What LaTeX Can Do for You

An Example Paper

Karl Ingo Friese Thomas Rausch

Welfen Laboratory – Division of Computer Graphics and Geometric Modelling
Institute of Computer Science, Department of Mathematics
University of Hannover, Welfengarten 1, 30167 Hannover, Germany
E-mail: cgi98@informatik.uni-hannover.de

Abstract

This paper is designed as an example paper for all authors who want to submit papers to the CGI '98. If you are using LaTeX you may want to take a look at the sources as well. In case you are using another word processor, you can take this as an example how the camera-ready version should look like if your paper is accepted.

1. Introduction

LaTeX is a high-quality typesetting system, with features designed for the production of technical and scientific documentation. LaTeX is the de facto standard for the publication of scientific documents. There exist several shareware applications for all common platforms and operating systems.

2. What LaTeX can do for you

The main reason why so many people use LaTeX is that every paper done with it looks brilliant. Unlike other (commercial) word processors, documents done with LaTeX look the same on every computer in the world, independent from hardware, operating system, printer or LaTeX version. Especially for technical and mathematical work LaTeX provides you with a large set of easy-to-use solutions.

3. About the style file `latex8.sty`

The use of the style file `latex8.sty` guarantees a uniform look of all submitted papers. It defines fonts, margins, headers, title etc. Please take the following informations into account:

- Using LaTeX2e your document should begin with

  \documentclass[times,10pt,twocolumn]{article}
  \usepackage{latex8}
  \usepackage{times}

  If you are using LaTeX 2.09 instead, it should start with

  \documentstyle[times,art10,twocolumn,latex8]{article}

With both versions you should

- Specify \pagestyle{empty} to omit page numbers in the final version
- Specify references as

  \bibliographystyle{latex8}
  \bibliography{...your files...}

- Use \Section and \SubSection instead of \section and \subsection.

The following section is an example how mathematical text with LaTeX does look like. It has been taken from *The Not So Short Introduction to LaTeX 2ε* by H.Partl, E.Schlegl, and I.Hyna (1987). The English Version was done by Tobias Oetiker (1994,1995). It has been modifed a bit to fit our needs. Non-LaTeX users should take a look at the formulas and try to get a similiar look for their paper.

4. Mathematics

LaTeX has a special mode for typesetting mathematics. Mathematical text within a paragraph is entered between \(and \), between $ and $ or between \begin{math} and \end{math}.

Add a squared and b squared to get c squared. Or using a more mathematical approach:

$$c^2 = a^2 + b^2$$

TEX is pronounced as $\tau\epsilon\chi$.

100 m^3 of water

This comes from my $\heartsuit$

It is preferable to display larger mathematical equations or formulas, that is to typeset them on separate lines. Therefore you enclose them between \[and \] or between *begin*{*displaymath*} and *end*{*displaymath*}. This produces formulas which are not numbered. If you want LATEX to number them, you can use the *equation* environment.

Add a squared and b squared to get c squared. Or using a more mathematical approach:

$$c^2 = a^2 + b^2$$

And just one more line.

With *label* and *ref* you can reference an equation within the text.

$$\epsilon > 0 \tag{1}$$

From (1) we gather ...

Note that expressions will be typeset in a different style if displayed:

$\lim_{n\to\infty} \sum_{k=1}^{n} \frac{1}{k^2} = \frac{\pi^2}{6}$

$$\lim_{n\to\infty} \sum_{k=1}^{n} \frac{1}{k^2} = \frac{\pi^2}{6}$$

There are differences between *math mode* and *text mode*. For example, in *math mode*:

1. Most spaces and linebreaks do not have any significance, as all spaces are either derived logically from the mathematical expressions or have to be specified using special commands such as, *quad* or *qquad*.

$$\forall x \in \mathbf{R}: \qquad x^2 \geq 0 \tag{2}$$

2. Empty lines are not allowed. Only one paragraph per formula.

3. Each letter is considered to be the name of a variable and will be typeset as such. If you want to typeset normal text within a formula (normal upright font and normal spacing) then you have to enter the text using the *textrm*{...} commands.

$$x^2 \geq 0 \qquad \text{for all } x \in \mathbf{R} \tag{3}$$

Mathematicians can be very fussy about which symbols are used: it would be conventional here to use '**blackboard bold**' symbols which is obtained by *mathbb* from the package *amsfonts* or *amssymb*.

4.1. Grouping in Math Mode

Most math mode commands affect only the next character. So if you want several characters affected by a command you have to group them together using curly braces: {...}.

$$a^{x+y} \neq a^{x+y} \tag{4}$$

4.2. Building Blocks of a Mathematical Formula

In this section the most important commands used in mathematical typesetting will be described.

Lowercase **Greek letters** are entered as *alpha*, *beta*, *gamma*, ..., uppercase letters are entered as *Gamma*, *Delta*, ...

$$\lambda, \xi, \pi, \mu, \Phi, \Omega$$

Exponents and Subscripts can be specified using the ^ and the _ character.

$$a_1 \qquad x^2 \qquad e^{-\alpha t} \qquad a_{ij}^3$$
$$e^{x^2} \neq e^{x^2}$$

The **square root** is entered as *sqrt*, the n^{th} root is generated with *sqrt*[*n*]. The size of the root sign is determined automatically by LATEX. If just the sign is needed use *surd*.

$$\sqrt{x} \qquad \sqrt{x^2 + \sqrt{y}} \qquad \sqrt[3]{2}$$
$$\surd[x^2 + y^2]$$

The commands *overline* and *underline* create **horizontal lines** directly over or under an expression.

$$\overline{m+n}$$

The commands *overbrace* and *underbrace* create long **horizontal braces** over or under an expression.

$$\underbrace{a + b + \cdots + z}_{26}$$

You can also handle mathematical accents easily. Wide hats and tildes, covering several characters are generated with *widetilde* and *widehat*. The ' symbol gives a prime.

$$y = x^2 \qquad y' = 2x \qquad y'' = 2$$

Often **vectors** are specified by adding small arrow symbols on top of a variable. This is done with the *\vec* command. To denote the vector from A to B the two commands *\overrightarrow* and *\overleftarrow* are useful.

$$\vec{a} \quad \overrightarrow{AB}$$

Names of log-like functions are often typeset in an upright font and not italic as variables. Therefore LaTeX supplies the following commands to typeset the most important function names:

```
\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
```

$$\lim_{x \to 0} \frac{\sin x}{x} = 1$$

For the *modulo function* there are two commands: *\bmod* for the binary operator "$a \bmod b$" and *\pmod* for expressions such as "$x \equiv a \pmod{b}$."

A built-up **fraction** is typeset with the *\frac*`{...}{...}` command. Often the slashed form $1/2$ is preferable, because it looks better for small amounts of 'fraction material.'

$$1\tfrac{1}{2} \text{ hours}$$

$$\frac{x^2}{k+1} \qquad x^{\frac{2}{k+1}} \qquad x^{1/2}$$

To typeset binomial coefficients or similar structures you can use either the command `{... \choose ...}` or `{... \atop ...}`. The second command produces the same output as the first one, but without braces.

$$\binom{n}{k} \qquad \begin{matrix} x \\ y+2 \end{matrix}$$

The **integral operator** is generated with *\int*, the **sum operator** with *\sum*. The upper and lower limits are specified with ^ and _ as with subscripts and superscripts.

$$\sum_{i=1}^{n} \qquad \int_0^{\frac{\pi}{2}}$$

For **braces** and other **delimiters** there exist all types of symbols in TeX (e.g. $[\langle \| \updownarrow$). Round and square braces can be entered with the corresponding keys, curly braces with `\{`, all other delimiters are generated with special commands (e.g. *\updownarrow*).

$$a, b, c \neq \{a, b, c\}$$

If you put the command *\left* in front of an opening delimiter or *\right* in front of a closing delimiter, TeX will automatically determine the correct size of the delimiter. Note, that you must close every *\left* with a corresponding *\right*. If you don't want anything on the right, use the invisible '*\right.*'!

$$1 + \left(\frac{1}{1 - x^2} \right)^3$$

In some cases it is necessary to specify the correct size of a mathematical delimiter by hand, therefore you can use the commands *\big*, *\Big*, *\bigg* and *\Bigg* as prefixes to most delimiter commands.

$$\Big((x+1)(x-1)\Big)^2$$
$$\Big(\Big(\Big(\Big\{\Big\}\Big\}\Big\} \quad \Big\|\Big\|\Big\|$$

To enter **three dots** into a formula you can use several commands. *\ldots* typesets the dots on the baseline, *\cdots* sets them centred. Beside that there are the commands *\vdots* for vertical and *\ddots* for diagonal dots.

$$x_1, \ldots, x_n \qquad x_1 + \cdots + x_n$$

4.3. Math Spacing

If the spaces within formulas chosen by TeX are not satisfactory, they can be adjusted by inserting special spacing commands. The most important are: `\␣` for a medium sized space (␣ stands for a "space" character) and *\quad* and *\qquad* for large spaces.

$$\iint_D g(x, y)\, \mathrm{d}x\, \mathrm{d}y$$

instead of

$$\int \int_D g(x, y) \mathrm{d}x \mathrm{d}y$$

Note that 'd' in the differential is conventionally set in roman type.

4.4. Vertically Aligned Material

To typeset **arrays**, use the *array* environment. It works somewhat similar to the *tabular* environment. The \\ command is used to break the lines.

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

The *array* environment can also be used to typeset expressions which have one big delimiter by using a "`.`" as a invisible *right* delimiter:

$$y = \begin{cases} a & \text{if } d > c \\ b + x & \text{in the morning} \\ l & \text{all day long} \end{cases}$$

For formulas running over several lines or for **equation system**s you can use the environments *eqnarray* and *eqnarray*`*` instead of *equation*. In *eqnarray* each line gets an equation number. In the *eqnarray*`*` no line numbers are produced.

The *eqnarray* and the *eqnarray*`*` environments work like a 3-column table of the form `{rcl}`, where the middle column can be used for the equal sign or the not-equal sign. Or any other sign you see fit. The \\ command breaks the lines.

$$f(x) = \cos x \tag{5}$$
$$f'(x) = -\sin x \tag{6}$$
$$\int_0^x f(y)\,dy = \sin x \tag{7}$$

Notice that there is too much space each side of the middle column, the equal signs. This can be reduced by setting *setlength**arraycolsep*`{`*2pt*`}` as in the next example.

Long equations will not be automatically divided into neat bits. The author has to specify where to break them and how much to indent. The following two methods are the most common ones used to achieve this.

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - - \frac{x^7}{7!} + \cdots \tag{8}$$

$$\cos x = 1 - \frac{x^2}{2!} + + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \tag{9}$$

The *nonumber* command causes LaTeX not to generate a number for this equation.

It can be difficult to get vertically aligned equations to look right with these methods; the package *amsmath* provides a more powerful set of alternatives.

4.5. Math Font Size

In math mode TeX selects the font size according to context. Superscripts for example get typeset in a smaller font. If you want to add roman text to an equation and use the *textrm* command, the font size switching mechanism will not work, as *textrm* temporarily escapes to text mode. Use *mathrm* instead to keep the size switching mechanism active. But pay attention, *mathrm* will only work well on short items. Spaces are still not active and accented characters do not work.

$$2^{\mathrm{nd}} \quad 2^{\mathrm{nd}} \tag{10}$$

Sometimes you need to tell LaTeX about the correct font size nevertheless. In math mode the fontsize is set with the four commands:

displaystyle (123), *textstyle* (123), *scriptstyle* ($\scriptstyle 123$) and *scriptscriptstyle* ($\scriptscriptstyle 123$).

Changing styles also affects the way limits are displayed.

$$\operatorname{corr}(X,Y) = \frac{\displaystyle\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\left[\displaystyle\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2\right]^{1/2}}$$

This is one of those examples in which we need larger brackets than the standard *left*`[` and *right*`]` provides.

4.6. Describing Variables

For some formulas you might want to add a section after the formula where you describe the variables used in the expression. The following example should help you to get this done.

$$a^2 + b^2 = c^2$$

Where: a, b – are adjunct to the right angle of a right-angled triangle.
c – is the hypotenuse of the triangle.

If you often need to typeset material like this, now the time is right to try out the *newenvironment* command. Use it to create a specialised environment for **describing variables**. Check the description at the end of the previous chapter.

4.7. Theorems, Laws, . . .

When writing mathematical documents, you probably need a way to typeset "Lemmas", "Definitions", "Axioms" and similar structures. LaTeX supports this with the command

newtheorem{*name*}[*counter*]{*text*}[*section*]

The *name* argument is a short keyword used to identify the "theorem". With the *text* argument you define the actual name of the "theorem" which will be printed in the final document.

The arguments in square brackets are optional. They are both used to specify the numbering used on the "theorem". With the *counter* argument you can specify the *name* of a previously declared "theorem". The new "theorem" will then be numbered in the same sequence. The *section* argument allows you to specify the sectional unit within which you want your "theorem" to be numbered.

After executing the *newtheorem* command in the preamble of your document, you can use the following command within the document.

begin{name}[*text*]
This is my interesting theorem
end{name}

This should be enough theory. The following examples will hopefully remove the final remains of doubt, and make it clear that the *newtheorem* environment is way too complex to understand.

Law 1 *Don't hide in the witness box*

Jury 2 (The Twelve) *It could be you! So beware and see law 1*

Law 3 *No, No, No*

The "Jury" theorem uses the same counter as the "Law" theorem. Therefore, it gets a number which is in sequence with the other "Laws". The argument in square brackets is used to specify a title or something similar for the theorem.

Murphy 4.1 *If there are two or more ways to do something, and one of those ways can result in a catastrophe, then someone will do it.*

The "Murphy" theorem gets a number which is linked to the number of the current section. You could also use another unit, like chapter or subsection for example.