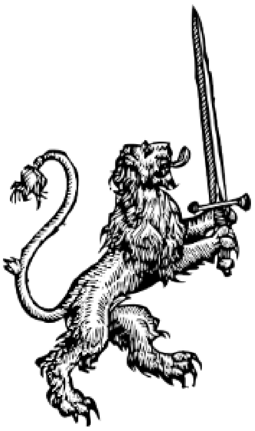
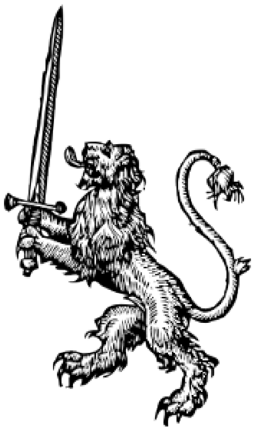


Baskerville



The Annals of the UK-TEX Users' Group
uk.tug.org
ISSN 1354-5930

Edited by
Jonathan Webley

Vol. 10.2

October 2009

Contents

UK-TUG Committee 2009

- Jonathan Fine (Chair)
- David Crossland (Secretary)
- David Saunders (Treasurer)
- Joseph Wright (Membership Secretary and Webmaster)
- John Trapp (Training Officer)
- Jonathan Underwood
- Charles Goldie
- Simon Dales
- Jonathan Webley (Baskerville Editor)

The committee can be contacted at:

uktug-committee@uk.tug.org

1 Editorial

Welcome to my second edition of *Baskerville.* I have been working on this edition, on and off, for months with various interruptions. It's been hard work and I've learnt a lot, but, finally, I've managed to pull everything together, and now we're up to twelve pages. I'm mentally planning the next edition for early next year, though that does depend on my availability and the quality and quantity of contributions.

Jonathan Webley
baskerville@uk.tug.org

2 Events

2.1 TUG 2010

TUG 2010 will be held in San Francisco, California, USA, from June 28–30, 2010, in the Sir Francis Drake Hotel in San Francisco.

Don Knuth and other members of the original Stanford TeX Project will be there. There will be several special events, including the conference banquet, to honour their work and celebrate TeX's 32nd anniversary.

Abstracts and presentation proposals are welcome at any time, and hotel reservations are available. The registration form will be posted soon.

The official website is:

tug.org/tug2010

3 News

3.1 The TEX FAQ

The UK List of TEX Frequently Asked Questions on the Web was updated in June this year and now contains 438 questions (and answers). The FAQ is maintained by Robin Fairbairns and is for English-speaking users of TEX. The questions cover a wide range of topics, but the actual typesetting issues are mostly covered from the viewpoint of a LATEX user.

The FAQ can be found at:

www.tex.ac.uk/faq

Any errors, corrections and potential new topics should be sent to:

faq-devel@tex.ac.uk

3.2 MathTran

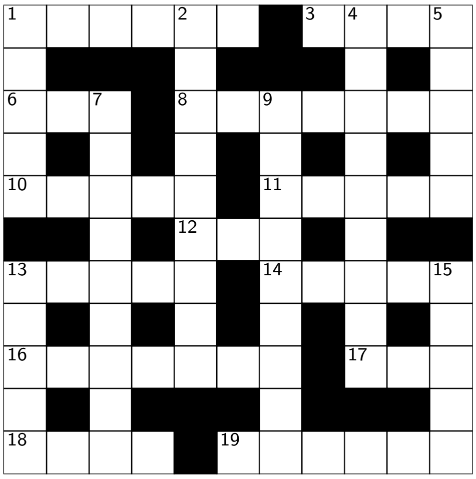
MathTran is a website that can be used to create, store and translate mathematical content. It translates mathematics entered using TEX notation into images for inclusion in web pages and on emails. The latest version now allows users with an account to save their formulas.

MathTran is funded by the OU and JISC. The website can be found at:

www.mathtran.org

4 The Hound

Jonathan Webley



Across

1 Inside the saucer, I see, is cherry-coloured. (6)

3 Requests are listed in the task sheet. (4)

6 I bought a bag in the Trossachs. (3)

8 Sadly, use a gas mask to relieve pain. (7)

10 Large hat, squashed, has okay from me. (5)

11 Species, drowning without a ship, have no part. (5)

12 This type of chart sounds like 3.14159 . . . (3)

13 There's an obelisk of steel, rusted. (5)

14 Lost in Iran, unknown at the end, wet. (5)

16 Quicken the best, but meant badly. (7)

17 So you want a coin? (3)

18 Duck mews annoyingly. (4)

19 Ah, the post was mislaid because it contains potassium. (6)

Down

1 Kegs for 100 demands. (5)

2 Sapper, with a double-O rating, watches a show. (4,5)

4 A skin pest disfigures in dangerous places. (5,4)

5 European sounds like leather. (5)

7 You! Extinguish the flames and stop fighting. (9)

9 Robin is found washed up on pure shore. (9)

13 Cormorants burp gas. Sh! (5)

15 Why art thou so young? (5)

5 siunitx: A LATEX Swiss army knife for units

Joseph Wright

5.1 Introduction

The siunitx package is a complete system for typesetting units in LATEX. It provides a large number of settings, allowing the user complete control of the output for a fixed input. The package grew out of SIunits and SIstyle, but covers many additional areas. This article provides a brief overview of why the new package was written and some of its key features.

5.2 Background

In November 2007, a seemingly simple query about a bug in the SIunits package [6] was posted to `comp.text.tex`. I suggested a bug fix, but on contacting the author of SIunits found he had no time for further maintenance. I therefore found myself somewhat accidentally as the new package maintainer. After fixing the bug at hand, I decided to have a good look over the code, and to ask for suggestions for improvements. It soon became very clear that unit support in LaTeX needed serious attention, and that tweaks to SIunits would not be enough. At that point, siunitx was born.

5.3 Units in print

The need for packages to control printing units is not immediately obvious. However, the existence of SIunits, SIstyle [3], fancyunits [1], unitsdef [4] and units [8] attests to the desire to go beyond direct input of units (and values).

Units are important, and without clear rules problems almost automatically arise. There are therefore internationally agreed units: the *Système International d'Unités* [2], usually referred to as SI units. To go with the agreed units are a set of rules on how to print both units and the associated values; the National Institute of Standards and Technology (NIST) have a good set of guidelines for authors [7]. Doing this properly and consistently is much better achieved using pre-build macros, rather than checking every single use by hand.

At the same time, different publishers have their own conventions. These do not always follow the 'official' rules, and authors do not want to change their source for every different use. This again makes the strong case for using adjustable macros for writing units; changing the style is then only a question of altering the definition.

Finally, values as well as units have rules, and these rules vary depending on publisher and country. The numprint package [5] provides a range of tools to automatically format numbers, to match the desired output style. It also includes basic abilities to include units with these numbers.

5.4 Enter siunitx

Given the existing range of packages, the question arises 'why another package?' The answer is that, although there are several other packages, none covers everything. For example, SIunits provides macros such as `\metre` to maintain consistency, but is poor on control of numbers, whereas SIstyle is stronger on numbers but does not provide unit macros. So a combined package, which can do everything, seemed desirable. Many of the ideas that I and others had were also well beyond anything available in the existing solutions. That said, a lot of the internals of siunitx is taken more-or-less verbatim from the existing packages. The numprint system has been extensively recycled here; most of the number-interpreting system is based on the numprint method. Indeed, while the initial aim of the new package was to deal primarily with *units*, much of the work I have undertaken has been concerned with *numbers*.

The design of siunitx is intended to be flexible. Almost everything is made available as an option, and given the number of options, key–value controls were the only way to do this. The package documentation covers (hopefully) all of them, but a few examples here will illustrate the ideas.

5.4.1 Some basics

The basic macro of the package is `\SI`. This takes a unit and a value, and typesets them together. The font used for this is controlled, and the space between them cannot be broken. Thus for example `\SI{10}{\metre}` gives '10 m'. The second argument is a unit, which can be given as a series of macros, or can be literal text: `\SI{2}{N.m^2}` gives '$2\,\mathrm{N\,m^2}$'. Here, there is no need to instruct the package to use maths mode for superscripts: this is handled automatically.

Both the number and unit parts of the input can be used independently, with macros `\num` and `\si`, respectively. The `\num` macro can interpret and format a wide range of numerical input. So `\num{1e2,3}` gives '$1 \times 10^{2.3}$' (when using U.K. settings, at least). As with the rest of the package, the way that numbers are interpreted can be adjusted by setting the appropriate package options. For angles, the `\ang` macro is available; it can work with angles as decimal and arcs: `\ang{1.23}` and `\ang{1;2;3}` give '1.23°' and '1°2′3″', respectively.

Package options can be set in three ways. As with most packages, load-time options are recognised. There is also a `\sisetup` macro, which adjusts settings for all subsequent input. Finally, all of the user macros accept an optional first argument; this allows changes for a specific item only.

5.4.2 The unit processor

One of the new ideas in siunitx is the unit processor. This takes macro-based units, and can reformat them depending on the desired output. 'Out of the box', `\SI{30}{\metre\per\second}` gives '$30\,\mathrm{m\,s^{-1}}$', whereas if we set the option `per=slash`, the same input gives '30 m/s'. This can be extended further, allowing the interpretation of powers, including reciprocals. Thus, given the input `\SI{20}{\per\Square\second}`, the result can be '$20\,\mathrm{s^{-2}}$', '$20/\mathrm{s^{2}}$', '$20\,\frac{1}{\mathrm{s^{2}}}$' or '$20\,{}^{1}\!/_{\mathrm{s^{2}}}$', depending on the options set.

5.4.3 Special effects

A lot of work has gone into making more complex ideas available for users of siunitx. Thus the idea of 'repeated' units is easily handled:

```
\SI{1 x 2 x 3}{\metre}
```

yields '1 m × 2 m × 3 m', for example. Setting option `repeatunits=false`, the alternative (and not entirely desirable) '1 × 2 × 3 m' results.

Similarly, the conversion of angles between decimal and degree–minute–second format is possible. So `\ang{1.23}` normally gives the expected '1.23°'; setting `angformat=arc` gives the alternative form '1°13′48.0″'.

In many parts of the natural sciences, errors in physical measurements are important. Two ways are common for showing these, as a separate error part, and as a bracketed error in the last digit: 1.23 ± 0.04 and 1.24(4), respectively. Using siunitx, both forms can result from the same input: `\num{1.24(4)}`. The separated form is obtained by setting the `seperr` flag. This works with exponents and units, for example:

```
\SI[seperr]{1.23(4)e5}{\candela}
```

gives '$(1.23 \pm 0.04) \times 10^{5}\,\mathrm{cd}$'. Notice the automatic addition of brackets to prevent ambiguity: this is an adjustable package option.

The final extra to highlight is the provision of a new column for tabular environments, the `S` column. This brings the ability to use the `\num` macro to tables, but also brings control of alignment (including exponents). A short example shows the effect:

```
\begin{table}
  \centering
  \begin{tabular}{cS[tabformat=3.1]}
    \toprule
    {Entry} & {Distance/\si{\metre}} \\
    \midrule
    1 & 102.3 \\
    2 & 123.2 \\
    3 &   2,3 \\
    4 & 145.  \\
    5 &   2.3 \\
    \bottomrule
  \end{tabular}
\end{table}
```

Entry	Distance/m
1	102.3
2	123.2
3	2.3
4	145
5	2.3

Here, the contents of the `S` column are centred under the column heading. Space is reserved for three digits before the decimal point, and one digit after. Thus, while the decimal points are aligned they are *not* at the centre of the column. A range of options are provided to allow control of positioning in `S` columns.

5.4.4 Emulation

Replacing the existing packages raises the issue of users moving existing manuscripts to siunitx. To aid this transition, emulation modes are available for all of the existing solutions. This means that users of the existing packages can experiment with siunitx without needing to learn new default settings or input conventions.

5.5 Conclusions

This short article can only provide a flavour of the abilities of siunitx. However, it hopefully highlights the possibilities made available by the new package. The package documentation includes a full range of examples, with almost all of the package options illustrated. I hope that I have covered a wide range of needs, and that siunitx will make it easier for LaTeX users to get units right without working too hard. Suggestions for new features are always welcome, particularly when they come with some interface ideas as well!

References

[1] Heiko Bauke. fancyunits, 2007. `http://www.mpi-hd.mpg.de/personalhomes/bauke/LaTeX/Tips_und_Tricks/fancyunits/index.php`.

[2] Conférence Générale des Poids et Mesures. The International System of Units (SI), 2008. `http://www.bipm.org/en/si/`.

[3] D. N. J. Els. The SIstyle package. Available from CTAN, `macros/latex/contrib/sistyle`, 2006.

[4] Patrick Happel. unitsdef—Typesetting units with LaTeX 2$_\varepsilon$. Available from CTAN, `macros/latex/contrib/unitsdef`, 2005.

[5] Harald Harders. The numprint package. Available from CTAN, `macros/latex/contrib/numprint`, 2008.

[6] Marcel Heldorn. The SIunits package. Available from CTAN, `macros/latex/contrib/siunits`, 2007.

[7] National Institute of Standards and Technology. International System of Units from NIST, 2008. `http://physics.nist.gov/cuu/Units/index.html`.

[8] Alex Reichert. units.sty—nicefrac.sty. Available from CTAN, `macros/latex/contrib/units`, 1998.

6 LaTeX for chemists: filling in the gaps

Joseph Wright

LaTeX is traditionally strongly favoured by mathematicians and physicists. Use by chemists has tended to remain on the 'physical' side of the subject. Support for the particular needs of chemists has therefore be somewhat variable. I have been involved with a selection of new or improved packages, which seek to address some of the gaps.

6.1 Introduction

LaTeX has a whole range of packages available, aimed at almost the entire range of (academic) pursuits. However, gaps still arise, and are filled by interested users. For chemists, despite the existence of some very useful tools, gaps have remained. As I have worked with LaTeX, I have worked to fill some of those gaps, as far as I have been able. This article gives an overview of the areas I have contributed to, as well as highlights from others, and showing some gaps that remain. All of my packages are available from CTAN in the usual manner: to keep the bibliography a little shorter, these are not formally cited although other people's packages are.

Some chemistry-focussed packages will be mentioned in the rest of this article. However, one which deserves particular mention here is mhchem [6]. This allows very simple input of chemical formulae (and simple in-line equations). Thus, it allows you to write `\ce{H2SO4}` to get H_2SO_4, `\ce{CH2=CH-C#CH}` to get $CH_2{=}CH{-}C{\equiv}CH$, or

```
\ce{2H2 + O2 -> 2H2O}
```

and get

$$2\,H_2 + O_2 \longrightarrow 2\,H_2O. \qquad (1)$$

It is a tool no chemist using LaTeX should be without.

6.2 Bibliography tools

6.2.1 BibTeX styles

One area where chemists seem to have unusual requirements is in creating bibliographies. To begin with, as a chemist the name is wrong: it is the *References* section. The most basic requirement for everyone using BibTeX is appropriate style files. When I started using LaTeX, I found the `pccp.bst` BibTeX style, based on the journal *Physical Chemistry Chemical Physics*. However, there were problems with the output. I therefore wrote my own style, `rsc.bst`, based on the general requirements of the Royal Society of Chemistry (RSC); as a U.K.-based worker, the RSC style is one I follow for my own documents. Over time, `rsc.bst` was joined by `angew.bst`, aiming at the requirements of *Angewandte Chemie* (arguably the 'top' general chemistry journal: I'm sure a lot of American chemists would disagree!). These files, plus some utility macros, ended up in a package called rsc. Later, the utility macros moved elsewhere, but the rsc survives in modified form.

The achemso package was originally written by Mats Dahlgren, and provided a BibTeX style following the requirements of the American Chemical Society (ACS). Once I started using the style, I spotted some issues. Contacting the author, I found he no longer had time for package maintenance. So, as much by accident as by design, I took over achemso. A complete re-write resulted, with improvement to the BibTeX style and the accompanying LaTeX package. More recently, I've rewritten the package again so that it fits in with the submission system at the ACS: this hopefully makes submitting articles a bit easier.

6.2.2 Bibliography packages

Beyond BibTeX styles, chemists have two important requirements for bibliographies. First, the idea of 'compound' references is common. Most chemistry journals use numerical citations, rather than the author–date system. It is very common to want a single reference number to refer to several related journal articles. The mcite package [9] can do this, but with very limited control of the results. In particular, it's common in chemistry to give each reference a letter inside a long list, such as

> [4] (*a*) G. Alberti, M. Casciola, U. Costantino, A. Peraio and E. Montoneri, *Solid State Ionics*, 1992, **50**, 315–322; (*b*) G. Alberti, M. Casciola, U. Costantino and R. Vivani, . . .

The mcite package cannot do this automatically. So I began to consider the issue, and to ask on `comp.text.tex` if anyone had contact details for the author of mcite. Luckily, Michael Shell was interested in the other extensions to mcite: the result was the mciteplus package [12], which *can* generate the desired output with control of formatting. Although I did not write any of mciteplus, the aim of making sub-lists inside each reference is included there specifically because I worked with him to get it working. Thus the example citation above can be given in the source simply as

```
\cite{Alberti1992,*Alberti1996}
```

which results in nicely-formatted output as above. This document uses my `rsc.bst` BibTeX style, which sets up mciteplus to follow the requirements of the RSC; for example, this instructs mciteplus to use a sub-list, and to make the sub-labels italic.

The second thing that chemists like to do is mix notes and references. This can be done by creating a BibTeX database of notes for every document you write to contain the information, but it is tedious. A much better idea would be to add the text directly into the body of the file, and have it move automatically to the bibliography section. To achieve this, I wrote the notes2bib package. The package works by creating a database for the notes during the LaTeX run, and then ensuring it is added to the list of files to be processed by BibTeX. This means it is 'neutral' with respect to sorting of bibliographies and packages used, such as cite [1], natbib [4], biblatex [7], etc. Using notes2bib implies using numerical citations: the citations do not make much sense with the author–year system! Using the package requires only inclusion of one or more `\bibnote`s in the source, although it is possible to convert `\footnote` or `\endnote` entries into `\bibnote` data automatically. For example, you could write

```
\bibnote{An example bibliographic note}
```

and this would give [2].

6.3 Utilities

The mhchem package is probably the most useful general utility package for chemists. However, this leaves a few gaps that can happily be filled. Initially, I provided a package with the rsc bundle to do this. However, later it became clear there was a better approach. The chemstyle package inherited the utilities from rsc, along with new functions. For example, this gives the macros `\tBu`, `\iPr`, *etc.* to give alkyl radicals: *t*-Bu, *i*-Pr, *etc.* It also provides items such as the 'standard state' symbol, to allow you to produce $\Delta H^{\ominus}$ easily.

The main aim of chemstyle is to help maintain consistency. By specifying the journal style to follow, the package allows float captions, cross-referencing and so on to follow the choices of the publication given. This is a lot easier than trying to remember the choices of every separate journal. It also loads a number of useful packages for the chemist, including my own chemscheme.

6.4 Graphics: the 'scheme'

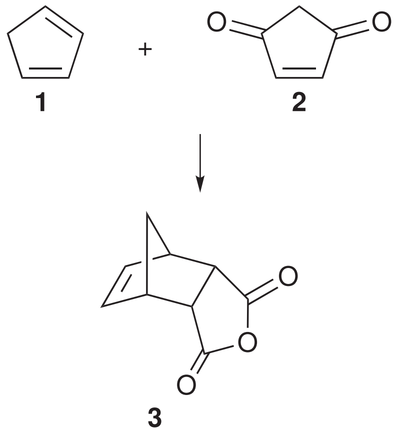
The concept of a 'scheme' is one that non-chemists find difficult to understand: they expect the name 'equation' to be used. An equation to most synthetic chemists is a simple, broadly non-graphical item, such as Equation 2.

$$2\,\mathrm{H_2} + \mathrm{O_2} \longrightarrow 2\,\mathrm{H_2O} \tag{2}$$

In contrast, a scheme is a more complex, graphically-rich item, such as Scheme 1. The example here is simple by the standards of many schemes in the research literature. Several packages are available for generating the graphics directly in LaTeX. However, producing anything beyond the most simple scheme becomes very difficult using text-based tools. I, like almost every synthetic chemist, use the commercial package ChemDraw[3] to produce my schemes.

Rather than trying to provide a package to tackle directly producing schemes in LaTeX, the chemscheme package aims to solve two lesser problems. The first aim of chemscheme is to provide an 'out of the box' float for schemes. Chemists expect the scheme to be near 'here' if possible, so this is the case with the `scheme` float. Thus the example scheme used here is produced using

```
\begin{scheme}
  \includegraphics{Scheme}
  \caption{A simple scheme}
  \label{sch}
\end{scheme}
```



Scheme 1: A simple scheme

The second aim is more complex. The use of reference numbers for chemicals in graphics is very common. Two packages exist to automate this in the text: bpchem [10] and chemcompounds [11]. However, neither can work with graphical content. Using PSfrag [5], the chemscheme package makes automatic substitution easy for `.eps` graphics. This is achieved by the `\schemeref` macro, which works with a temporary marker in the input. By using pst-pdf [8] this can also be used with PDFLaTeX.

6.5 biblatex bibliographies

The biblatex, currently available with beta status, is a completely new way to produce bibliographies from a database. The current version uses BibTeX, but does not need dedicated style files to do this. Instead, it requires LaTeX files containing formatting instructions. I've written some of these for chemists and other scientists: biblatex-chem to cover the same journals as rsc and achemso, biblatex-nature for *Nature*-like formatting and biblatex-science to emulate the journal *Science*.

6.6 Further afield

Beyond the focus on chemistry, one package in particular deserves mention here. Using units with numbers is common across the whole of science. The siunitx package provides a wide range of tools for typing units and values. For example, the input

```
$R = \SI[dp=3]{8.314472}
  {\joule\per\mole\per\kelvin}$
```

gives the typeset result $R = 8.314\,\mathrm{J\,mol^{-1}\,K^{-1}}$, while

```
$R = \SI[dp=5,per=slash]{8.314472}
  {\joule\per\mole\per\kelvin}$
```

gives $R = 8.314\,47\,\mathrm{J/(mol\,K)}$. This type of format control is available either on a per-macro basis or by setting package settings in the document. The aim of siunitx is therefore to allow units and values to have a single input syntax but give a range of output formats: this makes working with different publishing requirements much easier.

References

[1] Donald Arseneau. cite. Available from CTAN, macros/latex/contrib/cite, 2009.

[2] An example bibliographic note.

[3] See http://www.cambridgesoft.com.

[4] Patrick W. Daly. Natural Sciences Citations and References. Available from CTAN, macros/latex/contrib/natbib, 2009.

[5] Michael C. Grant and David Carlisle. The PSfrag system, version 3. Available from CTAN, macros/latex/contrib/psfrag, 1998.

[6] Martin Hensel. The mhchem Bundle. Available from CTAN, macros/latex/contrib/mhchem, 2007.

[7] Philipp Lehman. The biblatex package. Available from CTAN, macros/latex/exptl/biblatex, 2009.

[8] Rolf Niepraschk and Hubert Gäßlein. The pst-pdf package. Available from CTAN, macros/latex/contrib/pst-pdf, 2008.

[9] Thorsten Ohl. mcite: Multiple citations on one key. Available from CTAN, macros/latex/contrib/mcite, 2005.

[10] Bjørn Pedersen. The bpchem package. Available from CTAN, macros/latex/contrib/bpchem, 2004.

[11] Stephan Schenk. The chemcopounds package. Available from CTAN, macros/latex/contrib/chemcompounds, 2006.

[12] Michael Shell. Mciteplus: Enhanced Multicitations. Available from CTAN, macros/latex/contrib/mciteplus, 2008.

7 An Introduction to the Greek Alphabet

Jonathan Webley

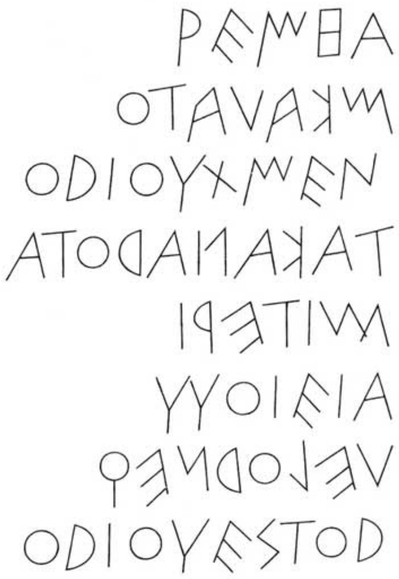
The first pure alphabet emerged around 2000 BCE in Egypt, based on alphabetic principles of the Egyptian hieroglyphs and is called the Proto-Sinaitic alphabet. Surprisingly, nearly every alphabet in the world today either descends directly from it or was inspired by it.

Essentially, writing was independently invented only a handful of times. Virtually every symbol we see today in the West had as its ultimate origins the work of, probably, a single person, living in, probably, ancient Egypt. That idea was then copied and evolved and developed to become the tens of thousands of symbols, for example, found in Unicode.

An important descendant of the Proto-Sinaitic alphabet was the Phoenician alphabet, which in turn evolved into the Arabic, Hebrew and Greek alphabets. Indian scripts such as Devanagari are also descendants of the Phoenician alphabet. All the modern scripts of Europe – such as Latin, Gothic and Cyrillic – are descended from the Greek alphabet. Our word *alphabet* derives from the Greek letters *alpha* and *beta.*

7.1 Math mode

Because of its widespread use in mathematics, science and engineering the Greek alphabet is a standard feature of TeX when in math mode. There are codes for every lowercase letter with the exception of omicron, because it has the same shape as the Latin "o". The math mode versions of the lowercase Greek letters are in italics whereas the uppercase letters are upright.



Greek Inscription, 5th century BCE

7.2 Babel

Using the `greek` option of the `babel` package is the normal way to write text in Greek. If the whole document is in Greek then the preamble would include:

`\usepackage[greek]{babel}`

Each Latin letter in the `tex` file is then automatically transliterated to its equivalent Greek letter. These equivalences are given in the tables below.

A document such as this one, containing mainly English with a smattering of Greek, would have:

`\usepackage[greek,english]{babel}`

and in the body when Greek is required:

`\foreignlanguage{greek}{Alpha}`

which renders as Αλπηα. LaTeX also has other ways of achieving this result which are not detailed here.

The `greek` option of this package also includes three obsolete letters as illustrated below. The LaTeX codes for these letters are: `\qoppa`, `\sampi` and `\stigma` and these are shown later.

7.3 The Letters

There are 24 letters in the modern Greek alphabet, with uppercase and lowercase variants. These letters are given in the following tables, and the columns in it are:

1. Greek letter produced in math mode by standard TeX.
2. The TeX code for this letter (the letter name).
3. Greek letter produced by the `babel` package.
4. The Latin letter used to produce this letter.

Lowercase letters

LaTeX letter	LaTeX code	Latin letter	Latin
α	`\alpha`	α	a
β	`\beta`	β	b
γ	`\gamma`	γ	g
δ	`\delta`	δ	d
ϵ, ε	`\epsilon`, `\varepsilon`	ε	e
ζ	`\zeta`	ζ	z
η	`\eta`	η	h
θ, ϑ	`\theta`, `\vartheta`	ϑ	j
ι	`\iota`	ι	i
κ	`\kappa`	ϰ	k
λ	`\lambda`	λ	l
μ	`\mu`	μ	m
ν	`\nu`	ν	n
ξ	`\xi`	ξ	x
o	o (omicron)	ο	o
π, ϖ	`\pi`, `\varpi`	π	p
ρ, ϱ	`\rho`, `\varrho`	ρ	r
σ, ς	`\sigma`, `\varsigma`	ς	s
τ	`\tau`	τ	t
υ	`\upsilon`	υ	u
ϕ, φ	`\phi`, `\varphi`	φ	f
χ	`\chi`	χ	q
ψ	`\psi`	ψ	y
ω	`\omega`	ω	w

Many of the uppercase letters are simply those used in the Latin alphabet, and for these there is no TeX code.

Uppercase letters

LaTeX letter	LaTeX code	Latin letter	Latin
A	A (alpha)	A	A
B	B (beta)	B	B

Uppercase letters (cont.)

LaTeX letter	LaTeX code	Latin letter	Latin
Γ	`\Gamma`	Γ	G
Δ	`\Delta`	Δ	D
E	E (epsilon)	E	E
Z	Z (zeta)	Z	Z
H	H (eta)	H	H
Θ	`\Theta`	Θ	J
I	I (iota)	I	I
K	K (kappa)	K	K
Λ	`\Lambda`	Λ	L
M	M (mu)	M	M
N	N (nu)	N	N
Ξ	`\Xi`	Ξ	X
O	O (omicron)	O	O
Π	`\Pi`	Π	P
P	P (rho)	P	R
Σ	`\Sigma`	Σ	S
T	T (tau)	T	T
Υ	`\Upsilon`	Υ	U
Φ	`\Phi`	Φ	F
X	X (chi)	X	Q
Ψ	`\Psi`	Ψ	Y
Ω	`\Omega`	Ω	W

Upright versions of the Greek letters can be found in `pxfonts`, `txfonts` and `upgreek`. There is also a variant of kappa found in the `amssymb` package: $\varkappa$ (`$\varkappa$`). The International Phonetic Alphabet uses various symbols, many of which are Greek letters or are derived from Greek letters. These can be found in various packages, including `phonetic`, `wsuipa` and `t4phonet`.

7.4 Bold Greek

The simplest way to produce bold Greek text is with the `babel` package:

`\textbf{\foreignlanguage{greek}{Lambda}}`
which produces:

Λαμβδα

In math mode things are more complicated: `textbf` is not valid; `mathbf` only makes bold the uppercase letters. In package `amsbsy` the command `boldsdymbol` will make all symbols bold. Command `pmb` produces heavy symbols which resembles bold. The `bm` package has a command `bm` which makes all maths bold. And finally, package `fixmath` has a command `mathbold` which can also be used.

7.5 Obsolete letters

There are a number of obsolete letters. The samples in the following table derive either from `amssymb` or from the `greek` option of the `babel` package.

The `arevmath` package is intended for presentations and posters and its use is not illustrated here. There are large and small forms of each letter, which otherwise seem identical.

Name	Letter	Packages
digamma	Ϝ	`amssymb`, `arevmath`
het		none known
koppa		`arevmath`
qoppa	ϟ	`arevmath`, `babel`
sampi	ϡ	`arevmath`, `babel`
san		none known
sho		none known
stigma	ϛ	`arevmath`, `babel`

Qoppa and koppa refer to the same letter: its shape evolved from that referred to as qoppa to that referred to as koppa.

All of these letters are available in Unicode.

7.6 Greek Diacritics

Modern Greek uses only two diacritics, however, before 1982 Greek orthography was more complex. Modern Greek is called monotonic (meaning “single accented”), while the older orthography is called polytonic (meaning “many accented”). The `babel` package by default uses the modern orthography, but the following preamble allows the older orthography to be used:

```
\usepackage[greek]{babel}
\languageattribute{greek}{polutoniko}
```

The latter option, “polutoniko”, is the Greek for “polytonic”.

Use of this option is not illustrated here.

7.7 Some uses

A couple of letters can be found in the `textcomp` and `gensymb` packages:

Symbol	`textcomp`	`gensymb`
Ω	`\textohm`	`\ohm`
µ	`\textmu`	`\micro`

This version of *mu* is used for the SI prefix micro- for 10^{-6}. For example, the symbol for a micron is µm.

Mathematics uses variants of *sigma* and *pi* as operators, such that in display mode subscripts and superscripts are placed beneath or above the letters respectively. These symbols are found in the `amssymb` package:

$$\sum_{i=0}^n \quad \texttt{\$\textbackslash sum\_\{i=0\}\^{}n\$}$$

$$\prod_{i=0}^n \quad \texttt{\$\textbackslash prod\_\{i=0\}\^{}n\$}$$

A couple of symbols are inverted, and are found in the `amssymb` package:

℧	`\mho`	This is ohm spelt backwards, and is an unofficial symbol for the siemens.
∇	`\nabla`	This symbol is the *del* operator in mathematics.

7.8 Greek Arithmetic

In Greek, there are no separate symbols, glyphs, for numbers. Modern Greek uses the same arabic numerals that we use. So, like the Romans, the ancient Greeks used letters to stand for numbers (unlike the arabic system where the symbols stand for digits).

The earliest system used was the Attic system, which is similar to the system used by the Romans:

Arabic	Greek	Roman
1	I	I
5	Π	V
10	Δ	X
100	H	C
1,000	X	M
10,000	M	

The letters used by the Greeks are the initial letters of the words for the numbers. Each letter represents a number, and the values are added (though the Romans also subtracted):

$$4 = \text{IIII (Greek)} = \text{IV (Roman)}$$

A later system was the Ionian or Athenian system, which uses 27 letters (the modern 24 letters plus 3 obsolete letters):

α	1	ι	10	ρ	100
β	2	κ	20	σ	200
γ	3	λ	30	τ	300
δ	4	μ	40	υ	400
ϵ	5	ν	50	ϕ	500
$\digamma$	6	ξ	60	χ	600
ζ	7	o	70	ψ	700
η	8	π	80	ω	800
θ	9	ϟ	90	ϡ	900

This system is also additive. So, $123 = \rho\kappa\alpha$. Extensions to this system allow for larger numbers.

Package `\athnum` can be used to convert numbers in arabic format to numbers in Ionian format.

8 The Hound Answers

Across

1. cerise, 3. asks, 6. sac, 8. assuage, 11. piece,
10. shako, 12. pie, 13. stele, 14. rainy,
16. animate, 17. sou, 18. smew, 19. potash

Down

1. casks, 2. soap opera, 4. snake pits,
5. suede, 7. ceasefire, 9. superhero, 13. shags,
15. youth

9 Contributions

All contributions to *Baskerville* should be sent to the editor at:

baskerville@uk.tug.org

Articles on any area of TEX or its friends, UK-TUG or related topics are very welcome. The Committee is particularly keen to publish articles with a UK *flavour*. Send in your comments on this issue; your suggestions, letters, thoughts, tips and hints, articles, jokes, questions, requests for help, jobs, cartoons or puzzles – anything relevant will be considered for publication.