determina o estilo do periódico indicado.

- `ancac3` : *ACS Nano*
- `chreay` : *Chem. Rev.*
- `nalefd` : *Nano Lett.*

Exercício

Exemplo mínimo de arquivo LaTeX.

- 1) Criar um documento da classe `article` (com suporte à Língua Portuguesa).
- 2) Invocar o pacote `lipsum`, que define o comando:
`\lipsum[paragrafo inicial - final]`.
- 3) Deve conter título e nome do autor.
- 4) Preencher `abstract` e corpo com o comando `\lipsum`.

Exercício

Seleção de classes de documentos

- 1) Alterar a classe do documento de `article` para `achemso`
 - a) `layout=traditional`, `journal=ancac3`
 - b) `layout=twocolumn`, `journal=ancac3`
- 2) Alterar a classe do documento para `revtex4-1`:
 - a) `aip,preprint`
 - b) `aip,reprint`

⁵ Outra possibilidade é o uso de comandos de acentuação. Esses comandos aparecem como `crian\c {c}a` ou `\' {a}gua`. Não se recomenda o uso de comandos de acento por dificultarem desnecessariamente a leitura do arquivo `tex` e por exigirem opções adicionais de corretores ortográficos.

Resumo

```
\documentclass[a4paper]{article}
\usepackage[brazilian]{babel} %pt_BR
\usepackage[utf8]{inputenc} %entrada
\usepackage[T1]{fontenc} %saida
\title{Meu título chamativo}
\author{nome}
%ambiente document
\begin{document}
\maketitle %título seguido do abstract
\begin{abstract}
Meu resumo bacana
\end{abstract}
%texto
\section{Introdução}
Começa aqui ...
\section{Conclusão}
... termina aqui.
\end{document}
```

Ambientes matemáticos

Documentos \LaTeX tendem a ser mais populares em áreas como Matemática, Física, Química, Computação, Engenharias e, mais recentemente, Biologia Quantitativa. Isto é devido a maior frequência e complexidade das fórmulas matemáticas, que são o ponto forte da marcação \LaTeX . Para o aproveitamento adequado desta seção, reiteramos que elementos matemáticos são *sempre* tratados como texto e, portanto, seguem as mesmas regras textuais (pontuação, espaçamento, etc).

Grosso modo existem dois modos de exibição para símbolos matemáticos. O primeiro, *inline*, permite o uso de símbolos e fórmulas lado a lado com texto. Isto se dá por meio do delimitador $\$ \dots \$$. Por exemplo, o trecho “O rato sabe determinar $\partial_t \psi(\vec{r}, t)$ ” é fruto de

```
O rato sabe determinar
\partial_t \psi(\vec{r}, t).
```

Note que $\$$ delimita o começo e fim da expressão matemática. Dentro da expressão matemática, é fácil reconhecer vários comandos como $\backslash partial$ que representa ∂ enquanto $\backslash vec{r}$ adiciona uma seta sobre a variável r , representando um vetor. Voltaremos aos símbolos em breve.

O segundo modo de exibição é o ambiente `equation` ou um de seus derivados como `align` e `\eqnarray`. No ambiente `equation`, o texto é interrompido e mostra-se a fórmula matemática numa linha separada, seguida de uma numeração (normalmente no lado direito). Por exemplo, considere o texto bruto

```
Uma das equações físicas mais importantes,
\begin{equation}
\nabla \times \vec{E}(\vec{r}, t) =
```

Tabela 1. Símbolos para modo matemático.

α	<code>\alpha</code>	∇	<code>\nabla</code>
ψ	<code>\psi</code>	$\frac{a}{b}$	<code>\frac {a}{b}</code>
\sum	<code>\sum</code>	\vec{a}	<code>\vec {a}</code>
\int	<code>\int</code>	∂	<code>\partial</code>

```
-\frac{1}{c}\frac{\partial}{\partial t}
\vec{B}(\vec{r}, t),
\end{equation}
```

raramente é apreciada.

resulta em "Uma das equações físicas mais importantes,

$$\nabla \times \vec{E}(\vec{r}, t) = -\frac{1}{c} \frac{\partial}{\partial t} \vec{B}(\vec{r}, t), \quad (2)$$

raramente é apreciada."⁶

Para dominar o modo matemático é necessário treino repetitivo. Claro, existe uma certa lógica por trás da maioria dos comandos matemáticos. Seu conhecimento facilita a familiarização dos comandos e identificação de erros. Dito isto, dentro do modo matemático,

- um símbolo pode ser tanto um carácter de entrada quanto um símbolo gerado por comandos. Por exemplo, a letra grega β é gerada pelo comando `\beta`;
- dois símbolos lado a lado descrevem produto, como visto em `\$a b\$`, que após processamento fica ab . Caso seja necessário explicitar o produto, recomenda-se os símbolos `\times` ($a \times b$) ou `*` ($a * b$);
- chaves são delimitadores. Isto quer dizer que chaves não são processadas como símbolos a menos que exigido de modo explícito. Por exemplo, `{a} = a`;
- símbolos subscritos são gerados mediante o comando `_ { }`. Por exemplo, `a_{b} = a_b`;
- símbolos supraescritos são gerados por `^ { }`. Por exemplo, `a^{b+c} = a^{b+c}`.

Exercício

Com o pacote `amsmath`:

- Adicionar ambiente `equation`
 - uma simples expressão algébrica
 - duas equações alinhadas
 - uma equação diferencial
 - uma integral
 - uma integral de caminho

⁶Vale a pena reforçar que fórmulas matemáticas devem ser sempre tratadas como elementos textuais. Por isso a Eq. (2) carrega uma vírgula.

Figuras e ambientes flutuantes

Figuras são parte integral de documentos. Tipicamente, uma figura contém uma ou mais imagens, gráficos ou ilustrações. Logo abaixo do conteúdo visual, coloca-se a legenda da figura, que nomeia e explica a informação nela contida. Tabelas são elementos semelhantes, mas geralmente a legenda posiciona-se acima dos valores da tabelados. Do ponto de vista funcional, ambos ambientes possuem dois elementos, um gráfico e outro textual. Outra característica importante é que suas posições não são fixas ao longo do texto. O motivo é bem simples: se o conteúdo textual anterior à figura for alterado, sua posição atualizada pode ocorrer em local indevido, quebrando a estrutura documental. Recebem, portanto, a denominação de ambientes flutuantes.

Figuras. O \LaTeX tenta posicionar objetos flutuantes da mesma maneira que textos, objetivando melhor aparência e uniformidade de pixels. Ele também observa a integridade de figuras ou tabelas, garantindo que não serão quebradas em páginas diferentes.

Outra questão importante é o gerenciamento de formatos de imagens. Suponha `pardosol.jpg` seja uma imagem a ser incluída. Seu formato é `.jpg` mas o documento `.tex` final pode ser, por exemplo, um arquivo `.pdf`. Portanto, é necessário executar a conversão do arquivo `.jpg` para o formato `.pdf`. Para evitar que o usuário execute manualmente essa conversão, costuma-se chamar o pacote

```
\usepackage{graphicx}
```

A conversão, entretanto, não garante que a qualidade da imagem final fique inalterada. De fato, a conversão automatizada entre diferentes formatos geralmente resulta em deterioração significativa da imagem.⁷ Isso é especialmente verdade para imagens não-vetoriais como `.jpg` e `.png`. Por esse motivo, recomenda-se a utilização de imagens vetoriais como `.pdf` ou `.eps`, cuja conversão entre formatos produz menores perdas e fragmentos. No caso do formato *Encapsulated PostScript* (EPS), a imagem em si não é um conjunto de pontos numa matriz bidimensional, mas sim um programa *PostScript* que quando interpretado reproduz pixels da imagem. Por ser um programa, as variáveis como largura, comprimento e resolução são facilmente obtidas pelo \LaTeX .

Regra de ouro

Utilize sempre que possível imagens EPS.

A inclusão mais comum de figuras ocorre através do ambiente `figure`. Dentro deste ambiente, utiliza-se o comando

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

para incluir conteúdo visual. Acima, consideramos uma imagem armazenada no arquivo `nomedoarquivo.eps`, no *mesmo* diretório onde se encontra o arquivo `.tex`. O extensão poderia ser JPG, PNG ou TIFF também. Caso a imagem estivesse em outro diretório, bastaria utilizar o caminho e nome do arquivo como argumento. Por exemplo, se a imagem `nuclearb.eps` estivesse no diretório `\home\topsecret\`, o comando ficaria

```
\includegraphics{\home\topsecret\nuclearb.eps}
```

Em princípio pode-se incluir quantas imagens necessárias desde que seu tamanho não exceda o tamanho da página. A organização das imagens dentro do ambiente `figure` é o mesmo que no texto. Várias comandos `\includegraphics` lado a lado irão se comportar da mesma maneira que texto num parágrafo. Se for necessário adicionar um novo parágrafo, pula-se uma linha. Para centralização utiliza-se o comando `\centering`, etc. Vejamos um exemplo mais concreto:

```
\begin{figure}
\centering
\includegraphics[width=0.5\textwidth]{fig1.eps}
\includegraphics[width=0.5\textwidth]{fig2.eps}
\includegraphics[width=0.5\textwidth]{fig3.eps}
\caption{Posicionamento de imagens. Aqui as
duas primeira imagens ficarão lado a lado,
enquanto a terceira será centralizada logo
abaixo das anteriores.}
\end{figure}
```

O comando `\caption` descreve a legenda da figura. Uma boa figura sempre vem acompanhada de um título e descrição apropriada do significado do conteúdo visual.

Note que, ao contrário do texto, o algoritmo nem sempre posiciona o `figure` na posição do desejada inicialmente. Para melhor direcionar o posicionamento, o ambiente `figure` admite um conjunto de opções

```
\begin{figure}[opções]
```

onde as opções são formadas pela concatenação de caracteres de posicionamento. Os caracteres mais importantes de posicionamento são:

- `h` : sugere posicionamento próximo ao local de definição;
- `t` : sugere posicionamento no topo da página;
- `b` : sugere posicionamento na base da página;
- `!` : ignora alguns parâmetros do \LaTeX nas sugestões seguintes.

⁷ Não confundir deterioração causada por conversão com deterioração causada por baixa resolução.

Assim, a opção `ht` irá sugerir o posicionamento da figura na região textual definida pelo autor. Caso falhe, tentará novamente no topo da próxima página. Já a opção `!ht` irá desconsiderar parâmetros internos estéticos (talvez a uniformidade de pixels, etc) e tentará colocar a figura na região de ocorrência ou no topo da próxima página. De certo modo, a opção “!” transmite urgência no posicionamento.

Exercício

Com o pacote `graphicx`:

- 1) Adicionar a figura `fig1.eps` com alguma legenda.
- 2) Dentro de um mesmo ambiente, adicionar duas imagens: `fig1.eps` e `imagem_descartada.pdf`.

Tabelas. O ambiente `table` insere um objeto flutuante que pode ser uma tabela propriamente dita ou um quadro, que diferem entre si pelas delimitações externas. Os objetos organizados, sejam eles números ou não, são tabulados pelo ambiente `tabular`. Este ambiente requer um conjunto de opções para especificar o número de colunas e seu alinhamento, sendo muito similar a uma matriz.

```
\begin{table}
  \caption{Exemplo de tabela. Elementos numa
    linha são separados por \&. Linhas
    extras são adicionadas com \verb+\\+
    Ela possui duas colunas com
    alinhamento à esquerda e uma barra
    vertical separando seus elementos.}
  \begin{tabular}{l | l} %duas cols
    entrada & saída \\
    $1$ & ovo \\
    feijão & $x^2$
  \end{tabular}
\end{table}
```

O comando `\hline` adiciona linhas horizontais onde o usuário desejar. Recomenda-se, porém, o uso mínimo de linhas, minimizando a poluição visual. Além dos três alinhamentos (l,c,r), também é possível alinhar números a partir do ponto decimal. Isto requer a definição do pacote `dcolumn`

```
\usepackage{dcolumn}
```

que define o alinhamento `D{.}{.}{-1}`, como mostra a Tabela 2.⁸

Resumo

```
\begin{figure}[ht]
  \centering
  \includegraphics{fig.eps}
  \caption{legenda}
```

⁸Colunas com alinhamento decimal são sempre tratadas como ambientes matemáticos.

```
\end{figure}
```

Estruturas químicas

Para os adeptos das artes da Química, o \LaTeX apresenta uma ferramenta muito útil. Trata-se do desenho bidimensional de estruturas químicas.⁹ Essencialmente trata-se de uma imagem construída inteiramente no próprio documento, com alta qualidade e independente de resolução.

Claramente, o uso da linguagem `tex` para essa tarefa seria uma tarefa muito árdua e, provavelmente, infrutífera. O pacote `chemfig` simplifica todo o processo. Ao autor cabe determinar a ordem dos átomos e suas ligações enquanto o \LaTeX se encarrega de desenhar a estrutura. Para isso, o pacote `chemfig` define um tipo especial de linguagem interpretada, ou *script*.

A partir de alguns poucos comandos de `script`, será possível construir estruturas complexas. Para começar, como sempre, definimos o comando principal

```
\chemfig{argumento}
```

Fácil! Agora basta determinar o argumento, mais precisamente, a estrutura molecular. Bem, primeiramente, moléculas são constituídas por átomos ligados entre si por meio de interações eletromagnéticas e spin. Átomos são representados por símbolos da tabela periódica enquanto ligações são representadas por barras simples, duplas ou triplas.

Dentro do comando `\chemfig`, dois objetos A e B compartilham uma ligação simples, dupla e tripla através dos `scripts`

```
\chemfig{A-B}           A — B
\chemfig{A=B}           A = B
\chemfig{A~B}           A ≡ B
```

Para adicionar mais átomos, basta adicionar mais letras e ligações

```
\chemfig{O=C=O}           O = C = O
```

Para enfatizar a ligação ou simplesmente diminuir a poluição visual, átomos podem ser suprimidos:

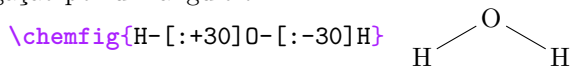
```
\chemfig{A_1~}           A1 ≡
\chemfig{=}               =
```

⁹Também é possível desenhar figuras tridimensionais mas nesse caso convém utilizar a imagem exportada de um software externo, pois 3D requer ângulo e iluminação correta para devida análise.

Tabela 2. Exemplo de alinhamento decimal. Primeira coluna possui alinhamento esquerdo enquanto a segunda e terceira colunas possuem alinhamento decimal. A terceira coluna substitui o ponto por espaço branco.

2.00	2.00	2 00
+2.00	+2.00	+2 00
-2.10	-2.10	-2 10
-10.99	-10.99	-10 99

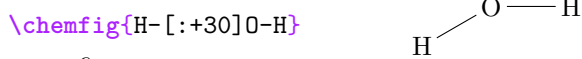
Naturalmente, nem toda molécula é linear e simples. Ao contrário, devido às interações eletrônicas, a minimização de energia requer estruturas mais flexíveis, capazes de suportar modos vibracionais e rotacionais. Dito isto, ligações com ângulos são geradas por meio de $[:\theta]$ concatenado à ligação correspondente. Nesse caso, o trecho adicional corresponde a um modificador, uma rotação da ligação por um ângulo θ :



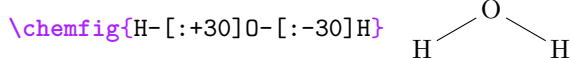
Pode parecer complicado mas a regra de construção sempre é feita por pequenos passos. Primeiro considera-se o caso da maior cadeia linear



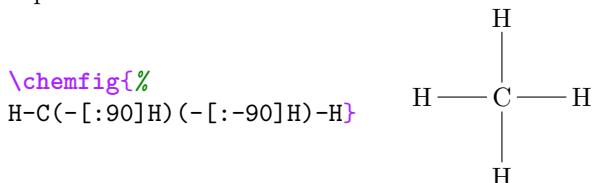
em seguida, modifica-se a primeira ligação com uma rotação de $\theta = \pi/6$



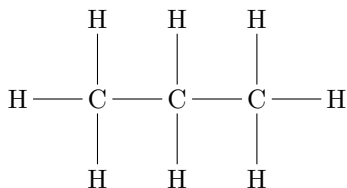
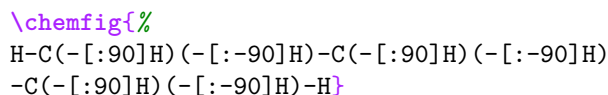
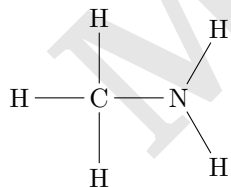
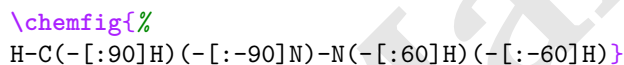
e, por fim,



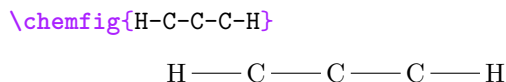
Para descrever ligações com múltiplos átomos, utiliza-se parênteses



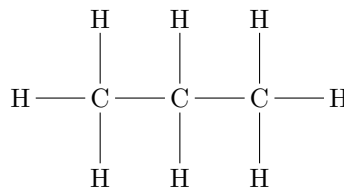
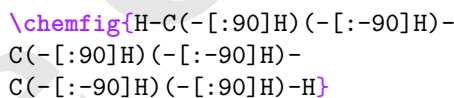
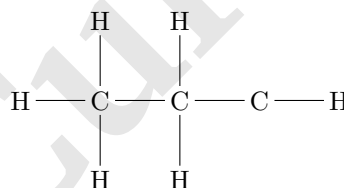
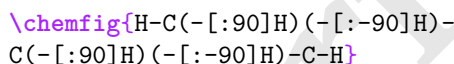
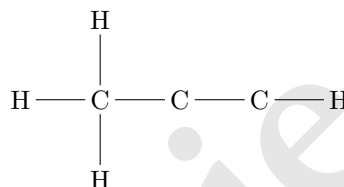
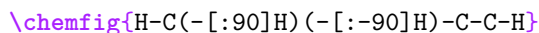
Neste caso, o átomo de carbono vem acompanhado por três pares de parênteses da forma (tipo_ligação[:ângulo]átomo). O primeiro elemento dentro do parênteses é o indicador do tipo de ligação, que pode ser -, = ou ~. A ligação é então modificada pelo ângulo correspondente e sua extremidade se conecta com o átomo. Outros exemplos mais complexos podem ser construídos



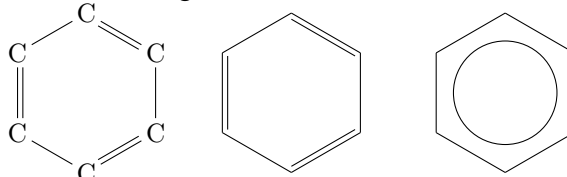
À primeira vista, o argumento de `chemfig` para o propano não parece fácil. Para apreciar a simplicidade do processo e construir estruturas ainda mais complexas é preciso compreender a lógica de montagem passo-a-passo. Novamente, a regra sempre é tentar escrever a maior cadeia linear



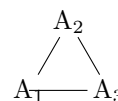
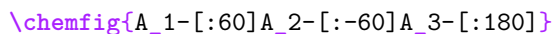
Em seguida, adiciona-se dois átomos H ao primeiro carbono, depois ao segundo e, por fim, ao terceiro:



Outro argumento muito importante para o comando `chemfig` descreve estruturas fechadas como anéis



Existem duas maneiras de especificar anéis de N átomos. A primeira consiste em especificar um cadeia linear adicionando uma ligação sujeita a rotação de $2\pi/N$. O detalhe importante é que o N -ésimo átomo deve se conectar com o primeiro, fechando a cadeia:



O problema dessa abordagem ocorre exatamente na última ligação, que é maior do que deveria e sobrescreve o primeiro átomo. Algumas correções consistem em manualmente ajustar a última ligação ou adicionar um átomo “extra”:

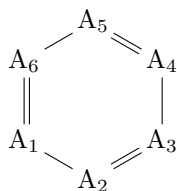
```
\chemfig{A_1-[:60]A_2-[:-60]A_3-[:180]A_1}
```



Nenhuma das soluções é completamente satisfatória pois quebram a lógica de construção das cadeias, impactando negativamente a utilidade do próprio comando.

Uma solução mais elegante contempla o uso da ideia de grupos “()” e dois novos símbolos. Cadeias de N átomos são descritas por `atomo*N(cadeia)`. O símbolo `*N` especifica que será uma cadeia fechada por N objetos. A variação `**N` desenha um círculo de tamanho apropriado no interior do anél, indicando a deslocalização eletrônica. Já `(cadeia)` especifica as ligações e átomos na cadeia. Por exemplo,

```
\chemfig{A_1*6(-A_2=A_3-A_4=A_5-_6=)}
```



Exercício

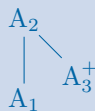
Com o pacote `chemfig`:

- 1) Criar um ambiente figura
- 2) Construir a molécula $C_6H_5NH_2$
- 3) Construir a molécula H_3PO_4

Resumo

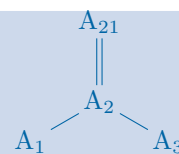
i. Ângulos:

```
\chemfig{A_1-[:90]A_2-[:-45]A_3^+}
```



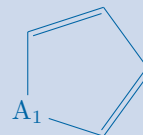
ii. Múltiplos átomos:

```
\chemfig{A_1-[:30]
A_2(=[:90]A_{21})-[:-30]A_3}
```



iii. Anéis:

```
\chemfig{A_1*5(-----)}
```



Citações e referências

Um dos aspectos mais úteis do \LaTeX é o mecanismo de referência e citações. A implementação da referência é bastante simples: envolve atribuição de identificadores, como a maioria dos sistemas de marcação. O grande vilão, como sempre, é o sistema de citação. Não que a implementação seja particularmente complexa: o problema das citações é burocrático. Isto se deve à atribuição adequada de créditos às fontes corretas. Além disso, citações também podem prover credibilidade a argumentos e hipóteses e, por isso, são alvos de normatizações rigorosas. Por exemplo, manuais tradicionais do estilo de citação Chicago têm tipicamente 1000 páginas. No \LaTeX , a ideia é remover essa carga burocrática do autor e passá-la diretamente ao compilador e/ou ferramentas auxiliares.

Identificadores e referência. Vamos então começar pela tarefa mais simples. Aqui, por referência entende-se a habilidade de indicar algum ambiente ou componente textual unicamente. Como exemplo, convém referenciar a Eq. (2). Isso é feito em dois passos. No primeiro passo, durante a criação do ambiente `equation`, colocou-se um comando adicional dentro do ambiente:

```
\label{eq:tut1}
```

Esse comando não possui nenhum tipo de representação visual. Ele apenas indica que o ambiente no qual está inserido pode ser referenciado pelo nome `eq:tut1`. Esse nome é de livre escolha do autor. Convém criar algum tipo de organização mas não é necessário.¹⁰ O nome escolhido poderia ter sido até mesmo `\label{1}` ou quem sabe algo mais exótico como `\label{ovomaltine}`. Nesse caso, teríamos

```
\begin{equation}
z=x^2+y^2
\label{ovomaltine}
\end{equation}
```

¹⁰A escolha deste autor para nomeação de ambientes não poderia ser mais desinteressante: “eq:tut1”. O primeiro termo “eq:” diz que o ambiente é uma equação e “tut1” indica que é a primeira equação do material que pode ser referenciada.

O comando `\label` pode ser incluído em qualquer ambiente. Para seções, capítulos e derivados, `\label` também pode ocorrer tanto dentro quanto após o comando de divisão correspondente. Veja por exemplo

```
\section{\label{secao1} Introdução}
...
\section{Conclusão}
\label{conclusao}
...
```

No segundo passo, executa-se a referenciação propriamente dita através do comando `\ref`. Esse comando indica ao \LaTeX para colocar, no local indicado, a numeração associada ao respectivo `\label`. Assim, `\ref{eq:tut1}` produz Eq. (2). Note que apenas o número foi substituído, “Eq.” e os parênteses ficaram inalterados. O mesmo raciocínio é aplicado a figuras e tabelas.

Citações. Inserir citações é uma tarefa bastante simples, basta utilizar o comando

```
\cite{nomecitacao}
```

Esse comando é muito similar ao comando `\ref`. Porém, no caso deste último, utilizava-se como argumento o nome previamente armazenado pelo comando `\label`. A pergunta natural a ser feita é: existe um comando equivalente para nomear os argumentos das citações? Claro! Mas existem duas implementações diferentes: uma exige que o autor lide manualmente com as referências (e normas!), enquanto a outra tem caráter automatizado. Aqui, veremos apenas a última, mais conhecida por **bibtex**.

Bibtex, assim como \LaTeX é um processador mas especializado na citação de obras literárias. Ele requer apenas três informações, todas passadas por meio de comandos:

i. `\bibliographystyle {estilo}`

Este comando especifica o estilo – e normas – que as citações devem obedecer, tanto no texto quanto na descrição da referência literária. Uma vez definido o estilo, o usuário precisa apenas decidir o que citar e o **bibtex** irá determinar a forma correta de inserção da citação. Lembre-se que citações inadequadas podem levar a rejeições de trabalho e estudos bem como atribuição incorreta.

Os estilos mais tradicionais dividem-se em numéricos (`unsrt`, `plain`, `abnt-num`) e alfabéticos (`chicago`, `apalike`, `abnt-alf`). Como exemplo, considere a Ref. [1] utilizando o comando `\cite{arg}`.

ii. `\bibliography {arquivo.bib}`

O comando `bibliography` insere a bibliografia no local indicado e seu formato depende da classe do documento `.tex`. Em formatos `report` e `book`, cria-se um capítulo Referências. Já para `article` e derivados, as referências são, em geral, apenas listadas. O

argumento `.bib` do comando `bibliography` contém as informações relevantes das referências como nome dos autores, título, editora, ano, etc.

iii. Arquivos `.bib`

Num arquivo `.bib`, cada obra literária é especificada de acordo com o tipo (`\@book`, `\@article`, etc); pelo seu nome de referência, isto é, o argumento a ser utilizado pelo comando `cite`; e por informações complementares como autoria, data, título, etc. Por ser de fácil catalogação, entradas `.bib` são geralmente disponibilizadas em sites de periódicos ou através de ferramentas de internet como o Google Scholar.¹¹ Grosso modo, cada entrada possui um conjunto de identificadores (`author`, `title`, `year`, etc) aos quais são atribuídos valores, sempre entre chaves, ou seja,

```
identificador={ valor da chave }
```

Os identificadores são separados por vírgula. Assim, um livro, contido num arquivo `.bib`, possui o seguinte formato

```
@book{einstein1956,
  title={Investigations on the Theory of
        the Brownian Movement},
  author={Einstein, Albert},
  year={1956},
  publisher={Courier Corporation}
}
```

Já artigos seguem o padrão

```
@article{einstein1935,
  title={Can quantum-mechanical
        description of physical
        reality be considered
        complete?},
  author={Einstein, Albert and Podolsky,
        Boris and Rosen, Nathan},
  journal={Physical review},
  volume={47},
  number={10},
  pages={777},
  year={1935},
  publisher={APS}
}
```

O número de identificadores varia de entrada para entrada. Como regra geral, título, nomes dos autores, ano são obrigatórios. Artigos ainda requerem o número e volume da publicação. No caso dos livros, recomenda-se a editora e edição, quando possível.

¹¹No Google Scholar, basta procurar o nome da referência e clicar em Cite. No link gerado, selecione `bibtex`. Passos semelhantes estão disponíveis no Web of Science, PubMed e no Inspire. No caso de periódicos, recomenda-se o uso das ferramentas disponibilizadas pela própria revista.

Exercício

- 1) Adicionar um label para cada seção
- 2) Adicionar um label para cada figura
- 3) Adicionar um label para cada equação
- 4) Utilizar o comando `ref` e criar uma frase que faça referência ao menos três vezes.
- 5) Adicionar uma seção de referências
- 6) Citar os artigos (criar `arquivo.bib`):
 - i. [EPR](#)
 - ii, [Grafeno](#)

Resumo

```
\documentclass{article}
\bibliographystyle{unsrt}
\begin{document}
...
\bibliography{arquivo.bib}
\end{document}
```

1. Nakamura G, Mulato M, Martinez A (2016) Spin gap in coupled magnetic layers. *Physica A* 451:313 – 319.

Marie Curie