TóPEXS

Tópicos para o usuário TEX

João Luiz Kohl Moreira

Departamento de Astrofísica – DAF

Observatório Nacional

1. Introdução

TEX é um processador de texto matemático escrito sob a supervisão de Donald Knuth da Universidade de Stanford, Estados Unidos nas décadas de 60 e 70. O objetivo principal visava os grandes computadores. Uma vez pronto e divulgado, o TEX conheceu uma enorme popularidade inicialmente entre os matemáticos (Knuth, por sinal é um dos melhores) do mundo inteiro, e em seguida nos laboratórios de maneira geral. Após o advento dos sistemas de comunicação eletrônica universais esse processador se instalou de maneira definitiva devido à sua fácil portabilidade. As grandes revistas científicas estão, mais e mais, adotando o TEX como processador oficial, aceitando cooperações escritas nesse sistema passadas por BITNET. Esse procedimento garante a agilização das publicações. As versões que se seguiram sofreram um grau de enxugamento tal que foi possível desenvolver uma versão para o PC-XT, com $640Kby$ de memória.

A motivação desse guia é principalmente didática. Na medida do possível será dada uma explicação sumária dos diferentes conceitos do TEX: suas variáveis, comandos, parâmetros, etc. A exposição será iniciada por um resumo da instalação em PCs, já que para os grandes sistemas é melhor consultar o analista do local, que estará mais preparado para isso. O manual de instalação parece não estar atualizado, causando alguns problemas para os usuários que pretendo sanar aqui.

É feita uma breve discussão sobre a instalação do PCTEX. A leitura dessa seção também introduz a organização dos diretórios e como o TEX trabalha. Se você já tem o TEX instalado ou não vai trabalhar em um PC, pode passar por cima dessa seção.

Em seguida, serão introduzidos os diferentes modos de utilização em ordem crescente de dificuldade, de maneira a introduzir o iniciante absoluto, como também tentar sanar problemas que aparecem no dia a dia, e questões pouco usuais.

Em princípio as seções 3, 4, 5 e 6 ensinam como trabalhar com o TEX na sua forma simples. No entanto não descarte a leitura das seções restantes. Em particular a seção 11 foi escrita para os momentos mais difíceis, ou seja, as mensagens de erro.

Propostas de formatos e "macros" serão dados no sentido de estabelecer alguns critérios de construção de figuras, tabelas e equações de uso constante. Os "macros" são de importância enorme pois proporcionam um maior conforto para o usuário. A aparente dificuldade de montar uma tabela se restringe, dessa forma, à primeira vez, pois esses "macros" poderão ser aproveitados posteriormente para outros textos.

Finalmente, deve-se deixar claro que o TEX representa mais do que um simples processador de textos encontrados normalmente no mercado. Ao contrário dos ditos *wyswyg* (*what you see is what you get*), sistemas em que se vê na tela o que sairá na impressora, o TEX se parece mais a um compilador em que primeiro se escreve o texto, ou *programa*, para se compilar depois, como se faz com o FORTRAN, por exemplo. Essa é uma característica dos utilitários de grandes sistemas. Talvez seja essa uma das razões da universalidade do TEX.

2. Instalação

A instalação do TEX (PCTEX) para PC-DOS é razoavelmente simples, se bem que o manual não descreve os passos exatamente como devem ser. O pacote se apresenta em dois conjuntos de disquetes. O primeiro, composto de 4 contém os programas e formatos necessários para o TEX funcionar. Este conjunto chama-se PCTEX. O segundo, geralmente com 6 disquetes contém os "fonts" necessários para a construção dos caracteres. O TEX desenha tudo, de forma que nenhum caracter da impressora será aproveitado.

No primeiro disquete do PCTEX estão os programas necessários para a sua instalação. A experiência mostrou que antes de lançar a instalação é bom criar um diretório no *hard disk* chamado PCTEX:

```
$C:
ou
$CD C:\
$MD PCTEX
```

Em seguida passe ao driver A onde já está o disquete de instalação. Teoricamente a instalação se ocuparia de criar o diretorio PCTEX. Para isso existe nesse disquete um comando chamado IFDIR.COM que testa a existência desse diretório. Esse comando não funciona em todos os casos de forma que é preferível manter o INSTALL.BAT intacto e criar por você mesmo um diretório PCTEX do teclado.

Para iniciar a instalação dá-se o comando:

```
$A:INSTALL
```

Aparecerá na tela um certo número de informações. Caso apareça uma mensagem de erro, procure se certificar se o diretório PCTEX está corretamente instalado. É imperativo que PCTEX seja um diretório da árvore principal. Reinicie o processo de instalação. Caso o erro persista, provavelmente o seu disco fixo ou o seu *driver* A: está com problemas.

Prosseguindo a instalação, o programa vai pedir que se troque os disquetes em sequência. Ao final dos 4 disquetes instalados o *prompt* DOS será colocado na tela esperando novas instruções. Será a hora de instalar os 6 disquetes *fonts* restantes.

Lance o comando:

```
$A:INSTALCM C
```

e siga as instruções de troca dos disquetes.

Após a instalação, o sistema ainda não está pronto para ser utilizado. Ele deve ser inicializado. O manual refere-se ao INITEX, utilitário para inicializar o pacote. Parece que as versões mais recentes, este INITEX está embutido no próprio TEX. Assim para inicializar, deve-se estar no diretório PCTEX recém criado e lançar o comando:

```
$tex /i
```

a opção /i indica que um novo sistema deve ser inicializado. O programa então dá um *prompt* após algumas informações:

```
**
```

Deve-se dar o nome do formato *default* do novo TEX. Pode-se escolher entre duas opções. Para aqueles que vão se servir do TEX propriamente dito, usa-se o formato *plain*. Para aqueles que vão se servir do LATEX , usa-se *lplain*. Um formato, seja o *plain* seja o *lplain* contém essencialmente alguns macros que definem o estilo dos textos, pois o TEX em si é *cru* para esses propósitos. Convida-se, inclusive os mais experimentados a criarem seus próprios formatos de acordo com as suas preferências. Por enquanto, os formatos existentes servem para os nossos propósitos e assim não vamos re-inventar a roda. Digamos que escolhemos o formato *plain*:

```
**plain [ENTER]
```

em seguida o programa dará um sem número de informações sobretudo as definições contidas em PLAIN.TEX que está no diretório C:\PCTEX\TEXINPUT. Ao final da operação ter-se-á o *prompt*:

```
*
```

onde se entrará com o comando:

```
* \dump
```

esse comando fará guardar o novo formato no diretório apropriado de forma que a cada vez que se lançar o TEX, esse formato será usado. Um arquivo chamado PLAIN.LOG no diretório PCTEX conterá todas as informações a respeito dessa inicialização.

O formato *plain* está exaustivamente descrito no livro TEXbook de D. Knuth. Pode-se fazer praticamente tudo que é necessário a um texto científico. No entanto têm-se a impressão de que um certo trabalho é necessário. Já o formato *lplain* segue as instruções do manual LATEX de Leslie Lamport. Nos grandes computadores tais como o Vax esse manual está disponível escrito em LATEX no diretório de documentação TEX. Vale a pena procurá-lo, copiá-lo e tirar uma cópia. Geralmente essa documentação está disponível ao usuário, isto é, sem proteção. Não há impedimento algum para esse procedimento, a não ser questões econômicas que devem ser resolvidas com os responsáveis locais, pois o manual usa cerca de 140 páginas. Já não é o caso do TEXbook. Esse livro têm distribuição exclusiva da Addison Wesley Publishing Company que detém os direitos de cópia. No entanto uma cópia do manuscrito em TEX pode ser encontrado no diretório pertinente. O arquivo está protegido e nem o analista de sistema local têm direito a mudar a proteção. Não tente. O resultado pode ser catastrófico. De qualquer forma nenhum desses manuais está disponível em forma de arquivo nas implementações de PCs.

Em princípio, basta dar o comando:

`$tex` [nome do arquivo em TEX]

Para quem escolheu o *lplain* eu recomendo criar uma arquivo chamado LATEX.BAT contendo o seguinte:

```
$tex &lplain %1
```

que é a forma que seria necessária de lançar o LATEX. Caso se fizer simplesmente:

```
$tex [nome do arquivo]
```

teria-se como resposta uma mensagem de erro algo como:

```
*** format plain not found
```

Uma vez criado o arquivo LATEX.BAT poderá se lançar:

`$latex` [nome do arquivo em LATEX]

Uma vez instalado o aspecto do diretório PCTEX, deve ter entre outras coisas algo parecido com:

```
 Directory of  C:\PCTEX

.            <DIR>      1-01-80  12:28a
..           <DIR>      1-01-80  12:28a
PIXEL        <DIR>      1-01-80  12:34a
PLAIN    LOG     2495   1-01-80  12:45a
TEX      EXE   225766  12-15-87   9:34a
BIBTEX   EXE    74321  12-15-87   1:14p
PCTEX    EXE     4096   8-06-87   3:22p
TEXFMTS      <DIR>      1-01-80  12:29a
TEXBIB       <DIR>      1-01-80  12:29a
TEXDOC       <DIR>      1-01-80  12:29a
TEXINPUT     <DIR>      1-01-80  12:29a
TEXTFMS      <DIR>      1-01-80  12:32a
```

O diretório C:\PCTEX\PIXEL deverá conter:

```
 Directory of  C:\PCTEX\PIXEL

.            <DIR>      1-01-80  12:34a
..           <DIR>      1-01-80  12:34a
DPI240       <DIR>      1-01-80  12:34a
DPI263       <DIR>      1-01-80  12:34a
DPI288       <DIR>      1-01-80  12:34a
DPI346       <DIR>      1-01-80  12:34a
DPI415       <DIR>      1-01-80  12:34a
DPI498       <DIR>      1-01-80  12:34a
DPI597       <DIR>      1-01-80  12:34a
```

Cada diretório "DPI" contém os "fonts" dos diferentes tamanhos dos caracteres. O primeiro corresponderá ao tamanho "\magstep0", o segundo a "\magstephalf", e do terceiro ao sexto os tamanhos 2 a 5, respectivamente. Se você der uma olhada no interior destes diretórios notará que os nomes dos arquivos tais como "cmit10.pk", etc, serão os mesmos em todos os diretorios. Cada arquivo corresponde a um "font" de diferentes formatos, ex. romano, itálico, matemático, etc. Cada diretório contém um tamanho diferente. Em outras palavras, o tamanho dos caracteres são definidos pelos nomes dos diretórios. Por isso, seria catastrófico copiar um arquivo de um diretório em outro. Caso houver perda de arquivo em um diretório desses, o melhor a fazer é re-instalar os "fonts" (segundo conjunto de disquetes).

Além disso no diretório C:\PCTEX\TEXFMTS deverá existir o arquivo PLAIN.FMT, que é o formato que foi instalado com o comando $TEX /i (ou LPLAIN.FMT, caso se instalou o LaTeX).

Outros diretórios contém arquivos diversos. Vale a pena, por exemplo, quando se estiver mais familiarizado com o TeX, dar uma olhada nos arquivos de \TEXINPUTS. Ali estão, por exemplo, os macros de que se servem PLAIN e LPLAIN para instalar seus formatos.

Até aqui, não se falou dos utilitários que vão fazer sair os textos já tratados numa impressora ou na tela, para visualização. Existem fabricantes de tais produtos, tal como a ARBORTXT que fornecem *drivers* para impressoras LASER (HP, PostScript, etc). Assim como o PTIVIEW ou MAXVIEW que se servem para mostrar na tela para um pré-exame, antes de imprimir. Existe também um utilitário para impressoras Epson chamado PCDOT. A instalação de cada um desses produtos deve ser feita segundo o manual de fabricante. Um comentário acerca de MAXVIEW é necessário. Ao instalar-se no diretório PCTEX, ele cria um arquivo VIEW.BAT que é o que deve ser utilizado para a sua execução. Deve-se modificar esse arquivo de forma que ele se torne:

```
DVI2SCR -A1 -R240 -"DPI -Q1000 -X0 -%2 -%3 -%4 -%5 %1
```

Note que o comando `-"DPI` foi introduzido. Esse parâmetro indica ao DVI2SCR (comando MAXVIEW para a tela) que deve-se procurar os "fonts" nos diretórios "DPI" que é onde eles estão, como vimos acima. Caso contrário, os "fonts" serão procurados em diretórios PXL, que nem existem.

3. Utilização

O primeiro aspecto da utilização diz respeito a algo que não faz parte do TeX: o editor de textos. Ao contrário dos pacotes comerciais, o TeX não possui um editor. Essa escolha depende do gosto de cada um. Qualquer editor que se presta a tratar textos para compiladores como o FORTRAN, PASCAL, etc, pode ser aproveitado. Exemplos: EDLIN, EDT, PMATE, TURBO BASIC, TURBO PASCAL, TURBO C, Word Star, Word Perfect, Word, dBase, Chi Writer e outros tantos. A única condição é que não sejam inseridos caracteres especiais. Exemplo, o Word Star, introduz automaticamente caracteres que ele aproveita para controlar tamanho de papel, margeamento, espaçamento, etc. A solução aqui seria editar um texto como *NON DOCUMENT*. O mesmo acontece com Wort Perfect, Word e Chi Writer. Todos eles dão opção para edição de textos "programas", para ser usado por um compilador. Essa é a opção para o TeX também.

Antes de qualquer coisa é preciso colocar o diretório PCTEX no *path* de seu PC. Edite o arquivo AUTOEXEC.BAT que deve estar no diretório "raiz" de seu disco e introduza no *path*:

```
.
.
$path=c:\pctex;....
.
.
```

onde os pontos indicam, se for o caso, outros comandos do AUTOEXEC.BAT.

É hora de uma primeira tentativa. Instale-se num diretório reservado para textos e usando o editor de sua preferência escreva o texto:

```
Deus nas alturas, estou aqui procurando um texto sem acentos, pois por
enquanto os meus leitores carecem do saber fazer isso. Agudos, tils
e cedilhas ainda ausentam, mas acredito, abundam no futuro.
\bye
```

Salve o arquivo com o nome de SACENTOS.TEX e saia do editor. Como pode-se notar, os acentos foram evitados às custas do estilo. Brevemente iremos voltar a essa questão. No DOS dê o comando:

```
$tex sacentos
```

Note que a extensão .TEX foi suprimida. Os arquivos TEX devem possuir essa que extensão e é assumida por *default* quando o arquivo é processado. É a mesma lógica de xxx.FOR para o compilador FORTRAN. Em resposta a esse comando a tela deverá apresentar as linhas seguintes:

```
This is TeX, Version 2.1 (preloaded format=plain 80.1.1)  1 JAN 1980 00:08
  (PCTeX 2.10, (c)Personal TeX, Inc 1987. S/N 10511)
**sacentos
(C:\TEXTS\SACENTOS.TEX [1]
Output written on C:\TEXTS\SACENTOS.DVI (1 page, 460 bytes).
```

Para verificar na tela o resultado use PTIVIEW ou MAXVIEW. Por exemplo:

```
$VIEW SACENTOS
```

Para imprimir numa impressora tipo Epson use:

```
$DVIEPS SACENTOS
```

Esse procedimento de processamento, visualização e impressão pode ser muito simplificado através do uso do utilitário PCTEX.EXE. Vá até o diretório C:\PCTEX\TEXFMTS e edite o arquivo PCTEX.CFG. Esse arquivo deve ter o seguinte conteúdo:

```
%E=epsilon %s
%C=tex %s
%V=view %s
%P=pcdot %s
%T=pclaser %s
```

Mude os parâmetros de acordo com sua conveniência. O parâmetro `E` define o editor, `C` deve permanecer como `tex`, e assim por diante. Em seguida, volte ao diretório de textos e faça o comando:

```
$PCTEX SACENTOS
```

Vai aparecer no canto superior direito da tela um quadro com um menu onde se desloca a vontade com as teclas de cursor. Assim vai-se editar, processar, visualizar apenas deslocando o cursor pelo menu.

Antes de continuar a discussão é preciso ficar claro que o TEX considera os espaços e um [Return] apenas como sinal de separação entre as palavras. Parágrafos são construídos a partir do comando `\par` ou 2 [Returns] consecutivos. Por isso é inútil colocar [Tab]s e 20 espaços para forçar uma separação maior entre as palavras pois nada disso será considerado. Da mesma forma é inútil colocar 20 [Return]s entre as linhas para forçar uma separação maior entre dois parágrafos. Existem formas de controlar isso que serão discutidas mais tarde. Os incrédulos podem tentar...

Avancemos um pouco no contrôle do *lay out* de nosso texto. Se você tirou uma impressão do texto vai perceber que a letra é um tanto pequena. Isso porque o TEX assume que o tamanho do "font" é `\magstep 0`, melhor dizendo que a dimensão adotada é de 10 pt. Para o TEX 1 pt vale 1/72.27 de polegada. Essa é a menor unidade para TEX. Os valores dos `\magsteps` , subsequentes seguem uma relação mais ou menos complicada dependente de $\sqrt{2}$. Para aumentar o tamanho das letras serão necessárias algumas providências:

1. Usar o comando interno TEX: `\magnification` ;
2. O TEX multiplica todos os valores *default* pelo valor adotado em *magnification*. Como consequência, o tamanho do papel, margem direita, margem esquerda, espaçamento etc, é multiplicado pelo mesmo valor. Como o papel sempre tem o mesmo tamanho, deve-se impor valores para outros parâmetros;
3. As unidades do TEX (in, cm, etc) também são modificadas, de maneira que é necessário impor unidades absolutas.

Edite o arquivo SACENTOS.TEX e insira no início deste:

```
\magnification=\magstephalf
\hsize=6.8 true in
\vsize=9.0 true in
\hoffset=0.0 true cm
\voffset=0.0 true cm
```

O significado de cada linha pode ser deduzido, no entanto vamos comentá-las afim de confirmar as deduções:

`\magnification =\magstephalf`	– ampliação de 1 `\magstephalf`
`\hsize =6.8 true in`	– dimensão horizontal da página: 6.8 in absolutos
`\vsize =9.0 true in`	– dimensão vertical da página: 9.0 in absolutos
`\hoffset =0.0 true cm`	– margem esquerda: 0 cm absolutos
`\voffset =0.0 true cm`	– margem superior: 0 cm absolutos

Com relação aos *offsets*, isto é, as margens, é preciso ter em mente que os utilitários de impressão tanto na tela quanto na impressora iniciam um texto a 1 polegada à esquerda e 1 polegada abaixo do início do papel. Por isso a adoção dos valores 0. Para fazer imprimir mais acima e à esquerda é necessário adotar valores negativos.

Uma vez feita a modificação faça imprimir o resultado por TEX. Em seguida modifique o valor de `\magnification` até `\magstep 5` e veja os resultados. Alerta-se para o fato que a partir de `\magstep 3` podem aparecer mensagens de alerta no processamento de TEX dando conta de `overfull box ...`. Isso se deve a fatores que discutiremos mais tarde.

Outros tantos valores podem ser modificados. Alguns principais podem ser:

`\baselineskip =12pt`	– espaçamento de 12pt
`\parskip =6pt`	– espaçamento entre parágrafos 6 pt a mais
`\parindent =12pt`	– valor do parágrafo 12pt

A título de treinamento, modifique esses valores, inclusive inserindo-os entre os parágrafos de seu texto e veja o resultado.

O sinal % representa um comentário. Significa que tudo o que vier naquela linha depois de '%' não será considerado pelo TEX podendo ser usado para se escrever um comentário ou um lembrete. Se você escrever o texto abaixo no seu arquivo ele não será tratado pelo TEX:

```
% Essa linha sera' esquecida pelo TeX.
```

4. Caracteres especiais e acentos

Os seguintes caracteres têm significação especial para o TEX, isto é, são entendidos como **comandos** e não como letras:

\	backslash	caracter "escape"
{	leftbrace	abre um grupo ou bloco
}	rightbrace	fecha um grupo ou bloco
$	dollar	modo matemático
&	ampersand	tabulação
#	flat	parâmetro
^	circunflex	circunflexo (modo texto) ou potência (modo matemático)
_	underscore	índice (modo matemático)
%	percent	comentário
~	tilde	espaço forçado. til (modo texto)

TEX foi escrito para a língua inglesa, por isso as letras acentuadas exigem um pouco mais de trabalho, se bem que tudo está previsto. Senão, como seria possível escrever a palavra francesa "cœur"? Aqui vai uma lista dos acentos possíveis com **qualquer letra**. Usa-se o o apenas para exemplificar:

`\‘o`	ò
`\’o`	ó
`\^o`	ô
`\"o`	ö
`\~o`	õ
`\=o`	ō
`\.o`	ȯ
`\u o`	ŏ
`\v o`	ǒ
`\H o`	ő
`\t oo`	o͡o

Além disso temos os "acentos" sob as letras:

`\c o`	o̧
`\d o`	ọ
`\b o`	o̲

Do que se deduz como fazer o 'c cedilha'. Deixo ao leitor a tarefa de descobrir como fazê-lo. Acima mostramos como fazer o 'o cedilha'.

E as letras especiais:

`\oe , \OE`	œ, Œ
`\ae , \AE`	æ, Æ
`\aa , \AA`	å, Å
`\o , \O`	ø, Ø
`\l , \L`	ł, Ł
`\ss`	ß

Ainda existem símbolos do tipo:

`\dag`	†
`\ddag`	‡
`\S`	§
`\P`	¶

Alguns comentários são necessários. Os acentos sobre a letra **i** feitos de maneira direta não são escritos corretamente. Por exemplo, ao fazermos `\'i` o resultado será í. É preciso fazer de tal forma que o pingo do i não apareça. A resposta é `\i` . Se fizermos \i o resultado será ı. Assim, um acento agudo em i pode ser feito da seguinte forma: `\'\i` . Outra questão diz respeito ao reconhecimento de um "comando" pelo TEX. A dita sequência "escape". "Escape" é representado no TEX pelo caracter '\'. O que segue é comando. Mas como o TEX saberá que esse comando terminou e que o que vem a seguir é texto novamente? A forma mais evidente é inserir a "sequência" dentro de um grupo, ou bloco. Vimos acima que os caracteres '{' e '}' são respectivamente sinais de abertura e fechamento de um grupo ou bloco. Um bloco, ou grupo, é caracterizado pelas definições que se fazem dentro e que não têm validade fora dele. Um grupo do tipo `{ \i }` pode definir o comando do tipo i-sem-pingo, mas no decorrer da datilografia ficaria por demais estafante escrever `{ \i }` a cada vez que fosse necessário um i-sem-pingo. Como solução aproveitamos as seguintes regras: 1) os comandos com caracteres alfabéticos terminam com um espaço; 2) os comandos com caracteres especiais só possuem um parâmetro, qual seja, o próprio caracter. Exemplo: \ae é reconhecido como tal porque o 'ae' terminou com um espaço; já \' pode ser seguido da letra a ser acentuada porque " ' " é um caracter especial. O comando só tem um parâmetro. Por isso, no 'oo' ligado, obtido por \t oo, o 't' deve imperativamente estar separado de 'oo'. Note que para fins do resultado final o espaço entre os dois foi ignorado. Deve-se tomar cuidado, pois, quando terminar uma palavra do tipo 'groß'. Se fizermos 'gro\ss papa', o resultado será 'großpapa' e se quizermos um 'groß papa' será necessário fazer '`gro\ss ~papa`'. Por outro lado, se fizermos 'gro\sspapa', o TEX nos responderá com uma mensagem de erro dizendo que não reconhece a sequência : \sspapa.

Comentários podem ser inseridos no texto sem que estes sejam impressos. O sinal '%' indica que o que vêm até o final da linha é comentário.

É comum também se desejar que % seja impresso e não interpretado como sinal de comentário, ou que, por exemplo, que o valor de um dado instrumento é US$ 5000.00. Na maioria dos casos basta fazer '\%' para fazer sair '%', '\$' para sair '$' e assim com os outros caracteres especiais do TEX. A não ser os sinais '{' e '}' que exigem um pouco mais de trabalho. Esses sinais podem ser facilmente produzidos no modo matemático que viremos a seguir. Em modo texto é um pouco mais complicado. Para ser sincero, eu precisei usar esses sinais no texto somente agora que estou escrevendo esse manuscrito.

Finalmente, quero dizer que achei um processador de texto que possa escrever, sem intervenções, o nome de um dos meus compositores preferidos: *Dvořák*

5. Letras, fontes e hífens

TEX não se restringe a um só tipo de letra durante o processamento do texto. Pode-se escrever em *itálico*, **negrito** ou *tombado*. Em modo matemático, que veremos a seguir, as letras têm um *itálico matemático* ou ainda $\mathcal{CALIGRAFICO}$. Esse último não admite letras minúsculas e acentos.

Para gerar o parágrafo acima o texto foi escrito da seguinte forma:

```
\TeX  n\~ao se restringe a um s\'o tipo de letra durante o processamento
do texto. Pode-se escrever em {\it it\'alico}, {\bf negrito} ou {\sl
tombado}. Em modo matem\'atico, que veremos a seguir, as letras t\^em um
$it\acute alico$ $matem\acute atico$ ou ainda $\cal CALIGRAFICO$.
Esse \'ultimo n\~ao admite letras min\'usculas e acentos.
```

Pode-se variar o tamanho das letras também. Para isso é preciso definir antes de bater o texto, os 'fonts' que se deseja utilizar. A seguir vem uma lista de 'fonts' que eu 'carreguei' no início do arquivo que gerou esse texto:

```
\font\tiny=cmr5 at 3pt
\font\littletenrm=cmr10 scaled\magstep1
\font\midtenrm=cmr10 scaled\magstep2
\font\bigtenrm=cmr10 scaled\magstep4
\font\huge=cminch
```

A interpretação desses comandos é mais ou menos a seguinte: além dos 'fonts' já existentes carregue **cmr5** com tamanho de 3pt e dê o nome "tiny", **cmr10** a um aumento de **\magstep1** (\magstep1 é uma constante pré-definida em PLAIN) e dê o nome de **\littletenrm**, a um

aumento de **magstep2** e dê o nome de **midtenrm** e assim por diante. Ao final defini o 'font' chamado \huge que é o **cminch** guardado no tamanho do 'font' corrente (lembre-se dos diretórios 'DPI'). Esses nomes poderiam ser diferentes pois o próprio usuário pode escolher. Por exemplo, em vez de \huge eu poderia ter escolhido \kingkohl, da seguinte forma:

```
\font\kingkohl=cminch
```

A definição do aumento pode ser feita de outra maneira:

```
\font\midmidrm=cmr10 at 18pt
```

O que definirá não um fator de aumento mas uma dimensão fixada a 18pt. É o caso de nosso 'font' \tiny, que produzirá o texto:

Essa é só para os míopes

Como consequência mudaremos o aspecto do nosso texto de acordo com o 'font' escolhido. Assim se batermos o seguinte texto:

```
Os patinhos da lagoa foram {\littletenrm crescendo},
{\midtenrm crescendo},
{\bigtenrm crescendo},
at\'e se tornarem enormes

\centerline{{\huge 22}}
```

O resultado será:

Os patinhos da lagoa foram crescendo, crescendo, crescendo, até se tornarem enormes

22

Aqui pode-se entender o papel dos blocos, definidos por tudo o que está entre '{' e '}'. Ao darmos o comando \it ou mesmo \huge, TEX instala o 'font' correspondente e assim ficará até que façamos instalar um outro. Se fizermos a seguir \it *tudo o que batermos estará em itálico mesmo que esse não seja o nosso desejo. As coisas poderão se modificar quando instalarmos outro 'font', por exemplo,* **bf, o que fará que todo o texto a seguir fique em negrito. Essas tentativas podem se interromper se fizermos re-instalar o 'font' original, no caso** \rm. Para não correrm os o perigo de ficarmos tentando recuperar o 'font' original é

mais seguro inserirmos o texto, inclusive o comando, dentro de um bloco. A modificação do 'font' ficará restrito ao texto no seu interior. Resumindo: quando quiser que *esse texto fique em itálico* faça: `{\it esse texto fique em it\'alico}` e não `\it {esse texto fique em it\'alico}`. A segunda forma está errada e vai fazer que todo o seu texto a seguir fique em itálico. A definição do bloco de nada adianta nesse caso.

Quanto à separação das palavras, TEX trabalha com um conjunto de variáveis entre elas o `\badness` e o `\tolerance` que controlam a "beleza" do texto em uma linha. Ele aperta e estica o quanto pode a separação entre as palavras de maneira que o texto se estabeleça sem que seja necessário separar as palavras. Quando isso não é possível, ele toma a decisão e separa as palavras segundo as regras da língua inglesa, ou seja, pela divisão semântica. Pelo que eu pude entender as regras de separação são bem definidas com apenas 14 exceções. No que concerne o português, enquanto a separação se dá entre duas consoantes, parece que as coisas funcionam bem. Há casos que aparecem erros evidentes. Por isso, não é incomum encontrarmos "sel-eção" no texto tratado pelo TEX. Uma maneira de consertarmos isso é impormos a regra de separação para o TEX, seja por introduzir mais exceções, no início do arquivo antes do texto, com o comando \hyphenation{se\-le\-\c c\ ao}, seja impondo a regra localmente apenas na palavra "seleção" que deu o problema, impondo ali no texto: se\-le\-ção. Eu prefiro essa última forma por uma razão simples, as exceções em TEX são guardadas no *buffer* de trabalho e encher o *buffer* com coisas não imprescindíveis pode levar o TEX a um *overflow* quando responde matreiramente:

```
TeX capacity exceeded, sorry.
```

que não seria problemático se fôsse essa a única causa possível para um *overflow*.

6. MaTEXmáticos

Como é um processador de textos para matemáticos, o TEX pode, sem dificuldade produzir a fórmula:

$$\int f(x)\,dx = F(x) + C \longrightarrow F'(x) = f(x)$$

Para produzir o parágrafo acima, eu bati o seguinte texto:

```
Como \'e um processador de textos para matem\'aticos, o \tex
pode, sem dificuldade produzir a f\'ormula: $$\int f(x)\;dx
= F(x) + C \longrightarrow F'(x) = f(x)$$
```

O TEX usa um processo mnemônico para produzir textos, em particular, matemáticos. O sinal '$' indica que o texto a seguir deverá ser interpretado como um texto matemático. Evidentemente se se quizer indicar o fim do modo matemático recolocamos o sinal '$'. A

maioria dos 'comandos' em modo matemático é exclusivo, não podendo ser reproduzido em modo texto. Por isso existem duas maneiras de inserir um texto matemático em um parágrafo. O primeiro, que eu acabei de usar, é iniciado com '$$' que indica que a fórmula matemática deve ficar destacado do texto, como se fosse um parágrafo individual. O segundo, inicia-se apenas com um '$', onde a fórmula será inserida no texto normalmente como se fosse uma palavra. Assim, a frase:

"Onde $\longrightarrow$ significa 'implica em', fazendo de $F(x)$ a função primitiva de $f(x)$. $\int f(x)\;dx$ é uma integral indefinida." Foi produzida pelo texto:

```
``Onde $\longrightarrow$ significa 'implica em', fazendo
de $F(x)$ a fun\c c\~ao primitiva de $f(x)$. $\int f(x)\;dx$ \'e
uma integral indefinida.''
```

Eis alguns exemplos cujos efeitos são interessantes:

`$x^2$`	x^2	`$x_2$`	x_2
`$2^x$`	2^x	`$x^2y^2$`	x^2y^2
`$x ^ 2y ^ 2$`	x^2y^2	`$x_2y_2$`	x_2y_2
`$_2F_3$`	${}_2F_3$	`$x^{2y}$`	x^{2y}
`$2^{2^x}$`	2^{2^x}	`$2^{2^{2^x}}$`	$2^{2^{2^x}}$
`$y_{x_2}$`	y_{x_2}	`$y_{x^2}$`	y_{x^2}
`$((x^2)^3)^4$`	$((x^2)^3)^4$	`${{{(x^2)}^3)}^4$`	$\left((x^2)^3\right)^4$
`$x^2_3$`	x_3^2	`$x_3^2$`	x_3^2
`$x^{expo}_{ind}+\pi $`	$x_{ind}^{expo}+\pi$		
`$x_{y^a_b}^{c^d}$`	$x_{y_b^a}^{c^d}$		

Operadores matemáticos são tratados pelo TEX, alguns de maneira especial. Já tivemos a oportunidade de saber como se cria o sinal '$\int$'. Através da sintaxe de índices e expoentes podemos escrever uma integral definida:

$$\int_0^1 x dx = 0.5$$

Existem ainda:

`\sum`	$\sum$	`\bigcap`	$\bigcap$	`\bigodot`	$\bigodot$
`\prod`	$\prod$	`\bigcup`	$\bigcup$	`\bigotimes`	$\bigotimes$
`\coprod`	$\coprod$	`\bigsqcup`	$\bigsqcup$	`\bigoplus`	$\bigoplus$
`\int`	$\int$	`\bigvee`	$\bigvee$	`\biguplus`	$\biguplus$
`\oint`	$\oint$	`\bigwedge`	$\bigwedge$		

Alguns operadores são especiais pela apresentação dos símbolos que devem resultar. Assim `$\sqrt 2$` resultará em: $\sqrt{2}$. Como pode ser deduzido da sintaxe das operações de índices

e expoentes, o operador `\sqrt` também segue a mesma regra. A expressão $\sqrt{x+2}$ é obtida através de `$\sqrt {x+2}$`. Note que $x+2$ está dentro de um "bloco". Da mesma forma temos:

`$\underline 4$`	$\underline{4}$
`$\overline {x+y}$`	$\overline{x+y}$
`$x^{\underline n}$`	$x^{\underline{n}}$
`$x^{\overline {m+n}}$`	$x^{\overline{m+n}}$
`$\sqrt {x^3+\sqrt \alpha }$`	$\sqrt{x^3+\sqrt{\alpha}}$
`$\root n \of {x^n+y^n}$`	$\sqrt[n]{x^n+y^n}$

As letras gregas, estão todas definidas no TEX:

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

Algumas letras são equivalentes no alfabeto mas têm aspecto caligráfico diferente. Esses casos são apresentados pelo prefixo '*var*'. As letras maiúsculas podem ser obtidas pelas maiúsculas do alfabeto latino. É o caso de 'A' que é entendido como maiúsculo de 'α'. A não ser os seguintes casos

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

Como vimos anteriormente letras caligráficas — apenas maiúsculas — são obtidas no modo matemático. Assim $\mathcal{G}$ é obtido de `$\cal G$`. Outros símbolos também são pré-definidos:

`\aleph`	$\aleph$	`\prime`	$\prime$	`\forall`	$\forall$
`\hbar`	$\hbar$	`\emptyset`	$\emptyset$	`\exists`	$\exists$
`\imath`	$\imath$	`\nabla`	∇	`\neg`	$\neg$
`\jmath`	$\jmath$	`\surd`	$\surd$	`\flat`	$\flat$
`\ell`	ℓ	`\top`	$\top$	`\natural`	$\natural$
`\wp`	$\wp$	`\bot`	$\bot$	`\sharp`	$\sharp$

\Re	$\Re$	|	$\|$	\clubsuit	$\clubsuit$
\Im	$\Im$	\angle	$\angle$	\diamondsuit	$\diamondsuit$
\partial	∂	\triangle	$\triangle$	\heartsuit	$\heartsuit$
\infty	∞	\backslash	$\backslash$	\spadesuit	$\spadesuit$

As chamadas operações binárias estão presentes:

\pm	$\pm$	\cap	$\cap$	\vee	$\vee$
\mp	$\mp$	\cup	$\cup$	\wedge	$\wedge$
\setminus	$\setminus$	\uplus	$\uplus$	\oplus	$\oplus$
\cdot	$\cdot$	\sqcap	$\sqcap$	\ominus	$\ominus$
\times	$\times$	\sqcup	$\sqcup$	\otimes	$\otimes$
\triangleleft		$\triangleleft$			
\oslash	$\oslash$	\star	$\star$	\odot	$\odot$
\triangleright		$\triangleright$		\diamond	$\diamond$
\wr	$\wr$	\dagger	$\dagger$	\circ	$\circ$
\bigcirc	$\bigcirc$	\ddagger	$\ddagger$	\bullet	$\bullet$
\bigtriangleup		$\bigtriangleup$		\amalg	$\amalg$
\ast	$\ast$	\div	$\div$		
\bigtriangledown		$\bigtriangledown$			

O sinal \setminus não é nada além do \backslash com diferente *glue*, variável que o TEX usa para separar as palavras e sinais entre si. Temos ainda os sinais relacionais. Os sinais $>, <$ e $=$ são facilmente obtidos nos teclados modernos. Além desses, temos:

\leq	$\leq$	\geq	$\geq$	\equiv	$\equiv$
\prec	$\prec$	\succ	$\succ$	\sim	$\sim$
\preceq	$\preceq$	\succeq	$\succeq$	\simeq	$\simeq$
\ll	$\ll$	\gg	$\gg$	\asymp	$\asymp$
\subset	$\subset$	\supset	$\supset$	\approx	$\approx$
\subseteq	$\subseteq$	\supseteq	$\supseteq$	\cong	$\cong$
\sqsubseteq	$\sqsubseteq$	\sqsupseteq	$\sqsupseteq$	\bowtie	$\bowtie$
\in	$\in$	\ni	$\ni$	\propto	$\propto$
\vdash	$\vdash$	\dashv	$\dashv$	\models	$\models$
\smile	$\smile$	\mid	$\mid$	\doteq	$\doteq$
\frown	$\frown$	\parallel	$\parallel$	\perp	$\perp$

Todos esses sinais relacionais aceitam um \not à sua frente de maneira a produzir sua negação. Exemplo: \not \leq irá produzir $\not\leq$. Da mesma forma os outros sinais. Faltam ainda os *arrows* ou "vetores", de especial importância em textos matemáticos:

\leftarrow $\leftarrow$ \longleftarrow $\longleftarrow$

`\Leftarrow`	$\Leftarrow$	`\Longleftarrow`	$\Longleftarrow$
`\rightarrow`	$\rightarrow$	`\Longleftarrow`	$\Longleftarrow$
`\Rightarrow`	$\Rightarrow$	`\Longrightarrow`	$\Longrightarrow$
`\leftrightarrow`	$\leftrightarrow$	`\longleftrightarrow`	$\longleftrightarrow$
`\Leftrightarrow`	$\Leftrightarrow$	`\Longleftrightarrow`	$\Longleftrightarrow$
`\mapsto`	$\mapsto$	`\longmapsto`	$\longmapsto$
`\hookleftarrow`	$\hookleftarrow$	`\hookrightarrow`	$\hookrightarrow$
`\leftharpoonup`	$\leftharpoonup$	`\rightharpoonup`	$\rightharpoonup$
`\leftharpoondown`	$\leftharpoondown$	`\rightharpoondown`	$\rightharpoondown$
`\rightleftharpoons`	$\rightleftharpoons$	`\uparrow`	$\uparrow$
`\Uparrow`	$\Uparrow$	`\downarrow`	$\downarrow$
`\Downarrow`	$\Downarrow$	`\updownarrow`	$\updownarrow$
`\Updownarrow`	$\Updownarrow$	`\nearrow`	$\nearrow$
`\searrow`	$\searrow$	`\swarrow`	$\swarrow$
`\nwarrow`	$\nwarrow$		

O símbolo $\overrightarrow{v}$ pode ser obtido de `$\buildrel \rightarrow \over v$`. Outros símbolos podem ser construidos dessa forma. Pelo que se pode notar o comando `\buildrel` "constroi" símbolos pela combinação de vários pré-definidos. Existem duas situações em que o comando `\over` pode ser utilizado. A primeira, que acabamos de ver, serve para auxiliar a construção de símbolos novos a partir de outros. A segunda permite construir frações. Por exemplo, a relação

$$l = \frac{1}{4\pi}$$

foi obtida da expressão: `$$ l = {1 \over 4\pi }$$`.

É preciso saber usar o `\over` nessas condições. Esse comando atua sobre tudo o que vêm antes e tudo o que vem depois. É preciso delimitar aquilo que ficará dentro da fração através de '`{ }`'. Caso contrário a expressão acima ficaria:

$$\frac{l = 1}{4\pi}$$

se fizéssemos simplesmente, `$$l = 1 \over 4\pi $$`.

Exemplos de símbolos obtidos por `\buildrel` :

`\buildrel \alpha \beta \over \longrightarrow`	$\buildrel \alpha \beta \over \longrightarrow$
`\buildrel \rm def \over =`	$\buildrel \rm def \over =$
`R^2 {\buildrel f(R) \over \longrightarrow } R^2`	$R^2 \buildrel f(R) \over \longrightarrow R^2$

A propósito, o aspecto da última expressão pode ser melhorado. Por exemplo:

$$R^2 \quad \buildrel f(R) \over \longrightarrow \quad R^2$$

que é obtido de

```
$$R^2 \quad{\buildrel f(R) \over \longrightarrow}\quad R^2$$
```

Existem outras maneiras de compor símbolos. Os físicos e astrofísicos relativistas gostam do símbolo $\Box$. Veja como foi composto esse símbolo inspirado na definição de `\llap` , pag. 353 do TEXbook: `$\sqcup $\hbox to 0pt{\hss $\sqcap $}`. Aqui os símbolos $\sqcup$ (`\sqcup`) e $\sqcap$ (`\sqcap`) foram colocados um em cima do outro de maneira a produzir o resultado desejado. É claro que essas definições podem se complicar enormemente. Por exemplo, já não é tão evidente produzir o símbolo '$\lesssim$' ou '$\gtrsim$', muitas vêzes necessário sobretudo entre os experimentais, mas sem muito significado entre os matemáticos. Contudo, estudaremos outras propriedades do TEX e outras definições, e com o conhecimento dos chamados "macros" faremos definições prévias desses símbolos e poderemos evocá-los no texto como se fosse um símbolo primitivo. Símbolos primitivos são aqueles pré-definidos no TEX.

Antes, vamos analisar algumas questões relativas ao *lay out* das fórmulas matemáticas. Se fizermos:

```
$$ {\bf F} = -{e^2 \over 4\pi\epsilon_0}\nabla({1-u^2/c^2 \over
s})$$
```

o resultado será:

$$\mathbf{F} = -\frac{e^2}{4\pi\epsilon_0}\nabla(\frac{1-u^2/c^2}{s})$$

No entanto fica muito mais apresentável a forma

$$\mathbf{F} = -\frac{e^2}{4\pi\epsilon_0}\nabla\left(\frac{1-u^2/c^2}{s}\right)$$

A diferença está em parênteses ajustáveis. A forma '`\left (`' e '`\right )`' permite esse ajustamento. Isso também é valido para '`\left [`' e '`\right ]`' bem como para '`\left {`' e '`\right }`'.

```
$${\bf F} = -{e^2 \over 4\pi\epsilon_0}\nabla\left({1-u^2/c^2 \over
s}\right)$$
```

É interessante fazer alguns comentários a respeito dessa fórmula para fixarmos as idéias: A forma `{\bf F}` visa gerar um 'F' em negrito. Coloca-se dentro de um bloco para não "contaminar" os outros caracteres com o 'font' *bold face* i.e. negrito. Existem ainda mais dois blocos nessa fórmula, ambos com o objetivo de fixar uma fração. Estude essa fórmula atentamente pois ela dá informações de como construir diversos tipos da simbologia matemática. Perguntas do tipo: "por que o sinal '-' está fora do bloco definindo a primeira fração: `{e^2 \over 4\pi \epsilon _0}`?", "qual o efeito de `e^2` ou `\epsilon _0`?", "você pode reconhecê-los

nas fórmulas?" se não puderem ser respondidas no exame da fórmula em TEX, podem ser respondidas modificando as fórmulas de diversas maneiras para se poder entender qual é a lógica que as rege.

Além dos delimitadores já citados ainda temos:

`\lbrack`	[(mesmo que '[')		
`\rbrack`	] (mesmo que ']')		
`\lfloor`	$\lfloor$	`\rfloor`	$\rfloor$
`\lceil`	$\lceil$	`\rceil`	$\rceil$
`\lbrace`	{ (mesmo que '{')		
`\rbrace`	} (mesmo que '}')	`\rangle`	$\rangle$
`\langle`	$\langle$		

Pode-se também, ao invés de `\left` ou `\right` usar `\bigl` ou `\bigr` além de `\biggl` `\Bigl` e `\Biggl` com seus respectivos pares "right" para produzir qualquer um dos delimitadores aumentados. Isto é interessante sobretudo quando existe mais de 1 nível de parênteses ou qualquer delimitador. Exemplo:

```
$$\Bigl(\left(y\in B\right)\to\bigl(\left(y\in
A\right)\land\left(y\in y\right) \bigr)\Bigr)\land\Bigl(\bigl(y\in
A\land y\not\in y\bigr)\to y\in B\Bigr)$$
```

produzirá:

$$\Big((y \in B) \to \big((y \in A) \land (y \in y)\big)\Big) \land \Big(\big(y \in A \land y \notin y\big) \to y \in B\Big)$$

Existem símbolos que podem ser definidos de forma alternativa, como é o caso de '[', ']' e outros citados acima:

$\neq$	`\ne`	`\neq`	`\not =`
$\leq$	`\le`	`\leq`	
$\geq$	`\ge`	`\geq`	
$\rightarrow$	`\to`	`\rightarrow`	
$\leftarrow$	`\gets`	`\leftarrow`	
$\ni$	`\owns`	`\ni`	
$\wedge$	`\land`	`\wedge`	
$\vee$	`\lor`	`\vee`	
$\neg$	`\lnot`	`\neg`	
$\vert$	`\vert`	`|`	
$\Vert$	`\Vert`	`\\|`	

Existem símbolos não matemáticos obtidos em modo texto que também podem ser gerados em modo matemático se forem usados os comandos corretos: †(`\dag` , `$\dagger $`) e ‡(`\ddag` , `$\ddagger $`). Outros símbolos tais como `\S` e `\P` , respectivamente, '§' e '¶' podem ser usados dentro do modo matemático salvo algumas precauções por exemplo em índices ou expoentes. Deve-se usar `$x^{\P }$` em vez de `$x^\P $`.

Em modo matemático os acentos são tratados de uma forma um pouco diferente e talvez um pouco mais trabalhosa. No entanto eles existem:

`$\hat a$`	$\hat{a}$
`$\check a$`	$\check{a}$
`$\tilde a$`	$\tilde{a}$
`$\acute a$`	$\acute{a}$
`$\grave a$`	$\grave{a}$
`$\dot a$`	$\dot{a}$
`$\ddot a$`	$\ddot{a}$
`$\breve a$`	$\breve{a}$
`$\bar a$`	$\bar{a}$
`$\vec a$`	$\vec{a}$

Note que a última definição dispensa a nossa composição que daria em $\vec{v}$. Há uma variante desses acentos no sentido de obter um efeito de acentos cobrindo até 3 letras:

`$\widehat x, \widetilde x$`	$\widehat{x}, \widetilde{x}$
`$\widehat {xy}, \widetilde {xy}$`	$\widehat{xy}, \widetilde{xy}$
`$\widehat {xyz}, \widetilde {xyz}$`	$\widehat{xyz}, \widetilde{xyz}$

A quantidade de símbolos disponíveis em TEX segue, aproximadamente, a definição de ∞. Os "TEXlólogos" mais experimentados ainda estão por esgotar todas as possibilidades. Por isso, não estranhe se a lista de símbolos descrita até aqui se mostrar incompleta no futuro.

Pode-se controlar, em modo matemático, a dimensão das letras sem que seja necessário mudar o 'font'. São comandos que são colocados automaticamente nas expressões matemáticas para controlar o tamanho dos símbolos em índices e expoentes, bem como em expressão inserida no texto, caso do $\frac{1}{2}$ (`$1\over 2$`) ou em expressão em destaque, caso do

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

(`$$1\over 2$$`) Em suma, existe o `\displaystyle` , `\textstyle` , `\scriptstyle` , e o `\scriptscriptstyle` . Cada um deles é usado e combinado de maneira a resultar na aparência mais conveniente, sem que o usuário se dê o trabalho de controlar isso. Um exemplo de como o tamanho varia de acordo com o uso desses comandos é

`$$n+\scriptstyle n+\scriptscriptstyle n$$`:

$$n+{\scriptstyle n+{\scriptscriptstyle n}}$$

Algumas vêzes esse tipo de variação não interessa como é o caso de:

$$a_0+\cfrac{1}{a_1+\cfrac{1}{a_2+\cfrac{1}{a_3+\cfrac{1}{a_4}}}}$$

obtido de

```
$$a_0+{1\over a_1+ {1\over a_2 + {1\over a_3 + {1\over a_4}}}}$$
```

A aparência ficará muito melhor se fizermos:

```
$$a_0+{1\over\displaystyle a_1+
        {\strut 1\over\displaystyle a_2+
          {\strut 1\over\displaystyle a_3+
            {\strut 1\over a_4}}}}$$
```

De maneira que o resultado seria:

$$a_0+\cfrac{1}{a_1+\cfrac{1}{a_2+\cfrac{1}{a_3+\cfrac{1}{a_4}}}}$$

`\strut` é um "caracter" de espessura nula (não é impresso) mas cuja altura é igual ao valor máximo do 'font' em uso no momento de sua utilização. Entre outras utilidades, `\strut` atua aqui no sentido de garantir uma altura conveniente para os denominadores.

Outros operadores seguem a mesma regra que o `\over` . É o caso de `\atop` : `$$x\atop y+2$$` para dar

$$\begin{matrix} x \\ y+2 \end{matrix}$$

que nada mais é que o `\over` sem o traço da fração, e `$$n\choose k$$` para dar

$$\binom{n}{k}$$

Por isso, se se quiser produzir:

$$\binom{n}{\frac{k}{2}}$$

deve-se fazer: `$$n\choose {k\over 2}$$` e não `$${n\choose k}\over 2$$`, que daria em

$$\frac{\binom{n}{k}}{2}$$

Cuidado: não use `$$ n\choose k\over 2 $$`. É errado. `\choose` e `\over` não admitem ficar no mesmo nível. Deve-se agrupar um deles num bloco. Por fim existe o `\above <dimen>`, onde `<dimen>` é um valor de uma certa dimensão. Esse comando fará aparecer um traço de fração de altura variável segundo o valor de `<dimen>`. Exemplo: a expressão `$$\displaystyle {a\over b}\above 1pt\displaystyle {c\over d}$$` dará em:

$$\frac{a}{\frac{b}{\frac{c}{d}}}$$

note a linha mais espessa.

Os diferentes "estilos" também modificam a apresentação de certos símbolos de forma a tornar, seja o texto, seja as fórmulas em evidência, mais "limpos". Assim ao fazermos: `$\sum _{n=1}^m$`, o símbolo aparecerá como: $\sum_{n=1}^{m}$, enquanto se fizermos: `$$\sum _{n=1}^m$$`, o símbolo aparecerá como:

$$\sum_{n=1}^{m}$$

A maneira de controlar isso, tanto em `\sum` como em `\int` é através de `\limits` ou `\nolimits`. Assim se fizermos: `$$\int _{-\infty }^{+\infty }$$`, teremos:

$$\int_{-\infty}^{+\infty}$$

enquanto que se fizermos: `$$\int \limits _{-\infty }^{+\infty }$$` o resultado é

$$\int\limits_{-\infty}^{+\infty}$$

Da mesma forma: `$$\sum \nolimits _{n=1}^m$$` vai dar em:

$$\sum\nolimits_{n=1}^{m}$$

De outra parte, é possível "numerar" nossas fórmulas através de `\eqno`, por exemplo:

`$$\sum_{n=1}^m a_i\sin i\omega t \eqno(1.0)$$`

produz:

$$\sum_{n=1}^{m} a_i \sin i\omega t \tag{1.0}$$

Existe uma classe de operadores que desempenham um papel especial. Exemplo: quando fazemos `$$sin x = x$$` o resultado será decepcionante:

$$sinx = x$$

O símbolo da função seno não se separa de x, de forma que a função não se "destaca" de seu parâmetro. Por isso o TEX define as funções como um 'font' especial, enquanto que toma precauções para que o parâmetro, ou os parâmetros se destaquem do símbolo dessa funções. Essa funções pré-definidas em TEX são:

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

De forma que ao fazermos `$$\sin x = x$$` teremos:

$$\sin x = x$$

Em matemática é corrente o uso das "reticências". TEX fornece opções para as formas de apresentação delas. Assim podemos fazer $f(x_1, \ldots, x_n)$ ou $x_1 = \cdots = x_n = 0$. Que foram obtidos por `$f(x_1,\ldots ,x_n)$` e `$x_1=\cdots =x_n=0$`. Essa preocupação com a apresentação se estende também para a forma de $(1 + x) \cdot (1 + y)$ `($(1+x)\cdot (1+y)$)`.

Seguindo as sinalizações de carater indutivo temos:

```
$$\overbrace{x+\cdots+x}^{k\rm\;times}$$
```

o que permite fazer:

$$\overbrace{x + \cdots + x}^{k \text{ times}}$$

Aqui o sinal `\;` serve para separar o k do *times*;

```
$$\underbrace{x+y+z}_{>\,0}$$
```

dará em:

$$\underbrace{x + y + z}_{>0}$$

O sinal `\,` provoca uma separação menor. Além de `\;` e o `\,` existe o `\!` que é uma separação de mesma dimensão que o `\,` mas negativa. Separações positivas maiores que as já citadas podem ser obtidas através de `\quad` ou `\qquad` .

Matrizes são facilmente criadas em TEX. No entanto, obviamente só podem ser geradas se for em evidência:

```
$$A=\pmatrix{x-\lambda &     1      &       0     \cr
                 0     & x-\lambda &       1     \cr
                 0     &     0     & x-\lambda \cr}$$
```

Dará em:

$$A = \begin{pmatrix} x-\lambda & 1 & 0 \\ 0 & x-\lambda & 1 \\ 0 & 0 & x-\lambda \end{pmatrix}$$

A forma de apresentação indutiva também está presente nas matrizes:

```
$$A=\pmatrix{a_{11}&a_{12}&\ldots&a_{1n}\cr
             a_{21}&a_{22}&\ldots&a_{2n}\cr
             \vdots&\vdots&\ddots&\vdots\cr
             a_{m1}&a_{m2}&\ldots&a_{mn}\cr}$$
```

Para dar:

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix}$$

7. Espaços, caixas, linhas e pontos

TEX adota e reconhece as seguintes unidades de medida:

`pt`	*point, unidade básica*
`pc`	*pica, 1pc = 12pt*
`in`	*inch, 1in = 72.27pt*
`bp`	*big point, 72bp = 1in*
`cm`	*centimeter, 2.54cm = 1in*
`mm`	*millimeter, 10mm = 1cm*
`dd`	*didot point, 1157dd = 1238pt*
`cc`	*cicero, 1cc = 12dd*
`sp`	*scaled point, 65536sp = 1pt*

Essas unidades servem para estabelecer diversos parâmetros como os que vimos na seção 3. Pode-se, por exemplo estabelecer um espaço horizontal:

```
Essa linha ter\'a \hskip 2cm um espa\c co de 2 cent\'{\i}metros.
```

Vai dar em

Essa linha terá um espaço de 2 centímetros.

Mas esse "2 centímetros" na verdade é esticado pelo valor do aumento (`\magnification`) adotado no texto. Para obter exatos "2 centímetros" precisamos fazer:

```
Essa linha ter\'a \hskip 2truecm um espa\c co de 2
cent\'{\i}metros verdadeiros.
```

Para dar:

Essa linha terá um espaço de 2 centímetros verdadeiros.

Pode-se também impor um espaço vertical, seguindo a mesma regra. Por exemplo: quando se escreve o texto:

```
Essa n\~ao \'e apenas uma linha, ela foi \vskip 1cm repartida
para dar em duas separadas de 1cm.
```

Ter-se-á o resultado:

Essa não é apenas uma linha, ela foi

repartida para dar em duas separadas de 1cm.

Obviamente o resultado será mais interessante se `\vskip` for colocado entre dois parágrafos.

```
Esse \'e o primeiro par\'agrafo.

\vskip 1cm
Esse \'e o segundo, separado de 1 cent\'{\i}metro a mais do primeiro
al\'em das separa\c c\~oes j\'a impostas.
```

Têm como resultado:

Esse é o primeiro parágrafo.

Esse é o segundo, separado de 1 centímetro a mais do primeiro além das separações já impostas.

`\vfill` e `\hfill` ou `\vfil` e `\hfil` * controlam o *glue* ou – cola como quer a língua portuguesa – e são particularmente importantes para os iniciantes. Esses comandos "enchem" de "vazio" seja na vertical, seja na horizontal. Por exemplo, se quisermos "encostar" uma linha no lado direito do papel – necessidade comum para quem quer colocar a data no cabeçalho de uma carta – podemos usar uma das duas opções:

```
\rightline{Rio de Janeiro, 28 de fevereiro de 2000}
```

ou

```
\line{\hfil Rio de Janeiro, 28 de fevereiro de 2000}
```

Que o resultado é o mesmo:

Rio de Janeiro, 28 de fevereiro de 2000

Para todos os efeitos, por enquanto vamos entender o comando `\line` como sendo aquele que indica ao TEX que o texto dentro do bloco deve ser tratado como uma linha.

Da mesma forma podemos construir um texto "centrado":

```
\centerline{Poderia ser um T\'{\i}tulo}
```

ou

```
\line{\hfil Poderia ser um T\'{\i}tulo \hfil}
```

Para dar:

Poderia ser um Título

Além disso com `\vfill` você pode forçar passar para a próxima página através de:

```
\vfill
\eject
```

TEX trabalha com base no conceito de "caixa", ou *box*. Existem caixas horizontais e verticais. A página é a maior caixa vertical possível. Uma linha é uma caixa horizontal do tamanho definido em `\hsize` . A menor caixa possível têm a dimensão de 1sp. Letras, números e símbolos são caixas individuais. Uma caixa tem altura, largura e profundidade. A

* `\hfill` e `\hfil` assim como `\vfill` e `\vfil` têm usos diferentes. Enquanto o `fill` é "hard", "fil" é "soft". O "fill" força o preenchimento sem "glue", enquanto "fil" admite o "glue" a fim de termos maior "jogo de cintura".

profundidade (*depth*) é uma altura contada para baixo a partir da linha de base. Essa linha de base é definida, por exemplo, de maneira que os 'g's e 'q's tenham suas "pernas" abaixo dela.

Se você definir uma caixa no meio do texto, poderá modificar certos parâmetros como margens e espaçamentos sem prejuizo do texto integral. O texto abaixo foi escrito dentro de uma caixa vertical em que `\baselineskip ,` e `\hsize` foram modificados:

> Thus the z component of angular momentum bears a fixed ratio to the energy radiated by a multipole of given m, and $m \neq 0$ corresponds to circularly polarized light. This is a purely classical result, but if the radiated energy is assumed to consist of photons each having energy $\hbar\omega$, the z component of the angular momentum per photom is mh. Equation (14-132) does not, however, distinguish between spin and orbital angular momentum, and its application to individual photons is not fully justified. (Panofsky & Phillips, *Classical Electricity and Magnetism*, 2nd. Edition, Addison-Wesley Eds, 1971)

Os valores de `\hoffset` e `\voffset` , que controlam as margens não podem ser modificados. Por isso, para jogar o texto mais para "dentro" lancei mão de `\hskip` . Veja como o texto acimo foi escrito:

```
\vskip 24pt
\hskip 1cm\vbox{\baselineskip=24pt \hsize=3.8in
Thus the $z$ component of angular momentum bears a fixed ratio
to the energy radiated by a multipole of given $m$, and $m \ne 0$
corresponds to circularly polarized light. This is a purely
classical result, but if the radiated energy is assumed to
consist of photons each having energy $\hbar\omega$, the $z$
component of the angular momentum per photom is $mh$. Equation
(14-132) does not, however, distinguish between spin and orbital
angular momentum, and its application to individual photons
is not fully justified. (Panofsky \& Phillips, {\it Classical
```

```
Electricity and Magnetism}, 2nd. Edition, Addison-Wesley Eds,
1971)}
```

O comando `\vskip 24pt` foi colocado para separar mais o texto do parágrafo precedente. Vejamos agora o que acontece se fizermos:

```
\vskip 24pt
\line{\hfil \vbox{\baselineskip=24pt \hsize=3.8in
Thus the $z$ component of angular momentum bears a fixed ratio
to the energy radiated by a multipole of given $m$, and $m \ne 0$
corresponds to circularly polarized light. This is a purely
classical result, but if the radiated energy is assumed to
consist of photons each having energy $\hbar\omega$, the $z$
component of the angular momentum per photom is $mh$. Equation
(14-132) does not, however, distinguish between spin and orbital
angular momentum, and its application to individual photons
is not fully justified. (Panofsky \& Phillips, {\it Classical
Electricity and Magnetism}, 2nd. Edition, Addison-Wesley Eds,
1971)}}
```

Ou seja, geramos uma "caixa" dentro de um comando típico de fazer "encostar" uma linha no lado direito. O resultado será:

Thus the z component of angular momentum bears a fixed ratio to the energy radiated by a multipole of given m, and $m \ne 0$ corresponds to circularly polarized light. This is a purely classical result, but if the radiated energy is assumed to consist of photons each having energy $\hbar\omega$, the z component of the angular momentum per photom is mh. Equation (14-132) does not, however, distinguish between spin and orbital angular momentum, and its application to individual photons is not fully justified. (Panofsky & Phillips, *Classical Electricity and Magnetism*, 2nd. Edition, Addison-Wesley Eds, 1971)

Ou seja, o texto todo ficou "encostado" à direita. Tanto `\line` quanto `\vbox` são tratados pelo TEX como "caixas" de tamanho ajustável. Assim ele não olha para ver se o "contexto" provoca uma caixa vertical dentro de uma caixa horizontal (é assim que `\line` é definido) maior que a "altura" desta última.

As caixas podem ter tamanho pré-definido. Se fizermos `\hbox to 150pt{...}` geramos uma caixa de 150pt de largura. Por exemplo, o texto escrito como abaixo:

```
Verifique: \hbox to 150pt{\hfil Semi-centrado \hfil} texto fora da
caixa.
```

Vai gerar:

Verifique: Semi-centrado texto fora da caixa.

Da mesma forma podemos fazer `\vbox to 50pt{...}` para gerarmos uma caixa de 50pt de altura.

Exemplo:

```
Texto superior

\vbox to 50pt{\vrule}

Texto inferior
```

Faz aparecer:

Texto superior

Texto inferior

O risco vertical que aparece acima é provocado pelo `\vrule` . Da mesma forma `\hrule` vai provocar um risco horizontal. Exemplo:

```
Esse texto

\vskip \baselineskip
\hrule
```

```
\'e separado por um tra\c co.
```

Aparecerá sob a forma:

Esse texto

é separado por um traço.

Você pode se perguntar o porquê de `\vskip \baselineskip` . O que eu fiz foi dizer ao TEX para avançar um espaço de 1 linha. Lembre-se do início do texto onde adotamos o valor de `\baselineskip = 15pt`. Esse é o valor do espaçamento entre as linhas.

Existem também os comandos `\hrulefill` e `\dotfill` que nos são úteis par escrevermos o índice de um livro, por exemplo:

Assunto Pag.

Capítulo I 9

Capítulo II 21

Foi obtido fazendo:

```
Assunto\hfill Pag.

Cap\'{\i}tulo I\hrulefill 9\par
Cap\'{\i}tulo II\dotfill 21
```

Alguns tipos de deslocamentos são pré-definidos, a saber:

1. Vertical: `\smallskip` , `\medskip` , `\bigskip` : valem respectivamente: (3 ± 1)pt, (6 ± 2)pt e (12 ± 4)pt, onde o '$\pm$' significa o "glue", ou seja o máximo que TEX pode "comprimir" ou "esticar" o texto na direção vertical.
2. Horizontal: `\quad` (1em), `\qquad` (2em), `\endskip` (0.5em), `\enspace` (0.5em)*, `\thinspace` (0.16667em) e `\negthinspace` (-0.16667em).

Com os conhecimentos que aprendemos até aqui, podemos nos aventurar num exercício interessante. Repare no texto abaixo:

---

* Use mais o `\endskip`

De repente, como geléia, o chão se partiu em dois nos separando em dois grupos. Tudo ruia como se fossem castelos de areia. As pessoas do meu grupo se precipitaram a uma espécie de ônibus que mais parecia uma daquelas antigas jardineiras. O motorista daquela geringonça não se preocupou em parar. As pessoas se lançavam como desesperados para dentro do veículo mesmo em movimento e em segundos não havia lugar para mais ninguém. Uma grande parte do grupo acabou pendurada nas portas, janelas ou em qualquer coisa onde pudesse agarrar. Diante de toda a balbúrdia e desespero postei meus olhos ao chão e perguntei aos meus botões: –E eu?

Foi produzido escrevendo-se:

```
\vskip 2\baselineskip
\vbox{\hrule\hbox{\vrule\vbox{%
De repente, como gel\'eia, o ch\~ao se partiu em dois nos
separando em dois grupos. Tudo ruia como se fossem castelos
de areia. As pessoas do meu grupo se precipitaram
a uma esp\'ecie de \^onibus que mais parecia
uma daquelas antigas jardineiras. O motorista daquela
geringon\c ca n\~ao se preocupou em parar. As pessoas se
lan\c cavam como desesperados para dentro do ve\'{\i}culo mesmo
em movimento e em segundos n\~ao havia lugar para mais
ningu\'em. Uma grande parte do grupo acabou pendurada
nas portas, janelas ou em qualquer coisa onde pudesse
agarrar. Diante de toda a balb\'urdia e desespero postei
meus olhos ao ch\~ao e perguntei aos meus bot\~oes: --E eu?
}\vrule}\hrule}
```

Foi criada uma caixa vertical "delimitada" por dois traços horizontais. O tamanho da caixa é ajustável (não foi definida a dimensão da caixa). Em seguida criou-se uma caixa horizontal "delimitada" por dois traços verticais. Dentro desta cria-se uma outra caixa vertical de maneira a se escrever um texto para ser separado e justificado numa aparência de parágrafo.

Contudo a apresentação final não é agradável pois os traços praticamente se superpõem às letras nas bordas.

De repente, como geléia, o chão se partiu em dois nos separando em dois grupos. Tudo ruia como se fossem castelos de areia. As pessoas do meu grupo se precipitaram a uma espécie de ônibus que mais parecia uma daquelas antigas jardineiras. O motorista daquela geringonça não se preocupou em parar. As pessoas se lançavam como desesperados para dentro do veículo mesmo em movimento e em segundos não havia lugar para mais ninguém. Uma grande parte do grupo acabou pendurada nas portas, janelas ou em qualquer coisa onde pudesse agarrar. Diante de toda a balbúrdia e desespero postei meus olhos ao chão e perguntei aos meus botões: –E eu?

Foi obtido acrescentando-se:

```
\vskip 2\baselineskip
\vbox{\hrule\hbox{\vrule\kern5pt\vbox{\kern5pt%
\strut De repente, como gel\'eia, o ch\~ao se partiu em dois nos
.
.
.
meus olhos ao ch\~ao e perguntei aos meus bot\~oes: --E eu?\strut
\kern5pt}\kern5pt\vrule}\hrule}
```

O comando `\kern n` onde `n` é uma dimensão faz com que o texto seja construido a `n` afastado dos traços. Já o comando `\strut` serve para garantir imprevistos. É preciso que se diga que o "caracter" `\hrule` não possui *glue*, ou cola, isto é, o TEX não calcula espaçamento de linhas ou parágrafos. O comando `\strut` garante o cálculo desses parâmetros.

De repente, como geléia, o chão se partiu em dois nos separando em dois grupos. Tudo ruia como se fossem castelos de areia. As pessoas do meu grupo se precipitaram a uma espécie de ônibus que mais parecia uma daquelas antigas jardineiras. O motorista daquela geringonça não se preocupou em parar. As pessoas se lançavam como desesperados para dentro do veículo mesmo em movimento e em segundos não havia lugar para mais ninguém. Uma grande parte do grupo acabou pendurada nas portas, janelas ou em qualquer coisa onde pudesse agarrar. Diante de toda a balbúrdia e desespero postei meus olhos ao chão e perguntei aos meus botões: –E eu?

O "cabeçalho" do último texto é assim:

```
\vbox{\hrule\hbox{\vrule\kern5pt\vbox{\hsize 4.0 truein
\baselineskip=6pt\kern5pt
.
.
.
```

A largura da "página" foi modificada de maneira a ficar mais estreita através de `\hsize` . Isso é possível dentro do `\vbox` como vimos anteriormente. O tamanho final da caixa é definida por esse valor.

Para terminar com essa questão. O leitor está convidado a fazer

```
$$\vbox{\hrule\hbox{\vrule\kern5pt\vbox{\hsize 4.0 truein
\baselineskip=6pt\kern5pt
.
.
.
\kern5pt}\kern5pt\vrule}\hrule}$$
```

E verificar o resultado. Esse exercício é ilustrativo do que pode fazer o comando de equação matemática em evidência e os *boxes*.

8. Lay-out

O comando `\settabs` controla a tabulação horizontal. É preciso dizer ao TEX que você está tabulando. Para isso usa-se o comando \+ para dizer que se está iniciando a tabulação e `&` para dizer que a tabulação continua na mesma linha. Por exemplo, imagine que você escreva a seguinte tabela:

```
\settabs 2 \columns
\+ Z\'elia & demitida \cr
\+ Ibraim & demitido \cr
\+ Ant\^onio & demitido \cr
\+ Marc\'{\i}lio & admitido \cr
\+ Fernando & abandono \cr
```

Será criada a seguinte tabela:

Zélia	demitida
Ibraim	demitido
Antônio	demitido
Marcílio	admitido
Fernando	abandono

Após o `\settabs` vêm `2 \columns` que indica que deve-se dividir a página em 2 e calcular a tabulação em função disso. Uma vez que se queira terminar a linha tabulada é preciso fazer `\cr` (*control carriage*). Imagine agora que se faça:

```
\+ Fernando & abandono & viajou \cr
```

O resultado será:

Fernando	abandono	viajou
Zélia	demitida	
Ibraim	demitido	
Antônio	demitido	
Marcílio	admitido	
Fernando	abandono	

A palavra "viajou" fica fora do contexto da página. É claro que no lugar do "2" pode ser colocado qualquer outro de maneira a produzir uma tabela de várias colunas.

Outra forma de definir uma tabulação é em vez de definir `2 \columns` é colocar um "template", por exemplo:

```
\settabs \+ ~Fernando~ & ~abandono~ \cr
```

.

.

e o resultado será:

Zélia demitida
Ibraim demitido
Antônio demitido
Marcílio admitido
Fernando abandono

Note que nesse caso o tabulador será exatamente do tamanho que o "template" estabeleceu. Note-se que o "template" ele mesmo não é impresso.

A característica desse tipo de tabulação é que a tabela ficará forçosamente à esquerda. Não é possível "centrar" um texto dentro de uma tabulação. Tampouco empurrá-lo à direita. Existe uma outra forma de tabular em TEX, usado sobretudo, para construir tabelas. Essa forma chama-se `\halign` (alinhamento horizontal). Imagine a tabela acima escrita como:

```
\halign{#\hfil&\quad#\hfil\cr
 Z\'elia & demitida \cr
 Ibraim & demitido \cr
 Ant\^onio & demitido \cr
 Marc\'{\i}lio & admitido \cr
 Fernando & abandono \cr}
```

O resultado apresentará uma pequena diferença:

Zélia demitida
Ibraim demitido
Antônio demitido
Marcílio admitido
Fernando abandono

O sinal '#' identifica um "parâmetro" na tabela que vem a ser o "dado" (Zélia, demitida, etc). O comando `\quad` faz com que as columas na tabela sejam afastadas uma da outra. A primeira linha depois de `\halign` indica o padrão que a tabela terá. Para centrar o texto em cada coluna usamos, então, o recurso dos `\hfil` :

```
\halign{\hfil#\hfil&\quad\hfil#\hfil\cr
 Z\'elia & demitida \cr
 Ibraim & demitido \cr
 Ant\^onio & demitido \cr
```

```
Marc\'{\i}lio & admitido \cr
Fernando & abandono \cr}
```

E teremos então:

Zélia	demitida
Ibraim	demitido
Antônio	demitido
Marcílio	admitido
Fernando	abandono

Muitas vêzes não se quer a tabela assim tão próxima da margem esquerda, como estas que vimos. Podemos usar o recurso do comando `\indent` que empurra o texto do valor exato do início de parágrafo. Assim, basta colocar no "cabeçalho" da tabela:

```
\halign{\indent\hfil#\hfil&\quad\hfil#\hfil\cr
```

Para obtermos:

Zélia	demitida
Ibraim	demitido
Antônio	demitido
Marcílio	admitido
Fernando	abandono

Pode-se também, dentro de `\halign` introduzir um texto de maneira a cobrir todas as colunas. Por exemplo:

```
\halign{\indent\hfil#\hfil&\quad\hfil#\hfil\cr
\multispan 2 \hfil \indent Planalto Inc. \hfil \cr
 Z\'elia & demitida \cr
 Ibraim & demitido \cr
 Ant\^onio & demitido \cr
 Marc\'{\i}lio & admitido \cr
 Fernando & abandono \cr}
```

Para produzir:

Planalto Inc.	
Zélia	demitida
Ibraim	demitido
Antônio	demitido
Marcílio	admitido
Fernando	abandono

Para traçarmos uma linha horizontal no meio da tabela é preciso dizer ao TEXque na linha onde o traço deve ser passado, não deve haver tabulação. Para isso devemos fazer:

```
\noalign{\hrule}
```

Usando um expediente que veremos mais tarde, podemos definir um "traço" horizontal, de maneira a que não tenhamos que dar esse comando a cada vez que queremos traçar uma linha horizontal:

```
\def\trule{\noalign{\hrule}}
```

O comando `\def` diz que a próxima palavra (`\trule`) deve ser substituida pelo que vem dentro das chaves. Assim, a tabela acima pode ser "melhorada" fazendo:

```
\halign{\strut\indent\hfil#\hfil&\quad\hfil#\hfil\cr
\def\trule{\noalign{\hrule}}
\trule
\multispan 2 \hfil \indent Planalto Inc. \hfil \cr
\trule
 Z\'elia & demitida \cr
 Ibraim & demitido \cr
 Ant\^onio & demitido \cr
 Marc\'{\i}lio & admitido \cr
 Fernando & abandono \cr}
 \trule
```

Cujo resultado podemos ver a seguir:

Planalto Inc.	
Zélia	demitida
Ibraim	demitido
Antônio	demitido
Marcílio	admitido
Fernando	abandono

Vale a pena discutir agora o que TEX classifica como o *glue*, mal traduzindo, "cola". Como vimos a "cola" controla a distância entre letras e palavras de maneira a fazer o texto apresentar-se dentro de um parágrafo. Dentro de uma tabela controlada por `\halign` , essa cola pode ser controlada através `\tabskip` . Por exemplo, `\tabskip =1em plus2em minus .5em` indica que a separação entre as colunas subsequentes é de $1em$ podendo ser "esticada" até $+2em$ e "comprimida" até $-.5em$*. `\tabskip` age para as colunas que seguem a separação '&' no cabeçalho da tabela. Não é válida para a coluna corrente. Exemplo:

* $1em$ =`\quad` no 'font' corrente

```
\halign{\indent\hfil#\hfil\tabskip=1em&\hfil#\hfil\cr
```

fará com que somente a segunda coluna seja regida por `\tabskip` . Você pode fazer, como indica o exemplo em TEXbook:

```
\tabskip=3pt
\halign{\hfil#\tabskip=4pt& ...
```

Ou seja, declarar `\tabskip` antes de `\halign` . Podemos, finalmente, fazer uma tabela dentro de um *box*, com traços, onde convém:

Planalto Inc.	
Zélia	demitida
Ibraim	demitido
Antônio	demitido
Marcílio	admitido
Fernando	abandono

Que foi produzido fazendo:

```
\vskip 24pt
\vbox{\tabskip=0pt \offinterlineskip
\halign to 150pt{\strut#& \vrule#\tabskip=1em plus1em&
\hfil#\hfil & \vrule# & \hfil#\hfil & \vrule# \tabskip=0pt\cr
\trule
&& \multispan3 \hfil Planalto Inc. \hfil &\cr
\trule
&& Z\'elia && demitida &\cr
&& Ibraim && demitido &\cr
&& Ant\^onio && demitido &\cr
&& Marc\'{\i}lio && admitido &\cr
&& Fernando && abandono &\cr
\trule}}
```

A aparente complexidade pode ser superada em fazendo apenas uma vez esse tipo de formato. Todas as outras tabelas podem seguir a mesma tática de maneira que você só seja obrigado a fazer isso uma vez. Nas outras vezes o trabalho será apenas de modificar os parâmetros.

Vamos iniciar por `\offinterlineskip` . É um comando que cancela aqueles espaços entre as linhas. O comando `\interlineskip` é automático no TEX. Ele usa o valor de `\baselineskip` . `\offinterlineskip` "desliga" o afastamento entre as linhas. Essa medida

é necessária pois caso contrário os traços verticais ficariam interrompida. Por isso a tabela deve ficar dentro de um `\vbox` . Em seguida `\tabskip =0pt` impõe que os traços verticais, em cada linha, fique lá no lugar onde eles devem estar. Sem compromisso de se "esticar" ou se "comprimir" na tabela. Deixa-se a primeira coluna em branco apenas com o "caracter" `\strut` que, como já vimos, é uma caixa de espessura nula e altura igual ao maior caracter do 'font' corrente. Isso deverá impor uma altura para o `\vrule` que segue logo após. O '#' que segue é necessário pois em `\halign` o TEX sempre pede um dado no cabeçalho de formato. Seguido a `\vrule` vem o `\tabskip =1em plus1em` que indica que o texto deve ser separado de $1em$ e esticado até $1em$. Tente outros valores e veja o resultado. É bastante educativo.

Além do alinhamento do texto, contamos com o alinhamento em modo matemático. Para o caso de equações em evidência usamos algo bastante parecido com `\halign` . Por exemplo,

```
$$\eqalign{E_x & = {e x_0\over4\pi\epsilon_0 s^3}
\left( 1 - {u^2\over c^2}\right), \cr
          E_y & = {e y_0\over4\pi\epsilon_0 s^3}
\left( 1 - {u^2\over c^2}\right), \cr
          E_z & = {e z_0\over4\pi\epsilon_0 s^3}
\left( 1 - {u^2\over c^2}\right). \cr}$$
```

Vai produzir em seu texto o seguinte:

$$\eqalign{E_x & = {e x_0\over4\pi\epsilon_0 s^3}
\left(1 - {u^2\over c^2}\right), \cr
          E_y & = {e y_0\over4\pi\epsilon_0 s^3}
\left(1 - {u^2\over c^2}\right), \cr
          E_z & = {e z_0\over4\pi\epsilon_0 s^3}
\left(1 - {u^2\over c^2}\right). \cr}$$

Note como se produz o conjunto de equações acima com numeração única.

```
$$\eqalign{E_x & = {e x_0\over4\pi\epsilon_0 s^3}
\left( 1 - {u^2\over c^2}\right), \cr
          E_y & = {e y_0\over4\pi\epsilon_0 s^3}
\left( 1 - {u^2\over c^2}\right), \cr
          E_z & = {e z_0\over4\pi\epsilon_0 s^3}
\left( 1 - {u^2\over c^2}\right). \cr} \eqno (19)$$
```

Ou seja:

$$\begin{aligned} E_x &= \frac{ex_0}{4\pi\epsilon_0 s^3}\left(1-\frac{u^2}{c^2}\right), \\ E_y &= \frac{ey_0}{4\pi\epsilon_0 s^3}\left(1-\frac{u^2}{c^2}\right), \\ E_z &= \frac{ez_0}{4\pi\epsilon_0 s^3}\left(1-\frac{u^2}{c^2}\right). \end{aligned} \tag{19}$$

Conjunto de equações como esse, em que cada uma tem sua própria numeração, deve ser feito um pouco diferente:

```
$$\eqalignno{E_x & = {e x_0\over4\pi\epsilon_0 s^3}
\left( 1 - {u^2\over c^2}\right),& (19-11) \cr
          E_y & = {e y_0\over4\pi\epsilon_0 s^3}
\left( 1 - {u^2\over c^2}\right),& (19-12) \cr
          E_z & = {e z_0\over4\pi\epsilon_0 s^3}
\left( 1 - {u^2\over c^2}\right).& (19-13) \cr} $$
```

Ou seja:

$$\begin{aligned} E_x &= \frac{ex_0}{4\pi\epsilon_0 s^3}\left(1-\frac{u^2}{c^2}\right), && (19-11) \\ E_y &= \frac{ey_0}{4\pi\epsilon_0 s^3}\left(1-\frac{u^2}{c^2}\right), && (19-12) \\ E_z &= \frac{ez_0}{4\pi\epsilon_0 s^3}\left(1-\frac{u^2}{c^2}\right). && (19-13) \end{aligned}$$

A sintaxe geral de `\eqalign` é:

```
\eqalign{<lado esquerdo> & <lado direito> \cr
.
.
.
         <lado esquerdo> & <lado direito> \cr}
```

O `\eqalign` não aceita mais de dois termos como, por exemplo, o `\setttabs` . Já o `\eqalignno` aceita três termos:

```
\eqalignno{<lado esquerdo> & <lado direito> & <N.> \cr
.
.
.
           <lado esquerdo> & <lado direito> & <N.> \cr}
```

No caso de uma só numeração para todas as equações, basta colocar `\eqno` fora do contexto de `\eqalign` .

Dentro dessas condições fica fácil demonstrar a integral de Gauss para a função de distribuição normal:

$$\eqalignno{\biggl(\int_{-\infty}^\infty e^{-x^2}\,dx\biggr)^2
&=\int_{-\infty}^\infty\int_{-\infty}^\infty
  e^{(-x^2+y^2)}\,dx\,dy\cr
&=\int_0^{2\pi}\int_0^\infty e^{-r^2}r\,dr\,d\theta\cr
&=\int_0^{2\pi}\biggl(-{e^{-r^2}\over2}
 \biggr|_{r=0}^{r=\infty}\,\biggr)\,d\theta\cr
&=\pi.&(11)\cr}$$

Que foi produzido fazendo:

```
$$\eqalignno{\biggl(\int_{-\infty}^\infty e^{-x^2}\,dx\biggr)^2
             &=\int_{-\infty}^\infty\int_{-\infty}^\infty
               e^{(-x^2+y^2)}\,dx\,dy\cr
             &=\int_0^{2\pi}\int_0^\infty e^{-r^2}r\,dr\,d\theta\cr
             &=\int_0^{2\pi}\biggl(-{e^{-r^2}\over2}
              \biggr|_{r=0}^{r=\infty}\,\biggr)\,d\theta\cr
             &=\pi.&(11)\cr}$$
```

Repare que não foi necessário colocar os três termos aonde não queríamos numerar a equação. Três termos existem apenas na última linha sendo que o primeiro está em branco.

Vejamos agora como se produz em as relações:

$$\begin{aligned}
E'_{||} &= E_{||},\\
B'_{||} &= B_{||},\\
\mathbf{E}'_{\perp} &= \gamma\left(\mathbf{E}_{\perp} + \mathbf{v}\times\mathbf{B}_{\perp}\right),\\
\mathbf{B}'_{\perp} &= \gamma\left(\mathbf{B}_{\perp} - \mathbf{v}\times\mathbf{E}_{\perp}/c^2\right),
\end{aligned}$$

Trata-se de algo diferente. Foi usado o comando `\displaylines` que permite a disposição da série de equações sem alinhamento pré-definido. Para que cada equação seja centrada é preciso usar o recurso de dois `\hfill` "rodeando-a":

```
$$\displaylines{\hfill E^\prime_{||} = E_{||},\hfill\cr
```

```
            \hfill B^\prime_{||} = B_{||},\hfill\cr
\hfill\hbox{\bf E}^\prime_\bot = \gamma\left(\hbox{\bf E}_\bot
   + \hbox{\bf v} \times \hbox{\bf B}_\bot\right),\hfill\cr
\hfill\hbox{\bf B}^\prime_\bot = \gamma\left(\hbox{\bf B}_\bot
   - \hbox{\bf v} \times \hbox{\bf E}_\bot/c^2\right),\hfill\cr}$$
```

Digamos que é o caso, também, de colocar uma numeração para cada equação:

$$E'_{||} = E_{||}, \qquad (18-40)$$

$$B'_{||} = B_{||}, \qquad (18-41)$$

$$\mathbf{E}'_\perp = \gamma\left(\mathbf{E}_\perp + \mathbf{v} \times \mathbf{B}_\perp\right), \qquad (18-42)$$

$$\mathbf{B}'_\perp = \gamma\left(\mathbf{B}_\perp - \mathbf{v} \times \mathbf{E}_\perp/c^2\right), \qquad (18-43)$$

Usamos aqui a facilidade dada pelo comando `\llap` , uma abreviação de *left overlap*, ou seja, tudo o que vêm depois de `\llap` deve ser "colocado em cima" do que já foi escrito. Essa foi a facilidade aproveitada para construir o sinal '□', visto anteriormente. Para produzir as equações acima foi feito:

```
$$\displaylines{\hfill E^\prime_{||} = E_{||},\hfill\llap(18-40)\cr
                \hfill B^\prime_{||} = B_{||},\hfill\llap(18-41)\cr
\hfill\hbox{\bf E}^\prime_\bot = \gamma\left(\hbox{\bf E}_\bot
   + \hbox{\bf v} \times \hbox{\bf B}_\bot\right),\hfill\llap(18-42)\cr
\hfill\hbox{\bf B}^\prime_\bot = \gamma\left(\hbox{\bf B}_\bot
   - \hbox{\bf v} \times \hbox{\bf E}_\bot/c^2\right),
   \hfill\llap(18-43)\cr}$$
```

Outras questões são de interesse do *lay out*. Por exemplo, as notas de pé de página. Como exemplo vamos colocar um sinal indicando uma nota que você pode achar na parte inferior dessa mesma página*. Esse texto foi produzido assim:

```
Outras quest\~oes s\~ao de interesse do {\it lay out}. Por exemplo, as
notas de p\'e de p\'agina. Como exemplo vamos colocar um sinal indicando
uma nota que voc\^e pode achar na parte inferior dessa mesma
p\'agina\footnote*{Agora voc\^e pode voltar l\'a para cima e continuar lendo
o texto para saber como \'e que eu vim parar aqui.}. Esse texto foi
produzido assim:
```

* Agora você pode voltar lá para cima e continuar lendo o texto para saber como é que eu vim parar aqui.

As vêzes é interessante não produzir o espaço no início de parágrafo. Quando essa necessidade é eventual usa-se o comando `\noindent` antes de escrever o texto. Quando essa necessidade é sistemática, vale mais a pena mudar o valor da variável `\parindent` no início do arquivo, depois da declaração de `\magnification` , se houver, através de:

```
\parindent=0pt
```

Da mesma forma pode-se mudar o valor do espaço entre os parágrafos através de:

```
\parskip=0pt
```

Pode-se, é claro atribuir um valor maior para essas variáveis segundo o critério de cada um. Vimos por outro lado que variáveis tais como `\magnification` , `\hoffset` , e `\voffset` não podem ser modificadas uma vez definidas. Como produzir um texto que "deborde" pela esquerda? Tal como:

Divida dois por três, em seguida por cem, tire o resultado de oito para em seguida dividir por três e depois por dez. Tire o que resultar de dez e divida por dez mais uma vez para outra vez tirar de dez. Depois divida por dez, por três e finalmente por dois para somar um. Tire a metade e subtraia de dez. Por três divida tudo. (Anônimo tentando obter o valor de π por frações)

A dica é dar um espaço horizontal negativo e em seguida colocar o texto dentro de um `\vbox` . O texto acima foi produzido com:

```
\hskip -0.7 truein \vbox{\noindent
Divida dois por tr\^es, em seguida por cem, tire o resultado de oito
.
.
.
(An\^onimo tentando obter o valor de $\pi$ por fra\c c\~oes)}
```

Colocar itens em evidência de maneira a ordená-los pode ser obtido através de `\item {...}`. Exemplo:

```
\item{1.} As causas que podem agir sobre a vontade para diminuir
ou anular a sua {\it espontaneidade} s\~ao as seguintes:
\item{2.} As causas que agem sobre a intelig\^encia, para diminuir
ou suprimir o conhecimento de um fim, s\~ao os diversos tipos de
{\it ignor\^ancia}. A ignor\^ancia pode ser ...
```

faz com que cada item seja destacado de forma que a continuação do parágrafo fique mais "para dentro":

1. As causas que podem agir sobre a vontade para diminuir ou anular a sua *espontaneidade* são as seguintes:

2. As causas que agem sobre a inteligência, para diminuir ou suprimir o conhecimento de um fim, são os diversos tipos de *ignorância*. A ignorância pode ser ...

Algumas vezes é necessário destacar um item sobre um item. Com `\itemitem` isso é possível:

```
\item{1.} As causas que podem agir sobre a vontade para diminuir
ou anular a sua {\it espontaneidade} s\~ao as seguintes:
\itemitem{a)} A {\it paix\~ao}, quer dizer, a atra\c c\~ao violenta
para um bem sens\'{\i}vel. A {\it paix\~ao que antecede} ao ato
volunt\'ario diminui ou suprime o uso da raz\~ao e, por conseguinte,
a responsabilidade...\footnote*{R. Jolivet, {\it Curso de Filosofia},
AGIR, Rio de Janeiro, 1968}
```

vai produzir:

1. As causas que podem agir sobre a vontade para diminuir ou anular a sua *espontaneidade* são as seguintes:

   a) A *paixão*, quer dizer, a atração violenta para um bem sensível. A *paixão que antecede* ao ato voluntário diminui ou suprime o uso da razão e, por conseguinte, a responsabilidade...*

9. Macros e definições

No TEX é fundamental o conceito dos macro-comandos, isto é, definições de comandos que vão substituir um conjunto de comandos chamados "primitivos". Aqui, primitivo é todo o comando que foi definido anteriormente, podendo ser um primitivo de fato, ou seja, um comando definido no interior do programa TEX , ou sendo ele mesmo um macro já definido. Sem os macros, compor um texto TEX seria uma tarefa profundamente estafante. Seria enlouquecedor se, a cada vez que se quisesse escrever o sinal '$\gtrsim$' tivéssemos que fazer `\lower 3pt\hbox {$\buildrel > \over \sim \;$.}` É muito mais prático definir no início do arquivo o comando `'\gappr '` da seguinte forma:

```
\def\gappr{\lower 3pt\hbox{$\buildrel > \over \sim\;$}}
```

* R. Jolivet, *Curso de Filosofia*, AGIR, Rio de Janeiro, 1968

e a cada vez que quisermos esse sinal, por exemplo, em $x \gtrsim 0$ basta escrevermos `$x\gappr 0$`. Outras tantas definições são passíveis de serem feitas nos diversos sinais que aparecem nesse texto. Por exemplo:

```
\def\etal{{\it et al} }
```

$=$*et al* , Ex: `Fisher \etal (1988)`: Fisher *et al* (1988)

```
\def\lappr{\lower 3pt\hbox{$\buildrel < \over \sim\;$}}
```

$=\lesssim$, Ex: `$x\lappr 0$`: $x \lesssim 0$

```
\def\lapla{\hbox{$\sqcup$\hbox to 0pt{\hss$\sqcap$}}}
```

$=\square$, Ex: `$\lapla b^n = 0$`: $\square b^n = 0$

Veja agora esse exemplo:

```
\def\abrev#1{$^{\rm #1}$}
```

```
1\abrev{o} de abril
```

Que dá em: 1º de abril

A definição `\abrev` possui um parâmetro de entrada que é identificado por `# 1`. O uso desse comando, no caso de nosso exemplo deve ser feito simplesmente: `1\abrev {o} de abril` onde o 'o' será colocado como expoente indicando abreviação. Um comando pode ter vários parâmetros de entrada desde que declarados na sua definição. Veja como eu defini a forma de apresentação de referências de artigos:

```
\def\refer#1#2#3#4{\vbox{\par   \hangindent=25pt \noindent
 #1, {\it #2}, {\bf #3}, #4} }
```

Existem 4 parâmetros de entrada, cada um sendo colocado de maneira a produzir, como resultado, uma referência. Por exemplo:

```
\refer{Fisher O., Fisher H., Fisher Jr.,O, (1988), ``First Trial
on Family Survival in a Domestic Space Craft Simulator''}{Family Journal}{44}
{623-666}
```

vai produzir:

Fisher O., Fisher H., Fisher Jr.,O, (1988), "First Trial on Family Survival in a Domestic Space Craft Simulator", *Family Journal*, **44**, 623-666

Note que o resto do parágrafo ficou "para dentro" como em `\item` . O controle disso é feito pelo comando `\hangindent =25pt` dado na definição de `\refer` . Note que é importante também fazer `\noindent` para que o início do texto não seja "empurrado" para dentro. Cada parâmetro declarado na definição é identificado pelos números de 1 a 4, indicando que o comando tem 4 entradas. Cada texto de entrada é definido pelos delimitadores `'{ }'`. O primeiro tem um aspecto normal, o segundo ficará em itálico, o terceiro em negrito e o último terá aspecto normal, novamente.

Letras acentuadas muitas vezes tornam cansativa a composição do texto pela quantidade de teclas fora do "campo" do teclado "normal". Podemos reduzir bastante esse trabalho se definirmos previamente essas letras por comandos mais fáceis de serem compostos. No entanto não se poderá evitar o caracter '\'. Eis um exemplo de definições:

```
\def\aa{\'a}            % a agudo
\def\ea{\'e}            % e agudo
\def\ia{\'{\i}}         % i agudo
\def\oa{\'o}            % o agudo
\def\ua{\'u}            % u agudo
\def\ag{\`a}            % a grave
\def\eg{\`e}            % e grave
\def\ig{\`{\i}}         % i grave
\def\og{\`o}            % o grave
\def\ug{\`u}            % u grave
\def\ac{\^a}            % a circunflexo
\def\ec{\^e}            % e circunflexo
\def\ic{\^{\i}}         % i circunflexo
\def\oc{\^o}            % o circunflexo
\def\uc{\^u}            % u circunflexo
\def\at{\~a}            % a til
\def\ot{\~o}            % o til
\def\ar{\"a}            % a trema
\def\er{\"e}            % e trema
\def\ir{\"{\i}}         % i trema
\def\or{\"o}            % o trema
\def\ur{\"u}            % u trema
\def\yr{\"y}            % y trema
```

É aconselhável colocar essas definições no início do texto para melhor estruturar o seu trabalho. Pode-se também colocar essas definições em um arquivo separado, chamado, por exemplo, ACENTOS.TEX. No arquivo do texto coloca-se o comando:

```
\input acentos
```

O comando `\input` permite incluir o arquivo que se queira, esteja ele com comandos ou com texto, desde que TEX o reconheça. Digamos que exista um arquivo chamado FORMATO.TEX com os seguintes comandos:

```
\magnification=\magstep1
\hsize=6.8 true in
\vsize=9.0 true in
\hoffset=0.0 true cm
\voffset=0.0 true cm
\parskip=6pt
\parindent=12pt
\baselineskip=12pt
```

seria o formato da página que você quer para o seu texto. Digamos que existe também uma arquivo com definições matemáticas chamado MATH.TEX. No início do arquivo, antes de iniciar o texto você coloca:

```
\input formato
\input math
\input acentos
```

Um macro pode aproveitar um outro, definido no mesmo texto. Por exemplo, podemos definir o comando `\cao` para de uma só vez escrever a sílaba "ção", tão usada no português:

```
\def\cc{\c c}
\def\at{\~a}
\def\ao{\at o}
\def\cao{\cc\ao}
```

Note apenas que o 'o' ficou separado do `\at` pois caso contrário o TEX poderia interpretar o comando `\ato`, que não definimos. Contudo essa definição de `\cao` não está completa visto que o "ção" vem sempre no final de uma palavra. Se após definirmos `\cao` escrevermos:

```
M\ao de mam\ao no cora\cao de S\ao Jo\ao
```

o resultado será:

Mãode mamãono coraçãode SãoJoão.

É preciso pensar um pouco mais e colocar um '\' seguido de um espaço ao final das definições, bem entendido antes do sinal '}'.

Macros também podem ser definidos dentro de outros macros. Podem também se auto recorrer representando uma certa recursividade. Veja, por exemplo, o que D. Knuth nos propõe no exercício 20.1, página 200 de seu TEXbook:

```
\def\a{\b}
\def\b{A\def\a{B\def\a{C\def\a{\b}}}}
\def\puzzle{\a\a\a\a\a}
```

Se fizermos `\puzzle` o resultado será: ABCAB.

10. Algumas dicas ordinárias

Podemos "construir" um espaço horizontal através de `\phantom {..texto..}` que produzirá um espaço em branco do tamanho exato do texto inserido entre os `'{ }'`. Às vêzes desejamos produzir um efeito do tipo:

Essa linha deverá
continuar logo abaixo.

Foi produzido da seguinte forma:

```
\noindent Essa linha dever\'a \hfil \break
\phantom{Essa linha dever\'a} continuar logo abaixo.
```

Ou então desejamos fazer:

```
\underbar{\phantom{coloque aqui o seu nome}}
```

que dará em: _____

****

Quando quiser se referir a uma "variável" cujo nome é uma letra do alfabeto, coloque-a dentro do modo matemático, isto é, `$...$`. Por exemplo a citação:

> "Combinando a fórmula da posição para aceleração constante e a da velocidade, que é obtida pela derivação da primeira, encontramos a equação de Torricelli:
>
> $$v^2 = v_0^2 + 2as$$
>
> onde v é a velocidade a, a aceleração e s, a variação na posição da partícula..."

refere-se no texto às variáveis 'v', 's' e 'a'. Como são letras do alfabeto não fica evidente no texto que elas se referem às variáveis da equação. Por isso escreveu-se no texto: `onde $v$ \'e a velocidade...` Por outro lado, o *itálico* em modo texto é diferendo do *itálico* do modo matemático que por sua vez é diferente do *deitado*.

****

Para gerar um parágrafo mais "fino" como o que vimos acima na descrição da equação de Torricelli, basta fazer `\narrower` no início deste. Separa bem do resto do texto dando

a entender que é algo distinto. No caso do texto acima usei uma outra opção que acaba produzindo o mesmo efeito mas de forma um pouco mais... digamos, prolixa:

```
$$\vbox{\hsize 4.7 truein
(texto do paragrafo)
}$$
```

****

`\raise` e `\lower` são comandos que permitem você fazer subir ou descer uma letra, ou um texto, numa linha. Na direção horizontal podemos usar o comando `\kern` . Como exemplo, veja como eu defini o logotipo desse manual com base nesses dois comandos:

```
\def\topics{T\kern-.125em \raise .75ex \hbox{\'op}\kern-.1667em \lower
 .5ex\hbox{E}\kern-.125em X\raise .75ex\hbox{s} }
```

Nesse exemplo, `\kern` foi usado para aproximar uma letra da outra, pois o valor da "separação" é negativo. Já vimos o uso de `\lower` na definição dos sinais '$\lesssim$ ' e '$\gtrsim$ '.

****

TEX define previamente o chamado `\footline` que indica o número da página embaixo no centro da linha. O comando `\nopagenumbers` é uma forma de subtrair essa numeração. Ela é incoveniente para casos de cartas e ofícios, por exemplo. Veja como é definido:

```
\def\nopagenumbers{\footline{\hfil}}
```

No lugar da numeração de páginas, para esse manuscrito, eu preferi indicar o nome e a versão:

```
\footline{{\eightrm\topics} \quad {\eightsl V 1.0}\hfill}
```

onde `\eightrm` e `\eightsl` são 'fonts' indicando o tipo romano e o tipo deitado de tamanho $8pt$. Esses 'fonts' foram definidos no inicio do texto. Eles só existem "pré-carregados" no TEX, você precisará definí-los com `\font` .

11. Algumas dicas necessárias

É possível e até provável que apareçam mensagens de erro ao executarmos o TEX. Como qualquer compilador você poderá se deparar com as desagradáveis mensagens com a interrupção da execução e o *prompt*: '?'. O mais aconselhável é ler atentamente a(s) mensagem(ns) e

tentar entender o mais que seja possível. Acontece que muitas vêzes as mensagens são indecifráveis mesmo para os mais experimentados. Não há uma regra do tipo receita de bolo para diagnosticar o problema. Seria muito bom se fosse possível listar aqui uma tabelinha do tipo:

Problema: A tela apagou e o computador não funciona. *Solução*: Verifique se têm luz.

Infelizmente isso não é possível. Por outro lado, é tentando interpretar as "abrobrinhas" que se aprende a manipular o TEX, em particular a "filosofia" TEX. Leve em conta que quando se roda o TEX, ao final, um arquivo com o mesmo nome de seu texto e a "extensão" .LOG é criado com todas as mensagens mostradas na tela. Serve como referência.

Contudo existem erros que se comete especialmente quando se é principiante. Tentarei abranger aqueles que cometi no início de minha "carreira".

a. Se o seu texto for em português, pode aparecer uma mensagem do tipo:

```
This is TeX, Version 2.1 (preloaded format=plain 80.1.1)  1 JAN 1980 00:57
  (PCTeX 2.10, (c)Personal TeX, Inc 1987. S/N 10511)
**teste
(C:\TEXTS\TOPICS\TESTE.TEX
! Please use \mathaccent for accents in math mode.
\'#1->{\accent
                19 #1}
l.1 $x=0 \'e
              quando o problema $
?
```

significará, provavelmente que você esqueceu de colocar um '$' ou '$$' ao final de uma fórmula matemática. Tudo o que vêm depois é considerado modo matemático e coisas do tipo `\'e` não são corretas nesse modo.

b. Se aparecer uma mensagem do tipo:

```
Overfull hbox(badness ...
```

podem ser várias coisas. Você pode ter colocado um texto muito grande dentro de um `\centerline` de maneira que este "estourou" a dimensão de uma linha. Pode ser também que você fez uma tabela por meio de `\halign` larga demais. Outras vezes um texto em outra língua que não o inglês foi colocado dentro de um `\vbox` com um `\hsize` muito pequeno. Lembre-se que o TEX têm dificuldade de separar sílabas no português. Nesse último caso, geralmente o TEX escreve a palavra que fez estourar o `\vbox` . Pode-se tentar forçar o TEX a uma separação particular através dos `\-` dentro da palavra. Mas isso não funciona para todos os casos. Finalmente, verifique se você colocou um texto muito grande dentro de um `\underbar`

. Para seu governo, o TEX não separa palavras dentro de um `\underbar` . Outras podem ser as causas dessa mensagem de erro. Compile você mesmo aquelas que forem aparecendo, depois escreva o seu próprio manual e divulgue.

c. As mensagens do tipo:

```
\Overfull vbox(badness ...
```

podem ser tratadas simplesmente mudando um pouco o valor de `\vsize` . Lembre-se que o TEX prefere unidades do tipo `in` ou `pt` para essas coisas. Outra causa mais grave é você ter construido uma tabela maior que o tamanho da página. O jeito é separar a tabela em duas ou mais.

d. Se você se aventurar a um TEX mais "profundo", cuidado com o `\begingroup` . Você pode se deparar com um

```
Overflow memory. Sorry...
```

Para isso não há solução. Procure outra forma. *Sorry...*

e. O mais comum dos erros é o de datilografia. Frequentemente acontece de um comando ser escrito errado. A consequência mais cômoda é o TEX anunciar:

```
Undefined control sequence: ...
```

o local do erro é apontado pela "quebra" da linha onde ele está. Por exemplo:

```
! Undefined control sequence.
l.1 Coisas est\ato
                    para acontecer...
?
```

você saberá que o comando `\ato` não foi definido.

f. Raramente pode acontecer de aparecer uma mensagem do tipo:

```
! Missing number, treated as zero.
<to be read again>
                    [uma letra qualquer]
l. xxx \nextnumber .....

?
```

Você (ou um colega seu) definiu um comando que está esperando um valor numérico. Às vezes vale a pena mudar o nome desse comando. Como sempre ele é discriminado na própria mensagem de erro. O mais garantido é finalizar o comando com um `\relax` .

g. Uma infinidade de mensagens indecifráveis. Mesmo para um criptologista experimentado. Não se desespere. Saiba que um pequeno erro pode se expandir provocando outros maiores como uma avalanche. Recomendo que escreva o seu texto em "módulos" e rode cada um deles separadamente para ao final executar todos de uma só vez. É mais prático.

h. A mensagem do tipo:

```
Underfull vbox (Badness 10000) ...
```

é apenas um aviso, não comprometendo a execução do TEX. Indica que o TEX não conseguiu ajustar direito os parágrafos na página. Ele avisa que a aparência da página não será tão boa quanto ele queria (se você não fizer questão, esqueça!). A maneira mais prática de evitar isso é introduzir no início do arquivo, antes do texto:

```
\raggedbottom
```

Assim ele vai "espalhar" os parágrafos de maneira a preencher a página de maneira "agradável".

z. Quando o TEX não consegue dar um "jeitinho" para continuar rodando ele pára a execução e dá um *prompt* do tipo:

```
?
```

Você poderá escolher responder com um 'h' e um [Enter]. TEX lhe dará uma pequena explicação do que está acontecendo e algumas sugestões de como corrigir o erro. Um [Enter] tão somente fará continuar a execução, mas, lembre-se que o erro pode se expandir. Um 'x' e [Enter] fará terminar a execução, permitindo que você reflita um pouco sobre tudo o que está acontecendo.

12. Dicas extraordinárias

Na página 182 do TEXbook, D. Knuth nos propõe um exercício delicioso. Como produzir a expressão?

$$2 \uparrow\uparrow k \stackrel{\rm def}{=} \left.2^{2^{2^{\cdot^{\cdot^{\cdot^2}}}}}\right\}k$$

Ele mesmo se encarrega de responder:

```
$$2\uparrow\uparrow k\mathrel{\mathop=^{\rm def}}
  2^{2^{2^{\cdot^{\cdot^{\cdot^2}}}}}
   \vbox{\hbox{$\Big\}\scriptstyle k$}\kern0pt}$$
```

`\mathrel` permite a construção do "sinal de definição".

****

Existem aqueles que necessitam trabalhar com um texto em coluna dupla, a exemplo dos grandes jornais, científicos ou não. LaTeX permite facilmente gerar esse tipo de recurso através de `\twocolumn` . Contudo, LaTeX não é do tipo de processador que "deixa" muita coisa. Um jeito para rodar textos em coluna dupla é fazer:

```
\newdimen\fullhsize
\fullhsize=6.5in \hsize=3.2in
\def\fullline{\hbox to\fullhsize}
\let\lr=L \newbox\leftcolumn
\output={\if L\lr
    \global\setbox\leftcolumn=\columnbox \global\let\lr=R
  \else \doubleformat \global\let\lr=L\fi
  \ifnum\outputpenalty>-20000 \else\dosupereject\fi}
\def\doubleformat{\shipout\vbox{\makeheadline
  \fullline{\box\leftcolumn\hfil\columnbox}
  \makefootline}
 \advancepageno}
\def\columnbox{\leftline{\pagebody}}
```

Essa bateria de comandos fará com que todo o seu texto seja composto em coluna dupla. Veja se reconhece que o tamanho da página será de $6.5in$, cada coluna terá $3.2in$. Esses valores, é claro podem ser modificados. Ao final do texto, por medida de segurança você deverá fazer:

```
\if R\lr \null\vfill\eject\fi
\supereject
\bye
```

antes que o texto tome uma forma esquisita na última página. Além disso, é bom impor `\nopagenumbers` e definir um `\headline` pois caso contrário a numeração sairá só do lado da coluna esquerda. Existe o problema do balanceamento do texto em duas colunas na última página. No formato que foi indicado, quando chegar à última página o texto será todo empurrado para a coluna da esquerda, ficando a direita completamente vazia.

Para criar um "artigo" com o resumo ocupando a linha inteira seguido do texto em duas colunas, o jeito é escrevê-lo dentro de um `\fullline {\vbox {... texto ...}}` contanto que não ultrapasse os limites de uma página.

****

Quem estudou o Espalhamento Rayleigh teve que se acostumar com equações do tipo:

$$\begin{aligned}\left(\frac{1}{\mu'}+\frac{1}{\mu}\right) S_{ll}^{(0)}(\mu,\mu') =& 2\left\{X_l(\mu)X_l(\mu')[1+\nu_4(\mu+\mu')+\mu\mu']-\right.\\ &- Y_l(\mu)Y_l(\mu')[1-\nu_4(\mu+\mu')+\mu\mu']-\\ &\left.-\nu_3(\mu+\mu')[X_l(\mu)Y_l(\mu')+Y_l(\mu)X_l(\mu')]\right\},\end{aligned}$$

Quem tentar escrever essa fórmula em TEX arrisca de se dar mal. É preciso escrevê-la dentro de um `\eqalign` para garantir o alinhamento dos termos no lado direito. Quando abrimos a expressão com um '`\{`' logo na primeira linha, `\eqalign` exigirá que coloquemos um '`\}`' antes do `\cr` . No entanto este deverá vir somente na última linha. Devemos "enganar" o TEX introduzindo um '`\right.`' antes de `\cr` na primeira linha e um '`\left.`' logo no após o '&' da última linha. '`\left.`' e '`\right.`' são delimitadores fantasmas em TEX feitos especialmente para essas ocasiões. Use sem culpa. D. Knuth também é obrigado a usá-los.

Para escrever a fórmula acima eu fiz:

```
$$\eqalign{\left( {1\over\mu^\prime}+{1\over\mu}\right) S^{(0)}_{ll}
(\mu,\mu^\prime) =
 &2 \left\{ X_l(\mu) X_l(\mu^\prime)
[1+\nu_4(\mu+\mu^\prime) +\mu\mu^\prime]-\right.\cr
 &-Y_l(\mu)Y_l(\mu^\prime) [1-\nu_4(\mu+\mu^\prime)+
\mu\mu^\prime]-\cr
 &\left. -\nu_3(\mu+\mu^\prime)
[X_l(\mu)Y_l(\mu^\prime)+ Y_l(\mu)X_l(\mu^\prime)] \right\} ,\cr}$$
```

****

Modificar a forma de um parágrafo às vezes é desejável, por exemplo, para introduzir uma figura que concerne apenas a ele. O exemplo abaixo foi produzido com o comando `\parshape` de maneira a produzir o efeito desejado:

```
\parshape 21 0.45in 5.25in 1.0in 4.2in 1.0in 4.2in 1.1in 4.0in 1.2in
3.8in 1.4in 3.4in 1.6in 3.0in 1.9in 2.4in 2.2in 1.8in 2.6in 1.015in
3.0in 0.20in 2.6in 1.015in 2.2in 1.8in 1.9in 2.4in 1.6in 3.0in 1.4in
3.4in 1.2in 3.8in 1.1in 4.0in 1.0in 4.2in 1.0in 4.2in 0.4in 5.6in
\noindent Tempo, tempo, tempo\dots Entidade abstrata mas por certas
vezes t\~ao paup\'avel. O tempo nos leva a vida que nos leva a morte...
```

Tempo, tempo, tempo... Entidade abstrata mas por certas vezes tão paupável. O tempo nos leva a vida que nos leva a morte. O que nos dá a noção do inexorável, do eterno, do imortal. Essa é a angústia do tempo que vivemos e que depois nos eternizamos. O eterno que significa nada mais do que o nada, pois quando se eterniza, deixa-se de ser. Tivesse o homem ignorado o desfilar dos sóis, das luas e entendido que se tratavam das mesmas entidades, e que é o tempo que avança, talvez não perceberíamos que vivemos – e que morremos. É a terrível constatação que a Morte é viúva do Tempo. O tempo que não se sabe se virá, mas que já se foi. Não existe o tempo presente, somente aquele que, talvez, haverá e aquele que já houve. Podemos vizualizar um ponto no espaço porque já vimos um corpo no espaço, uma partícula, um minúsculo grão de poeira. Mas não podemos vizualizar um ponto de tempo porque nunca vimos sequer um corpo no tempo. Somente o nada, o nada, o nada... Danese! Da vida não se leva nada mesmo! É o que se come, o que se bebe, o que se vive, yeh!

O primeiro número, seguindo `\parshape` indica o número de linhas no parágrafo a serem tratadas de maneira especial. Em seguida vem uma sequência de pares de dimensões indicando onde a linha vai começar e a sua extensão. O número de pares não deve ser inferior nem superior ao número indicado no inicio. Se o parágrafo possui mais linhas que o indicado por `\parshape` , a forma das linhas restantes é dada pela forma da última linha indicada.

****

Vimos na seção 10 como a linha inferior da página, indicando o nome e a versão desse "livro", foi gerada. Vejamos agora como foi criada a linha superior, com o nome da seção e a numeração:

```
\headline{\ifodd\pageno\rightheadline\else\leftheadline\fi}
\def\rightheadline{\eightrm \hfil \currentsection \hfil \folio}
\def\leftheadline{\eightrm\folio\hfil \currentsection\hfil}
```

se o número da página (`\pageno`) for ímpar, executa-se o comando `\rightheadline` que coloca a numeração (`\folio`) no canto superior direito. Caso contrário a numeração ficará no canto esquerdo. O nome da seção está guardado na variável `\currentsection` . Vejamos agora como foi que eu defini essa variável. Não se pode exemplificar sem também re-definir o comando `\beginsection` que usei em todo o texto.

```
\def\beginsection #1. #2.
 { \mark{#1\noexpand\else #1} \def\currentsection{#1. #2}
  \sectionbreak
  \leftline{\sectionfont #1. #2}
  \nobreak\smallskip\noindent}
\def\sectionfont{\bf}
\def\sectionbreak{\penalty-200 \vskip18pt plus4pt minus6pt}
\def\currentsection{}
```

`\beginsection` toma dois parâmetros. Ambos terminam com um ponto. Por exemplo, essa seção foi iniciada com o comando:

```
\beginsection 12. Dicas extraordin\'arias.
```

****

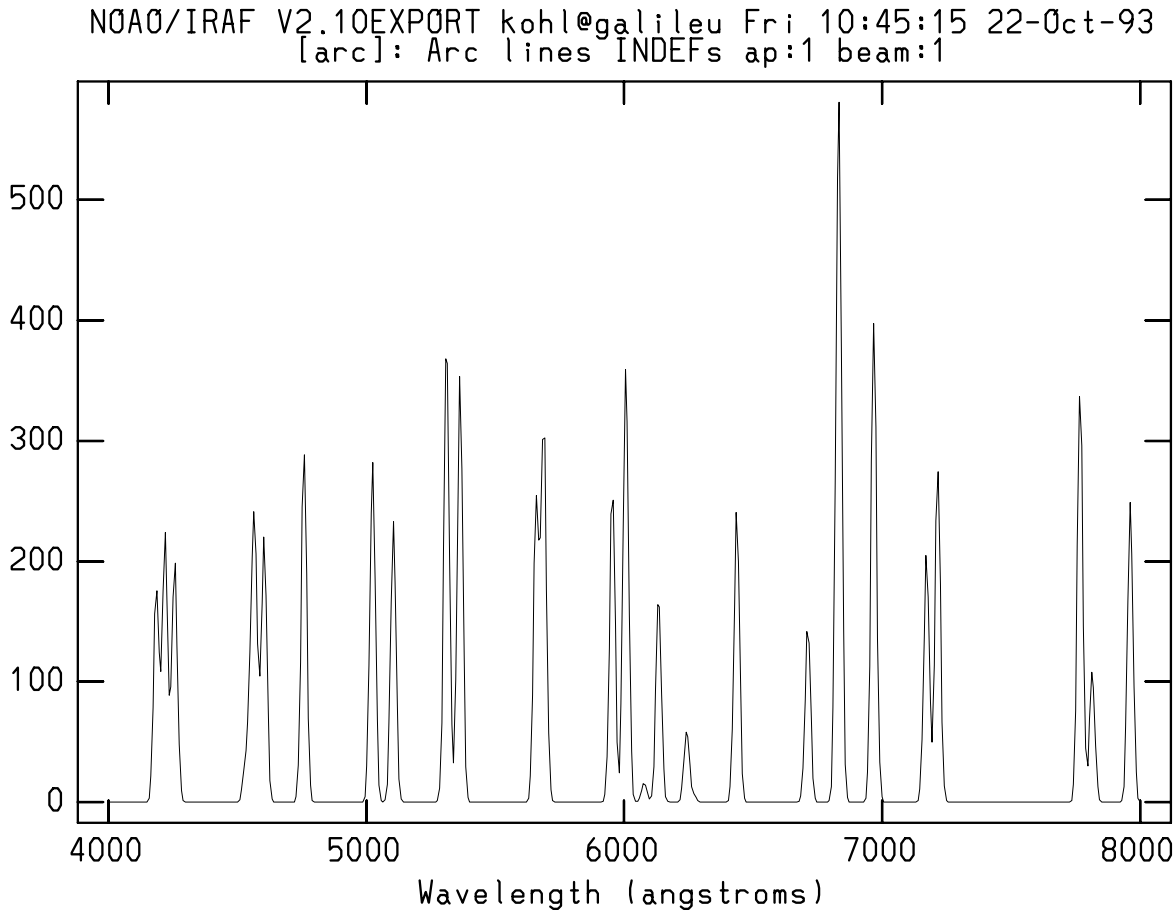
Introduzir figuras propriamente ditas no TEX envolve um processo que depende da impressora e de seu *driver*. No entanto, pode-se deixar espaço especialmente para figuras, tabelas ou efeitos especiais. Uma vez com o espaço obtido, pode-se lançar mão da famosa goma arábica, ou dependendo da impressora, imprimir na mesma página a figura desejada. Veja como foi que defini um comando chamado `\figure` para escrever artigos:

```
\def\figure#1#2{\topinsert
    \vbox{\baselineskip 12pt
      \vskip 6.5 truein
      {\bf Figure #1.} #2}
    \endinsert}
```

No meio do texto basta fazer: `\figure {2.2}{Gr\'afico mostrando a correla\c c\~ao ...}` que será gerado um espaço em branco na parte superior da próxima página em que isso for possível, seguido de uma legenda explicativa da figura. Um espaço será imposto entre essa "figura" e o texto normal de maneira a impedir qualquer confusão. A "figura" ficará na parte superior da página por causa do `\topinsert` . Com `\midinsert` teríamos uma inserção no meio da página e finalmente com `pageinsert` garantimos a inserção de uma página inteira. Uma definição parecida pode ser dedicada para tabelas.

****

Se sua impressora é "PostScript" com um driver do tipo "dvips", você pode capturar uma figura seja em "PostScript", seja em "Encapsulated PostScript". Alguns programas gráficos produzem esse tipo de formato o que dá grande flexibilidade na captura.



Veja, por exemplo, a figura a acima. Ela foi produzida no IRAF com a figura na tela e o comando gráfico:

```
:.snap epsf
```

Assim, criou-se um arquivo chamado "sgi436.eps". No início do texto eu introduzi o comando:

```
\input epsf
```

E para gerar a figura eu fiz:

```
\topinsert
\epsffile{sgi436.eps}
\endinsert
```

Algumas vezes a figura é muito grande e "transborda" pelas margens. Uma forma de se prevenir contra isso é, antes do comando `\epsffile` acrescentar:

```
\epsfxsize=\hsize
```

Isso vai forçar a figura a um valor máximo horizontal do tamanho exato do `\hsize` . Se caso o transborde é vertical então troque o "x" pelo "y" no comando acima. Evidentemente o valor adotado para esses casos é arbirtrário, ou seja, não é preciso que seja forçosamente `\hsize` . Pode ser 50 mm ou 10 pt.

O caso de uma figura do MONGO é um pouco mais complicado. MONGO não gera arquivo tipo ".eps". Trata-se da primeira situação, isto é, de um arquivo PostScript comum. Para isso eu gerei um macro que captura um arquivo PS do MONGO, gerado em "Landscape":

```
% Usage: \topmongo{[file]}{[number]}{[caption]}
\def\topmongofig#1#2#3{%
\topinsert
\special{psfile=#1 voffset=-360 hoffset=480 vscale=60 hscale=60 angle=90}
\vskip 4.8 truein
$$\vbox{\hsize=5.5truein%
\noindent{\bf Figure: #2. }{#3}
}$$
\endinsert
}
```

A maior parte das vezes será necessário mudar os valores dos parâmtros. Com prática a operação torna-se mais fácil. Se você tem legendas e muitos gráficos o melhor é definir um comando análogo ao "`\figure` " acima e introduzir um parâmetro que seria o nome do arquivo.

****

Alguns usuários TEX gostam de "imitar" o formato de suas revistas preferidas. Por exemplo, algumas revistas, como a Astronomy & Astrophysics, gostam de apresentar as equações "encostadas" no lado esquerdo, tal como:

$$\text{Res } f(a) = \frac{1}{2\pi i} \oint_C f(z)\, dz,$$

outras preferem "indentar" as equações, tal como:

$$\text{Res } f(a) = \frac{1}{2\pi i} \oint_C f(z)\, dz,$$

Para produzir as equações acima eu fiz, primeiramente:

```
$$\leftline{$\displaystyle{\rm Res}\;f(a)
= {1\over 2\pi i}\oint_C f(z)\;dz,$}$$
```

e em seguida:

```
$$\leftline{\indent$\displaystyle{\rm Res}\;f(a)
= {1\over 2\pi i}\oint_C f(z)\;dz,$}$$
```

No entanto, existe uma forma mais geral do que estas de maneira a evitar que a cada vez que se escrever uma equação, seja necessário repetir essa bateria de comandos. Além do mais, essa forma de produzir equações não funciona se for preciso numerar. Se fizermos, no início do texto, as definições seguintes:

```
\newif\ifeqno \newif\ifleqno \everydisplay{\displaysetup}
\def\displaysetup#1$${\displaytest#1\eqno\eqno\displaytest}
\def\displaytest#1\eqno#2\eqno#3\displaytest{%
 \if!#2!\ldisplaytest#1\leqno\leqno\ldisplaytest
  \else\eqnotrue\leqnofalse\def\eqn{#2}\def\eq{#1}\fi
  \generaldisplay$$}
 \def\ldisplaytest#1\leqno#2\leqno#3\ldisplaytest{\def\eq{#1}%
  \if!#3!\eqnofalse\else\eqnotrue\leqnotrue\def\eqn{#2}\fi}
\def\generaldisplay{\ifeqno\leftdispleq\else\indentdispeq\fi}
\def\leftdispleq{\leftline{\indent$\displaystyle\eq\hfill\llap\eqn$}}
\def\indentdispeq{\leftline{\indent$\displaystyle\eq$}}
```

a forma de apresentação das equações em evidência (isto é, tudo aquilo que for colocado entre `$$...$$`) será modificada, no caso para "indentada". Se fizermos:

```
$${\rm Res}\;f(a) = {1\over 2\pi i}\oint_C f(z)\;dz \eqno(2.12) $$
```

o resultado será

$$\mathrm{Res}\, f(a) = \frac{1}{2\pi i} \oint_C f(z)\, dz \qquad (2.12)$$

Se você quizer "encostado" na esquerda, ao invés de "indentado", basta retirar o comando `\indent` de `\leftdispleq` e `\indentdispeq` . Há, contudo, uma limitação nessa nova definição de equação em evidência: a numeração do lado esquerdo das equações não vai funcionar. Supõe-se que nesse formato, essa forma de numeração seja "feia".

****

Para quem tem o driver em PostScript, Filipe* passou-me um macro de autor anônimo que pode vir a interessar. Veja abaixo como fica a tabela do "Planalto Inc." (Seção 8) com a ajuda desse macro:

* Filipe de Moraes Paiva, CBPF, Rio, fmpaiva@brlncc.bitnet

Planalto Inc.	
Zélia	demitida
Ibraim	demitido
Antônio	demitido
Marcílio	admitido
Fernando	abandono

Para obter essa tabela "deitada" eu fiz:

```
\input rotate
\vskip \baselineskip
\setbox1=\vbox{\tabskip=0pt \offinterlineskip%
\def\trule{\noalign{\hrule}}
\halign to 150pt{\strut#& \vrule#\tabskip=1em plus1em&
\hfil#\hfil & \vrule# & \hfil#\hfil & \vrule# \tabskip=0pt\cr
\trule
&& \multispan3 \hfil Planalto Inc. \hfil &\cr
\trule
&& Z\'elia && demitida &\cr
&& Ibraim && demitido &\cr
&& Ant\^onio && demitido &\cr
&& Marc\'{\i}lio && admitido &\cr
&& Fernando && abandono &\cr
\trule}}

\rotl1
```

Com `\setbox 1` eu defini um *box* com numeração (1) cujo conteúdo é a nossa tabela. Através de comandos especiais recupera-se as dimensões dessa "caixa" com os quais pode-se trabalhar. Os comandos `\rotl rotr \rotu` e `\rotf` seguidos do número do "box" executam a rotação. Repare que eu introduzi um arquivo chamado "rotate.tex", que é onde esses macros estão definidos. Na próxima página você pode ver uma reprodução desse arquivo.

```
%
%   These macros allow you to rotate or flip a \TeX\ box.  Very useful for
%   sideways tables or upsidedown answers.
%
%   To use, create a box containing the information you want to rotate.
%   (An hbox or vbox will do.)  Now call \rotr\boxnum to rotate the
%   material and create a new box with the appropriate (flipped) dimensions.
%   \rotr rotates right, \rotl rotates left, \rotu turns upside down, and
%   \rotf flips.  These boxes may contain other rotated boxes.
%
\newdimen\rotdimen
\def\vspec#1{\special{ps:#1}}%  passes #1 verbatim to the output
\def\rotstart#1{\vspec{gsave currentpoint currentpoint translate
   #1 neg exch neg exch translate}}% #1 can be any origin-fixing transformation
\def\rotfinish{\vspec{currentpoint grestore moveto}}% gets back in synch
%
%   First, the rotation right. The reference point of the rotated box
%   is the lower right corner of the original box.
%
\def\rotr#1{\rotdimen=\ht#1\advance\rotdimen by\dp#1%
   \hbox to\rotdimen{\hskip\ht#1\vbox to\wd#1{\rotstart{90 rotate}%
   \box#1\vss}\hss}\rotfinish}
%
%   Next, the rotation left. The reference point of the rotated box
%   is the upper left corner of the original box.
%
\def\rotl#1{\rotdimen=\ht#1\advance\rotdimen by\dp#1%
   \hbox to\rotdimen{\vbox to\wd#1{\vskip\wd#1\rotstart{270 rotate}%
   \box#1\vss}\hss}\rotfinish}%
%
%   Upside down is simple. The reference point of the rotated box
%   is the upper right corner of the original box. (The box's height
%   should be the current font's xheight, \fontdimen5\font,
%   if you want that xheight to be at the baseline after rotation.)
%
\def\rotu#1{\rotdimen=\ht#1\advance\rotdimen by\dp#1%
   \hbox to\wd#1{\hskip\wd#1\vbox to\rotdimen{\vskip\rotdimen
   \rotstart{-1 dup scale}\box#1\vss}\hss}\rotfinish}%
%
%   And flipped end for end is pretty ysae too. We retain the baseline.
%
\def\rotf#1{\hbox to\wd#1{\hskip\wd#1\rotstart{-1 1 scale}%
   \box#1\hss}\rotfinish}%
```

13. Exemplos

Alguns exemplos de uso do TEX em aplicações variadas são ilustrativos do que se pode e se deve fazer. D. Knuth descreve alguns formatos possíveis que vão desde cartas até cartazes de concertos de orquestras.

A seguir eu mostro algumas formas de compor texto das maneiras que eu achei mais convenientes, não significando que sejam as melhores. Cada um poderá escolher o seu próprio estilo. O que vêm são apenas idéias.

a) Carta.

O estilo "Carta" é composto de algumas definições necessárias para colocar o cabeçalho, o destinatário, o remetente e o final da carta contendo um "Atenciosamente". Dou um exemplo abaixo e você pode admirar o resultado na página seguinte:

```
\settabs 2\columns
\def\hoje{\number\day ~de \ifcase\month\or
   janeiro\or fevereiro\or mar\c co\or abril\or maio\or
   junho\or julho\or agosto\or setembro\or outubro\or novembro\or dezembro\fi
   ~de \number\year}
\def\depara{\vskip 2\baselineskip}
\def\prez#1{\line{\noindent Prezado #1,\hfil}}
\def\atencio#1{%
\depara
\depara
\rword{Atenciosamente,}\par
\depara
\depara
\rword{$\overline{\hbox{\phantom{xxx}#1\phantom{xxx}}}$}}
\def\rword#1{%
\hskip 0.5\hsize \hbox to 0.4\hsize{\hfil #1 \hfil}}

\+ & Rio de Janeiro, \hoje\cr
\depara
\+ Ant\^onio Pedro de Alc\^antara & \cr
\+ Rua da Matriz, 29 & \cr
\+ 01 256 S\~ao Paulo - SP \cr
\depara
\+ & Jos\'e Dias da Sica \cr
```

```
\+ & Rua Padre C\'esar Firmino, 148 \cr
\+ & 21 110 Rio de Janeiro - RJ \cr

\depara

\prez{Ant\^onio}

\depara

Venho por meio desta solicitar que V.S. me receba em entrevista,
oportunidade esta em que poderei apresentar as minhas intens\~oes com
vistas a um poss\'{\i}vel trabalho de doutorado.

Tenho acompanhado nos \'ultimos anos o trabalho que V.S. tem
desenvolvido na \'area de bioqu\'{\i}mica, em especial no tratamento de
conservantes para alimentos enlatados.

Posso adiantar que eu tamb\'em tenho investido nessas pesquisas, em
particular -- se V.S. notou -- no desenvolvimento do espessante Es7
extraido de esterco de cavalo.

Como V.S. deve saber, esse trabalho n\~ao desembocou em bons
resultados, desconfio, pela necessidade de conhecimento de um
conservante eficaz.


\atencio{Jos\'e Dias da Sica}
```

Inicialmente impus um `\settabs 2\columns` porque eu sei que dessa forma a página fica dividida em 2 colunas de dimensões iguais. Se por um lado a declaração de destinatário não é empurrado pelo espaço de parágrafo (`\indent`), por outro o início de "Rio de ..." e do bloco do rementente ficam alinhados.

O comando `\depara` serve para separar os blocos de 2 vezes o espaçamento vertical. O comando `\prez` é tratamento do início do texto da carta (Prezado...).

Defini o macro `\hoje` de maneira a escrever a data em que escrevi a carta, tomada do relógio do máquina que estou usando. Alguns não gostam de usar esse expediente porque perde-se a data em que a carta foi escrita, se se quiser guardar o texto.

Finalmente dedica-se um comando do "Atenciosamente", com um traço horizontal em cima do nome do remetente. Essa é a oportunidade de "semi-centrar" um texto dentro de um `\hbox` de dimensão fixa. Tanto o "Atenciosamente" quanto o "José ..." ficam alinhados de maneira mais elegante. Eles são empurrados para a esquerda através do artifício `\hskip` dentro do comando `\rword` .

Rio de Janeiro, 31 de setembro de 1993

Antônio Pedro de Alcântara
Rua da Matriz, 29
01 256 São Paulo - SP

José Dias da Sica
Rua Padre César Firmino, 148
21 110 Rio de Janeiro - RJ

Prezado Antônio,

Venho por meio desta solicitar que V.S. me receba em entrevista, oportunidade esta em que poderei apresentar as minhas intensões com vistas a um possível trabalho de doutorado.

Tenho acompanhado nos últimos anos o trabalho que V.S. tem desenvolvido na área de bioquímica, em especial no tratamento de conservantes para alimentos enlatados.

Posso adiantar que eu também tenho investido nessas pesquisas, em particular – se V.S. notou – no desenvolvimento do espessante Es7 extraido de esterco de cavalo.

Como V.S. deve saber, esse trabalho não desembocou em bons resultados, desconfio, pela necessidade de conhecimento de um conservante eficaz.

Atenciosamente,

José Dias da Sica

b) Cartaz.

TEX permite fazer cartazes de seminários, cursos, conferências e congressos. Veja como eu escrevi um arquivo TEX para produzir um poster dos seminários na astronomia:

```
\magnification=\magstep3
\nopagenumbers
\font\cmrmed=cmr10 scaled \magstep5
\font\cmrlit=cmr7 scaled \magstep2
\font\thick=cminch scaled 300
\font\lit=cmsl10 scaled \magstep4
\hsize=6.0 true in
\vsize=8.0 true in
\hoffset=0.0 true cm
\voffset=4.8 true cm
\parskip=12pt
\parindent=12pt
\baselineskip=12pt

{\noindent \thick SEMINARIOS}

\vskip 12pt
\rightline{\lit na astronomia}

\vskip 64pt
\centerline{\bf Augusto Damineli }
\centerline{{\cmrlit IAG -USP }}

\vskip 24pt
\centerline{\sl ``Espectroscopia de Estrelas Quentes no}
\centerline{\sl Infravermelho pr\'oximo'' }

\vskip 64pt
\cmrlit
Data: 4 de julho de 1991 \`as 14:00 hs.

Local: Audit\'orio do Observat\'orio Nacional
\bye
```

O resultado pode ser visto na próxima página.

c) Quadro de controle.

Um computador ou um terminal de uso público necessita de um quadro de controle para que cada usuário marque sua hora. Quando foi necessário fazer dois desses quadros, um para um PC-XT que controlava uma impressora Laser e outro para um PC-AT/386, eu montei um sistema de maneira a mudar somente o nome de cada PC e a semana e fazer sair os dois quadros de uma só vez. Bastava a seguinte sistemática:

```
$tex
**\relax
*\def\semana{11-10-1991 a 16-10-1991}
*\input pclaser
*\input schedul
*\input pc386
*\input schedul
*\bye
```

Aqui eu ignorei as mensagens que TEX dá depois de cada comando. Inicialmente eu chamei `$tex` sem o nome do arquivo de maneira que tanto o arquivo 'DVI' quanto o 'LOG' terão o nome 'TEXPUT'. O *prompt* '**' que o TEX dá indica que ele está esperando automaticamente o nome de um arquivo a ser tratado. Com `\relax` esse *prompt* é abandonado, passando ao '*' simples onde pode-se entrar tanto com um texto como com um comando da maneira que estamos acostumados. Defini um comando chamado `\semana` , que será aproveitado dentro do texto em SCHEDUL.TEX. Dentro de PCLASER.TEX e PC386.TEX estão comandos que definem o nome do computador a sair no quadro. O quadro propriamente dito é construido em SCHEDUL.TEX. Assim, acompanhando a sequência de operações, temos:

`$tex`	chama o TEX com saída em TEXPUT.DVI e TEXPUT.LOG.
`**\relax`	"relaxa" a entrada de arquivo e entra comando.
`\def \semana {11-10-1991 a 16-10-1991}`	define a semana corrente.
`\input pclaser`	entra o arquivo PCLASER.TEX definindo o nome do quadro.
`\input schedul`	entra o arquivo SCHEDUL.TEX que constroi o quadro.
`\input pc386`	entra o arquivo PC386.TEX definindo o nome do outro quadro.

SEMINARIOS

na astronomia

Augusto Damineli
IAG -USP

"Espectroscopia de Estrelas Quentes no Infravermelho próximo"

Data: 4 de julho de 1991 às 14:00 hs.

Local: Auditório do Observatório Nacional

`\input schedul`	entra novamente o SCHEDUL.TEX para construir o segundo quadro.
`\bye`	sai do TEX e guarda os resultados.

O conteúdo dos arquivos são:

PCLASER.TEX:

```
\def\pc{PC-HP LaserJet}
```

PC386.TEX:

```
\def\pc{PC - AT 386}
```

SCHEDUL.TEX:

```
\magnification=\magstep3
\hsize=16.1 true cm
\vsize=25.0 true cm
\hoffset=0.0 true cm
\voffset=-0.5 true in
\parskip=12pt
\parindent=0pt
\baselineskip=11pt
\raggedbottom
\nopagenumbers

\font\bigtenrm=cmr10 scaled \magstep5

\vbox{\tabskip=0pt \offinterlineskip
\def\tablerule{\noalign{\hrule}}
\def\hgrosrule{\noalign{\hrule height3pt}}
\def\vgrosrule{\vrule width3pt}
\halign to 380pt{\strut#& \vgrosrule#\tabskip=1em plus2em&
 # & \vgrosrule#\tabskip=0pt\cr\hgrosrule
&&\strut&\cr
&&\hfil {\bigtenrm \pc }\hfil&\cr
&&\strut&\cr
&&Semana: \semana \hfil&\cr\hgrosrule}}

\vskip 12pt
\vbox{\tabskip=0pt \offinterlineskip
\def\tablerule{\noalign{\hrule}}
```

```
\def\hgrosrule{\noalign{\hrule height3pt}}
\def\vgrosrule{\vrule width3pt}
\def\hmidrule{\noalign{\hrule height1pt}}
\def\dias{
&&8 - 10&&&&&8 - 10&&&&\cr\tablerule
&&10 - 12&&&&&10 - 12&&&&\cr\tablerule
&&12 - 14&&&&&12 - 14&&&&\cr\tablerule
&&14 - 16&&&&&14 - 16&&&&\cr\tablerule
&&16 - 18&&&&&16 - 18&&&&\cr
}
\halign to 380pt{\strut#& \vgrosrule#\tabskip=1.0em&
  \hfil# & \vrule#\tabskip 1em plus2em& \hfil#\hfil&
  \hfil#& \vgrosrule#\tabskip=1.0em&\hfil#&\vrule#\tabskip 1em plus2em&
\hfil#\hfil&
  \hfil#& \vgrosrule#\tabskip=0pt\cr\hgrosrule
&&\multispan4 \hfil Segunda\hfil&&\multispan4 \hfil Ter\c ca\hfil&\cr\hmidrule
\dias
\hgrosrule
&&\multispan4 \hfil Quarta\hfil&&\multispan4 \hfil Quinta\hfil&\cr\hmidrule
\dias
\hgrosrule
&&\multispan4 \hfil Sexta\hfil&&\multispan4 \hfil S\'abado\hfil&\cr\hmidrule
\dias
\hgrosrule
}}
\eject
```

Na próxima página você poderá admirar o exemplo de um quadro para o PCLASER.

d) Tabela.

A listagem abaixo é um exemplo do que se pode fazer com relação a tabelas de estrutura complexa. Pode-se colocar *boxes* dentro de *boxes* e dentro deles, textos. São usados diferentes sistemas de unidades, textos que cobrem todas as colunas revezando com colunas que por sua vez contém um parágrafo inteiro. Você pode ir tentando diversas formas de fazer esse tipo de tabela. Muitas vezes acontece de haver aparente incompatibilidade de dimensões que devem ser corrigidas, pelo menos no início, através de tentativa e erro.

```
\vbox {\hoffset=5pt \tabskip=0pt \offinterlineskip \parindent=12pt
```

PC-HP LaserJet

Semana: 21-10-91 a 28-10-91

Segunda		Terça	
8 - 10		8 - 10	
10 - 12		10 - 12	
12 - 14		12 - 14	
14 - 16		14 - 16	
16 - 18		16 - 18	
Quarta		**Quinta**	
8 - 10		8 - 10	
10 - 12		10 - 12	
12 - 14		12 - 14	
14 - 16		14 - 16	
16 - 18		16 - 18	
Sexta		**Sábado**	
8 - 10		8 - 10	
10 - 12		10 - 12	
12 - 14		12 - 14	
14 - 16		14 - 16	
16 - 18		16 - 18	

```
\def\tablerule {\noalign{\hrule}}
\halign to 514.5 truept{\strut#&\vrule#& \hfil\hbox{\kern 0.1
truein\vbox to 5.0 truecm{\hsize 3.0 true in
\noindent\strut#\strut\vfill}\kern 0.45 truein}\hfil & \vrule#&
\hfil\hbox{\kern 0.1 truein\vbox to 5.0 truecm{\hsize 3.0 true
in\noindent\strut#\strut\vfill}\kern
0.45truein}\hfil & \vrule# \tabskip=0pt\cr \tablerule
&& \item {}EST\'AGIOS LINGU\' ISTICOS \item{--} Comunica\c c\~ao
pr\'e-lingu\' istica por cho\-ro, \item{~~} gestos. \item{--}
Est\'agio holofr\'astico--uso da pa\-la\-vra - fra\-se. \vfil
&& \item {}EST\'AGIOS FONOL\'OGICOS \item {}Est\'agio pr\'e-verbal
\item {--} Vocaliza\c c\~ao pr\'e-lingu\' istica e per\-cep\-\c c\~ao
\item {~~} (0--1 ano) \item {--} Fonologia das 50 primeiras
pa\-la\-vras \item{} (1--1 ano e meio).
&\cr\tablerule
&& \multispan 3 \hfil\hbox{\kern 0.1truein\vbox to 5.0 truecm{\hsize
6.5 truein\noindent \strut \item {} COGNI\c C\~AO: -- Est\'agios de
Piaget \item {$\bullet$} Per\' {\i}odo pr\'e-operat\'orio (1 ano e
meio -- 12 anos) \item {--} Subper\' {\i}odo pr\'e-conceitual (1 ano
e meio -- 4 anos) \item {}Nascimento da representa\c c\~ao
simb\'olica.  A cri\-an\-\c ca refere-se agora ao passado e futuro,
ainda que quase toda sua atividade esteja no presente.
Predomin\^ancia do jogo simb\'olico.\vfill}\kern 0.45truein}\hfil
&\cr\tablerule
&& \item {}Est\'agio telegr\'afico \item {} Come\c ca o uso de
palavras em com\-bi\-na\-\c c\~oes. \item {} Aumenta de 3 a 4
com\-bi\-na\-\c c\~oes na mai\-o\-ria das fra\-ses, quan\-do chega
bem per\-to da \'e\-po\-ca da frase simples, bem formadas.
&& \item {}Fonologia do fonema simples \item {} Expande invent\'ario
dos sons lin\-gu\' {\i}s\-ti\-cos.  O processo fonol\'ogico
resultante ainda \'e incorreto na produ\c c\~ao a\-t\'e os 4 anos,
quan\-do as pa\-la\-vras de es\-tru\-tu\-ra mor\-fo\-l\'o\-gi\-ca
sim\-ples s\~ao be\-m reduzidas.
&\cr\tablerule
&& \multispan 3 \hfil\hbox{\kern 0.07truein\vbox to 6.0 truecm{\hsize
6.5 truein\noindent \strut \item {}EST\'AGIOS LINGU\' ISTICOS \item
{}COGNI\c C\~AO: \item {}Est\'agios de Piaget \item {$\bullet$}
```

```
Subper\' {\i}odo intuitivo (4--7 anos) \item {} A cri\-an\-\c ca
repousa na sua percep\c c\~ao imediata para resolver problemas.
Co\-me\-\c ca a desenvolver o conceito de reversibilidade.  Come\c ca
a ser envolvida nos jogos sociais.\vfill}\kern 0.07truein}\hfil
&\cr\tablerule \cr}} \vfill
```

ESTÁGIOS LINGUÍSTICOS

– Comunicação pré-linguística por choro, gestos.
– Estágio holofrástico–uso da palavra - frase.

ESTÁGIOS FONOLÓGICOS

Estágio pré-verbal

– Vocalização pré-linguística e percepção (0–1 ano)
– Fonologia das 50 primeiras palavras (1–1 ano e meio).

COGNIÇÃO: – Estágios de Piaget

• Período pré-operatório (1 ano e meio – 12 anos)

– Subperíodo pré-conceitual (1 ano e meio – 4 anos)

Nascimento da representação simbólica. A criança refere-se agora ao passado e futuro, ainda que quase toda sua atividade esteja no presente. Predominância do jogo simbólico.

Estágio telegráfico

Começa o uso de palavras em combinações.

Aumenta de 3 a 4 combinações na maioria das frases, quando chega bem perto da época da frase simples, bem formadas.

Fonologia do fonema simples

Expande inventário dos sons linguísticos. O processo fonológico resultante ainda é incorreto na produção até os 4 anos, quando as palavras de estrutura morfológica simples são bem reduzidas.

ESTÁGIOS LINGUÍSTICOS

COGNIÇÃO:

Estágios de Piaget

• Subperíodo intuitivo (4–7 anos)

A criança repousa na sua percepção imediata para resolver problemas. Começa a desenvolver o conceito de reversibilidade. Começa a ser envolvida nos jogos sociais.

Referências:

1. The TEXbook, Donald E. Knuth, Ed. Addison Wesley Publishing Company, 1989. Leitura quase que obrigatória se você quer ser um profundo TEXnólogo.

2. Além do TEXbook, poucas são as referências adicionais. Como leitura introdutória existe também uma "INTRODUÇÃO AO SISTEMA TEX" de Curt Roloff e Carlos Bertulani, publicação do Instituto de Física da UFRJ. Outra introdução é a de Filipe de Moraes Paiva: "Notas introdutórias sobre LaTeX", 1993, Notas Técnicas do CBPF. O que existe são publicações internas nos laboratórios e que não chegam ao grande público.

3. Por US$ 60.00 você pode se juntar ao TUG (TEXUsers Group) e receber informações atualizadas ou ter direito a consulta via "e-mail".

TEXUsers Group
P.O. Box 869
Santa Barbara, CA 93102
E.U.A.
e-mail: tug@math.ams.org
Tel.: 1 805 963-1338
FAX: 1 805 963-8358

A. Índice Remissivo