LaTeX maths and graphics

Tim Love

March 26, 2002

This handout assumes that you have already read the *Advanced LaTeX* handout, so if you're unsure about 'environments', read no further. It heavily uses LaTeX 2_ε features, though the maths section will be useful to LaTeX 2.09 users. 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 - AMS-LaTeX. Type `handout amstex` for details.

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

Contents

1 Maths

There's more to maths typesetting than meets the eye. Many conventions used in the typesetting of plain text are inappropriate to maths. LaTeX goes a long way to help you along with the style. For example, in a LaTeX maths environment, letters come out in *italics*, `-` as '$-$' (minus) instead of the usual '-' (dash), '*' becomes $*$, `'` becomes $'$ and spacing is changed (little around '/', more around '+').

Many of the usual LaTeX constructions can still be used in maths environments but their effect may be slightly different; eg `\textbf{ }` only affects letters and numbers. '`{`' and '`}`' are still special characters; they're used to group characters.

As usual in LaTeX you can override the defaults, but think before doing it: maths support in LaTeX has been carefully thought out and is quite logical though the LaTeX source text may not be very readable. It's a good idea to write out the formulae on paper before you start LaTeXing, and try not to overdo the use of the '`\frac`' construction; use '/' instead.

1.1 Environments

There are 2 environments to display one-line equations.

equation:- Equations in this environment are numbered.

```
\begin{equation}
 x + iy
\end{equation}
```

$$x + iy \tag{1}$$

displaymath:- These won't be numbered. `\[, \]` can be used as abbreviations for `\begin{displaymath}` and `\end{displaymath}`.

```
\begin{displaymath}
 x + iy
\end{displaymath}
```

$$x + iy$$

Never leave a blank line before these equations; it starts a new paragraph and looks ugly. '`\displaystyle`' is the font type used to print maths in these *display* environments. Other relevant environments are:-

math:- For use in text. `\(` and `\)` can be used to delimit the environment, as can the TeX constructions `$` and `$` . For example, `$x=y^2$` gives $x = y^2$.

eqnarray:- This is like a 3 column tabular environment. Each line by default is numbered. You can use the `eqnarray*` variant to suppress numbering altogether.

```
\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`
$\propto$	`\propto`	$\models$	`\models`	$\perp$	`\perp`
$\mid$	`\mid`	$\parallel$	`\parallel`	$\bowtie$	`\bowtie`
$\smile$	`\smile`	$\frown$	`\frown`	$\aleph$	`\aleph`
$\hbar$	`\hbar`	$\imath$	`\imath`	$\jmath$	`\jmath`
ℓ	`\ell`	$\wp$	`\wp`	$\Re$	`\Re`
$\Im$	`\Im`	$\prime$	`\prime`		`\empty`

∇	`\nabla`	$\surd$	`\surd`	$\top$	`\top`
$\bot$	`\bot`	$\Vert$	`|`	$\angle$	`\angle`
$\forall$	`\forall`	$\exists$	`\exists`	$\neg$	`\neg`
$\flat$	`\flat`	$\natural$	`\natural`	$\sharp$	`\sharp`
$\backslash$	`\backslash`	∂	`\partial`	∞	`\infty`
$\triangle$	`\triangle`	$\sum$	`\sum`	$\prod$	`\prod`
$\coprod$	`\coprod`	$\int$	`\int`	$\oint$	`\oint`

1.3 Arrows

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

Calligraphic

These characters are available if you use the `\cal` control sequence.

```
${\cal A B C D E F G H I J K L M N O P Q R S T U V W X Y Z}$
```

gives ${\cal A B C D E F G H I J K L M N O P Q R S T U V W X Y Z}$

Character Modifiers

`\hat{e}`	$\hat{e}$	`\widehat{easy}`	$\widehat{easy}$
`\tilde{e}`	$\tilde{e}$	`\widetilde{easy}`	$\widetilde{easy}$
`\check{e}`	$\check{e}$	`\breve{e}`	$\breve{e}$
`\acute{e}`	$\acute{e}$	`\grave{e}`	$\grave{e}$
`\bar{e}`	$\bar{e}$	`\vec{e}`	$\vec{e}$
`\dot{e}`	$\dot{e}$	`\ddot{e}`	$\ddot{e}$
`\not e`	$\not e$		

Note that the wide versions of `hat` and `tilde` cannot produce very wide alternatives. The '`\not`' operator hasn't properly cut the following letter. The *Fine Tuning* section on page 8 describes how to adjust this.

If you want to place one character above another, you can use `\stackrel`, which prints its first argument in small type immediately above the second

```
$ 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 $\textstyle\sum_{i=0}^2$ (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.

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

1.6 Roots

```
$\sqrt{4} + \sqrt[3]{x + y}$
```

gives $\sqrt{4} + \sqrt[3]{x + y}$.

1.7 Fractions

Three constructions for putting expressions above others are

frac:- `$\frac{1}{(x + 3)}$` produces $\frac{1}{(x+3)}$.

choose:- `${n + 1 \choose 3}$` produces ${n+1 \choose 3}$.

atop:- `${x \atop y}$` produces ${x \atop y}$.

These constructions can be used with ones described earlier. Eg,

```
\[ \sum_{-1\le i \le 1 \atop 0 < j < \infty} f(i,j)\]
```

gives

$$\sum_{-1\le i \le 1 \atop 0 < j < \infty} f(i,j)$$

1.8 Delimiters

these	are made by these	*and these*	are made by these
$($	`(`	$)$	`)`
$[$	`[`	$]$	`]`
$\{$	`\{`	$\}$	`\}`
$\lfloor$	`\lfloor`	$\rfloor$	`\rfloor`
$\lceil$	`\lceil`	$\rceil$	`\rceil`
$\langle$	`\langle`	$\rangle$	`\rangle`
$/$	`/`	$\backslash$	`\backslash`
$\vert$	`|`	$\Vert$	`|`
$\uparrow$	`\uparrow`	$\Uparrow$	`\Uparrow`
$\downarrow$	`\downarrow`	$\Downarrow$	`\Downarrow`
$\updownarrow$	`\updownarrow`	$\Updownarrow$	`\Updownarrow`

This table shows the standard sizes. To get bigger sizes, use these prefices

(for left delimiters)	(for right delimiters)	magnification
`\bigl`	`\bigr`	a bit bigger, but won't overlap lines
`\Bigl`	`\Bigr`	150% times `big`
`\biggl`	`\biggr`	200% times `big`
`\Biggl`	`\Biggr`	250% times `big`

For example,

```
$\Biggl\{2\Bigl(x(3+y)\Bigr)\Biggr\}$
```

gives $\Biggl\{2\Bigl(x(3+y)\Bigr)\Biggr\}$.

You can 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.

Use `\label{}` to label an equation (or figures, 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}
```

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

Now the following text

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

refers back to equation 3 by number, and

```
 refers back to the equation on page \pageref{LABELLING}
```

refers back to the equation on page 7.

A file will have to be LaTeX'ed twice before the references, both forwards and backwards, will be correctly produced.

1.10 Matrices

The `array` environment is like LaTeX's `tabular` environment except that each element is in `math` mode. The default style used is text style but you can override this by using `\displaystyle`.

```
\begin{math}
\begin{array}{clrr} %
      a+b+c & uv & x-y & 27 \\
       x+y  & w  & +z  & 363
\end{array}
\end{math}
```

produces $\begin{array}{clrr} a+b+c & uv & x-y & 27 \\ x+y & w & +z & 363 \end{array}$

The rows are arranged so that their centres are aligned. You can align their tops or bottoms instead by using a further argument when you create the array.

```
\begin{array}{clrr}[t]
```

would produce top-aligned lines, and `'[b]'` would produce bottom-aligned ones. The *Delimiters* section of this document shows how to bracket matrices.

TEX has a few maths facilities not mentioned in the LATEX book. The following TEX construction might be useful.

```
\begin{math}
\bordermatrix{&a_1&a_2&...&a_n\cr
          b_1 & 1.2  & 3.3  & 5.1  & 2.8  \cr
          c_1 & 4.7  & 7.8  & 2.4  & 1.9  \cr
          ... & ...  & ...  & ...  & ...  \cr
          z_1 & 8.0  & 9.9  & 0.9  & 9.99  \cr}
\end{math}
```

$$\begin{array}{cc} & \begin{array}{cccc} a_1 & a_2 & ... & a_n \end{array} \\ \begin{array}{c} b_1 \\ c_1 \\ ... \\ z_1 \end{array} & \begin{pmatrix} 1.2 & 3.3 & 5.1 & 2.8 \\ 4.7 & 7.8 & 2.4 & 1.9 \\ ... & ... & ... & ... \\ 8.0 & 9.9 & 0.9 & 9.99 \end{pmatrix} \end{array}$$

1.11 Macros

These aid readability, save on repetitive typing and offer ways of producing stylistic variations on standard LATEX formats.

```
\def\bydefn{\stackrel{def}{=}}
\def\convf{\hbox{\space \raise-2mm\hbox{$\textstyle
    \bigotimes \atop \scriptstyle \omega$} \space}}
```

produce $\stackrel{def}{=}$ and $\underset{\omega}{\bigotimes}$ when `$\bydefn$` and `$\convf$` are typed.

1.12 Fine tuning

It's generally a good idea to keep punctuation outside math mode; LATEX's normal handling of spacing around punctuation is suspended during maths. Sometimes you might want to adjust the spacing in a formula (*e.g.*, you might want to add space before a *dx*). Use these symbols:-

`a\, b` ($a\, b$) thin space
`a\> b` ($a\> b$) medium space
`a\; b` ($a\; b$) thick space
`a\! b` ($a\! b$) negative thin space

In an `eqnarray` environment you may want to break a long line. You can do this by putting

```
y & = & a + b \nonumber \\
  &   & + k
```

to give

$$\begin{aligned} y &= a+b \\ & +k \end{aligned} \quad (4)$$

but the spacing around the '+' on the 2nd line is wrong because LaTeX thinks it's a unary operator. You can fool LaTeX into treating it as a binary operator by inserting a hidden character.

```
y & = & a + b \nonumber \\
  &   & \mbox{} + k
```

gives

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

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 + l + m + n +\nonumber\\
 & & o + p \nonumber
\end{eqnarray}
```

$$x + iy = \quad (6)$$
$$a+b+c+d+e+f+g+h+i+j+k+l+m+n+$$
$$o+p$$

1.13 Examples

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

  gives

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

- ```
  \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}) +
  {\cal B}^2(\hat{\theta}).
  \end{eqnarray}
  ```

  gives

  $$\begin{aligned} {\cal M}^2(\hat{\theta},\theta) &= E[(\hat{\theta} - \theta)^2] \\ {\cal M}^2(\hat{\theta},\theta) &= {\rm var}^2(\hat{\theta}) + {\cal B}^2(\hat{\theta}). \end{aligned} \quad (8)$$

- ```
  \begin{equation}
  \hat{W}_{s}(t,\omega;\phi) \bydefn
  \int\limits_{-\infty}^{\infty}
  {\hat{\cal R}_s(t,\tau;\psi) e^{-j\omega \tau}
  \, d \tau }
  \end{equation}
  ```
  gives
  $$\hat{W}_s(t,\omega;\phi) \stackrel{def}{=} \int\limits_{-\infty}^{\infty} \hat{\mathcal{R}}_s(t,\tau;\psi) e^{-j\omega\tau} \, d\tau \tag{9}$$

- ```
  \begin{eqnarray}
  {\cal B}(t,\omega) & \approx &
  {1 \over 4\pi}
  {\cal D}_t^2 W_{\bf S}(t, \omega)
  {{{\scriptstyle \infty} \atop
  {\displaystyle \int \! \int \!
  }}\atop {\scriptstyle -\infty}}
  t_1^2
  \phi(t_1,\omega_1) \, dt_1 d\omega_1
  \nonumber \\
  && +
  {1 \over 4\pi}
  {\cal D}_\omega^2 W_{\bf S}(t, \omega)
  {{{\scriptstyle \infty} \atop
  {\displaystyle \int \! \int \!
  }}\atop {\scriptstyle -\infty}}
  \omega_1^2
  \phi(t_1,\omega_1) \, dt_1 \, d\omega_1.
  \label{F4}
  \end{eqnarray}
  ```
  gives
  $$\begin{aligned} \mathcal{B}(t,\omega) \approx{}& \frac{1}{4\pi}\mathcal{D}_t^2 W_{\mathbf{S}}(t,\omega) \iint\limits_{-\infty}^{\infty} t_1^2 \phi(t_1,\omega_1)\, dt_1 d\omega_1 \\ &+ \frac{1}{4\pi}\mathcal{D}_\omega^2 W_{\mathbf{S}}(t,\omega) \iint\limits_{-\infty}^{\infty} \omega_1^2 \phi(t_1,\omega_1)\, dt_1\, d\omega_1. \end{aligned} \tag{10}$$

- ```
  \newsavebox{\DERIVBOXZLM}
  \savebox{\DERIVBOXZLM}[2.5em]{$\Longrightarrow\hspace{-1.5em}
  \raisebox{.2ex}{*}
  \hspace{-.7em}\raisebox{-.8ex}{\scriptsize lm}\hspace{.7em}$}
  \newcommand{\Deriveszlm}{\usebox{\DERIVBOXZLM}}

  \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

```
\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 `xfig` and the LaTeX `picture` environment

If you *do* want `xfig` to produce LaTeX code it's helpful to constrain `xfig` by selecting LaTeX fonts and by setting the `Angle Geom` mode so that only sloping lines that LaTeX can replicate are drawn. An `Angle Geom` box appears on the bottom line when you select a drawing option.

To convert your picture into a latex file, select the '`Export...`' option. There are various LaTeX options available. The simplest to use are *LaTeX box* (if, for drafting purposes, you just want an empty frame) or *LaTeX figure*. Using *epic* will reduce the resulting file size, but you'll need to have

```
\addpackage{epic}
```

in your LaTeX file. The resulting file can be inserted into your document using something like

```
\include{transfig}

...

\begin{figure}[htbp]
\begin{center}
\input{comp.latex}
\end{center}
```

```
\caption{Compilation Stages}
\end{figure}
```

You may need to edit this picture file, but it's a start.

2.2 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 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. Simply use

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

then include the postscript file using the following commands

```
\begin{figure}[htbp]
\includegraphics{yourfile.ps}
\end{figure}
```

Just about all of the following facilities use postscript. You'll need to run `latex` to generate '`foo.dvi`'. This file can be viewed by the latest `xdvi` program, which can cope with embedded postscript. Run `dvips -t a4 -o foo.ps foo.dvi` to convert the resulting `DVI`/postscript file to pure postscript. This will produce a file that can be previewed with `ghostview` or `gs`. On the teaching system this file can be printed out using `plotview` or

```
lp -dljmr1 -opostscript foo.ps
```

See the *Creating and Printing graphics on PC, Mac, SUN and HP machines* and *Xfig* handouts for more details.

2.2.1 `xfig`

`xfig` can load in `Postscript` and also write `Postscript` files that can be put into a LaTeX document. Use `Export`'s `Encapsulated Postscript` option.

It's possible (but tedious) to use LaTeX fonts and maths in an `xfig` drawing. Do the text on `xfig` as if it you were writing a LaTeX document (i.e., put maths within $ delimiters). Then

- mark all the text as *special*.
- export as `PS/TeX` to produce `unnamed.pstex_t`
- export as `PS/TeX` to produce `unnamed.pstex`
- Do `mv unnamed.pstex_t unnamed.tex; mv unnamed.pstex unnamed.ps` and change `unnamed.tex` so that the `\epsfig` loads in the renamed file (in this example, `unnamed.ps`)
- Add lines like those below to your LaTeX file

  ```
  \documentclass[dvips]{article}
  \usepackage{epsfig}
  \begin{document}
  \input{unnamed.tex}
  \end{document}
  ```

2.2.2 Mac

Modern Macintosh application should generate a conforming EPS file. If you have problem, then use the LaserWriter 8 driver to produce EPS files. Platform-specific considerations do crop up. The EPS generated on the Macintosh will use |ASCII 13— line terminators, while Unix will use |ASCII 10— (and DOS will use both). If this causes trouble, use `emacs` to convert, or try (in Unix)

```
tr "\015" "\012" <original.ps >new.ps
```

2.3 Scaling, rotation, wrap-around and shadows

To scale, use `\resizebox` which takes 2 optional arguments

```
\resizebox{5cm}{10cm}{\includegraphics{yourfile.ps}}
```

would rescale the postscript so that it was 5cm wide and 10cm high. To make the picture 5cm wide and scale the height in proportion use

```
\resizebox{5cm}{!}{\includegraphics{yourfile.ps}}
```

To rotate anticlockwise by the specified number of degrees, use

```
\rotatebox{120}{\includegraphics{yourfile.ps}}
```

The following examples demonstrate how to combine these features and how to use the `subfigure` package to have more than one graphic in a figure.

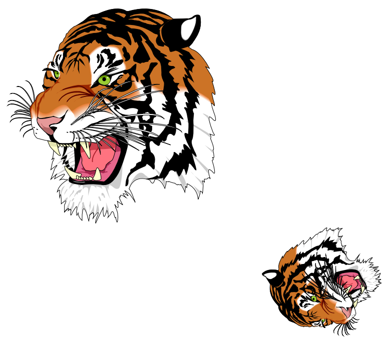
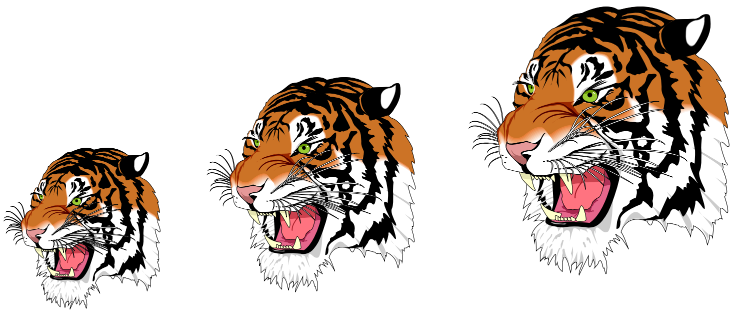


Figure 2: Tigers

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

```
\begin{figure}[hbtp]
\begin{center}
% an mbox for alignment
\mbox{\subfigure[Small]
{\resizebox{!}{30mm}{\includegraphics{tiger.eps}}}\quad
\subfigure[Medium]
{\resizebox{!}{40mm}{\includegraphics{tiger.eps}}}\quad
\subfigure[Large]
```

```
{\resizebox{!}{50mm}{\includegraphics{tiger.eps}}}
}% end of mbox
\caption{3 more tigers}
\end{center}
\end{figure}
```



(a) Small (b) Medium (c) Large

Figure 3: 3 tigers

```
\begin{wrapfigure}{l}{2cm}
\resizebox{2cm}{!}{\includegraphics{tiger.eps}}
\end{wrapfigure}
```

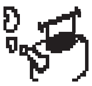
The `wrapfig` package lets you insert a graphic and have the text wrap around it. You provide 2 arguments to the `wrapfigure` command: the first (`l` or `r`) selects whether you want the graphic to be on the left or right of the page. The 2nd argument gives the width of the graphic. Not all text will flow perfectly around (for example, `verbatim` text fails) so check the final output carefully.

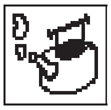
Using the `shadow` package lets you do the following – `\shadowbox{shadow package}` – which produces shadow package.

2.4 Bitmaps

If you want to include a small bitmap into your document, you can use the `Bitmap` macro.

```
\usepackage{bitmap}
\begin{document}
\Bitmap{kettle.px}{1pt}
Now in a frame
\vspace{.5cm}
\fbox{\vbox{\Bitmap{kettle.px}{1pt}}}
\end{document}
```



Now in a frame 

2.5 Graphs

Probably the easiest way to put complicated graphs into LaTeX is to use `gnuplot`. See the relevant online handout by typing `handout latex_gnuplot`.

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