Baskerville

The Annals of the UK TEX Users' Group **Editor: Editor: Sebastian Rahtz** **Vol. 4 No. 4**

ISSN 1354–5930 **September 1997**

Articles may be submitted via electronic mail to `baskerville@tex.ac.uk`, or on MSDOS-compatible discs, to Sebastian Rahtz, Elsevier Science Ltd, The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, to whom any correspondence concerning *Baskerville* should also be addressed.

This reprint of *Baskerville* is set in Times Roman, with Computer Modern Typewriter for literal text; the source is archived on CTAN in `usergrps/uktug`.

Back issues from the previous 12 months may be ordered from UKTUG for £2 each; earlier issues are archived on CTAN in `usergrps/uktug`.

Please send UKTUG subscriptions, and book or software orders, to Peter Abbott, 1 Eymore Close, Selly Oak, Birmingham B29 4LB. Fax/telephone: 0121 476 2159. Email enquiries about UKTUG to `uktug-enquiries@tex.ac.uk`.

Contents

I Editorial

1 *Baskerville* articles needed

Baskerville has been getting good articles this year, and I am very grateful to all the contributors. But I need more! Please delight fellow TEX users with your words of wisdom. *Please note the following schedule of copy deadlines*:

Issue	Submit material for publication	Submit last-minute notices	Anticipated posting date
4.5	Oct 17	Oct 24	Nov 10
4.6	Dec 19	Dec 22	Jan 9

Please also note the changed email and paper mail addresses for the editor in the banner heading above.

Each issue of *Baskerville* will have a special theme, although articles on any TEX-related subject are always welcome. Contributions on the themes for the remainder of 1994 are eagerly solicited: *Baskerville* 4.5 will try and go beyond TEX, to see what is on the horizon, and *Baskerville* 4.6 will be about font-encoding if past history is anything to go by . . .

1.1 TEX goes CD-ROM

In the last *Baskerville*, the Dutch-produced 4TEX CD was advertised, and shortly afterwards a box of them arrived in the UK. They were promptly snapped up by discerning members, and back-orders to Holland from around the world soon accounted for all the copies which were made. If you do manage to find one, it's a real treasure trove (you can see one of my 'finds' later) of fonts, macros, programs, articles, all piled together moderately higgledy-piggledy. NTG and the 4TEX team are to be enthusiastically thanked for this product. I couldn't get too excited about 4TEX itself (it's a DOSsy shell for TEX), but I have used the disk over and over again to find odd files. Are any *Baskerville* readers who bought the CD willing to write a full review?

If that wasn't enough, those of us who attended TUG94 were given another CD, 'TEXcetera', courtesy of Prime Time Freeware. This is an almost-complete copy of the CTAN archives as of mid June (they left out a few monolithic items like the Archimedes TEX setup to make it fit a single disk), collected and compressed into (usually) meaningful bundles. They couldn't just dump the whole archive since a) its too big, and b) the ISO 9660 file system on the CD couldn't cope with the names and the level of subdirectories. This CD is a Really Useful Thing! I recommend all or any TEX persons reading this to buy a copy now, and encourage Prime Time Freeware to issue regular editions. Details of suppliers are given in the regular section at the back of *Baskerville*.

1.2 The archive is dead, long live the archive. . .

Later in this issue, Martyn Johnson and Robin Fairbairns explain why and how the UK's TEX Archive has moved to Cambridge. I join them in a tremendous vote of thanks to Peter Abbott for the way he stood behind the archive for years at Aston; without him we would have none of today's fancy CTANs. At the same time, I would like to record again the hallowed names of those pioneer archivists who worked so hard on the old archive: Adrian Clark, Malcolm Clark, Brian Hamilton Kelly, Niel Kempson, David Osborne, Sebastian Rahtz, Chris Rowley and Phil Taylor. David's (ongoing) work on the `uktex` and `texhax` bulletins also deserves the fullest recognition here.

1.3 Colophon

This issue of the journal was created entirely with the new standard LATEX and printed on a Hewlett Packard LaserJet 4. *Baskerville* is set in ITC New Baskerville Roman and Gill Sans, with Computer Modern Typewriter for literal text. Production and distribution was undertaken in Cambridge by Robin Fairbairns and Jonathan Fine.

Example: before and after

Economic forecasts for 1992

variable	forecast Grecon	forecast CPB (MEV '92)
	% mutations w.r.t. 1991	
real consumption (c)	1.1	1.25
price index consumption (p_c)	2.6	3.25
real investments (i_m)	1.4	-2.5
export price index (p_b)	3.5	3.25
real import of goods (m)	4.1	3
real output of goods (v')	2.5	2.1[a]
real domestic production (bpr)	1.4	1.6
private employment (a)	0.32	0
wage rate (l)	4.0	4
government income (%) from output of goods (iso')	-0.1	–[b]
	absolute quantities	
unemployment ($\times$1000 persons)	510[c]	525
balance of payments (10^9 Hfl)	24.4	25.0

[a]The quantities v' and bpr aren't given as such by the CPB. The CPB data presented here are computed using their GRECON definitional equations. For details, see appendix D.
[b]Not available.
[c]Not a model outcome: see text in par. 3.1 and 3.2.

Economic forecasts for 1992

	Grecon	CPB[a]
mutations w.r.t. 1991		
real consumption (c)	1.1	1.25
price index consumption (p_c)	2.6	3.25
real investments (i_m)	1.4	−2.5
export price index (p_b)	3.5	3.25
real import of goods (m)	4.1	3
real output of goods (v')	2.5	2.1[b]
real domestic production (bpr)	1.4	1.6
private employment (a)	0.32	0
wage rate (l)	4.0	4
government income (%)		
from output of goods (iso')	−0.1	–[c]
absolute quantities		
unemployment ($\times$1000)	510[d]	525
balance of payments (10^9 Hfl)	24.4	25.0

a. MEV '92
b. The quantities v' and bpr aren't given as such by the CPB. The CPB data presented here are computed using their GRECON definitional equations. For details, see appendix D.
c. Not available
d. Not a model outcome: see text in par. 3.1 and 3.2.

II Table design

Siep Kroonenberg
N.S.Kroonenberg@eco.rug.nl

[Editor's note: I am grateful to Siep Kroonenberg and Gerard van Nes (editor) for permission to reprint this article from MAPS, the journal of the Nederlandstalige TEX Gebruikersgroep.]

LATEX users generally seem unaware of current ideas on table design. The following table is a typical LATEX production:

LATEX table design		
	1991	1992
Unemployment ($\times$1000)	500	600
Balance of Payments (10^9 Hfl)	24	25

In a professionally-designed publication, the above table would probably look more like this:

Common sense table design

	1991	**1992**
Unemployment ($\times$1000)	500	600
Balance of Payments (10^9 Hfl)	24	25

If you read a book on typography, *e.g.* [Treebus 1982] or [McLean 1980]: you'll find that they use rules and boxes with far more restraint, and rely more on white space and variation in typefaces for organization.

The table examples in [Lamport 1986] were (I hope) merely intended to demonstrate techniques. However, their style was almost unanimously adopted by LATEX users.

So I think that some design education is in order. I am not a design professional. However, many people never even think about table design; so if I set them thinking and they start reading books on typography by real professionals then this paper has served its purpose.

Note. This is not meant to be a technical exposition. [Goossens *et al.* 1994] and [Lamport 1986] tell you most of the technical things you need to know. All the same, I have indicated here and there with what codes or constructs you might accomplish certain effects.

1 Basics of table design

A table should present its information as clearly as possible. Typographic means to organize this information includes rules, white space, choice of typefaces and appropriate headings and captions. But if a feature doesn't help to make a table clearer, it had better be left out.

Macroeconomic memoranda

1.	Karl Läusche, Maria Vader, Theo Zernike	Money illusion and savings illusion; an illusionistic look on neo-Hegelian monetary theory
2.	Hendrik Kooyker, Johan Zonderlink	BIGTHUMB, a software package for handling missing and politically incorrect data
3.	Anneke Draaijer	Consumer behavior, expectation formation and the long-term economic effects of risk-aversion

Rules and boxes

Rules have their uses. They can emphasize headings. They can also separate different items and unite the several data for one item, as in the table above. Vertical rules, as in the table below, would have the opposite effect and would be no help at all in making the table easier to read.

Macroeconomic memoranda

1.	Karl Läusche, Maria Vader, Theo Zernike	Money illusion and savings illusion; an illusionistic look on neo-Hegelian monetary theory
2.	Hendrik Kooyker, Johan Zonderlink	BIGTHUMB, a software package for handling missing and politically incorrect data
3.	Anneke Draaijer	Consumer behavior, expectation formation and the long-term economic effects of risk-aversion

But even in the earlier example one might wonder whether white space wouldn't have been more effective than rules.

A table may also be boxed to set it off from the surrounding text. But LaTeX users normally don't go through the trouble of wrapping text around tables and figures; therefore, there is little reason to box in a table.

In all cases, there should be enough space between rules and text. A rule too close to text interferes with readability and makes the text look cramped.

An alternative to rules or boxes is a shaded background, preferably in a second colour. This is not supported by LaTeX as far as I know, although with PostScript some tricks are possible (see *e.g.* [Goossens *et al.* 1994] section 11.6). This formatting device requires high output quality in order to look good.

Alignment and justification

A column of text labels can be left- or right-aligned, or centered. If the table has any length at all, a centered column can easily look sloppy. With left- or right-alignment there is at least one straight edge to give the column structure. Think twice before centering a column in a longer table.

A column of figures is usually decimally aligned (see below for some technical issues). If the figures are unrelated, you may consider right- or left-alignment instead.

Don't justify text inside a narrow column or you'll end up with large distracting holes between words. This is easier said than done, but see further below.

Headings

Headings may get added emphasis by setting them bold, italic, at a larger point size or in a different typeface. Don't go overboard, though. The heading of a centered or decimally-aligned column may need some manual adjustment.

2 An example

We illustrate some of these points with the 'before-and-after' example. It is sufficiently complex to illustrate a number of points; I am not implying that it is any worse than other LaTeX tables I have seen. The 'before' table is a LaTeX remake of a table from [DV91]. At an earlier occasion, it has been used as a demonstration of LaTeX's table-making capabilities.

The example table contains footnotes; therefore it is enclosed in a minipage environment.

Rules

The most conspicuous shortcoming of the 'before' example is the tight spacing between horizontal lines and text. I am not aware of a parameter which controls this distance; however, the '`\\`' command takes an optional length parameter, also in a `tabular` environment.

In this case, as in most cases, the vertical rules are better left out. It is advisable to begin and end the column specification with `@{}`:

```
\begin{tabular}{@{}l@{}r@{}lr@{}l@{}}
```

Without vertical rules, no white space needs to be reserved at the left- and righthand sides.

Actually, I used a `tabular*` environment, which allowed me to set the width to `\linewidth`: exactly the width of the minipage.

Another unfortunate detail is the footnote rule next to the bottom rule. I solved this by dropping the bottom rule. Also, I redefined in a separate style file several aspects of minipage footnotes: among others, the footnote rule now stretches across the width of the minipage.

The rule under the title is not part of the `tabular` environment, but is constructed as a '`\rule`'-rule. This made it easy to give it a custom thickness. Again, the length was set to `\linewidth`.

Headings

As to the various headings: the word *forecast* repeated information from the table header and was dropped. The word *variable* could also safely be omitted.

Aligning the Grecon- and CPB headings at the bottom instead of the top would have been an improvement, but moving the text '*MEV '92*' to a footnote was even better. Their horizontal positioning was adjusted by hand, adding '~' here and there.

The '*% mutations...*' and '*absolute quantities*' headings looked rather jarring in the figures columns, and were moved to the left column.

Fonts

Sans serif faces are especially appropriate for tabular material. At small sizes serifed faces easily look fussy, especially if the output quality is not top notch. Sans serif faces suffer much less from scaling down. A sans serif face also helps to set off the table from the surrouding text.

Several sizes and weights are used (typographers talk about an italic *weight*; the TeX community should realize that they entertain rather off-beat ideas about font families). And hyphens are replaced by proper minus-signs.

3 Technical issues

Some things in LaTeX are harder than they should be. Two notorious examples are table-related: aligning a column of figures on the decimal point, and setting text in a table cell ragged right.

Decimal alignment

There are at least three ways in LaTeX to accomplish decimal alignment:

- If all numbers have the same number of digits after the decimal point, decimal alignment coincides with right alignment, since in most fonts all digits have the same width.

- Split the numbers right before the decimal point, i.e. put an ampersand '`&`' before the decimal point (or after the number, if it has none). The column formatting for the resulting two columns should be `r@{}l`: right-align the part before the decimal point, left-align the remainder, and put no white space in between.
- Use the `dcolumn` package by David Carlisle. This is documented in [Goossens *et al.* 1994] section 5.5.1.

Ragged right justification

You may have noticed that `\raggedright` simply doesn't work in a tabular environment. Again, let me suggest a couple of brute-force workarounds.

- Divide the text manually between rows. Of course, this is practical only in very simple cases.
- Put a parbox around the text, *e.g.*
  `\parbox{1in}{\raggedright` *text...*`}`
  This is simple enough, but not very elegant since it involves specifying column widths outside the `\begin{tabular}` command.

Goossens *et al.* give a more sophisticated solution in section 5.3.1, '*Typesetting Narrow Columns.*' As in the last of the above two workarounds, it adds code to make `\raggedright` operational again.

4 The trouble with LaTeX

It took me a lot of time to prepare the examples in this paper. Even the standard LaTeX `tabular` environment has plenty of quirks, and extension packages such as `array` or `tabularx` only add to them. Too often, it was a matter of trial and error what would work and what wouldn't, and that might depend on the package used. In the end I didn't use any of the table extension packages for this paper.

In LaTeX, some aspects of layout and typography can be controlled by changing a few parameters or by replacing some simple code out of a style file. But there are quite a few rough spots: sometimes the code is too cryptic for easy modification and sometimes the code is not in the style file at all. When typesetting tables one tends to run into such rough spots.

Besides LaTeX, I use high-end wordprocessors and low-end desktop publishing software. I am exceedingly frustrated that simple things that you just do in a commercial program, require hours or days of study and experimentation in LaTeX.

Still, LaTeX can't be beaten (yet) for long documents or for automation. It remains robust and efficient whatever the size and complexity of the job. So I keep using it for certain types of work.

I hope that (LA)TeX developers are seriously addressing LaTeX's shortcomings. What is really needed is a more accessible basic LaTeX system, which doesn't require wizardry to tailor to one's own preferences, and which can put an end to the current proliferation of style files to patch up its defects.

Finally I want to mention that [Goossens *et al.* 1994] was a great help in preparing this paper, even though the solutions proposed there didn't always work out.

References

[Treebus 1982] Treebus, K. F. *Tekstwijzer.* SDU 1982.

[McLean 1980] McLean, Ruari. *Typography.* Thames and Hudson 1980.

[Lamport 1986] Lamport, Leslie. *LaTeX, A Document Preparation System.* Addison-Wesley 1986.

[Goossens *et al.* 1994] Goossens, Michel, Frank Mittelbach, Alexander Samarin. *The LaTeX Companion.* Addison-Wesley 1994.

[DV91] Dietzenbacher, H.W.A., W. Voorhoeve. *Het model GRECON 91-D. Septembervoorspellingen voor 1992*. Onderzoeksmemorandum no. 450. Economics Department, Groningen University 1991.

III Maths in LaTeX: Part 1, Back to Basics

R. A. Bailey
Goldsmiths' College, University of London

1 Introduction

The book *LaTeX: A Document Preparation System* by Leslie Lamport is rather coy about Mathematics. It simply does not reveal the full range of Mathematical expressions that can be correctly typeset without going outside LaTeX. The result is that some Mathematical authors, while attracted to the generic mark-up of LaTeX, believe that they need to use plain TeX or AMSTeX to write their documents.

This sequence of tutorials seeks to correct that impression, by explaining what Mathematical expressions can be typeset with LaTeX without the need for the `amstex` package. Perhaps this will provoke someone else to write a tutorial on that package. The first part is mostly, but not entirely, devoted to things which you can find in *The Manual*, even though you may have overlooked some of them. Succeeding parts (in the next and later issues of *Baskerville*) will be mostly about Mathematical goodies provided by TeX but upon which *The Manual* is silent, even though they are necessary and quite easy to use. The final part will deal with arrays, concentrating on their use in Mathematics.

These are tutorials, so I expect you, the reader, to do some work. Every so often comes a group of exercises, which you are supposed to do. Use LaTeX to typeset everything in the exercise except sentences in italics, which are instructions. If you are not satisfied that you can do the exercise, then write to me with hard copy of your input and output (no email address before we go to press, I'm afraid): I will include a solution in the following issue of *Baskerville*.

A word on fonts. Fonts in Mathematics are handled differently in LaTeX 2.09, in NFSS, and in LaTeX 2ε. Rather than compare these systems every time that I mention fonts, I shall limit myself to LaTeX 2.09. With any luck, this will enrage some knowledgeable person enough to write an article on handling of Maths fonts in different flavours of LaTeX.

2 What does it look like?

2.1 Maths Mode

(LA)TeX has a special state, called *Maths mode*, which it must be in to recognize Mathematical expressions and typeset them properly. Maths mode in LaTeX is everything between `\(` and `\)`, or, alternatively, everything between `$` and `$`. The parentheses are better for trapping errors, because it is obvious whether the left or right one is missing, if any. A missing `$` causes (LA)TeX to swap Maths mode and ordinary mode from then onwards, giving strange output but no errors until it eventually meets something like `x^2` that it cannot interpret in the wrong mode. On the other hand, the dollar signs are easier to type, and easier to see in your input file.

In Maths mode most symbols are typeset as if they represent single-letter variables. A string of three letters will be set as if those three variables should be multiplied together. Fancy features like kerns and ligatures, which are used in normal text to help the reader interpret letter-strings as words, are turned off. Letters are set in the special font known as *Maths italic* which is usually used for variables.

Almost all spaces that you type are ignored. (LA)TeX thinks that it knows better than you do how Mathematics should be spaced, and it is probably right to think so.

Don't stay in Maths mode for too long just because you are too lazy to type a few `$` signs. Everything between the `$`s should be Maths. A common mistake by beginners is to forget that a punctuation sign, like a comma, may have a different meaning in Maths from its meaning in text. In

the scalars a, b and c

we have a textual list containing three mathematical objects, so the input file contains

```
the scalars $a$, $b$ and $c$
```

That comma is a textual one. The lazy typist types

```
the scalars $a, b$ and $c$
```

and obtains

 the scalars a,b and c

On the other hand, in

 the vector (a,b,c)

there is a single Mathematical object, so it is correct to type

```
the vector $(a,b,c)$
```

or, equally well,

```
the vector $(a, b, c)$
```

These commas are part of the Mathematical notation.

2.2 Basic symbols

The basic symbols are the numerals `1`, `2`, ..., the Latin letters `a`, `b`, ..., `z`, `A`, ..., `Z`, and the Greek letters `\alpha`, `\beta`, `\gamma`, ..., `\omega`, `A`, `B`, `\Gamma`, ..., `\Omega`. If you don't know the standard English spellings of Greek letters, look on page 43 of *The Manual*. Upper-case Greek letters which are conventionally the same as their Latin equivalents do not have special commands. Some Greek letters have variants: `\varepsilon`, for example.

The obvious symbols for operators are the keyboard symbols + and −. If you forget to go into Maths mode (a common temptation when typing a table of data), the symbol − will not look like a minus sign. Outside Maths mode the + will look like a plus sign, but the spacing will be wrong. In Maths mode (LA)TEX knows what is the proper spacing to put around binary operators like + and −; it also knows the proper spacing to surround binary relations like =. Try typing the following both inside Maths mode and outside it, and compare the results.

```
1 +2 =  3          4-1 = 3
1 -4 = -3         -2+7 =+5
```

Also try > outside Maths mode: you may be surprised.

2.3 Sub- and superscripts

Subscripts are introduced with `_`: for example, `x_n` gives x_n. If there is more than one thing in the subscript you have to use braces, as in `x_{n+1}` for x_{n+1}. You can type `x_{n}` for x_n if you want, but it makes your input file less readable.

Superscripts are done similarly, using `^`: thus `y^3` for y^3 and `y^{-1}` for y^{-1}.

A sub- and superscript can be put on the same symbol in either order: `x_n^2` and `x^2_n` both produce x_n^2. Double subscripts or superscripts are obtained by using braces in the obvious way: `x_{n_2}` and `n^{m^2}`.

To put a sub- or superscript *before* a symbol, precede it with `{}`. Otherwise the sub- or superscript attaches itself to the previous thing, which may well be something like $+$ or $=$.

In an expression such as $(X+Y)^2$, strictly speaking TEX thinks it is putting the superscript on the right parenthesis if you type `(X+Y)^2`, and it positions the superscript in accordance with that thought. If this really offends you, you can force TEX to share your logic by typing `{(X+Y)}^2`, but you may not always prefer the result.

2.4 Modifying symbols

To turn x into x' type `x'`. You do not need to think of the prime as a superscript.

Some common modifiers are exemplified in

`\bar{x}`	$\bar{x}$	`\tilde{x}`	$\tilde{x}$
`\hat{x}`	$\hat{x}$	`\vec{x}`	$\vec{x}$

A few more such decorations are shown on page 51 of *The Manual*. If any of them is used over an i or a j then the dotless versions of those letters should be used: `\imath` and `\jmath`.

There are wide versions of `\hat` and `\tilde`:

`\widehat{a+b}`	$\widehat{a+b}$
`\widetilde{1-\theta}`	$\widetilde{1-\theta}$

There are also wide versions of `\bar` and `\vec` but with less obvious names: I'll cover these in a later tutorial.

Logically, a decoration such as `\hat` may modify the whole of a subscripted expression such as x_2; you usually mean 'the estimate of x_2' rather than 'the second part of $\hat{x}$'. However, both $\hat{x_2}$ and $\widehat{x_2}$ simply look wrong, so you have to let aesthetics triumph over logic and type `\hat{x}_2`.

2.5 Dots

To get a line of dots to show that some items have been missed out, use `\ldots` if the missing items are normally aligned on their baselines, such as letters, or `\cdots` if the missing items are normally aligned on the centreline, such as binary operators. If the missing items are part of a textual list, don't forget to come out of Maths mode and to put a comma at the end of the dots.

```
for $i=1$, $2$, \ldots, $10$
the vector $(x_1, x_2, \ldots, x_n)$
$a_1 + a_2 + \cdots + a_n$
$y_1 = y_2 = \cdots = y_7$
```

If you think that the dollar signs round the numerals in the first example are unnecessary, try embedding that phrase in a piece of italic text.

2.6 Square roots

Type `\sqrt{2}` to obtain $\sqrt{2}$. The same technique works for more complicated expressions than 2: you don't have to do anything to make the root sign the right size. For example,

`\sqrt{n^2+6}` $\sqrt{n^2+6}$

Other roots, such as cube roots, are obtained by putting in an optional argument:

`\sqrt[3]{8} = 2` $\sqrt[3]{8}=2$

The simple symbol for a square root is `\surd`.

Don't abuse TeX's wizardry by using `\sqrt` for a large expression in text or in a complicated display. The mess obscures the message.

2.7 Displayed Maths

To get a single line of displayed Maths, type the contents between `\[` and `\]`. You should not start a paragraph with displayed Maths, but may end one. If the displayed Maths is in the middle of a paragraph, remember not to leave blank lines around it in your input file.

Displayed Maths may also be typed between `$$` and `$$`, but the effect is not quite the same. For example, the document option `fleqn` aligns displayed Maths on the left if you use `\[` and `\]`, but not if you use `$$`.

To put a short piece of text in displayed Maths, insert it in `\mbox`, remembering to include any necessary spaces that would be ignored in Maths mode.

```
\[ a=b \mbox{ if } c=d \]
```

Don't try to use `\mbox` in a similar way to put short text between pieces of Maths in text: it inhibits line-breaks.

2.8 Words as labels

Sometimes you want to attach natural-language words to Mathematical symbols to label them. For example, you might have analogous quantities associated with the rows and columns of a rectangular array, and wish to indicate this by using the same symbol, say Q, with different subscripts. It simply will not do to type `Q_{rows}`, because this gives Q_{rows}, where the subscript looks like the product of r by o by .... And it is no good putting `rows` in an `\mbox`, because it will come out too big. Once something has been put in a box, it doesn't change size. You have to type `Q_{\rm rows}` to get $Q_{\rm rows}$. (Did you remember the caveat about fonts?)

If this seems too much trouble, you might decide to abbreviate to Q_r and Q_c. But this will not do either, because the subscripts look like variables into which numbers, say, could be substituted. If you don't want to mislead your readers, you should type `Q_{\rm r}`.

2.9 Fractions

A built-up fraction is made with `\frac`:

`\frac{n}{m}` $\frac{n}{m}$

This comes out larger in displayed Maths than in text. It is better to use the solidus, as in n/m, for most fractions in text, with the exception of a few simple common fractions like $\frac{1}{2}$.

Of course, fractions can be put inside other fractions with no bother:

```
\frac{a(b+c)}{5 + \frac{1}{xy}}
```

$$\frac{a(b+c)}{5+\frac{1}{xy}}$$

2.10 Binary operators

In the golden olden days of golf-ball typewriters, it was a luxury to a Mathematician to have the symbol for direct sum, or for union. (LA)TEX not only has the symbols; it knows that they are operators, and gives them the correct spacing for infix operators, and has reasonably good ideas about where to break lines near them. A few of the common ones are:

`+`	$+$	`-`	$-$	`\pm`	$\pm$
`\times`	$\times$	`\div`	$\div$	`\oplus`	$\oplus$
`\cup`	$\cup$	`\cap`	$\cap$	`\wedge`	$\wedge$.

There are many more on page 44.

In fact, (LA)TEX is even cleverer than this. If a binary operator doesn't find itself between two things it can operate on then it becomes a simple symbol, and spaces and line-breaks adjust accordingly. You should have noticed this if you did the exercise suggested above.

2.11 Binary relations

(LA)TEX also knows about infix relations, such as

`=`	$=$	`\in`	$\in$	`\subset`	$\subset$
`<`	$<$	`\leq`	$\leq$	`\perp`	$\perp$.

More are shown on page 44. Don't confuse $\in$ with either of the epsilons.

Compare `\mid` with $|$. The former is a relation, while the latter is just a symbol. So which should you use for 'divides'?

Relations can be negated by preceding them with `\not`:

`Z_2 \times Z_2 \not\cong Z_4` $\quad Z_2 \times Z_2 \not\cong Z_4$

This doesn't work quite right for $\in$, so there is the special command `\notin`. Also, `\ne` is a useful shorthand for `\not=`.

2.12 Fonts in Maths

(Did you remember the caveat about fonts?)

For something like script letters use `\cal`, as in `${\cal F}(x)$` for $\mathcal{F}(x)$. The braces give the scope of `\cal`: for a single Mathematical letter such as $\mathcal{H}$ you can get away with `$\cal H$`. Only upper-case Latin letters may be modified by `\cal`.

In some branches of Mathematics, constants are shown in Roman type. So the base of natural logarithms is `{\rm e}`.

For bold letters, you can use `\bf` to modify Latin letters and upper-case Greek ones:

`{\bf Mv} = a{\bf w}` $\quad \mathbf{Mv} = a\mathbf{w}$

For lower-case Greek letters, and for non-letters, you have to use a cumbersome construction:

`\mbox{\boldmath $\lambda$}` $\quad \boldsymbol{\lambda}$

Because of the box, this does not change size properly in sub- and superscripts.

2.13 Writing Maths

The ability to produce beautiful Mathematical formulae is no licence to produce poor Mathematical writing. Remember that relations are verbs. It is impossible to parse the sentence

Therefore $n = 56$ is the sample size.

but

The equation $x^2 + 9 = 0$ has no real roots.

is fine.

Don't start a sentence with notation: the reader doesn't get the right visual clue. If possible, avoid putting notation immediately after *any* punctuation, unless it is part of a list. This saves the reader from having to work out if the punctuation is Mathematical or textual. Similarly, avoid abbreviations like 'iid' and 'e.g.' which might be mistaken for notation at a first glance.

3 Exercises

Exercise 1 The zeros of the quadratic ax^2+bx+c are

$$\frac{-b \pm \sqrt{b^2-4ac}}{2a}.$$

Exercise 2 The upper 5% point of the χ^2_6 distribution is 12.592.

Exercise 3 If $\nu = n_1+n_2-2$ and

$$s^2 = \frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}$$

then

$$\frac{\bar{X}_1-\bar{X}_2}{s\sqrt{(\frac{1}{n_1}+\frac{1}{n_2})}}$$

is distributed as t_ν.

Exercise 4 By choosing bases, it follows that the subspaces $Z_1, \ldots, Z_r$ span V; hence it follows that V is the direct sum $V = Z_1 \oplus \cdots \oplus Z_r$, as asserted.

Exercise 5 If $\mathcal{M}$ and $\mathcal{N}$ are subspaces of a finite-dimensional inner product space $\mathcal{V}$ then

$$(\mathcal{M}+\mathcal{N})^\perp = \mathcal{M}^\perp \cap \mathcal{N}^\perp$$

and

$$(\mathcal{M}\cap\mathcal{N})^\perp = \mathcal{M}^\perp + \mathcal{N}^\perp.$$

Moreover, $\mathcal{M}^\perp \cong \mathcal{V}/\mathcal{N}$.

Exercise 6 The sum of squares for the linear model $V_{\text{protein}}+V_{\text{fishmeal}}$ is 1559378.

Exercise 7 The usual regression equation is $\mathbf{Y} = \mathbf{X}\beta+\varepsilon$, where $\mathbf{Y}$ is an $n\times 1$ vector, $\mathbf{X}$ is an $n\times p$ matrix, β is the $p\times 1$ vector of unknown parameters, and ε is the $n\times 1$ vector of random errors. The least-squares estimate $\widehat{\beta}$ of the parameters is given by

$$\widehat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}.$$

Exercise 8 The T-orders are $p(x)^{e_1}$, $p(x)^{e_2}$ and $p(x)^{e_3}$, where $e_1 > e_2 \geq e_3$. This implies that $p(x)^{e_1} \mid \eta(x)^{e_1-d}$ and hence that $\eta(x) = \psi(x)p(x)^d$ for some polynomial $\psi(x)$.

Exercise 9 We have $t \in A\setminus B$ if and only if

$$t \in A \text{ and } t \notin B.$$

Exercise 10 Pascal's triangle is based on the identity

$${}^{n-1}\mathrm{C}_k + {}^{n-1}\mathrm{C}_{k-1} = {}^{n}\mathrm{C}_k.$$

IV Backslash—Mathematical Activity

Jonathan Fine
J.Fine@uk.ac.cam.pmms

First an apology. I did not allow time to proof my last article and so did notice many small errors. None were TEXnically important, except the solution to exercise 3. I thank David Carlise for pointing out that although `\par` and `\xyz` might have the same meaning, only `\long` macros will accept a `\par` in their parameter text. I guess I also forgot to mention that when used as a macro parameter delimiter, the meaning of a control sequence has no bearing but the name is everything. And in the middle of page 17, right column, the line

```
\spaceit \endspaceit
```

should be deleted.

The themes of this *Baskerville* issue are mathematics and tables. Siep Kroonenberg's article on tables is excellent. Here is a little trick for use within mathematics. It involves active characters. The sort of thing one might wish to do is have, say, `[[` act as a sort of ligature for a compound math character, such as ⟦.

For every character code 0–255 there is a mathcode, which controls just how that character should be typeset, when in mathematics. More exactly, it gives the class or part of mathematical speech, the font family to use, and the location with the font family.

A little known and little used feature of TEX is,

> A `\mathcode` can also have the special value `"8000"`, which causes the character to behave as if it has catcode 13 (active). Appendix B uses this feature to make `'` expand to `^{\prime}` in a slightly tricky way.

Knuth writes on [155] (this means page 155 of the *TEXbook*). This feature is not used by `plain` for any other purpose. This remark is flagged as a 'double dangerous bend' and so this article may not be suitable for all readers.

As a result of this magic value for mathcode, a character can be made to act as if it were active when it is in mathematics mode, but not in text mode. This is done without changing the catcodes, and so even if ordinary letters are so made special, formation of control sequence names proceeds as usual. Moreover, math macros such as `\matrix` and `\eqalign` read their text as a parameter, and this fixes the category codes. (The `plain` footnote macro goes to some length to avoid this [363], so as to allow category code changes to occur within the text of the footnote. This enable verbatim text to there appear.)

Knuth [48] "discourage[s] people from making extensive use of `\catcode` changes except in unusual circumstances" precisely because "when the arguments to a macro are first scanned … their categories are fixed once and for all at that time." A `\matrix` may contain math and ordinary text, or may itself be the argument to another macro (this is why verbatim does not work properly within LATEX section titles). Thus, to achieve a smart `[[` by category code changes would be difficult, and create many unwelcome side effects.

However, mathcode `"8000` does not have these problems, because it is not a category code change. To understand the use of this unusual mathcode, let us change the math code of `[` to this new value in such a way that ordinary documents will process exactly as before. The line

```
\mathcode '\[ "8000
```

will change the mathcode to the magic value, but the previous value—which controlled its conduct—is now lost. So we shall first save it. In fact the code below

```
\ifnum \mathcode`\[ = "8000
\else
  \begingroup
    \catcode `\[ =13
    \global \mathchardef [ \mathcode`\[
  \endgroup
  \mathcode `\[ = "8000
\fi
```

will first test that we haven't monkeyed with it before, and if safe to do so, will `\mathchardef` an *active* `[` to the original mathcode value, and finally set the mathcode of `[` to the magic value.

These changes (unless `[` already has an active meaning, say for use within ordinary text) should have no effect whatsoever on the processing of manuscripts. So what have we gained?

Previously the mathcode of `[` caused the appropiate character to be looked up from the appropiate font, and used as a mathematical part of speech of the appropiate class. Now the mathcode of `[` will cause the meaning of active character `[` to be looked up. The current value of this meaning is a mathchar which causes the previous appropiate action. *This meaning can now be changed*, to produce new behaviour. This is the gain.

Our example is that we wanted `[[` to produce $[\![$. This compound symbol fragment was produced using `$[\![$`, where `\!` gives a negative thin space. To obtain this same result, but using `[[` as input, we must reset the value for active `[`.

Here's how. Active `[` must inspect the next token. To avoid `\futurelet` complications, I will assume is not a brace or a space, and so can be read as a parameter. If it is another `[` we produce the compound symbol, otherwise we produce a single [and restore the parameter to the input stream.

```
\def \next #1%
{%
  \ifx #1[%
    \lbrack@\!\lbrack@
  \else
    \lbrack@
    \expandafter #1%
  \fi
}
\begingroup
  \catcode`\[=13  % active
  \global \let [ \next
\endgroup
```

The control sequence `\next` is used to hold the value until we change the catcode of `[` to access active `[`. If we tried to make the definition all at once, we would find that we would no longer have access to regular `[`. The command `\lbrack@` has been freshly introduced, to hold the customary mathcode of `[`. This could have been obtained via

```
\mathchardef \lbrack@ \mathcode`[
```

if we had though to *before* we started changing things. As it now is, we can use

```
\mathchardef \lbrack@ "405B
```

which value comes from `plain.tex` (see [344]).

The `\expandafter` in the above definition is to prevent code such as

```
$ [ \mathmacro { argument } ] $
```

producing a disaster, where `\mathmacro` takes a single parameter. Stepping through the above code for active `[` we will we get

```
\lbrack@ \expandafter
\mathmacro \fi { argument }
```

as an intermediate result. Without the `\expandafter`, the `\mathmacro` would get `\fi` as its argument, and that is totally wrong. As it is, the `\expandafter` causes the token *after* the `\mathmacro`, which is the `\fi`, to be expanded *before* `\mathmacro` does its piece. When `\fi` is expanded [213],

> TEX reads to the end of any text that ought to be skipped. The "expansion" of a conditional is empty.

and this is just what we want. The `\fi` is gone, and so now `\mathmacro` gets its proper argument.

This device, which I call *active mathematical characters* makes all sort of dirty trickery possible. Mathematicians have a wide range of complicated symbols, diagrams, matrices, and so forth. Perhaps use of this device will allow for improved input syntax for at least some of these devices.

Finally, problems and solutions. Problem 5 from last issue has a short solution (six lines of 80 column code) but seems to require a long explanation. The solution to Problem 6 will be given in the next issue. There are two new problems for this issue. The solution to Problem 7 is in the *TEXbook*. Problem 8 asks a question about possible `\mathchar` values.

Solution 5. *The problem was to write a macro which will trim the leading and trailing spaces from user supplied text.* Assume that `\text` is a macro whose expansion is the user-supplied text, such as

```
\def \text { apples and oranges }
```

and that `\trim \text` is to redefine `\text` as

```
\def \text {apples and oranges}
```

which is as before, but without leading and trailing spaces. However, the original value of `\text` may contain macros, nested braces, and perhaps even conditionals.

Here is the solution, with comments as we go along.

```
\catcode`\@=11  % @ is a letter
\def\trim #1{%
  \expandafter\trim@
  \expandafter{#1 }%
  #1%
}
```

If `\text` is the argument to `\trim`, the expansion of `\trim` will result in `\trim@` being called with two parameters. The first will be, enclosed in braces, the user supplied text *but with an additional trailing space* (the reason for which will be given later) and the second the name of the control sequence (`\text`) whose redefinition is sought.

We now set things up to remove the leading space, if any. We use `@` as a private delimiter, for it cannot occur *with category code 11* in user supplied text. The expansion of

```
\def\trim@ #1{\trim@@ @#1 @ #1 @ @@}
```

will cause `\trim@@` to see before it *two* copies of the user-supplied text, both with (another) additional trailing space, the first copy without and the second with an additional leading space. This whole mess is closed with `@@`, which functions as a delimiter.

The trick now is to have `\trim@@` look for text delimited on both left and right by the pair `@␣` of tokens (being an `@` followed by a space). If the user-text has a leading space, such occurs around the first copy. If not, around the second copy. The parameter delimiters of

```
\def\trim@@ #1@ #2@ #3@@{%
  \trim@@@\empty #2 @%
}
```

select the appropiate copy of the user-text to be parameter `#2`. The rest of the arguments can be thrown away, all the way up to the `@@` delimiter. The parameter `#2` will be the user-text, with a trailing space added twice (by `trim` and by `trim@` also), and with the leading space (if present) stripped.

We are nearly done now. The purpose of the `\empty` (a macro which expands to nothing) will be explained later. We copy the user supplied text with yet another trailing space (that's the third time we've done this) and call `\trim@@@` with `@` as a delimiter.

Here come the final and amusing macro. We wish to strip the trailing space, if present. Perversely, we have three times added a trailing space. Now *in regular user defined text, by virtue of TEX's reading rules [37], it is impossible for user supplied text to contain two successive explicit space characters*. So we use two successive spaces characters as a delimiter, to strip trailing spaces. This is why we have been so assiduously been building them up at the end.

We need a helper macro

```
\def\unbrace#1{#1}
```

to allow the construction of the final macro

```
\unbrace{\def\trim@@@ #1 } #2@#3
{%
  \expandafter\def
  \expandafter #3%
  \expandafter {%
  #1}%
}
\catcode`\@=12  % restore @
```

whose first parameter `#1` is delimited by *two space characters*. This strips the trailing space, and we discard any other spaces there may be, up to the trailing `@`. The third parameter `#3` is the control sequence (`\text` in our case) whose stripped redefinition we seek.

By now, `#1` is stripped of leading and trailing space, and has an `\empty` prepended. This is 'stripped' via the `\expandafter`'s. The macro is finished.

Some further explanations are required. A trailing space is added *three* times when it might seem that twice is enough, to cover the case that `\text` is empty. In that situation the first added *trailing* space will also be a *leading* space, and will be treated as such. The purpose of the `\empty` is to forestall TEX's (usually helpful) custom of stripping "the outermost braces enclosing the argument" [204]. Without this sweet nothing, the macros produce from

```
\def\text {{well wrapped}}
```

the new value

```
\def\text {well wrapped}
```

which is wrong! Earlier in the expansion the trailing space(s) stopped this happening.

Finally, an acknowledgement. The basic ideas for dealing with the leading space are due to Donald Arseneau, but the trailing double space trick is all my own work.

Exercise 7. What reason does Knuth give for choosing `$` as the math bracket. Hint: mathematics and tables are known as 'penalty work' because they will attract an extra charge from the typsetter. The solution in on [127].

Exercise 8. Why should the value `"8000` be forbidden [155] as mathchar (rather then mathcode) value? And why not?

V Hyphenating British English

Philip Taylor
RHBNC
P.Taylor@Vax.Rhbnc.Ac.Uk

Many members of UKTUG will already be aware that an enormous debt of gratitude is owed to Dominik Wujastyk, who undertook the initial generation of a set of hyphenation patterns for TEX which were based on a British (as opposed to American) dictionary. That debt of gratitude is also owed to Oxford University Press, who donated their internal word-list of some 160 000 entries with primary, secondary and tertiary breakpoints shewn as well as a 'frequency-of-use' index for each word.

Dominik struggled against seemingly insuperable odds to process this vast word-list; the standard *Patgen* simply wasn't up to the task, and despite the best efforts of Peter Breitenlohner, Wayne Sullivan and many others, an attempt to build a suitably large DOS/Pascal version was doomed to failure. In the end, Dominik discovered the *web2c* implementation of Karl Berry, and this, together with D J Delorie's DJGPP C compiler, eventually enabled him to build a version of *Patgen* which could cope with a 160 000-entry word-list.

But although he did not know it, his troubles were but starting: once he could read the word-list, he had to supply values for three of the most cryptic and arcane variables in the known TEX world: `good_wt`, `bad_wt` and `threshold`. These three variables control the entire pattern generation process, yet even their inventor, Frank Liang, was forced to confess in his Ph.D thesis ("Word Hy-phen-a-tion by Com-put-er") that he was unable to justify the values which he had used to generate the American patterns other than by purely empirical means. And so Dominik, too, used Frank's values, and produced patterns which, statistically at least, were as valid as Frank Liang's. Dominik recorded his experiences in a talk which he gave to the UK TEX Users' Group Easter meeting which was held at RHBNC last year.

However, the generation of patterns is not a once-and-forever task: those patterns which Dominik had produced were larger than the American equivalent, requiring for some systems at least either a specially 'large' TEX or at least a TEX tuned to accommodate a larger pattern set. Furthermore it correctly hyphenated only 90% of the words in the 160 000-entry word-list, missing about 10% completely. There were also a few words which it was known would be hyphenated incorrectly using Dominik's patterns, and which were subsequently documented in the distributed `ukhyphen.tex`.

With a sabbatical year in India on the horizon, Dominik felt that it was time to hand over the baton; he had created a viable set of patterns, and if someone else wanted to improve on them, that was up to them. As Dominik knew that I had a considerable interest in pattern generation, and that I had, in fact, offered to run *Patgen* on my VAX/VMS system if he had been unable to get a copy working on any of the systems to which he had access, he asked if I would like to become 'custodian of the patterns', and I willingly agreed. After all, Dominik had done all the hard work — acquired a suitable machine-readable dictionary, created the initial pattern set, ascertained suitable values for `good_wt`, `bad_wt`, `threshold`... So my task should be infinitely more straightforward: just build on what Dominik had already done.

But of course, life is rarely that straightforward: as soon as I came to build a large *Patgen* for VAX/VMS, I discovered that the Kellerman & Smith changefile which I had no longer worked. Furthermore, K&S were unwilling to allow it to pass into the public domain, so any development work on it would have been futile. My saviour turned out to be Christian Spieler, who had already ported the remainder of the standard TEX distribution to Alpha/VMS; only *Patgen* remained, and once I had explained to him the importance of that little-known utility, he willingly and promptly undertook an Alpha/VMS port, including as standard the additional workspace which it was known would be required. Within 24 hours a test version was ready, and it worked beyond my wildest dreams: no second version was needed, the very first version went straight into production, and that same day I was able to produce a set of patterns which, statistically at least, were as good as those produced by Dominik.

But just as Dominik had had to battle with `good_wt`, `bad_wt` and `threshold`, I too had my own windmills at which to tilt: in my case the problem came about because Christian had, very reasonably, based *his* implementation on *Patgen2* (Peter Breitenlohner's 8-bit modifications to DEK's standard 7-bit Patgen). And Patgen2 has four new variables with which to cope: `hyph_start`, `hyph_finish`, `pat_start` and `pat_finish`!

Fortunately for me, these are nowhere near as arcane as `good_wt` and its ilk: the two `hyph_` parameters allow multiple passes through the dictionary to be subsumed into a single run, whilst the two `pat_` parameters allow the minimum and maximum length of pattern for each pass to be separately specified. I do not pretend for one instant that I *fully* understand these, and I certainly don't pretend to have more than the vaguest comprehension of the full implications of `good_wt`, etc., but at least I can now generate patterns to my heart's content, and the Alpha is busy doing that at the very time that I am writing this report. . .

Between now and the time of publication of the next *Baskerville*, I hope to have a much clearer understanding of the possible interactions between the various parameters to *Patgen*. And I hope, too, to have prepared a new set of patterns which the UK community will be able to adopt as a standard, together with a minimal set of exceptions which I am sure will still be necessary. But work will not then stop: I have already enlisted the help of a friend and sometime colleague, Chris McManus, who I hope will be able to define some *rules* for the choice of values for the various parameters (Chris is a medic, statistician and polymath *extraordinaire*, and if anyone can formulate rules for this problem, I am convinced that it is he); and between us I hope that we will be able to publish some guidelines for the use of *Patgen2* — guidelines which are sadly lacking at the moment.

And finally I hope that you, too — the UK TEX Community — will contribute to this project: for someone has to identify the mistakes which the patterns allow, and such a task is far beyond the ability of any one individual to undertake. Once a new definitive set of patterns is announced, I will ask you all to look carefully at every document that you typeset thereafter; and note whenever a hyphenation looks strange; and to check it with a definitive list of valid hyphenation points (I am using "The Oxford Minidictionary of Spelling and Word-Division", but pointers to other definitive sources will be most welcome); and if you find a genuine instance of a wrongly-hyphenated word, then *please* report it to me. I will probably set up an e-mail list solely for this purpose, since I lose paper mail almost by definition whilst e-mail remains accessible in perpetuity.

So, to summarise: building on previous work by Don Knuth, Frank Liang, Peter Breitenlohner, The Oxford University Press, Dominik Wujastyk and Christian Spieler (doubtless among many others), I am now in a position to generate British English hyphenation patterns. In conjunction with Chris McManus, I hope that we will be able to formalise much that has been heuristic, or at best stochastic, in the past. And with your help, I hope to be able to produce not only a definitive set of British English patterns, but an equally definitive (but, one hopes, very small!) set of exceptions. I look forward to this challenge very much indeed.

VI A METAFONT of ‘Simpsons’ characters

Raymond Chen
rjc@math.princeton.edu

[Editor’s note: I found this issue’s ‘stocking filler’ on the NTG CD-ROM; correspondences between Simpsons characters and well-known TEXxies do tend to come to mind.]

The author can type `\Lisa`, `\Homer`, `\Bart`, or `\Marge` to produce the corresponding character. The default is to draw the character facing to the right and looking directly at you. To modify this, you can prefix the macro `\Left` to get the character face left instead of right, *e.g.* `\Left\Lisa`.

You can also prefix the macro `\Goofy` and suffix two pairs of coordinates, which modify how the pupils are drawn. E.g., `\Goofy\Lisa(7,5)(5,5).` The first pair of coordinates is applied to the right pupil (which is the one on the left when printed) and the second pair to the left pupil. The units are relative to the size of the character. (So if you say `\font\simpsons=simpsons scaled 1200` you don’t have to modify all the coordinates in the `\Goofy`’s.)

If you uses both prefixes, as in `\Goofy\Left`, then the mirror-image-reversal takes place *after* the goofiness is applied. This is so that you can just say `\Goofy\Left\Lisa(7,5)(5,5)` to get a mirror image of `\Goofy\Lisa(7,5)(5,5)`.

Some sample Simpsons:

ΠΓ

Ξ D’oh!

ΠΓ

Θ This is Lisa Simpson. She’s smart, she’s sweet, she’s sensitive. . . but don’t hold that against her.

ΓΓ

Σ I’m Bart Simpson. Who the hell are you?

ΓΓ

Ψ Mmmm. . .

ΠΓ

ff Suck. Suck.

The characters were obtained from:

Lisa Simpsons Illustrated, Summer 91, cover
Homer Simpsons Illustrated, Fall 91, cover
Bart Simpsons Illustrated, Fall 91, article on Dan Castellaneta
Marge Simpsons Illustrated, Fall 91, article on Dan Castellaneta
Burns Simpsons Illustrated, Fall 91, article on Dan Castellaneta
Maggie Simpsons 1992 calendar, “Phone pranks”
SNPP Simpsons Illustrated, Fall 91, Homer’s job file

They were traced and transferred to graph paper, then magnified fourfold.

VII The 15th Annual TEX Users Group Meeting

Michel Goossens
CERN, Geneva
m.goossens@cern.ch

1 Introduction

July 31st, Santa Barbara, California, USA.[1] Just the right combination of sunshine, temperature, and sea breeze. The mountains in the background, the beach nearby, the food nearly perfect. The ideal setting for a conference. And here we were, some 120 TEX enthusiasts, coming from many countries and cultures, to meet each other, and talk about and listen to presentations of the latest developments in the area of high quality typesetting.

We were not disappointed. The quality of the presented papers was uniformly good, or even outstanding, so many Birds of a Feather (BoFs) were going on in parallel that it was impossible to keep track of the many hot topics being debated by specialists and users in these informal meetings that took place when there were no formal presentations.

The formal theme of the conference was “Innovation”. Malcolm Clark and Sebastian Rahtz brought together a tremendous programme that clearly showed how TEX is now making inroads in many areas of book production, like colour support, more flexible page layouts, scholarly and non-Latin alphabet editions. Several groups are working on extending TEX or LATEX so that these tools become ever better adapted to the demands of present-day document handling and are integrated more readily into electronic distribution networks or databases. Several new approaches introduce object-oriented programming techniques, and hence show that TEX forms an integral part of a modern computing development environment.

I hope that the following detailed overview will give you a flavour of all these developments, and that it will convince you that you want to know more about one or more points. You can obtain the proceedings of the Conference by becoming a TUG member for $60, which entitles you to four issues of TUGboat and of TEX and TUG News, or else for $30 you can obtain a copy of the Proceedings only. For more details contact the TUG office.

It all started on Saturday July 30th in the evening with the traditional Welcome Party. This where one meets old friends and colleagues or discovers new faces; the latter are at first looking around with somewhat anxious eyes, but are quickly surrounded by reassuring oldies, shaking hands, and being welcomed to the “Family”. The Californian wine, beer, or lemonade flowed freely, and by the end of the evening all ice was broken and the atmosphere was one of harmonious warmth and unity.

The Conference was formally opened the next day by TUG’S Executive Director, and local organizer, Patricia Monohon, and Christina Thiele, TUG’s President also spoke a few words of welcome.

2 Publishing, languages, literature and fonts.

It was Charles (Chuck) Bigelow who had the honour to present the first paper. He started by looking back at letter forms over the past 2500 years or so, and then discussed work—together with Kris Holmes—on the Lucida Sans Unicode font, that contains at present some 1700 alphabetic and mathematical symbols and is or will be available with the multi-byte operating systems Windows/NT, Apple GX and AT&T Plan 9.

Frank Mittelbach then discussed some of the dos and don’ts that he learned while preparing the *LATEX Companion*. From the discussions following the talk it seemed that his impressions were shared by many other authors/editors who are in the publishing business.

Just before tea it was Yannis Haralambous who showed off his artistic talents using METAFONT when he presented his work on typesetting the Holy Bible in biblical Hebrew using his *Tiqwah* system, that will make it possible, for the first time, to use the typographic powers of TEX to typeset high-quality Bible editions. Together with his work on typesetting the Holy Koran using several thousand ligatures, and his font developments for many other scripts, (as

[1]For another view of TUG94, see *Malcolm’s Gleanings* later in this *Baskerville*.

described at earlier conferences, and later in the present one) this will allow scholars in many disciplines to typeset their works at affordable prices using TEX and any computer.

Michael Cohen, an American teaching at the University of Aizu in Japan, explained how his *Zebrackets* system of meta-METAFONTs can generate striated parenthetical delimiters on demand. This offers the reader a more complete graphical picture of the relationship between various document elements by augmenting the information content of their representation.

Yannis Haralambous then came back on stage to present "Humanist", his new system to "humanize" LATEX. Document input, markup and editing is performed using any word processor that supports RTF output (like Word, WordPerfect), that will then be turned into LATEX code by the Humanist system. A user can thus work on a text in the most friendly and natural way (*i.e.* without a single LATEX command), but will get syntactically correct LATEX output so that the powerful TEX engine can be used to obtain high-quality typeset output.

The final paper of the Sunday was by Basil Malyshev, on converting METAFONT fonts automatically into PostScript Type 1 outlines. It was read by Alan Hoenig in the author's absence. Various techniques to perform the conversion in question were presented and the one chosen for the creation of the *Paradissa Fonts Collection* was described. This collection offers a freely available set of PostScript Type 1 renderings of all Computer Modern, Euler, CM Cyrillic and LATEX fonts.

3 Colour, and LATEX

Leslie Lamport started the presentations of the second day. He gave us his ideas on "LATEX4", a WYSIWYG-like, though structured text editor, well integrated into the user environment.

James Hafner gave a short historical overview of how colour was first implemented in Tom Rokicki's dvips `.dvi` driver to provide an efficient and simple method for specifying colour with TEX. Tom Rokicki then discussed a new implementation of colour support and proposed a standard way for specifying colour and colour-like specials, implemented by modular C-code, that can be easily integrated into the `.dvi` drivers. Angus Duggan described his program DVISep, a simple colour separator for `.dvi` files, as well of some other tools for working with `.dvi` files. Sebastian Rahtz provided an introduction to the colour commands available in LATEX 2_ε and showed some interesting examples. Michel Goossens discussed some of the more basic issues concerning the use of colour in documents. He emphasized that the colour dimension has to be used with great care, so as not to distract the reader from the main message. Colour, like typography, has a set of rules, that have to be learnt and applied for greater effectiveness. Friedhelm Sowa presented his original and device-independent approach to colour support and showed some results obtained using BM2FONT on a Hewlett Packard inkjet printer. Michael Sofka gave an overview of the various stages in the production of a colour book. He addressed the issues involved in professional colour separation, and demonstrated how TEX, with a suitable driver, can be used to produce high-quality custom and process colour books. Then Sebastian Rahtz returned to the spotlight, with a presentation of PSTricks, a paper by Denis Girou and Timothy van Zandt, who could not be present. Sebastian, in his usual clear style, showed how PSTricks provides a convenient interface to PostScript from within TEX. It allows one to draw any kind of graphics object, like circles, polygons, curves, springs. It offers several drawing tools, grids and has various commands to place text along a path. Objects and text can be rotated, scaled and tilted, and 3-D effects are available. Framing and clipping are supported, as is a general tree-drawing package. A package for generating slides, `seminar`, exists, and an early version of a plotting package is also ready.

After the presentations on colour our attention turned to the subject of general LATEX-related developments. First, Jon Stenerson showed us his system for creating customized LATEX style files via a graphical user interface, composed of menus, windows, and dialog boxes. It is at present closely linked to the Scientific Word text processor, although, in principle, it could be used with any LATEX environment. Johannes Braams provided a clear introduction to classes and packages and LATEX 2_ε. He started by relating the LATEX 2_ε packages and classes to LATEX 2.09 major and minor styles. Then he discussed how old styles can be most easily upgraded. In the last part of his talk he gave a concise overview of the document classes and packages that come with LATEX 2_ε. The last talk of the day was by Alan Jeffrey, who covered the subject of using PostScript fonts with LATEX 2_ε. He described the LATEX 2_ε font packages `psnfss` and `mathptm` and some of the design decisions made in their development.

Before the dinner "on the beach" several BoF sessions took place. One was on "colour", coordinated by David Carlisle, another on "practical indexing", coordinated by Nelson Beebe, and one on "font encoding", coordinated by Alan Jeffrey. Many of the discussions in the BoFs carried over into the beach dinner time, but, as families were also present, other more mundane subjects were also addressed. It was one more golden occasion to get to know each other in a more personal context, without reference to glue, (coloured) boxes or other TEX speak.

4 TEX Tools

Tuesday morning was devoted to "Tools", and started with a presentation by Oren Patashnik, the author of BIBTEX. He first took a look back and explained why some of the design decisions of BIBTEX were made. Then he discussed some of the features that he plans to include in the new version, such as an easier interface to create non-standard bibliographies, support for national languages and the possibility of multiple bibliographies in a single document. The next talk was by Pierre MacKay, who presented his typesetter's toolkit, which includes tools for remapping fonts and generating composite glyphs, and a program for generating AFM PostScript metric files for the Computer Modern fonts. Michael Barnett described a remarkable application where a combined use was made of electronic typesetting and symbolic computations. His work seems to indicate that a considerable amount of time and effort can be saved when complex formulae are obtained symbolically by a computer program, like MATHEMATICA. Minato Kawaguti, of Japan, proposed a new and efficient method to edit (LA)TEX source files by combining an emacs-type editor and a special version of `xdvi`, where the two windows (emacs and `xdvi`) are displayed simultaneously, and pointing to a portion of the document in the `xdvi` window positions the text in the editing window in the same region.

After coffee Yannis Haralambous showed his work on the Indica system, and a completely new TEX system for Sinhalese. The Indica system is a generalized preprocessor for Indic scripts (scripts of languages used on the Indian subcontinent, plus Sanskrit and Tibetan). Urdu, where the Arabic script is used, is not supported. Various input encodings are accepted and with the help of `flex`, a UNIX-based lexical analyser generator, are translated into TEX commands. Identical input encodings can be used for different languages, thus minimizing user retraining when inputting in different languages. The Sinhalese TEX system is a complete typesetting workbench for that language, containing specially designed fonts. Jean-luc Doumont explained how pretty-printing of Pascal programs can be done entirely within TEX, without the need of a preprocessor. He showed how this approach of "preprocessing within TEX", using two-token tail-recursion, can also be applied to other situations, *e.g.* , for an elementary chemistry mode.

After lunch we had the afternoon off and most of us spent it in the nice town of Santa Barbara. In fact, during the Tuesday afternoon we were supposed to go and have a look near the Santa Barbara Channel Islands, that provide a shelter for the area between the islands and the mountains, thus giving Santa Barbara its unique sub-tropical climate. The plan was to go and spot a few whales, but the sea was somewhat rough, and the captain preferred to take us on a 3-hour tour along the coast. Even so quite a few of our passenger-colleagues felt sick, and it was with some relief that many of us set foot ashore again around 7 pm, and set off to go and pick a restaurant to enjoy the local food.

5 Futures

The next day's theme was "Futures", and Joachim Schrod thought that interactivity was the way forward. He emphasized that Knuth already very early on thought that an interactive TEX would be useful. Many TEX systems have been built that contain some interactivity. To better understand the actions of TEX he proposes that a formal approach should be used since, according to his views, informal descriptions have failed. As part of a solution he presented, after developing an abstract decomposition, a formal description for TEX's macro language. The latter can be interpreted by a Common Lisp system and the resulting Executable TEX Language Specification (ETLS) can be used as the basis for a debugger of TEX macros. Chris Rowley then reviewed some of the investigations of the LATEX3 team in the area of modeling and specifying page layouts. One of the questions that they asked themselves was how well LATEX can cope with that job compared to other text processing software systems, and whether a complete redesign of the system is needed. He also mentioned the wider question of how these aspects should be addressed in future typesetting systems. Don Hosek gave an overview of various page layouts he had tried for his new magazine *Serif*, and showed how he could massage TEX into doing (almost) everything he wanted, mainly using code from the infamous Appendix D of *The TEXbook*. John Plaice then reported on the present status of the Omega project, which is a series of extensions to TEX to improve its multi-lingual abilities. It supports multiple input and output character sets and allows any input encoding. Transformations from one coding to the other are supported. Even scripts requiring a very complex contextual analysis, such as Arabic or Khmer, can be handled elegantly using 16-bit or 32-bit virtual fonts.

After a short break Arthur Ogawa showed ways of combining within TEX the descriptive markup and object-oriented programming (OOP) paradigms. He discussed an extension to LATEX's markup scheme that more effectively addresses the needs for a production environment, and for implementing such a system he heavily relied on the use of OOP techniques, where LATEX environments can be thought of as objects, and several environments can share functionality of a common, more general object. In his companion talk to Ogawa's, William Baxter went on to describe the actual implementation of an OOP system in TEX, where formatting procedures and markup are strictly decoupled, so that, indeed, designers can fully benefit from the OOP techniques available.

The afternoon started with the TUG Business meeting, where decisions taken by the TUG Board of Directors for the coming year were presented, explained, and discussed. These decisions will be presented elsewhere. The Knuth Scholar was also announced: Shelly Lee Ames of the University Manitoba, where she works for the Canadian Mathematical Society (Société mathématique du Canada) preparing formats and proofing all papers published by the society in their Journal and Bulletin. This involves handling submissions in many different flavours of TEX, and initiating the development of macros to implement their formatting requirements.

After the meeting Yannis Haralambous, in a companion paper to Plaice's on the Omega project, showed a few applications for fully diacriticized scholarly Greek, vowelized Arabic, properly kerned Khmer, and for Adobe's calligraphic Poetica font. Then Phil Taylor reported progress on the NTS project. This project was started in 1992 by the German-speaking TEX user's group, DANTE, and has as its main task the development of a successor to Donald Knuth's now frozen TEX system. In fact two paths, one evolutionary, with e-TEX, and one more revolutionary, with NTS (New Typesetting System) are at present being investigated. As the TEX typesetting system consists of a rather complex set of tools, the group proposes to define a "canonical TEX kit", which is assumed to be present at every installation. The status of the e-TEX project was reviewed by Peter Breitenlohner. At present this involves improved control over tracing, additional math delimiters, improved access to the current interaction mode, checking for the existence of a control sequence, alternative ligature/kerning, extensions to the set of valid prefixes for macro definitions (*e.g.* , `\protect` and `\bind`), support for colour. Finally it was Jiří Zlatuška who told us about the team's present thinking on the more ambitious NTS project. He sees essentially a two-phase approach, namely first a re-implementation in a rapid-prototype language such as CLOS or Prolog, so that one can experiment easily with various modular representations of the present TEX engine. Using this model one will try and identify functionally independent units, for which various alternate ways of extensions can then be proposed and tested. Based on the knowledge gained in phase one, the second phase will then see the step-by-step re-implementation of the functional units in a more efficient and widely available programming language, such as C++. Initially only e-TEX will be implemented in NTS, but later on alternate algorithms can be included to perform some of the typesetting tasks better. The long-term aim of NTS is thus to make maximum use of the phase-1 test bed to investigate and evaluate possible approaches to overcome various of TEX's perceived shortcomings. A lively discussion followed these presentations, and then the participants went off into one of the three BoF sessions. The first was on WWW servers, coordinated by Peter Flynn and Norman Walsh, where the latter discussed at some length his paper describing his WWW interface to the CTAN archive, which provides an attractive means to combine different views of the archive into a single view. Marko Grobelnik coordinated a BoF on database publishing, while Oren Patashnik discussed extensions to BIBTEX in his BoF. At the Banquet, that started at 19:30, all participants had one last chance together with their families to socialize, and enjoy the good food, wine (some had original 16 year old cask Caol Ila malt whisky. . .), and the music.

6 Publishing and design

It was a little difficult for some of the participants to get up on time for the last morning. Yannis Haralambous and Maurice Laugier discussed some of the tools used at the Louis-Jean Printing house in Gap (France) to typeset books. The TradTEX-SGML program was introduced. It is used to convert TEX and LATEX files into SGML. The tool is presently implemented on a Macintosh and is in real-life production. eDVItor is a program that allows interactive editing of a `.dvi` file, using a mouse-driven cursor to move blocks of text, insert illustrations, change colours, etc. It runs on both DOS and Macs. Michel Downes stated that the American Mathematical Society produces almost all its publications (a couple of dozen journals and book series) with TEX using AMS-developed macro packages. About two years ago a major overhaul of the macros package was decided, one of the goals being to ease revisions to the visual design. In this new approach the design specifications are kept outside of the TEX code in an element specification template that is relatively easy to understand and modify by traditional book designers. Alan Hoenig then showed us some examples of visually pleasing page layouts, which most TEX users only thought possible with PageMaker or Quark Express. His secret is to turn off some of the TEX functions, like vertical glue or tall characters, and all lines are assumed to have the same height and depth. It is to be said that this arguably restrictive set of conditions still allows one to typeset probably at least 99% of all printed material in the world. And, indeed, the model is not so limited as it seems, since with some work one can include section heads, display material, and so on. Just before the coffee break, Malcolm Clark presented Jonathan Fine's paper in his absence. He described first some historic aspects of the TEX typesetting program, leading to a discussion of strategies for possible future extensions. He strongly believes that with improved macro packages and `.dvi` processors many of the present problems will be solved. Also imposing a more rigorous syntax for input compuscripts should help. This will not only allow the source to be used with a possible future successor of TEX, but also ensure re-use with other, not-necessarily typesetting, applications.

Marko Grobelnik presented a TEX-based system developed in Slovenia for publishing dictionaries, lexicons and encyclopedia. The TEX macros are augmented with many special purpose written editing tools to assist the editor, who looks after the contents and form of the publications. The final talk was by Henry Baragar, who showed how special purpose ("small") languages can be used for documenting Knowledge bases so that LATEX can be augmented by adding expressiveness for specific tasks. He introduced the language TESLA, that allows Expert System analysts to mark up groups of rules into tables so that the logical structure of the database becomes clear. The system generates LATEX tables, that can be typeset in tabular form to be used by expert system programmers or typeset as text, to be used by Domain experts, thus yielding presentation forms adapted to the targeted audience.

The conference was brought to a close by Christina Thiele, but not before Mimi Burbank, coordinator of next year's TUG meeting, gave us a short outline of plans for the 1995 meeting, to be held during the week of July 24–28th 1995 in the Trade Winds Hotel in Florida. It was also the occasion to honour the winners of the trophies for the best papers, namely Alan Hoenig, Yannis Haralambous and Tom Rokicki, who were presented with EPODD CD-ROMs by Nelson Beebe.

7 Conclusion

I think that I can safely suppose that at the end of our five day conference all participants left the University of California, Santa Barbara Campus satisfied to have taken part in this unique event. Even though most of us, Internet addicts, were a little surprised to find only very limited access to the Internet, this fact might indeed have been more of a blessing than a shortcoming, since in this way we were not distracted by having to answer e-mail or otherwise respond to "urgent requests" from home. In any case it certainly benefitted contacts between the participants and hence contributed to the friendly atmosphere. Another positive factor was the hard work of John Berlin and Janet Sullivan of the TUG office, who did their best almost 24 hours per day to help solve problems, or better, trying to prevent them before they occurred. Their kindness and helpfulness were truly appreciated by all those present. Thanks once again to John, editor of the *The TUGly Telegraph* (and his partner in crime, Malcolm Clark), which kept us informed of the latest conference news, and to Katherine Butterfield, Suki Bhurji, and Wendy McKay for helping with staffing the on-campus TUG office.

VIII The National Typesetter Users' Forum (NTUF)

Philip Taylor
Chairman, National Typesetter Users' Forum
`P.Taylor@Vax.Rhbnc.Ac.Uk`

Although the majority of TEX users are content to produce their final copy using a laser printer or similar, those who are preparing so-called 'camera ready copy' for use by professional publishing houses, printers, etc., need to be able to produce their final copy to a somewhat higher standard. A typical laser printer operates at 300 dpi, which will produce acceptable results only if (a) the typeface is not too small, and (b) the typeface does not exploit exceptionally thin lines (Computer Modern at 300 dpi is noticeably poor in this respect). A better quality laser printer operates at 600 dpi, and at this resolution both small fonts (say down to 5 pt) and thin lines (as in Computer Modern) can be resolved reasonably well, although an unfortunate combination of both a small font and thin lines will still usually lead to break-up.

Phototypesetters start where laser printers leave off; the lowest resolution of a typesetter is of the order of 635 dpi, and resolutions of 1270 and even 2540 dpi are by no means uncommon. At 1270 dpi, fonts as small as 3 pt, and extremely fine lines, can both be resolved reasonably well, and for normal textual work there is usually no need to consider higher resolutions. However, if gently sloping lines (usually from a graphic or from a custom glyph) are to be resolved without the eye detecting a disturbing step function in their rendering, then the highest possible resolutions, of 2540 dpi or more, are required.

The National Typesetter Users' Forum provides an opportunity for both existing and potential users of a phototypesetter to meet to discuss problems of common interest. The meetings take place both physically (the group meets once per term) and electronically (there is an e-mail list, `Typesetting@Vax.Rhbnc.Ac.Uk`); at the physical meetings there are regular reports both from service providers (*e.g.* the Phototypesetter support group at the University of London Computer Centre) and from what would elsewhere be termed 'special interest' groups (*e.g.* TEX, PostScript, Apple Macintosh, IBM PC, etc.) The most recent meeting was also addressed by a guest speaker (on this occasion, Ian Chivers speaking on Adobe Acrobat), and it is hoped to arrange further speakers for forthcoming meetings.

All members of the UK TEX community, whether or not they are already users of a phototypesetter, are welcome to join the group; those with access to e-mail may send their electronic subscriptions to `Listserv@Vax.Rhbnc.Ac.Uk`, in the normal Listserv form (Subscribe typesetting *given name SURNAME*), whilst those restricted to more traditional means of communication should send a note or fax to Ian Chivers, NTUF Secretary, The Computer Centre, Kings College, University of London (E-mail: `I.Chivers@Kcl.Ac.Uk`; telephone: 0171-333 4339; fax: 0171 937 7783).

The next (physical) meeting is scheduled for 14:15 on Tuesday 18th October at the University of London Computer Centre; anyone wishing to take part in a pre-meeting *dim sum* lunch is invited to contact me personally for further information. I hope to see many of you there.

IX Malcolm's Gleanings

Malcolm Clark
m.clark@warwick.ac.uk

1 TUG94, The Conference

My impressions of the TUG94 conference in Santa Barbara will be pretty general:[2] I did not sit through every session and listen to every talk. To be frank, that's not really what I go to these events for. Since we had the preprints as part of the conference pack, I could (if I wanted) flick through and pick out the potentially interesting ones. Or better, see what was really dire, and ignore them. Since I seemed to be roped in to other conference stuff anyway, I kept having to disappear and find people.

One distressing feature I did note was the inability of many speakers to address an audience. We are in a fairly large auditorium. Fortunately there are microphones, but in US style these are fixed rather than throat or lapel mikes. This does make mobility a problem, especially when you are trying to use overheads. So many people turn to the projected slide and point to it instead of pointing to the slide on the ohp and talking to the audience. A microphone simply does not catch your voice if you turn the back of your head to it. Honest. One other thing I notice is that the TEX Users Group (or perhaps TEX users) have little charisma. I suppose when the material is so worthy (*i.e.* high in *content*), the presentation (*i.e.* the *form*) shouldn't matter. I'm sorry, but it does. But again, when addressing an audience of presumed converts, perhaps we shouldn't worry about a lack of presentation skills. Again, I think not. It does make us look very amateurish, and not everyone in the audience is a convert.

The conference had a number of 'big names'. At least, it had some people who were well known, but not frequent attendees at the annual meeting. The first coup was Chuck Bigelow, who gave an entertaining enough talk, but its relevance to TEX was not clear. Leslie Lamport's contribution was interesting, although when he started talking about LATEX4 a shudder seemed to run through the LATEX3 team. He had something to say about structure editors, but informal discussions later suggested that he maligned them unfairly. If you want to visualise LL as you read the LATEX book, the Bibby lion cartoons in it are remarkably similar. Oren Patashnik also talked about BIBTEX. I had imagined someone at least seven feet tall. Perhaps the other 'newcomer' I was hoping to see was Norm Walsh whose book 'Making TEX work' had just been published. (He nearly is seven feet tall.) Apart from that it was the usual gaggle of TEXies.

In general the conference seemed to run smoothly, or at least, not many people saw the hitches. The most obvious hitch was the lack of tea or coffee on the afternoon of the first day. The overhead projectors could have been better. The vendors could also have had a better deal. To get to the vendors you had to poke your way through an apparent dead end, past a few bins and through a nondescript door. And all they had were a few tables.

The social programme was slightly chaotic: it started with a reception where keg of Sierra Nevada turned out to be Michelob, but the bowling turned out well, with some pleasant surprises (the usual performances from Nelson Beebe and Ken Dreyhaupt, and a cute native American rain dance from Don 'do people really think I'm a nerd' Hosek); the barbecue at the beach benefited from some real Sierra Nevada (as well as copious quantities of other comestibles); the boat trip was apparently a success, despite some upchucking and no whales – and much confusion on how or when to get to the boat; the banquet (a buffet, actually) was limited in choice, but agreeable enough, and the music improved enormously as the evening progressed – enough to get a surprising number of people on their feet, notably Tom 'party animal' Rokicki.

Now, I wouldn't like you to get the idea that we're only here to have a good time, but the social side really is valuable. You end up talking to all sorts of people and probably learn more useful stuff this way than in the rest of the conference. I toy with the idea of having one single parallel session and devoting the rest of the time to constructive socialisation.

The 'Tugly Telegraph' made its appearance each day. It's useful, since it has a more accurate daily programme, as well as instructions on how to get to 'events', and other general bits and pieces. It is perhaps less successful than last year's at Aston, but then, its editor, John Berlin, is doing other jobs too (unlike last year's editor), and only occasional

[2] Readers who want a different view can peruse Michel Goossens' article earlier in this issue of *Baskerville*.

extra help with the newsletter. In any event, he manages to get each edition out before midnight on the preceding day. The crossword flops: no correct entries are submitted. Peter Flynn is obviously too subtle or devious.

I'm told that the TUG general meeting overruns. This was one event I was determined to miss. The next two talks are more or less cancelled. As a result, there is a proposal that next year's general meeting will be open ended. This is naive. Given a choice between a general meeting at (say) four o'clock, which might run on till the evening, and catching a few rays on the beach, I know where I will be. On the other hand, slotting the meeting in at the beginning of the afternoon, I might just be carried along by inertia and attend. Of course, I'm jaded by the TUG board stuff. I've been there and I know that nothing changes, no matter how strongly you feel about it, and how sincerely you want to get things done. By not attending, I surrender my rights to comment. But I had a wonderful afternoon instead. How often do you get the chance to swim with dolphins?

Apparently there is also some bizarre notion to reduce the membership fees, but to make *TUGboat* optional. Somehow TTN will become a more general 'journal', carrying some of *TUGboat*'s present material. What present material you may ask? It is now August 12th and no sign has been seen of the second edition of 1994 (volume 15 number 2). *TUGboat*'s calendar suggested that this edition would be mailed on May 23rd. When last year's final copies came out more or less on time I had supposed that it had finally managed to get its act together and was to be produced on a regular and reliable basis. Clearly I was deluded. What is the problem? I refuse to accept the usual story that it is a complex journal and that to achieve the standards required the devoted and underpaid or unpaid editorial volunteers have to devote limitless time and energy to it. *TUGboat* is dying at the altar of quality. If the journal is to have any credibility it has to come out regularly. Maybe it really is too complex and TeX is not really up to the production. Commercial publishers – to whom we direct much encouragement to use TeX – could not allow themselves to be sucked into this cuckoo's nest. TUG has to try to be realistic and trim the sails of *TUGboat* so that it can leave port. There are enough enemies of TUG, inside and outside the user group, who wish to see it dismembered, and who do not need to be able to point to *TUGboat* to see graphic demonstration (or non-demonstration) of the health of the whole organisation.

Another canard flies: despite the manifest evidence that this is an international conference (add up the speakers from outside the US) the old bogey that TUG is essentially a North American organisation reappears in discussions with some board members. They want some umbrella organisation to be formed from representatives of TUG (North America TUG), and the other user groups, which will somehow 'direct' TeX research and development. A likely tale. However, if TUG does uncouple itself from *TUGboat*, this could be a serious proposal. If *TUGboat* is separate, I won't buy it, because the package of TUG plus *TUGboat* membership will be too expensive. The only benefits that remain of TUG membership are cheaper fees to the annual meeting, and TTN. Only a very small proportion of TUG members go to the annual meeting (about 140 this year), and frankly, *Baskerville* is a far better deal than TTN (and similarly for most of the other user group newsletters). Anything important will appear in the local newsletters. So membership of TUG will decline further, since there are no perceived benefits.

Eventually the conference winds down. Christina Thiele — out-going (no pun intended, or even possible) President of TUG — makes the closing announcements, failing to thank any of the local people who actually did make the conference work. Let me then record a sincere vote of thanks to John Berlin and Janet Sullivan of the TUG office, who were the 'official' TUG representation, and who held the whole thing together. Similarly, the volunteer helpers of Suki Bhurji, Wendy Mckay and Katherine Butterfield were indispensable. Conferences don't run themselves. Since John is now leaving to continue his studies at UCLA (doing a course on multimedia) he will be sorely missed at the TUG office.

What does next year hold? St Petersburg: the one in Florida, not the revisionist Leningrad or Petrograd. We are promised a hotel venue and an appeal to the publishing fraternity. My heart sinks into the alligator infested swamps.

I think the conference was, on the whole, good value. It was probably too long. There is always a problem about fitting the talks in, and thoughts are expressed that some of the talks should not have been presented. This would of course cut down on the overall length. I honestly don't know. The written abstract which speakers submit is rarely a good yardstick for selecting the papers. The best suggestion I have heard was from Angus Duggan, who suggested a day in which speakers had ten minutes each to present their abstracts, then a massive set of parallel sessions the following day(s). You choose what to go to on the basis of the ten minute abstract. It might be worth trying. At least we would then have some time for the informal discussions and scheduled workshops. The venue was certainly good, the residences were fair, the food edible, the lack of a bar was a blow, the lecture theatre was too far away, the beach was excellent, the sun shone relentlessly, there were plenty meeting rooms/common rooms in the residence, conference services tended to verge towards the non-sexist airhead quality, the TUG helpers were overstretched. I do think it gelled pretty well. It's the most enjoyable TUG conference I've been to. An experience worth sharing.

2 Offizin

Whenever I pontificate about publishing with TEX, someone will always bring me to earth by pointing out that the proceedings of the 1988 TEX conference in Exeter took an interminable time to hit the bookshops. The figure is about two years (I was busy…). It was therefore a pleasant relief to receive *Offizin* earlier this year This is a production of DANTE, the german-speaking TEX group. It is a publication designed to disseminate some of the lectures given at the group's 'TEX days'. I worked out just when I presented the paper which is produced in translation: it was February 1991. That makes the TEX88 book look much less laggardly! Of course, what I had to say, about *TEX in Europe and America*, is hopelessly out of date, but when it appears in my list of publications, no-one will know that!

Putting this schadenfreude aside, it is an interesting volume. It should be the first in a series, a series published by Addison Wesley (Germany). According to other bits of Addison Wesley, they don't do conference proceedings, so someone did some fancy footwork to get this through. Well done.

One quote I managed to extract was 'typography has its experts, but they have no audience'.

X Topical Tip: making the TOC tick

R. A. Bailey
Goldsmiths' College, University of London

Question 1 I have a problem that I have not been able to solve by reading *LaTeX: A Document Preparation System* by Leslie Lamport. How can I force a table of contents to have entries for 'preface', 'bibliography' and 'index' (for example, like the table of contents of *The Manual* itself has)? For example, if I use the `\chapter*{preface}` sectioning command, no entry for the table of contents is generated; if I use explicit commands such as `\addcontentsline{toc}{chapter}{Preface}`, it works for the preface but it generates incorrect page numbers for the index and bibliography (maybe I put the commands in the wrong place, but it is not obvious to me where exactly I should put them).

Answer The best way to get headings of funny 'sections' like prefaces in the the table of contents is to use the counter `secnumdepth` described on pages 157 and 160 of *The Manual*. I use

```
\setcounter{secnumdepth}{-1}
\chapter{preface}
```

Of course, you have to set `secnumdepth` back to its usual value (which is 2 in the standard styles, I think) before you do any 'section' which you want to be numbered.

This is why it works. `\chapter` without the star does

1. put something in the `.toc` file;
2. write the chapter title;
3. if `secnumdepth` ≥ 0 then increase the counter for the chapter and write it out.

The above behaviour is much more predictable than `\addtocontents`, which, in my opinion, should be avoided if at all possible.

XI Moving the UK CTAN

Martyn Johnson (maj@cl.cam.ac.uk)
and
Robin Fairbairns (rf@cl.cam.ac.uk)

1 The background (RF)

The UK node of the Comprehensive TEX Archive Network () has a long and honourable history, which starts long ago in the recognition (by Peter Abbott at Aston — see [Abbott 1990]) of the need to provide an archive of TEX-related material within the UK. At the time that the Aston archive was created, TEX-related material was mostly made available *ad hoc* by its originators — there was no site with ambitions to provide a *complete* set of systems, macros, and so on. Furthermore, access from within the UK to overseas material was less than straightforward (access to ftp, using an account at UCL, was severely restricted).

The Aston archive was originally a VMS-based facility, offering connection via the Janet coloured-book protocols to machines that were part of Aston's centrally-provided service. Later, the archive group were given a second-hand VAX, and later still a parallel version of the archive was established on a SparcStation that sat on Peter Abbott's desk. This machine (with the net name `ftp.tex.ac.uk`) eventually became part of , offering access via anonymous ftp to all and sundry throughout the world.

At the beginning of this year, Peter Abbott told your committee that he would be retiring (early) at the end of July; the implication was that it would be unlikely that we could count on Aston's willingness to offer a home to the archive beyond that date.

The committee discussed whether it was reasonable even to consider maintaining a CTAN node in the UK (covering the world with two sites in Europe and one in the USA is hardly a convincing approach); and if the node was to be retained, the implications of so doing. After much soul-searching, we decided that it wasn't reasonable to spend UK TEX Users' Group funds on a new home for the archive; we concluded that our membership is such a small proportion of the TEX community within the UK[3], that group funds really *shouldn't* be expected to cough up for support of the community at large.

In parallel with these discussions, the committee investigated alternative new sites for the archive; candidates were the Universities of Warwick, Sussex and Cambridge, the National Typesetting service (at Oxford) and the Hensa[4]. While the typesetting archive looked promising at first, they eventually suggested that we should approach Hensa (which had already been on our list of possible candidates). Hensa in fact maintains two archives, one for micros and one for Unix; since is a cross-architecture service, offering support for micros, Unix machines and others, it was obvious that Hensa couldn't maintain a node. Sebastian Rahtz, who's the only volunteer effort that the UK node has, was unwilling to maintain an archive that wasn't , so we decided to look elsewhere.

Warwick (in the person of Malcolm Clark — he of the *Gleanings*) concluded that they would need to be provided with some extra disk space before they could offer to host the service. Sussex would very much like to host the service, but were unwilling to do so until their connection to SuperJanet went live. Cambridge expressed an early interest, whereafter very little (that was visible to the committee) happened for some time while Martyn Johnson established what was necessary and acquired agreement from the head of the department and from the rest of the Computer Laboratory's systems group.

2 The Archive Operational Requirement

Towards the end of June Sebastian Rahtz wrote an 'operational requirement', which described the facilities provided by the Aston archive, and explained which were essential and which might, at a pinch, be dropped.

The complexity of it all was quite a surprise. The primary service is an ftp server, but it needs to be able to perform

[3] In contrast to the situation, say, in Germany, where DANTE owns the archive machine.
[4] Higher Education National Software Archive.

many automatic transformations, such as packing a directory into a zip image. Gopher and WWW services were also provided at Aston. Behind the scenes, there was a mechanism which kept the main nodes in step with each other whilst maintaining a peer relationship between them. The archive maintainers needed remote access to the machine and the ability to manipulate the archive. Finally, there were several well-publicised mail addresses at ftp.tex.ac.uk.

3 Meeting the Operational Requirement (MAJ)

Whilst it was clear that we had the resources available to run the archive, there were a number of awkward decisions to be made. The main question was "which machine?"; we certainly didn't have a machine available to dedicate to the task. My initial assumption was that I would put the service on one of our main fileserver machines, a Digital Alpha/AXP 3000/500S, since that machine had plenty of disc space available. But I was worried about security — we don't even allow our own users to log in to our fileservers. Then I realised that one of our most public machines (another Alpha) had a 1Gbyte disc which could be made available, and the decision was made. A welcome side effect of this choice was that it made it easier for me to provide external access to the archive through NFS without compromising the security of the internal services.

Transferring the data from Aston to Cambridge was the least of the problems. Not very long ago, moving such a large amount of data over the network would have been unthinkable. I just did it, using a standard "mirror" script running in the background. It took about a day to pull the archive across, and a few minutes each day thereafter to keep it up to date.

It was obvious that the standard ftp daemon supplied with OSF/1 wasn't up to the job of running the archive. The only sensible choice was wu-ftpd, as used at the other nodes. This compiled easily enough for the Alpha, and initially appeared to work. However, none of the document conversion scripts would function at all, and the bug[5] was only found after a great deal of detective work. Once found, the problem was easy to fix, but even then, one of the document conversion scripts still didn't work[6].

After those and other bugs[7] had been mended, there were still the Gopher and WWW services to consider. We had not previously run a Gopher server, and I was not very keen on doing it, but Sebastian assured me that the service was well used and would be missed if we didn't provide it. In the end the Gopher service turned out to be straightforward to make available, though we decided to not to attempt to set up WAIS indexing initially.

WWW was even easier, since we already ran a suitable server on another machine, and we merely had to copy the data into it. Or did we? All of the Aston archive's services were published as being available from the machine ftp.tex.ac.uk, and this could obviously not be made an alias for two different machines. I really did not want to run a second WWW server, so we decided that the world would just have to change. Each service offered by the archive now has its own name: ftp.tex.ac.uk, gopher.tex.ac.uk, nfs.tex.ac.uk and www.tex.ac.uk. While we were at it, we thought that a better name for the mail domain would be simply tex.ac.uk, though we intend to continue to accept mail using the old addresses for some time.

Of course we cannot force people to use the correct name. In fact only the mail and WWW services are provided on machines other than ftp.tex.ac.uk. Mail is not a problem since there are well-established mechanisms for redirecting it to a site hub. In order that the WWW service should not seem to have disappeared to people following old links, I wrote a tombstone service. No matter what you ask for, it always returns a fixed page, which explains that the server has moved, and offers a link to the new place.

The mail facilities were provided by configuring a logically separate mail system within our mail hub, which runs the PP mailer. We did not want to do anything which would make a further move of the archive more difficult, so we were very keen to ensure that users could not confuse the mail domains tex.ac.uk and cl.cam.ac.uk. We cannot prevent them being confused, but at least we can ensure that they will be told that they are confused by having incorrectly addressed mail returned to them.

It was always part of the agreement that the remote management of the archive by Sebastian Rahtz, David Osborne and others should continue. It is easy enough to give those users login access to the archive machine, but they needed to be able to manipulate the archive without needing "root" privilege. The archive is owned by a pseudo-user "ctan", and the archivists do their work using a local command which allows them to pretend to be ctan, provided that they

[5] Overwriting the daemon's own argument and environment strings for cosmetic reasons, so that subprocesses start up with a corrupt environment which the OSF/1 shell takes umbrage at.

[6] But it didn't work at Aston either, and has been mended.

[7] Notably the bug that made zoo's 'portability library' not port to 64-bit machines.

can quote the password. Much of the maintenance of the archive is handled automatically, by passing mail around and running periodic jobs. All we had to do was arrange to deliver the mail — Sebastian did the rest!

All of a sudden, it seemed that we were ready to go live. We had already taken over control of the `tex.ac.uk` domain of the DNS (a saga in itself) and simply had to flip over the addresses and watch the traffic come in. And that is exactly what happened.

Three days before he retired, Peter Abbott sent a message to the committee mailing list saying that Aston did indeed want to reallocate the Archive machine from the day that Peter left; the move was 'just in time'!

4 The first six weeks (MAJ)

We received remarkably little mail concerning the changeover, so it must be considered a success. A handful of people had bound to the old address and needed to be told the new one, but it seems that the vast majority of clients simply did not notice any change. One person complained that the Gopher service was not quite what it was, and this is indeed an area which needs some more work. So far, after nearly 6 weeks of operation, the new archive has shipped some 11Gbytes of stuff via ftp. Equally encouragingly, there have been no complaints from local users of the machine, for whom it is a compute server, that the archive activities are having any impact on their work. In fact the Alpha is a very fast machine, and provided that there is enough memory, a few ftp sessions are hardly noticeable.

By a curious twist of fate, the service offered in the first few weeks has not been as good as we might have hoped. One of the reasons we chose to use the Alpha machine we did was that in practice its hardware and software have been highly reliable. Unfortunately there have been three serious problems since we transferred the archive.

The first was a spectacular thunderstorm in the Cambridge area which caused widespread disruption to our equipment. The server itself recovered with little difficulty, but the network problems were more serious and our whole department was cut off from the outside world for some time.

The next weekend, the archive machine itself failed. The front panel lights were dim and flickery, but this 'obvious' power supply fault was actually being provoked by a faulty system disc, so a replacement had to be sent for, and the system had then to be restored from backup tapes. This meant that the service was down for most of the weekend and the following Monday.

The third failure again started during a weekend. The machine simply froze, and when rebooted, did exactly the same thing again. The problem was tracked down to the NFS server, which was being fed "poison packets" by a machine in Norway. The machine's administrator didn't respond to mail, but back-door intervention on the archive machine had the required effect. We're told that the bug in OSF/1 which made it crash on receiving the bad packet is mended in the next release.

Fortunately the user community has been very tolerant of these early problems, and we very much hope that the next few weeks will be better.

5 Conclusion (RF)

The archive has grown from humble beginnings as a side service on a VAX/VMS machine using protocols only (significantly) available within the UK, to become part of an internationally-coordinated group of archive sites offering services to anyone who chooses to use it in the world at large. As a community, we have many reasons to thank Peter Abbott for his foresight, and Aston University for hosting the archive for many years.

For the time being, Cambridge University has taken over the torch; let us hope that the network can continue to serve as a beacon leading progress in the use of TEX world-wide.

References

[Abbott 1990] Peter Abbott. *UKTEX and the Aston Archive*. In Malcolm Clark, editor, *TEX Applications, Uses, Methods*, pages 109–114. Ellis Horwood, 1990. Proceedings of TEXeter (1988).