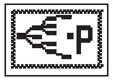
```
\end{document}
```

Now in a frame 

2.5 Graphs

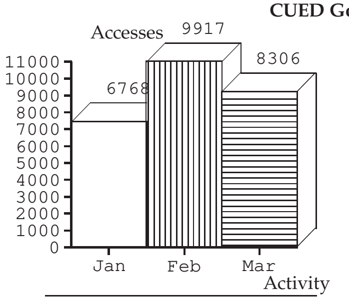
Probably the easiest way to put complicated graphs into LaTeX is to use `gnuplot` or Matlab[17], and produce encapsulated postscript output.

The `ebar` package (a local version of the `bar` package) is ok for simpler graphs. See *The LaTeX Companion* for more details.

```
\begin{center}
\textbf{CUED Gopher Usage, 1994}

\begin{figure}[htbp]
\begin{barenv}
% set the 'z' axis depth
\setdepth{10}
%Scale the height
\setstretch{.01}
% Put the numbers above the bars. Other options include 'empty',
\setnumberpos{up}
% Set type of x labels (numbers by default) but could also be 'days'
\setxvaluetyp{month}
% Set the x origin, end, and step size
\setxaxis{1}{3}{1}\setxname{Activity}
% Set the y origin, end, and step size
\setyaxis{0}{10000}{1000}\setyname{Accesses}
\ebar{6768}{1}
\ebar{9917}{2}
\ebar{8306}{3}
\end{barenv}
\end{figure}
\end{center}
```

CUED Gopher Usage, 1994



[17] http://www-h.eng.cam.ac.uk/help/tpl/programs/matlab.html

LaTeX maths and graphics

Tim Love

September 11, 1997

This handout assumes that you have already read the Advanced LaTeX[1] document handout, so if you're unsure about 'environments', read no further. LaTeX produces maths text well, but graphic support is less good. Fortunately there's a graphics editor called `xfig` whose output can be used in a LaTeX file.

Note that there's an alternative way of producing maths in LaTeX - $\mathcal{AMS}$-LaTeX. See the online manual[2] for details.

Comments and bug reports to Tim Love (tpl@eng.cam.ac.uk).

Contents

[1] http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/latex_advanced/latex_advanced.html
[2] http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/amslatex.dvi

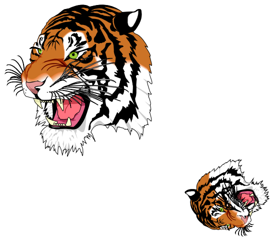


Figure 2: Tigers

```
\begin{figure}[hbtp]
\resizebox{!}{40mm}{\includegraphics{/export/ghostfonts/tiger.eps}}
\rotatebox{120}
{\resizebox{!}{20mm}{\includegraphics{/export/ghostfonts/tiger.eps}}}
\caption{Tigers}
\end{figure}

% remember to do \usepackage{subfigure} at the top of the document!
\begin{figure}[hbtp]
\begin{center}
% an mbox for alignment
\mbox{\subfigure[Small]
{\resizebox{!}{30mm}{\includegraphics{/export/ghostfonts/crest.eps}}}\quad
\subfigure[Medium]
{\resizebox{!}{40mm}{\includegraphics{/opt/latex/ps/color_CU_arms.ps}}}\quad
\subfigure[Large]
{\resizebox{!}{50mm}{\includegraphics{/export/ghostfonts/crest.eps}}}
}% end of mbox
\caption{3 crests}
\end{center}
\end{figure}
```

The easiest way to clip the postscript image is to use the `graphicx`[16] package instead of `graphics`. The functionality of the 2 packages is essentially equivalent, but the command style is different. The following fragment displays only part of the image. The viewport coordinates are in the same units as the bounding box.

```
% Use the graphicx package
\begin{figure}[htbn]
\includegraphics[viewport=200 400 400 600,width=5cm,clip]%
{/opt/latex/ps/color_CU_arms.ps}
\end{figure}

% Use the wrapfig package
\begin{wrapfigure}{l}{2cm}
\resizebox{2cm}{!}{\includegraphics{/export/ghostfonts/crest.eps}}
\end{wrapfigure}
```

[16]http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/graphicx.dvi


```
\begin{eqnarray}
a1 & = & b1 + c1\nonumber\\
a2 & = & b2 - c2
\end{eqnarray}
```

$$\begin{aligned} a1 &= b1+c1 \\ a2 &= b2-c2 \qquad (2) \end{aligned}$$

Maths in these 2 sorts of environments have different default sizes for some characters and other behavioural differences so that a line of maths won't impinge on text lines below or above. If you want to put some non-maths text in amongst maths then enclose it in an `\mbox{...}`

1.2 Special Characters

Greek

α	`\alpha`	β	`\beta`	γ	`\gamma`	δ	`\delta`	ϵ	`\epsilon`
ζ	`\zeta`	η	`\eta`	θ	`\theta`	ι	`\iota`	κ	`\kappa`
λ	`\lambda`	μ	`\mu`	ν	`\nu`	ξ	`\xi`	o	`o`
π	`\pi`	ρ	`\rho`	σ	`\sigma`	τ	`\tau`	υ	`\upsilon`
ϕ	`\phi`	χ	`\chi`	ψ	`\psi`	ω	`\omega`	Γ	`\Gamma`
Δ	`\Delta`	Θ	`\Theta`	Λ	`\Lambda`	Ξ	`\Xi`	Π	`\Pi`
Σ	`\Sigma`	Υ	`\Upsilon`	Φ	`\Phi`	Ψ	`\Psi`	Ω	`\Omega`

Miscellaneous

$\ldots$	`\ldots`	$\cdots$	`\cdots`	$\vdots$	`\vdots`
$\ddots$	`\ddots`	$\pm$	`\pm`	$\mp$	`\mp`
$\times$	`\times`	$\div$	`\div`	$\ast$	`\ast`
$\star$	`\star`	$\circ$	`\circ`	$\bullet$	`\bullet`
$\cdot$	`\cdot`	$\cap$	`\cap`	$\bigcap$	`\bigcap`
$\cup$	`\cup`	$\bigcup$	`\bigcup`	$\uplus$	`\uplus`
$\biguplus$	`\biguplus`	$\sqcap$	`\sqcap`	$\sqcup$	`\sqcup`
$\bigsqcup$	`\bigsqcup`	$\vee$	`\vee`	$\bigvee$	`\bigvee`
$\wedge$	`\wedge`	$\bigwedge$	`\bigwedge`	$\setminus$	`\setminus`
$\wr$	`\wr`	$\diamond$	`\diamond`	$\bigtriangleup$	`\bigtriangleup`
$\bigtriangledown$	`\bigtriangledown`	$\triangleleft$	`\triangleleft`	$\triangleright$	`\triangleright`
$\oplus$	`\oplus`	$\bigoplus$	`\bigoplus`	$\ominus$	`\ominus`
$\otimes$	`\otimes`	$\bigotimes$	`\bigotimes`	$\oslash$	`\oslash`
$\odot$	`\odot`	$\bigodot$	`\bigodot`	$\bigcirc$	`\bigcirc`
$\amalg$	`\amalg`	$\leq$	`\leq`	$\prec$	`\prec`
$\preceq$	`\preceq`	$\ll$	`\ll`	$\subset$	`\subset`
$\subseteq$	`\subseteq`	$\sqsubseteq$	`\sqsubseteq`	$\in$	`\in`
$\vdash$	`\vdash`	$\geq$	`\geq`	$\succ$	`\succ`
$\succeq$	`\succeq`	$\gg$	`\gg`	$\supset$	`\supset`
$\supseteq$	`\supseteq`	$\sqsupseteq$	`\sqsupseteq`	$\ni$	`\ni`
$\dashv$	`\dashv`	$\equiv$	`\equiv`	$\sim$	`\sim`
$\simeq$	`\simeq`	$\asymp$	`\asymp`	$\approx$	`\approx`
$\cong$	`\cong`	$\neq$	`\neq`	$\doteq$	`\doteq`

```
\Deriveszlm
```

gives

$\overset{*}{\underset{\mathrm{lm}}{\Longrightarrow}}$

2 Graphics

LaTeX has a `picture` environment in which pictures can be drawn, but you'll find graph paper handy. `xfig` can create code for the `picture` environment but the resulting graphics still suffer several limitations: only certain slopes and circles can be reproduced. The best method presently available is to use `xfig` to produce postscript files, which have no such limitations, but require a postscript printer or equivalent.

Whatever graphics you want to add, you should use the `figure` environment so that LaTeX can cope sensibly with situations where, for example, you attempt to insert near the bottom of a page a graphic that's half a page high. The `figure` environment will *float* the graphic to the top or bottom of the page, or on the next page, with preferences that you can provide.

- `h` here
- `t` top of page
- `b` bottom of page
- `p` on a page with no text

Putting `!` as the first argument in the square brackets will encourage LaTeX to do what you say, even if the result's sub-optimal. See the online hints about floats in LaTeX[11] for further details.

```
\begin{figure}[htbp]
   \vspace{0.5in}
   \caption{0.5 inch of space}
\end{figure}
```

Figure 1: 0.5 inch of space

It's possible to have more than one graphic in a *figure*. See the example later on.

2.1 Postscript

It's easy to incorporate Postscript files as long as they have a proper bounding box comment; *i.e.* LaTeX requires full Encapsulated Postscript[12] as produced by (for example) `xv` and `xfig` on the `HP`s. If the file hasn't got a `BoundingBox` line near the top, you can use `ps2epsi` to generate one. Wherever the postscript file comes from, simply use

```
\documentclass[dvips]{article}
\usepackage{graphics}
```

then include the postscript file using the following commands

[11] http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/float_hint.html
[12] http://www-h.eng.cam.ac.uk/help/tpl/graphics/postscript.html


```
$ a \stackrel{def}{=} b + c $
```

gives $a \stackrel{def}{=} b + c$

See the *Macros* section for how to stack characters using `atop`.

Common functions

In a maths environment, LaTeX assumes that variables will have single-character names. Function names require special treatment. The advantage of using the following control sequences for common functions is that the text will not be put in math italic and subscripts/superscripts will be made into limits where appropriate.

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

1.4 Subscripts and superscripts

These are introduced by the '^' and '_' characters. Depending on the base character and the current style, the sub- or superscripts may go to the right of or directly above/below the main character. With letters it goes to the right.

```
$F_2^3$
```

produces 'F_2^3'. Note that the sub- and superscripts aren't in line. To make them so, you can add an invisible character after the 'F'. `$F{}_2^3$` produces $F{}_2^3$.

With $\sum$ the default behaviour is different in display and text styles.

```
$\sum_{i=0}^2 $
```

produces $\sum_{i=0}^2$ (text style) but

```
\[\sum_{i=0}^2 \]
```

produces (in display style)

$$\sum_{i=0}^2$$

This default behaviour can be overridden, if you really need to. For example in text mode,

```
$\sum\limits_{i=0}^2$
```

produces $\sum\limits_{i=0}^2$

1.5 Overlining, underlining and bold characters

```
${\underline{one}} {\overline{two}}$
```

produces ${\underline{one}} {\overline{two}}$. This is not a useful facility if it's used more than once on a line. The lines are produced so that they don't quite overlap the text; lines over or under different words won't in general be at the same height.

`\mathbf{}` can be used to create bold characters. `$\mathbf{F}_2^3$` produces $\mathbf{F}_2^3$.

To be able to reproduce bold greek characters you can try the `amsbsy` package then use something like `$\pmb{\Gamma}$`. Alternatively, you can use the following idea

gives

$$\begin{aligned} y &= a+b \\ &\quad +k \end{aligned} \tag{5}$$

You can use the `\lefteqn` construction to format long expressions so that continuation lines are differently indented.

```
\begin{eqnarray}
\lefteqn{x+ iy=}\\
 & & a + b + c + d + e + f + g + h + i + j + k +\nonumber\\
 & & l + m \nonumber
\end{eqnarray}
```

$$\begin{aligned} &x + iy = \\ &\quad a+b+c+d+e+f+g+h+i+j+k+ \\ &\quad l+m \end{aligned} \tag{6}$$

If you want more vertical spacing around a line you can create an invisible vertical "struct" in LaTeX. `\rule[-.3cm]{0cm}{1cm}` creates a box of width 0, height 1cm which starts .3cm below the usual line base. By adjusting these values you should be able to create as much extra space below/above the maths as you like. Here's an example

1.14 Maths and Postscript fonts

It's easy to use a postscript font (like helvetica) for the text of a LaTeX document. What's harder is using the same font for maths. An easy, reasonable option is to use the `mathptm` package to put the maths into the postscript *Times* font.

1.15 Matlab and LaTeX

Matlab[10] has some support for LaTeX production. For example

```
latex('(sin(x)+2*x+3*x^2)/(5*x+6*x^2)','math.tex')
```

puts the LaTeX representation of the expression into a file called `math.tex`. Type "`help latex`" inside matlab for details.

1.16 Examples

- ```
  \begin{equation}
  \hat{\theta}_{w_i} = \hat{\theta}(s(t,{\cal U}_{w_i})).
  \end{equation}
  ```

  gives

  $$\hat{\theta}_{w_i} = \hat{\theta}(s(t,\mathcal{U}_{w_i})). \tag{7}$$

- ```
  \begin{eqnarray}
  {\cal M}^2(\hat{\theta},\theta) &=& E[(\hat{\theta} - \theta)^2]
  \nonumber \\
  {\cal M}^2(\hat{\theta},\theta) &=& {\rm var}^2(\hat{\theta}) +
  ```

[10] http://www-h.eng.cam.ac.uk/help/tpl/programs/matlab.html

gives $\left\{2\Big(x(3+y)\Big)\right\}$. If you're not using the default text size these commands might not work correctly. In that case try the `exscale` package.

It's preferable to let LaTeX choose the delimiter size for you by using `\left` and `\right`. These will produce delimiters just big enough for the formulae inbetween.

```
$\left( frac{(x+iy)}{\{\int x\}} \right)$
```

gives $\left(\frac{(x+iy)}{\{\int x\}}\right)$

The left and right delimiters needn't be the same type. It's sometimes useful to make one of them invisible

```
\[ z = \left\{
             \begin{array}{ll}
                  1 & (x>0)\\
                  0 & (x<0)
             \end{array}
       \right.
\]
```

produces

$$z = \left\{ \begin{array}{ll} 1 & (x>0) \\ 0 & (x<0) \end{array} \right.$$

Over- and underbracing works too.

```
$\overbrace{\alpha \ldots \omega}^{\mbox{greek}}
 \underbrace{a \ldots z}{_\mbox{english}}$
```

produces $\overbrace{\alpha\ldots\omega}^{\mbox{greek}} \underbrace{a\ldots z}_{\mbox{english}}$. The use of `\mbox` stops the text appearing in math italic.

1.9 Numbering and labelling

Numbering happening automatically when you display equations. If you *don't* want an equation numbered, use `\nonumber` beside the equation. Equation numbers appear to the right of the maths by default. To make them appear on the left use the `leqno` class option (i.e., use `\documentclass[leqno,....]{....}`).

Use `\label{}` to label an equation (or figure, section etc) in order to reference from elsewhere.

```
\begin{equation}
W_{\bf S}(t,\omega) = \int\limits_{-\infty}^{\infty} {
  {\cal R}_{\bf S}(t,\tau) e^{-j\omega\tau} \,d \tau }
\label{LABELLING}
\end{equation}
```

$$W_{\mathbf{S}}(t,\omega) = \int\limits_{-\infty}^{\infty} \mathcal{R}_{\mathbf{S}}(t,\tau) e^{-j\omega\tau} \,d\tau \tag{3}$$

Now the following text

```
refers back to equation \ref{LABELLING}
```