

# Baskerville

The Annals of the UK  $\TeX$  Users' Group  
ISSN 1354-5930

Editor: Editor: Sebastian Rahtz

Vol. 5 No. 3  
June 1995

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   | 3  |
| 1    | Calling on you  | 3  |
| 2    | Looking different                                     | 3  |
| 3    | Sad resignation of UKTUG committee member             | 3  |
| 4    | Whither $\LaTeX$ ?                                    | 3  |
| II   | Cathy Booth Memorial                                  | 5  |
| III  | Cathy Booth 1952-1991                                 | 6  |
| IV   | Honorary Member                                       | 7  |
| V    | Letter to the editor                                  | 7  |
| VI   | Using Adobe Type 1 Multiple Master fonts with $\TeX$  | 8  |
| 1    | Introduction  | 8  |
| 2    | Multiple Master overview                              | 8  |
| 2.1  | Multiple Master Design Space                          | 8  |
| 3    | Multiple Master Font Programs                         | 9  |
| 3.1  | Multiple Master Keywords                              | 9  |
| 3.2  | The <code>makeblendedfont</code> Operator             | 10 |
| 3.3  | Multiple Master <code>findfont</code> Procedure       | 10 |
| 4    | Using Multiple master fonts with $\TeX$               | 10 |
| 5    | In practice   | 12 |
| VII  | Maths in $\LaTeX$ : Part 5, Getting started on arrays | 14 |
| 8    | Mathematical arrays                                   | 14 |
| 8.1  | Basics  | 14 |
| 8.2  | Matrices and vectors                                  | 15 |
| 8.3  | Arrays of arrays                                      | 15 |
| 8.4  | Changing the row spacing                              | 15 |
| 8.5  | Changing the column spacing                           | 16 |
| 8.6  | Words in arrays                                       | 16 |
| 8.7  | Inter-column constants                                | 17 |
| 8.8  | Positioning   | 17 |
| 8.9  | Cases   | 17 |
| 8.10 | Binomial coefficients                                 | 17 |
| 9    | Tables of data  | 19 |
| 9.1  | General   | 19 |

|      |   |    |
|------|---|----|
| 9.2  | Horizontal rules . . . . .  | 19 |
| 9.3  | Vertical rules . . . . .  | 20 |
| 9.4  | Spanning several columns . . . . .  | 20 |
| 9.5  | Changing the type of an entry . . . . .                                   | 21 |
| 9.6  | Decimal tabulation . . . . .  | 22 |
| 9.7  | Things that stick out . . . . .   | 22 |
| 9.8  | Columns specifiers . . . . .  | 22 |
| VIII | Report of visit to CERN to attend presentation of $\Omega$ . . . . .      | 24 |
| 1    | Introduction . . . . .  | 24 |
| 2    | What <i>is</i> $\Omega$ ? . . . . .                                       | 24 |
| 3    | Why Unicode? . . . . .  | 24 |
| 4    | Virtual Metafont and Fonts to Support Unicode . . . . .                   | 26 |
| IX   | A.N.Other's Gleanings . . . . .   | 28 |
| 1    | Framed, or a boojum? . . . . .  | 28 |
| 2    | Lettres . . . . .   | 28 |
| 3    | To be taken daily . . . . .   | 28 |
| 4    | Web . . . . .   | 28 |
| 5    | Deathwish 1996 . . . . .  | 29 |
| 6    | Another famous Belgian . . . . .  | 29 |
| 7    | Thank you . . . . .   | 29 |
| X    | New PostScript font metrics for L <sup>A</sup> T <sub>E</sub> X . . . . . | 30 |
| XI   | Standard for T <sub>E</sub> X Directory Structures . . . . .              | 31 |
| XII  | The 1995 Annual General Meeting . . . . .                                 | 32 |
| XIII | The UK T <sub>E</sub> X Users Group . . . . .                             | 34 |
| 1    | Group aims and activities . . . . .                                       | 34 |
| 2    | Programme of meetings for 1995 . . . . .                                  | 34 |
| XIV  | Obtaining T <sub>E</sub> X . . . . .                                      | 35 |
| XV   | Membership Data . . . . .   | 37 |
| 1    | Membership Services . . . . .   | 37 |

---

## I Editorial

---

### 1 Calling on you

It seems to have been a while since we had a ‘normal’ issue of *Baskerville*; after the effort last autumn to get the FAQ issue to our readers, events keep overtaking us, and the last issue was so full it omitted almost all our regular information. Even now, with summer upon us, and T<sub>E</sub>X conferences coming up thick and fast, *Baskerville* still needs quality articles. Please send in material to benefit fellow T<sub>E</sub>Xxies.

The deadline for *Baskerville* is always the same — the beginning of the last week of odd months.

### 2 Looking different

Have you noticed anything strange about this *Baskerville* yet? Yes, it is not typeset in the eponymous Baskerville font. Since the article later in this issue by Michel Goossens and Sebastian Rahtz discusses how to create instances of Adobe’s Multiple Master fonts, and T<sub>E</sub>X metrics for those instances, it seemed a useful idea to set a whole issue in Minion, as a practical demonstration. We do not claim that the variables chosen to make the instances are especially good (our bold is rather weedy, for instance), but we hope it will interest some readers. Make of it what you will — Baskerville itself will be back next time.

### 3 Sad resignation of UKTUG committee member

At the committee meeting of UKTUG in May, Jonathan Fine announced his resignation because of ongoing disagreement with other committee members, and that he would not be standing for re-election. Admirers of *Baskerville*’s regular appearance over the last 18 months have reason to regret his departure, as he played a great part in the *Baskerville* distribution and (like all the committee) worked hard at proof-reading. We hope his regular supply of articles (interrupted for this issue) will not cease.

### 4 Whither L<sup>A</sup>T<sub>E</sub>X?

Earlier this year, UKTUG wrote to all 1994 members who had not renewed to ask if they were sure; Richard Preston of Oxford University Press took the trouble to write back and made some interesting points which are worth answering publicly. He said: “. . . While I have the opportunity, I might as well air one or two frustrations:

1. I don’t know what form L<sup>A</sup>T<sub>E</sub>X3 is going to take, but it seems to me that if T<sub>E</sub>X is not going to become the Betamax of DTP systems then it has to have a WYSIWYG interface. The current palaver that one has to go through even to change fonts really isn’t acceptable to 99% of the people who use these systems. Ideally it should be possible to view a WYSIWYG window and a window containing the source code, with changes made in either window being immediately reflected in the other (I believe Borland’s Delphi works on this principle for Pascal programming).

2. I now work in OUP’s electronic publishing department, and many of our products are simultaneous book and CD-ROM publications, using essentially the same text data. There is an obvious need for a typesetting system that can produce typeset pages based on tagged ASCII files, which can then be used in the CD-ROM version. We currently have to translate typesetters coded files into SGML tagged text, and while this is not difficult, a system that could translate L<sup>A</sup>T<sub>E</sub>X’s generic codes into SGML would be a godsend.

3. T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X appear to have become the DTP equivalent of UNIX — an extremely powerful whole built up from many powerful sub-units, but usable only by those with enough time and enthusiasm to really learn it. As an example, we bought Y&Y T<sub>E</sub>X’s setup for use here; it is extremely good and has served us very well, but the initial installation and messing around with font encoding nearly drove me insane. I know this is a commercial product and is nothing to do with TUG, but it does reflect the willingness of the TeX community to accept that using TeX should be a struggle. Compare this with the ‘install and away you go’ use of Quark or Pagemaker.

This letter seem to have turned into something of a rant, and I’m sorry for that, but it is born out of a genuine concern that a work of genius like T<sub>E</sub>X is being denied the usage it deserves because of a lack of awareness within TUG. I’ve reprinted from *Baskerville*

sat through too many lectures on how to use Cyrillic or Hebrew, or the merits or de-merits of `\frenchspacing` to believe that this letter will alter T<sub>E</sub>X's course one iota, but at least it's made me feel better."

in reply to which your *Baskerville* editor and the UKTUG chairman made the following points:

1. Indeed, you can have this effect of immediate feedback if you use Blue Sky Research's excellent Lightning Textures product. I think the interface you get should be separated from the underlying structure of T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X<sub>3</sub>, and that style design is not a matter of daily usage. You don't change fonts in a document, you change it in a style file
2. There are various projects to translate L<sup>A</sup>T<sub>E</sub>X to SGML reliably; interested parties may like to contact Seamus McCague of ICPC, Dublin ([seamus@icpc.ie](mailto:seamus@icpc.ie)) who have developed a sophisticated solution.
3. It is true that some publications need, and can afford, the detailed level of hand-crafted design that an interactive system can provide when used by a skilled operator; however I imagine that, for example, you do not typeset your dictionaries on such a system—there is an important part of publishing where highly automated and high quality typesetting is needed and this includes a lot of technical publishing where accuracy of layout is too vital to be left to individual operators.

We appreciate Richard's points, but what exactly do you (our members) and those of TUG, want us to do? The various T<sub>E</sub>X user groups are aware of these issues, but they do not control software development, and largely act as communication channels and meeting points. We (the UKTUG committee) very much welcome suggestions and guidance about actions you think we can take.

---

## II Cathy Booth Memorial

---

At the 1992 AGM it was agreed that the UKTUG would establish the Cathy Booth Memorial Fund, to be used “to support education and research in electronic publishing in general and in the use and development of  $\text{\TeX}$  and its relatives in particular, and for other charitable purposes connected with education”.

The committee intended to set up a charitable trust to administer the fund, and then to collect money for it (cheques cannot be written to a trust which does not yet exist). However, both trusts and charities are covered by various laws, and the legal hurdles proved too much for the committee. Accordingly, at its meeting in February 1995, the committee decided to adopt a simpler way of honouring Cathy Booth’s memory, but with the same aim. Specifically, it was decided to do something in Cathy Booth’s name, at least once in each calendar year, supported by money if appropriate, for the purpose “of furthering research and education in relation to electronic publishing in general and the use and development of the public domain typesetting program  $\text{\TeX}$  and related programs in particular”. (This wording is taken from the draft trust document.)

As it’s first act in Cathy Booth’s memory, the committee gave support, in two ways, to the April 1995 GUTenberg meeting on the  $\Omega$  project. It donated £200 to the meeting. It also sent Robin Fairbairns to the meeting as its delegate. A report of that meeting appears elsewhere in this issue.

R. A. Bailey  
Honorary Secretary 1991–1995

---

### III Cathy Booth 1952–1991

Chris Rowley and Malcolm Clark

---

Cathy Booth<sup>1</sup> died on June 27th, 1991. She had known that she had cancer of the kidneys for just about a year. She had undergone a major operation the previous summer, but sadly the extensive surgery performed at that time was too late to control the cancer. During that last year she fought the disease consistently, cheerfully and resolutely; she was determined that she should not be treated any differently, despite the gradual decline in mobility and energy which ensued. Ultimately she was confined to a wheel chair and she moved into a hospice at Tiverton, near her Exeter home, in the hope that her house would be modified to allow her to return to it and continue her life as normal. However, this was not to be: she died, peacefully and gracefully, at the hospice with most of her close family at her side.

Cathy was born on June 19th, 1952. She attended Exeter University, where she completed a BSc in Geography. She fell in love with Exeter and spent the rest of her life that area. She spent some time after University working on Exmoor, and later had her own business as a furniture restorer. But in 1984, after a period in hospital where she began to do some computing, she took a post as Computing Development Officer back at Exeter University. One of her first responsibilities was support of T<sub>E</sub>X. As a result, she attended the IUSC conference on Typography and Design Methods for Computer Typesetting held at Oxford University at Easter 1985. Following that meeting, she attended a T<sub>E</sub>X course held at Imperial College. Her involvement and concern with T<sub>E</sub>X, and more generally with Electronic Publishing, grew consistently from that time.

She attended many of the T<sub>E</sub>X conferences in Europe (Strasbourg 1986, Karlsruhe 1989 and Cork 1990), as well as TUG meetings in North America (Montreal 1988 and Stanford 1989), and some of the GUTenberg meetings in Paris. But it will be the T<sub>E</sub>X88 conference held at Exeter University for which she will be best remembered by many in the T<sub>E</sub>X world. Cathy was the local organiser for this meeting and its huge success was due in no small way to, on the one hand, her vivacity and personal magnetism and, equally important, a lot of detailed planning and hard work that she undertook during the run up to the conference. She thus ensured the smooth running of the conference and various associated events: in particular, the memorable excursion to Totnes which included travel by both steam train and river cruiser.

Cathy was an excellent teacher of T<sub>E</sub>X. She taught a number of TUG courses on Beginning and Intermediate T<sub>E</sub>X, and other T<sub>E</sub>X courses at Exeter University and she was very energetic in organising other TUG courses in Europe. At the University she also developed courses which united her skill with T<sub>E</sub>X and her design abilities. This facility with design was just one side of more general artistic skills; these later led to her completion of an Art History course. The course results were published only a few days before she died and, characteristically, hers were excellent.

She was a founding member of the UK T<sub>E</sub>X Users Group and of its committee; she was also a member of the committee of the BCS Electronic Publishing Specialist Group. Those of us lucky enough to be on either of those committees at that time will recall her very real contribution to the work of these groups: her range of innovative ideas, coupled with her down to earth attention to detail, resulted in significant improvements to the activities of both these organisations.

---

<sup>1</sup>This article is based on one that appeared in T<sub>E</sub>Xline in 1991

---

## IV Honorary Member

---

In November 1991 the Committee of the UKTUG decided to award Honorary Life Membership of the Group to Malcolm Clark in recognition of his efforts and achievements in the creation and establishment of the UKTUG.

It had been planned to announce this at the 1992 AGM. Unfortunately, Malcolm, who knew nothing about this, did not attend that AGM, so the announcement was postponed. For various reasons, including the chair's continuing illness, the secretary's incompetence and the arrangements for TUG '93 in Aston, the issue was shelved. However, we are now delighted to announce the award, however belatedly, and we apologize to Malcolm for the delay.

Members may note that Clause 15 (2) of the constitution empowers the Committee to award Honorary Membership (so long as the total number of such members does not exceed 25); and also that the Committee at the time consisted of the following persons: Peter Abbott (chair), Ian Hall (treasurer), Rosemary Bailey (secretary), David Penfold (membership secretary), Sue Brooks, Geeti Granger, David Eckersley, Carol Hewlett, James Petts, Philip Taylor.

R. A. Bailey  
Honorary Secretary 1991–1995

## V Letter to the editor

Sir – many congratulations on an excellent and informative issue of Baskerville (Vol. 5, No. 2); my only queries concern the article on “Portable documents: Why Use SGML” by Professor David Barron, in which he writes: “Traditionally, a document was a file (or a deck of cards), and consisted solely of text”. I am surprised that Professor Barron is apparently unaware of a far older tradition, in which a document consisted of a series of sheets of paper (earlier, vellum or papyrus) of uniform size, bound between two slightly larger and more rigid sheets of board, and containing not only text but also illustrations and other entities; this style of document, apparently common until at least the end of the twentieth century, was called a “book”, and was highly portable, requiring very little in the way of support hardware apart from some equally portable optical enhancement devices (“glasses”, or “spectacles”) for the visually challenged. Professor Barron then goes on to write: “[. . .]: such a document is a *multimedia* document”: surely such a document is better termed a *multimedium* document, by analogy with existing *multi*. . . words; I am sure Professor Barron has never used a *multi*passes compiler, plugged in his computer to a *multi*ways socket outlet, or augmented his computer system with a *multi*purposes adapter.

Yours very sincerely,

Philip Taylor  
RHBNC

*Professor Barron replies:* Sir – Philip Taylor is to be complimented on a fine display of pedantry in the best academic tradition, the kind of tradition that gives academics a bad name amongst normal folk. In computing we use lots of everyday words with specialised meanings, and most of us find no difficulty in using the context of an utterance to achieve any necessary disambiguation. With regard to his criticism of my use of the term “multimedia document”, I agree that I don't plug my computer into a multiways socket. But then, I don't attend a performance of an operum at Covent Garden, either.

---

## VI Using Adobe Type 1 Multiple Master fonts with T<sub>E</sub>X

Michel Goossens  
CERN, Geneva  
m.goossens@cern.ch  
and  
Sebastian Rahtz  
Elsevier Science Ltd  
s.rahtz@elsevier.co.uk

---

### 1 Introduction

The multiple master font format is an extension of the Type 1 font format, which allows the generation of a wide variety of typeface styles from a single font program. This capability allows users and applications control over the typographic parameters of fonts used in their documents, in a manner reminiscent of Knuth's ground-breaking METAFONT. This article describes the multiple master system in some detail, and describes the procedures needed to make instances, and create the appropriate font metrics for use with T<sub>E</sub>X.

### 2 Multiple Master overview

A multiple master font program contains two or more outline typefaces called *master designs*, which describe one or more *design axes*. The master designs that constitute a design axis represent a dynamic range of one typographic parameter, such as the weight or width. This range of styles is defined in a multiple master font program by specifying one master design to represent each end of an axis, such as a *light* and *extra-bold* weight, as well as any *intermediate master designs* that are required. The maximum number of master designs allowed is sixteen.

A *font instance* consists of a font dictionary derived from the multiple master font program (or from another font instance). It contains a `WeightVector` array with  $k$  values that sum to 1.0 and which determine the relative contributions of each master design to the resulting interpolated design.

All derived font instances share the `CharStrings` dictionary and `Subrs` array of the main multiple master font program, making it relatively economical to generate a variety of font instances. Multiple master fonts can be made compatible with the installed base of PostScript language interpreters by including several PostScript language procedures and a new set of `OtherSubrs` routines in the font program. The procedures include the new `make-blendedfont` operator, the interpolation procedure `$Blend` and a new definition of the `findfont` operator.

#### 2.1 Multiple Master Design Space

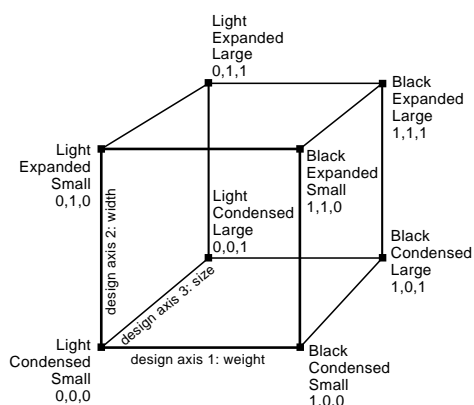
It is possible to think of the master designs as being arranged in a 1, 2, 3, or 4 dimensional space with various font instances corresponding to different locations in that space. The entries in the `FontInfo` dictionary specify what this space is and where the master designs are located in it. This information is necessary for interactive programs that allow users to create new font instances, and should be included in the font's Adobe Multiple Font Metrics (AMFM) file.

Fig. 1 illustrates an example of the design space of a three axis multiple master font. In this example, the axes are *weight*, *width*, and *optical size*. It is recommended that a font program be organized to have the lightest weight, narrowest width, and smallest design size mapped to the origin of the blend space.

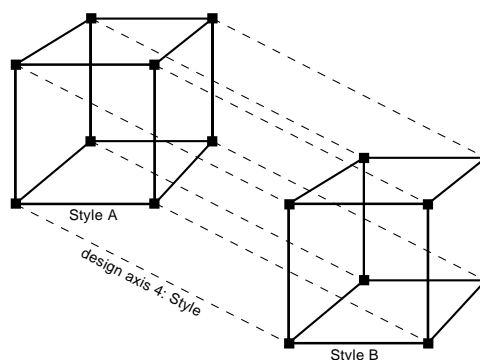
Multiple master coordinates are of two types: those which represent the design space and those which represent the blend space. Design coordinates are integers whose range for a particular typeface is chosen by the designer. They are used in font names and in the user interface for software which creates and manipulates multiple master font programs. The theoretical range for a weight or width axis is from 1 to 999 design units; however a typical typeface, with styles ranging from light to black, might have a dynamic range of from 200 (for light) to 800 units (for black).

Another type of optional axis would be for optical size, in which the character design changes with the point size. The design coordinates for the optical size axis might have a dynamic range of from 6- to 72-point, which represents the practical extremes of sizes for typefaces designed for publishing purposes.





**Figure 1.** Multiple master typeface design space arrangement



**Figure 2.** Arrangement of the multiple master design space for a four axis font

*Blend coordinates* are normalized values, in the range of 0 to 1, which correspond to the minimum and maximum design space coordinates. They are used by the Type 1 rasterizer because they are more convenient for mathematical manipulations. The linear space of blend coordinates is related to the (potentially) non-linear space of the design coordinates by the `BlendDesignMap` entry in the font dictionary.

Fig. 2 illustrates how a four axis design might be represented. An example of a fourth axis would be having an axis for a typographic style (serif/sans serif) or contrast (high/low: the ratio of thick to thin stem widths). This diagram illustrates that if four axes are defined, sixteen master designs are required. Also, since sixteen is the maximum number of designs allowed, there can be no intermediate designs with four axes.

### 3 Multiple Master Font Programs

Multiple master typefaces may contain from two to sixteen master designs, organized as having from one to four design axes. The maximum number of master designs allowed in a multiple master font is expressed by the equation  $2^n + x = 16$ , where  $n$  is the number of design axes,  $x$  is the number of intermediate designs, and 16 is the maximum allowed number of master designs.

The values used for calculating the weighted average are stored in the font dictionary in an array named `WeightVector`. The multiple master font program, as shipped by the font vendor, can have a default setting for the `WeightVector`; it is recommended that it is set so the default font instance will be the normal roman design for that typeface.

#### 3.1 Multiple Master Keywords

`BlendAxisTypes` is a (required) array of  $n$  PostScript language strings where  $n$  is the dimensionality of the design space and hence the number of axes. Each string specifies the corresponding axis type. In the case of 3-axis example, this value would be:

```
/BlendAxisTypes [/Weight /Width /OpticalSize]
```

`BlendDesignPositions` is a (required) array of  $k$  arrays giving the locations of the  $k$  master designs in the design space. Each location subarray has  $n$  numbers giving the location of the design in the  $n$  dimensions of the design space, with a minimum value of zero and a maximum value of one. Table 1 with eight master designs is based on the example shown in Fig. 1. It corresponds to the design space for the Minion 3-axis multiple master font.

The `BlendDesignPositions` array corresponding to this case is:

```
/BlendDesignPositions
[[0 0 0][1 0 0][0 1 0][1 1 0]
[0 0 1][1 0 1][0 1 1][1 1 1]] def
```

`BlendDesignMap` is a required entry consisting of an array of  $n$  arrays where  $n$  is the dimensionality of the design space. Each array contains  $m$  subarrays that describe the mapping of design coordinates into normalized coordinates for that design axis. The minimum value allowed for  $m$  is two, and the maximum is twelve. The order of the subarrays corresponds to the order of design axes in `BlendAxisTypes`. In the case of the Minion font this array is three dimensional ( $n = 3$ ) and has the following form:

| <i>Design label</i>             | <i>Blend space coordinates</i> |
|---------------------------------|--------------------------------|
| design 1: light condensed small | 0 0 0                          |
| design 2: black condensed small | 1 0 0                          |
| design 3: light expanded small  | 0 1 0                          |
| design 4: black expanded small  | 1 1 0                          |
| design 5: light condensed large | 0 0 1                          |
| design 6: black condensed large | 1 0 1                          |
| design 7: light expanded large  | 0 1 1                          |
| design 8: black expanded large  | 1 1 1                          |

**Table 1.** Design labels and blend space values for the Minion 3-axis multiple master font

```
/BlendDesignMap [
  [[345 0] [620 1]] [[450 0] [600 1]]
  [[6 0] [8 0.35] [11 0.5] [18 0.75] [72 1]] ]
```

The first number in an individual subarray is in design coordinates with a minimum value of 1 and a maximum value of 999. The second number in the subarray is in normalized coordinates, that is, in the range of 0 to 1. In the above example, the weight ranges from 345 to 620, while the width ranges from 450 to 600 in design space. The third axis, optical size, ranges from 6 to 72 (corresponding to the point sizes for which the typeface can be adjusted for optimal legibility).

### 3.2 The `makeblendedfont` Operator

```
blendedfontdict weightvector makeblendedfont blendedfontdict
```

This operator creates a font dictionary with blended entries. The `blendedfontdict` argument is a font dictionary of an existing multiple master font; it can be from either the original multiple master font itself, or from an interpolated font instance since any Blend dictionary contains all elements needed to derive additional font instances.

The `weightvector` argument is an array of numbers summing to 1.0 to be used as the weights for creating the new font instance. The value of `WeightVector` in `blendedfontdict` is set to the values in the array `weightvector`. Blended values are calculated for entries in the `Private` and `FontInfo` dictionaries. The result is a font dictionary that can be used as an argument to `definefont`. The resulting dictionary and its contents are still read-write, so the caller of `makeblendedfont` can make further modifications if necessary.

The Blend dictionary data structures provide the information needed by the `makeblendedfont` operator, without needing to have the specific list of entries to be blended built into the procedure. This allows a single copy of the procedure to be used even if the set of entries to be blended varies in future fonts.

### 3.3 Multiple Master `findfont` Procedure

Multiple master font programs from Adobe include a procedure which redefines the `findfont` operator in `sys-temdict`. This is necessary because of the need to generate font instances on-the-fly to satisfy multiple master font references in a PostScript language document. The procedure creates all necessary font instances before calling the standard `findfont` procedure.

Two procedures, `NormalizeDesignVector` and `ConvertDesignVector`, which are referenced in `findfont`, must be configured for the number of axes and master designs in the font program in which they are used. The `NormalizeDesignVector` procedure must calculate the normalized equivalent of the design coordinates in the `FontName`, using the values in the `BlendDesignMap` array. These normalized coordinates must be left on the stack for the `ConvertDesignVector` procedure. This procedure should take the normalized coordinates, generate `WeightVector` values, and leave them on the stack for the `makeblendfont` operator.

## 4 Using Multiple master fonts with $\TeX$

Multiple master fonts come with a set of multiple master AFM files, which are called “AMFM” (Adobe Master Font Metrics) files. This file contains information about the number of master designs, the number of axes, the `BlendDesignPositions` and `BlendDesignMap` arrays, as well as the names, and `weightvector` for the master designs, from which all font instances are derived.

To get the actual metric information for the characters in a font instance, one has to combine the metric information



deriving font-instance parameters in normalized coordinate space in function of the four master designs are given by:

$$\begin{aligned} w_1 &= (1-x)(1-y) & w_2 &= (1-x)y \\ w_3 &= x(1-y) & w_4 &= xy \end{aligned}$$

where  $x$  is the normalized weight and  $y$  the normalized width. The corresponding mapping parameters between design space and normalized coordinates are:

```
BlendDesignPositions [ [0 0] [1 0] [0 1] [1 1] ]
BlendDesignMap [ [[215 0][830 1]][[300 0][700 1]] ]
BlendAxisTypes [ /Weight /Width ]
```

Now one can extract any of the boundingbox and kern entries for a given font instance by getting the element in question from the eight (or four, in the case of Myriad) master files and calculating the interpolated value. To make matters simpler an explicit example will be given for the Myriad font, since it involves only four numbers in each case. Fig. 6 shows some parts of the four master-design afm files

When the instance afm file has been created, a suitable metric for T<sub>E</sub>X can be built with `afm2tfm` or the *fontinst* package.

## 5 In practice

We have instantiated the ideas outlined above by developing Unix shell scripts, and adapting an AFM-parsing program distributed by Adobe. The main script takes the following actions:

1. create a small PostScript file to invoke multiple master operators with values passed to the script;
2. run GhostScript on this file to derive normalized weights, and write them to a temporary file;
3. run our “mmafm” program to read master AFM files, write a new instance AFM file, and create a T<sub>E</sub>X metric (our initial setup uses `afm2tfm` to create 8r base-encoded metrics, and EC-encoded virtual fonts for actual use);
4. write a `dvips` map entry and header file to tell the driver about the new font.

Thus a call to our script consists of the parameters `MinionMM zmn18ac6 360 460 6`; this creates a metric file called `zmn18ac6`, using Karl Berry’s scheme to name “Minion, light weight, 8a-encoded, condensed, at 6pt design size”. The entry in the map file reads `zmn18rc6 zmn18ac6 " TeXBase1Encoding ReEncodeFont "` `<8r.enc <MinionMM.pfb <zmn18ac6.pro`, and `zmn18ac6.pro` contains the code:

```
/zmn18ac6 /MinionMM findfont
dup begin [
  360 460 6   NormalizeDesignVector
  ConvertDesignVector
] end makeblendedfont definefont
```

In addition, we hand-wrote “fd” files to tell L<sup>A</sup>T<sub>E</sub>X how to match up the various weight and width instances we created to its notions of series and shape. The only complication here was that the Minion font has an optical size axis, and we built four instances which we wanted L<sup>A</sup>T<sub>E</sub>X to use at different user sizes:

```
\DeclareFontShape{T1}{zmn}{lc}{n}{%
  <-7>zmn18tc6 %
  <7-10>zmn18tc8 %
  <10-15>zmn18tc11 %
  <15->zmn18tc18}
{}
```

The effect of the optical sizes is demonstrated by Fig. 7 which shows the 6pt and 18pt instances scaled to the same size. The differences in design are as apparent as a corresponding examples from Computer Modern would be.

The tools we developed served to test the ideas, and build a set of metrics; they are available from us on request, but users should beware that they are neither intuitive in use, nor robust. It is to be hoped that a more functional, portable, solution will be developed in time. The keen T<sub>E</sub>Xie may be interested in developing a `MakeTeXTFM` script for Unix `web2c` systems to apply the programs on the fly from within T<sub>E</sub>X.

*Maths in L<sup>A</sup>T<sub>E</sub>X: Part 5, Getting started on arrays*

```
FontName MyriadMM-LightCn
FamilyName Myriad MM
Weight Light
ItalicAngle 0
IsFixedPitch false
FontBBox -52 -250 970 818
...
StartKernPairs 974
KPX A z 10
KPX A y -31
KPX A x 4
KPX A w -36
KPX A v -42
KPX A u -9
KPX A t -17
KPX A s 0
KPX A r -4
KPX A quoteright -90
KPX A quotedblright -90
KPX A q -9
KPX A p -4
KPX A o -12
...
EndKernPairs

StartFontMetrics 2.0
FontName MyriadMM-BlackCn
FamilyName Myriad MM
Weight Black
ItalicAngle 0
IsFixedPitch false
FontBBox -64 -250 970 843
...
StartKernPairs 974
KPX A z 10
KPX A y -10
KPX A x 0
KPX A w -10
KPX A v -10
KPX A u 0
KPX A t 0
KPX A s 10
KPX A r 0
KPX A quoteright -20
KPX A quotedblright -20
KPX A q 0
KPX A p 0
KPX A o 0
...
EndKernPairs

FontName MyriadMM-LightSemiEx
FamilyName Myriad MM
Weight Light
ItalicAngle 0
IsFixedPitch false
FontBBox -58 -250 1100 825
...
StartKernPairs 974
KPX A z 25
KPX A y -10
KPX A x 0
KPX A w -10
KPX A v -25
KPX A u -10
KPX A t 0
KPX A s -10
KPX A r 0
KPX A quoteright -30
KPX A quotedblright -30
KPX A q -10
KPX A p 0
KPX A o -10
...
EndKernPairs

FontName MyriadMM-BlackSemiEx
FamilyName Myriad MM
Weight Black
ItalicAngle 0
IsFixedPitch false
FontBBox -48 -250 1432 867
...
StartKernPairs 974
KPX A z 7
KPX A y -44
KPX A x -6
KPX A w -47
KPX A v -62
KPX A u -22
KPX A t -32
KPX A s -6
KPX A r -10
KPX A quoteright -90
KPX A quotedblright -90
KPX A q -18
KPX A p -10
KPX A o -18
...
EndKernPairs
```

**Figure 6.** The four afm files for the Myriad master designs



---

## VII Maths in L<sup>A</sup>T<sub>E</sub>X: Part 5, Getting started on arrays

R. .A. Bailey  
Queen Mary and Westfield College  
University of London

---

### Recall

This is the fifth part of a sequence of tutorials on typesetting Mathematics in L<sup>A</sup>T<sub>E</sub>X. The first four appeared in issues 4.4, 4.5, 5.1 and 5.2 of *Baskerville*. The series includes some things which can be found in [3], but I am working in more things which, while straightforward and necessary for Mathematical work, are not in [3] or [4]. In this tutorial and the next I cover arrays: matrices, tables of data, aligned equations, and other items with a two-dimensional layout.

I do not cover the extra array goodies provided by packages such as Frank Mittelbach's `array.sty`, David Carlisle's plethora of array add-ons or the new packages from the American Mathematical Society. I hope that someone else will write a follow-on article describing some or all of those, particularly if they can give easier methods of doing some of the more cumbersome items in this tutorial.

In case you missed the previous tutorials, I remind you that I expect you, the reader, to do some work. Every so often comes a group of exercises, which you are supposed to do. Use L<sup>A</sup>T<sub>E</sub>X 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 tell me. Either write to me at

School of Mathematical Sciences  
Queen Mary and Westfield College  
Mile End Road  
London E1 4NS

with hard copy of your input and output, or email me at `r.a.bailey@qmw.ac.uk` with a copy of the smallest possible piece of L<sup>A</sup>T<sub>E</sub>X input file that contains your attempt at the answer. In either case I will include a solution in the following issue of *Baskerville*: you will remain anonymous if you wish.

## 8 Mathematical arrays

### 8.1 Basics

The `array` environment is used to set all manner of aligned Mathematics. A simple example is:

```
\begin{array}{lrrr}
x & 1 & 2 & 5 \\
x^2 & 1 & 4 & 25 \\
\end{array}
```

This array has four columns. The alignment of these columns is shown in the argument `lrrr` of the environment. The first column is left-aligned; the other three are right-aligned. It is also possible to have a centred column, shown by a `c`. Typically columns of Mathematical symbols are centred, integers are right-aligned and powers of a single symbol are left-aligned. >From now on I shall refer to the argument such as `lrrr` as the *columns specifier*.

Entries in each row of the `array` are separated by ampersands. The end of a row (except the last one) is shown by `\\`. If the last few entries in a row are blank there is (usually) no need to put in all the ampersands.

The `array` environment can be used only in Maths mode. All its entries are automatically in Maths mode, in `\textstyle`. If you have fractions or summations in an array you may want to precede them by `\displaystyle`.

Each entry is in a separate 'box', which is typeset before the array is built up. Thus switches such as `\bf`, `\displaystyle` and `\raggedright` extend no further than the end of the current entry. If a binary relation or operator, such as `=` or `+`, occurs at the beginning or end of an entry (L<sup>A</sup>)T<sub>E</sub>X cannot give it the proper spacing relative to the previous or succeeding entry.

$$\begin{pmatrix} 2 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 4 \end{pmatrix} \begin{pmatrix} 5 \\ -3 \\ 8 \end{pmatrix} = \begin{pmatrix} 5 \\ -22 \\ 30 \end{pmatrix}$$

Figure 1. A matrix equation

$$\det \begin{bmatrix} \frac{\partial h_1}{\partial y_1} & \frac{\partial h_1}{\partial y_2} & \dots & \frac{\partial h_1}{\partial y_n} \\ \frac{\partial h_2}{\partial y_1} & \frac{\partial h_2}{\partial y_2} & \dots & \frac{\partial h_2}{\partial y_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_n}{\partial y_1} & \dots & \dots & \frac{\partial h_n}{\partial y_n} \end{bmatrix}$$

Figure 2. The Jacobian of  $h$

### 8.2 Matrices and vectors

The most obvious use for `array` is for a matrix. Use `\left` and `\right` to surround the matrix with fences of the right size. For example, the matrix in Figure 1 begins

```
\left(
\begin{array}{rrr}
```

The commands `\vdots` and `\ddots` are useful in matrices. They are both used in Figure 2.

An array can have a single column, as in the column vectors in Figure 1, or a single row, which is useful for a permutation or a row vector.

### 8.3 Arrays of arrays

Try to think logically, rather than visually, about the contents of an array. For example, if you have six  $5 \times 5$  Latin squares set out in a  $2 \times 3$  rectangle then you might think that you have 10 rows and 15 columns. However, it is more logical to do `\begin{array}{ccc}` and then make each entry in this array a new array with five centred columns. One advantage of this approach is that it makes editing the file much easier when you decide to change the order of the Latin squares, or to lay them out in a  $3 \times 2$  rectangle.

### 8.4 Changing the row spacing

The default spacing between the rows of an `array` is fine for simple matrices with numbers in, but is often insufficient for large entries, particularly if you use `\displaystyle`. To adjust the amount of space after the current row, put an explicit length in square brackets after the `\\`: for example, `\\[20pt]`.

I find it useful to work in multiples of `\jot`: this length is the usual extra space between lines of displayed Mathematics. Curiously, `\jot` is equal to 3 points in all of L<sup>A</sup>T<sub>E</sub>X's three point sizes, so an adjustment in a 10 pt document may not be correct if you change the document to 11 pt. If you work in multiples of the length `\baselineskip` then no extra adjustment is needed if the point size of the document is changed.

Because of the complicated way in which the row-spacing in an array is adjusted to fit the contents, the length given in `[ ]` is not exactly an *addition*. You have to experiment to find out what works best: doubling the length given does not usually double the space between the rows.

The existence of this optional length parameter to `\\` implies that no row of an array may start with `[`. If you need to do this, precede it with an empty pair of braces: `{}[ . . .`

Changing the inter-row spacing for the whole array is much simpler than changing it for a single row. This spacing is governed by the value of the number `\arraystretch`, whose default setting is 1. Between the rows of an array, the length `\baselineskip` is multiplied by `\arraystretch`. In Figure 2, the rows have been spread out by putting

```
\renewcommand{\arraystretch}{2}
```

before the `\begin{array}`. However, this command was given *after* the start of the displayed Maths, so that it remained local to that particular display. If you experiment with other values of `\arraystretch` in this example you will find that there is virtually no separation between the first two rows when `\arraystretch` is smaller than 2. Figure 3 also shows an array in which `\arraystretch` has been altered.

Arrays inside arrays can have different values of `\arraystretch`. For example, in

```
\[
```

|            |  |              |   |
|------------|--|--------------|---|
| $W_{1_G}$  | $V_0$  | 1            | one stratum                             |
| $W_{\Psi}$ | $(V_{\text{rows}} + V_{\text{cols}}) \cap V_0^\perp$ | $2(q^2 + q)$ | direct sum of two unidentifiable strata |
| $W_{\eta}$ | $(V_{\text{rows}} + V_{\text{cols}})^\perp$          | $q^3$        | one stratum                             |

Figure 3. A table of mathematical information

$$\alpha_1: p + q\omega + r\omega^2 + s\omega^3 + t\omega^4 \mapsto p + q\omega + r\omega^2 + s\omega^3 + t\omega^4$$

$$\alpha_2: p + q\omega + r\omega^2 + s\omega^3 + t\omega^4 \mapsto p + s\omega + q\omega^2 + t\omega^3 + r\omega^4$$

$$\alpha_3: p + q\omega + r\omega^2 + s\omega^3 + t\omega^4 \mapsto p + r\omega + t\omega^2 + q\omega^3 + s\omega^4$$

$$\alpha_4: p + q\omega + r\omega^2 + s\omega^3 + t\omega^4 \mapsto p + t\omega + s\omega^2 + r\omega^3 + q\omega^4$$

Figure 4. Four automorphisms of a splitting field

```
\renewcommand{\arraystretch}{2}
\begin{array}{ccc}
A & & B & & C
\end{array}
```

the outer array has `\arraystretch` equal to 2, while the inner array which is the third entry in the first row has `\arraystretch` equal to 1.2.

### 8.5 Changing the column spacing

The default inter-column spacing in an array is one quad. To change the spacing between a pair of adjacent columns, use `@{...}` between their alignment codes in the columns specifier, putting inside the braces the actual horizontal space required. For example, `\begin{array}{cc@{\quad}c@{ }c}` produces an array with four centred columns in which the inter-column spaces are, in order, one quad, two quads and zero.

At the start and end of the array there is usually a space of width half the inter-column space. These can be suppressed by putting `@{ }` at the start and end of the columns specifier.

The overall inter-column spacing is controlled by the value of the length `\arraycolsep`, which is half the default inter-column space. This can be changed by using `\addtolength` in the usual way. I like to alter `\arraycolsep` by multiples of itself. For example, to treble the size of `\arraycolsep` do

```
\addtolength{\arraycolsep}{2\arraycolsep}
```

while to halve it make the last parameter `-0.5\arraycolsep`. In Figure 3, `\arraycolsep` has been multiplied by 2.5.

Finally, the command `\extracolsep` inside an `@{...}` can be used to insert extra space to the left of all subsequent columns: for example, `@{\extracolsep{1em}}`. This is not suppressed by a subsequent `@{...}`.

My remarks about quads above are not strictly true unless your current text font is Computer Modern roman in 10 pt. A quad length is defined to be 1 em, whose size depends on the current font, whereas `\arraycolsep` is defined to be 5 pt, irrespective of the point size of the document. I find it very odd that L<sup>A</sup>T<sub>E</sub>X, which has been so carefully set up so that the user can apparently change painlessly between 10 pt, 11 pt and 12 pt, does not change the size of Maths lengths like `\jot` and `\arraycolsep` in the different point sizes.

### 8.6 Words in arrays

The odd word or two in an array can be put in by using `\mbox`. But what should you do when you have a verbal phrase extending over more than one line, as in the final column of Figure 3?

Most people's first thought is to have a separate line of the array for each line of text. As usual, this visual approach is inferior to the logical one when it comes to revising your file. Hard experience has shown me that you have to



$$3(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) = (x - 1, y + 2) \begin{bmatrix} 4 & 1 \\ 1 & 1 \end{bmatrix} \begin{pmatrix} x - 1 \\ y + 2 \end{pmatrix}$$

Figure 5. Matrices aligned on their top rows

keep altering which words go on which lines as you alter other parts of the array. It is better to use the fourth type of column code `p`, which creates a paragraph box of a specified width. Thus the columns specifier in Figure 3 is actually `cccp{1.5in}`. Just guess a length to put inside `p{...}` at first: changing it later is much simpler than shuffling words around.

Figure 3 demonstrates another good reason to use the `p` code. The rows are far apart, because `\arraystretch` has been magnified. But we *don't* want the lines of text at the end of the second row to be that far apart. We could make an inner array with a smaller value of `\arraystretch`, but it's simpler to use `p`.

The only problem with the naked `p` columns is that they are set justified on both sides, which is absurd in narrow columns. I usually put `\centering` or `\raggedright` at the beginning of every entry in such a column. Unfortunately, you cannot do this in the final column of an array, because L<sup>A</sup>T<sub>E</sub>X gets confused about whether `\` refers to the row of the array or the line in the paragraph. If necessary, I get round this by adding a dummy unused column at the end of the array. Thus I told a lie above: the columns specifier in Figure 3 is really `cccp{1.5in}c`.

### 8.7 Inter-column constants

The code `@{...}` can be used for putting anything between two columns that must appear in every row, not just spaces. This is convenient for any repeated items. For example, the columns specifier in Figure 4 is

```
{l@{\colon p + q\omega + r \omega^2 +
s\omega^3 + t\omega^4 \mapsto p +
{}}r@{\omega + {}}r@{\omega^2 +
{}}r@{\omega^3 + {}}r@{\omega^4}}
```

while the first row is just

```
\alpha_1 & q & r & s & t
```

Notice the pairs `{}` that have been inserted to make `+` behave as a binary operator.

### 8.8 Positioning

By default, each array is positioned so that it is centred vertically. In Figure 1, the equals sign is aligned with the centre of each of the three arrays. To align an array on its top row, start it with `\begin{array}[t]{...}`; to align it on its bottom row, use `[b]` in place of `[t]`.

These positionings work well for simple tables of numbers. However, the logical interaction between them and `\left` and `\right` fences produces bizarre results (try it!). If you want matrices aligned on their top rows, as in Figure 5, you have to do something quite complicated, which I shall come back to later.

### 8.9 Cases

We often write equations where the value on the right-hand side depends on some condition, as in

$$X(\omega, t) = \begin{cases} 1 & \text{if } \phi(\omega) = t \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

You can do the right-hand side of this using an array, but it is simpler to use the direct plain T<sub>E</sub>X command `\cases`. The right-hand side of Equation (1) was produced with

```
\cases{1 & if $\phi(\omega) = t$\cr
0 & otherwise.}
```

The argument of `\cases` is like a two-column array, with its rows separated by `\cr` instead of `\`. The first entry in each row is in Maths mode; the second is in non-Maths mode. Both columns are left-aligned.

### 8.10 Binomial coefficients

If you write your binomial coefficients as  $\binom{6}{2}$  rather than  ${}^6C_2$  you may be tempted to build them with an array. Don't. Use the plain T<sub>E</sub>X infix command `\choose`, as follows.

$$6 \ \text{\choose} \ 2 \quad \binom{6}{2}$$

It comes out rather differently in `\textstyle` and in `\displaystyle`.

Because `\choose` does not have its two arguments in braces, it is safest if you enclose the whole expression in grouping braces, as in `{6 \choose 2}`, to ensure that `\choose` does not search too far back or too far forward for its arguments. If you need to use this a lot, make a L<sup>A</sup>T<sub>E</sub>X-style macro such as

```
\newcommand{\binom}[2]{\{#1\choose#2}}
```

The command `\atop` works similarly, but omits the parentheses.

**Exercises**

**Exercise 55** The matrix  $\begin{bmatrix} 1 & 3 \\ 3 & 5 \end{bmatrix}$  is symmetric.

**Exercise 56** The determinant is given by

$$\det A = \begin{vmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{vmatrix}$$

**Exercise 57** For a rotation,

$$\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x\cos\theta + y\sin\theta \\ y\cos\theta - x\sin\theta \end{pmatrix}.$$

**Exercise 58** Here is a pair of mutually orthogonal Latin squares:

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| A | B | C | α | β | γ |
| B | C | A | γ | α | β |
| C | A | B | β | γ | α |

Square 1      Square 2

**Exercise 59** A few real functions:

- $x \mapsto x^2$       differentiable everywhere
- $x \mapsto |x|$       differentiable everywhere except at the origin
- $x \mapsto \lfloor x \rfloor$       continuous everywhere except at the integers

**Exercise 60** Typeset the table in Figure 3.

**Exercise 61** Pascal's Triangle begins

|  |   |   |    |    |    |   |   |
|--|---|---|----|----|----|---|---|
|  |   |   | 1  |    |    |   |   |
|  |   | 1 |    | 1  |    |   |   |
|  | 1 |   | 2  |    | 1  |   |   |
|  | 1 | 3 |    | 3  | 1  |   |   |
|  | 1 | 4 | 6  |    | 4  | 1 |   |
|  | 1 | 5 | 10 | 10 | 5  | 1 |   |
|  | 1 | 6 | 15 | 20 | 15 | 6 | 1 |

**Exercise 62** The Stirling numbers of the second kind are defined by

$$S(n, k) = \frac{1}{k!} \sum_{j=1}^k (-1)^{k-j} \binom{k}{j} j^n.$$

|        |        |        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|--------|--------|
| Year 1 | Year 2 | Year 1 | Year 2 | Year 1 | Year 2 | Year 1 | Year 2 |
| 269    | 466    | 134    | 590    | 211    | 505    | 209    | 352    |
| 100    | 219    | 74     | 137    | 102    | 363    | 29     | 254    |
| 197    | 421    | 88     | 356    | 191    | 563    | 19     | 106    |
| 216    | 708    | 25     | 212    | 67     | 338    | 44     | 268    |

Figure 6. Number of eelworms in two successive years

**Exercise 63** If the random variable  $X$  is symmetric about 0 and if  $Y = X^2$  then

$$f_Y(y) = \begin{cases} 0 & \text{if } y \leq 0 \\ \frac{f_X(\sqrt{y})}{\sqrt{y}} & \text{if } y > 0. \end{cases}$$

**Exercise 64**

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \cdots & \ddots & \vdots \\ \sigma_{1n} & \cdots & \cdots & \sigma_{nn} \end{bmatrix}$$

**Exercise 65**

$$\begin{array}{ccccccc} M_0 & \subseteq & M_1 & \subseteq & M_2 & \subseteq & \cdots \subseteq M_m \\ \uparrow & & \uparrow & & \uparrow & & \\ \text{null} & & \text{linear} & & \text{quadratic} & & \end{array}$$

## 9 Tables of data

### 9.1 General

The extra topics that I cover in this section are most pertinent to tables of data, but do sometimes arise in Mathematical arrays. Likewise, many considerations about Mathematical arrays also apply to tables of data. For example, the table in Figure 6 does not really have eight columns: it has two long columns but has been broken down into four parts to fit on the page nicely. So it is set as an array with one row and four columns, each of whose entries is an array with two columns.

L<sup>A</sup>T<sub>E</sub>X has another environment, called `tabular`, which is very similar to `array`. It may be used inside or outside Maths mode. By default, each of its entries is set in ordinary mode. However, any extra space on either side of the ampersands has no effect. The length `\arraycolsep` is replaced by the length `\tabcolsep`. Tables of data often have a mix of words and symbols, and you have to decide which is less trouble: using an `array` and putting the words in `mboxes` or `p` columns, or using a `tabular` and putting the symbols in `$. . . $`. If the latter, don't forget to put numbers in `$. . . $` if they involve a minus sign.

### 9.2 Horizontal rules

To obtain a horizontal rule stretching the full width of the array, use `\hrule`. Don't forget to finish the previous row with `\`, even if it is the last row of the array, as at the end of Figure 7. Two or more `\hrules` can follow each other with no intervening `\`. A little vertical space is put between two such horizontal rules. However, any vertical rules in the array (see below) will be suppressed between the successive horizontal rules.

For a horizontal rule spanning only columns 2–4, say, use `\cline{2-4}`. Again, finish the preceding row with `\`. Figure 7 shows an example of this. Successive `\clines` will be interpreted as being in the same row, as in `\cline{2-4}\cline{7-7}`, for example, so that `\cline{2-4}\cline{2-4}` does *not* produce two horizontal lines with a little space between them. Likewise, there is no space between an `\hline` and an immediately preceding or succeeding `\cline`.

|  | 1961 | 1971 | 1981 |
|--|------|------|------|
| Billion passenger kilometres travelled by: |      |      |      |
| Air  | 1    | 2    | 3    |
| Rail                                       | 39   | 36   | 34   |
| Road                                       |      |      |      |
| Buses and coaches                          | 67   | 51   | 42   |
| Cars, taxis and two-wheeled motor vehicles | 142  | 284  | 380  |
| Bicycles                                   | 10   | 4    | 4    |
| Total                                      | 259  | 377  | 406  |

Figure 7. Passenger transport (from [1])

| Premiums |      |
|----------|------|
| 1 Month  | £ 18 |
| 3 Months | £ 48 |
| 6 Months | £ 95 |
| 9 Months | £136 |

Figure 8. Travel insurance

|                    |        |
|--------------------|--------|
| Bootstrap          | 0.301  |
| Jackknife          | 0.314  |
| Delta method       | 0.244* |
| Half-samples       | 0.364† |
| Random subsampling | 0.423‡ |
| Normal theory      | 0.302  |

Figure 9. Sticking out to the right (from [2])

### 9.3 Vertical rules

A vertical rule is obtained by putting | at the appropriate place in the columns specifier. For example, the columns specifier for the tabular in Figure 8 is

```
|r@{ Month}l@{\quad\pounds}r|
```

Two or more |s may come next to each other.

For a vertical rule in a single item, stretching from the top to bottom of its row, use \vline. This cannot be used in a p column, but may be put inside @{ . . . }.

If you omit the trailing ampersands in a row with few items the trailing vertical rules will also be omitted. This can be a nuisance in a sparse table. If you want a vertical rule at each side of a sparse table, consider doing this:

```
\begin{array}{|c|}
\begin{array}{@{}...@{}}
...
\end{array}
\end{array}
```

The length \arrayrulewidth controls the thickness of all the horizontal and vertical rules just described. The width of the space between the vertical rules created by || in the columns specifier is controlled by the length \doublerulesep, as is the vertical space between two successive \hlines.

### 9.4 Spanning several columns

The \multicolumn command is used to create an entry spanning several columns. It takes three arguments. The first is the number of columns to be spanned; the second is the code for the type of column to be used; the third is the text of the entry.

```
\begin{tabular}{crrrr}
Type & \multicolumn{4}{c}{Grade}\\
\cline{2-5}
& 12 & 16 & 24 & 30\\
\hline
$A$ & 0 & 1 & 21 & 24\\
```

```
$B$ & 1 & 6 & 24 & 13
\end{tabular}
```

| Type | Grade |    |    |    |
|------|-------|----|----|----|
|      | 12    | 16 | 24 | 30 |
| A    | 0     | 1  | 21 | 24 |
| B    | 1     | 6  | 24 | 13 |

Any | in the columns specifier is regarded as belonging to its preceding column, and may be overwritten by \multicolumn unless it is included in the second argument. The exception is any | to the left of the first column, which is regarded as part of the first column. For example, the first row in Figure 8 is

```
\multicolumn{3}{|c|}{Premiums}\
```

Expressions @{ . . . } in the columns specifier are similarly assigned to columns and similarly overwritten.

Indented row labels for subheadings can also be conveniently obtained by using \multicolumn. The tabular in Figure 7 begins with three l columns. The first row after the second horizontal rule is a \multicolumn spanning three columns; the next begins with & followed by a \multicolumn spanning two columns.

Unfortunately, if the width of a multicolumn entry spanning three or more columns is more than the total width of the spanned columns, L<sup>A</sup>T<sub>E</sub>X cannot adjust the spacing between those columns to make it equal: it puts all the extra space to the left of the last of the spanned columns. The only way that I know around this is to use @{ . . . } to put the same amount of extra space between each adjacent pair of the spanned columns.

### 9.5 Changing the type of an entry

Although its name does not suggest it, the command \multicolumn is also used to change the type of a single entry. This is most often used to give a centred heading to a column of right-aligned data.

```
\begin{array}{rrr}
\multicolumn{1}{c}{n} & & \\
\multicolumn{1}{c}{n^2} & & \\
\multicolumn{1}{c}{\phi(n)} & & \\
9 & 81 & 6 \\
10 & 100 & 4 \\
11 & 121 & 10 \\
\end{array}
```

| $n$ | $n^2$ | $\phi(n)$ |
|-----|-------|-----------|
| 9   | 81    | 6         |
| 10  | 100   | 4         |
| 11  | 121   | 10        |

As this example shows, this works fine when the centred heading is narrower than the widest other entry in the column. However, when it is wider then all the other entries in the column are aligned with its right-hand edge, when probably what you wanted was to have them right-aligned among themselves and then all centred under the heading. How can this be fixed?

If most of the numbers are the same length, a reasonable solution is to make the whole column centred and put in \phantom{0}s to make the shorter numbers as long as the longest. This works because all digits have the same width. That is what I did in Figure 6.

I adopted a different solution in Figure 7, where I decided that the year headings, not being numbers to be added to other numbers in their columns, should be centred. This time I used the columns specifier rrr for each column of data. In the headings I used \multicolumn{3}{c}{ . . . } and in the data rows I left one blank entry either side of each data entry. The result is rather widely spaced: space could be saved by using @{ . . . } between each data column and its blank neighbours.

The command \multicolumn{1} can also be used to remove a vertical rule or the contents of an @{ . . . } from one particular row. For example, if the columns specifier is l | r then the vertical rule is missing from any row whose first entry is made with \multicolumn{1}{l}.

### 9.6 Decimal tabulation

Sometimes a column of numbers with decimal points should be aligned on those decimal points. If they all have the same number of digits after the point, there is no problem: simply use an `r` column. If only a few numbers are missing a few digits after the point, fill in the spaces with `\phantom{0}` and still use an `r` column. Only if you have a long column of numbers with very variable numbers of digits after the point is it worth going to the trouble of using the columns specifier `r@{.}l` and entering a number such as 3.142 as `3 & 142`.

### 9.7 Things that stick out

Sometimes you have a column of numbers that should be right-aligned but some of the entries have accretions that should stick out to the right, as in Figure 9, or a left-aligned column with pieces sticking out to the left. There are three ways of dealing with this, each appropriate in different circumstances.

If the pieces that stick out are all digits, use `\phantom{0}` as described above.

Sometimes the piece that sticks out is very small and occurs in only one row. If a displayed Mathematical array ends the sentence then sometimes the full stop needs to come outside the array, vertically centred (with `\end{array}.`); but sometimes the final row of the array needs a full stop at the end of it without upsetting the column alignment. Then the trick is to add the full stop in such a way that (L<sup>A</sup>)T<sub>E</sub>X thinks that it doesn't take up any space. According to [3, page 98], you do this with `\makebox[0pt][l]{.}`, but the plain T<sub>E</sub>X `\rlap{.}` is shorter and achieves the same effect. For a small item sticking out on the left use `\makebox[0pt][r]{ }` or `\llap{ }`. Of course, the content of `\makebox{ }` or `\rlap` or `\llap` is not in Maths mode.

Statisticians sometimes put a question mark to indicate dubious data, and scientists often decorate numbers with stars to show statistical significance. In such cases it is best to use the columns specifier `r@{ }l` for the column: the `r` part is used for the numbers and the `l` part is used for any decorations. In Figure 9 the columns specifier is `lr@{ }l` and the fourth line is

```
Half-samples & 0.364 & \dag\
```

To see the difference between these last two solutions, compare the following.

```
\begin{array}{|r@{ }l|}
20 & ++ & | 20++ |
\end{array}

\begin{array}{|r|}
20 \rlap{ $++$ } & | 20|+
\end{array}
```

### 9.8 Columns specifiers

We have seen that a columns specifier is a string of the following objects: `r`, `l`, `c`, `p{...}`, `@{...}` and `|`. It is easy to lose track when setting up the column specifier for a complicated array. However, repeated elements may be simplified by using `*`. Thus an array with 15 centred columns can be specified with `*{15}{c}`. This shorthand is really advantageous for something complicated like `l|*{4}{r@{ }lrr|}`. Moreover, `*` expressions can be nested within each other.

## Exercises

**Exercise 66**

Values of  $v_{\gamma\chi}$  for the design

| $\gamma$      | $\chi_0$ | $\chi_1$               | $\chi_2$               | $\chi_3$      | $\chi_4$               |
|---------------|----------|------------------------|------------------------|---------------|------------------------|
| $\mu$         | 1        | 0                      | 0                      | 0             | 0                      |
| $\sigma$      | 1        | 0                      | 0                      | 0             | 0                      |
| $\rho$        | 1        | 0                      | 0                      | $\frac{1}{4}$ | 0                      |
| $\kappa$      | 1        | $\frac{3+e_2-e_1}{36}$ | $\frac{3+e_4-e_2}{36}$ | 0             | $\frac{3+e_1-e_4}{36}$ |
| $\varepsilon$ | 1        | 1                      | 1                      | 1             | 1                      |

**Exercise 67** Table of results:

|       | $B_1$ | $B_2$ |     |
|-------|-------|-------|-----|
| $A_1$ | 37    | 53    | 90  |
| $A_2$ | 41    | 48    | 89  |
|       | 78    | 101   | 179 |

**Exercise 68** Typeset the table in Figure 6.

**Exercise 69**

| Source      | df | SS     | MS     | VR     |
|-------------|----|--------|--------|--------|
| $x_1$       | 1  | 104474 | 104474 | 1220.5 |
| $x_2   x_1$ | 1  | 2284   | 2284   | 26.68  |
| regression  | 2  | 106758 |        |        |
| residual    | 15 | 1284   | 85.6   |        |
| 'total'     | 17 | 108042 |        |        |

**Exercise 70** Typeset the table in Figure 7.

**Exercise 71**

| Date        | Miles  | Gallons | Cost   |
|-------------|--------|---------|--------|
| 27 December | 46674  | 4.18    | £ 5.56 |
| 3 January   | 46038  | 3.93    | £ 5.00 |
| 6 January   | 47956? | 7.1     | £10.50 |
| 10 January  | 47292  | 7.89    | £10.50 |
| 17 January  | 47464  | 4       | £ 5.00 |
| .....       |        |         |        |
| 3 February  | 48112  | 6.88    | £ 9.00 |

**References**

- [1] CENTRAL STATISTICAL OFFICE: *Key Data 1986*, Her Majesty's Stationery Office, London, (1986).
- [2] EFRON, B.: *The Jackknife, the Bootstrap and Other Resampling Plans*, Society for Industrial and Applied Mathematics, Philadelphia, (1982).
- [3] LAMPORT, L.: *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*, first edition, Addison-Wesley, Reading, Mass., (1986).
- [4] LAMPORT, L.: *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*, second edition, Addison-Wesley, Reading, Mass., (1994).

---

## VIII Report of visit to CERN to attend presentation of $\Omega$

Robin Fairbairns  
University of Cambridge Computer Laboratory

---

### 1 Introduction

As you will have seen in earlier *Baskervilles* of this year, the Francophone T<sub>E</sub>X users' group, Gutenberg, arranged a meeting in March at CERN (Geneva) to 'launch'  $\Omega$ . UK T<sub>E</sub>X Users' Group responded to Gutenberg's plea for support to enable T<sub>E</sub>X users from impoverished countries to attend, by making the first disbursement from UK T<sub>E</sub>X Users' Group's newly-established Cathy Booth fund. That money (together with some of the outstanding surplus from T<sub>E</sub>Xeter '88), was used in the meeting's fund that supported the attendance of four representatives of CyrTUG (which covers Russian and other users of the Cyrillic alphabet) and of one representative of CSTUG (Czech republic and—still—Slovakia). Apart from these, there was a large contingent from France, several from Switzerland (including one German-speaking Swiss and one Englishwoman working in Lausanne), and one each from Germany, the Netherlands, Spain, Australia<sup>2</sup> and the UK (me).

The speakers at the meeting were Michel Goossens (the president of Gutenberg, as host for Gutenberg and as an expert on background to, and the use of Unicode), and Yannis Haralambous and John Plaice,  $\Omega$ 's two developers.

The meeting can be accounted a success; all that attended enjoyed themselves, and also learnt a lot. This article is the first of (at least) two in which I will describe the thinking that led to the production of  $\Omega$ , the problems that it addresses and the ways it solves those problems.

### 2 What is $\Omega$ ?

$\Omega$  is an extension of T<sub>E</sub>X and related programs that has been designed and written by Yannis Haralambous (Lille) and John Plaice (Université Laval, Montréal). It follows on quite naturally from Yannis' work on exotic languages, which have always seemed to me to be bedevilled by problems of text encoding.

Simply,  $\Omega$  (the program) is able to read scripts that are encoded in Unicode (or in some other code that is readily transformable to Unicode), and then to process them in the same way that T<sub>E</sub>X does. Parallel work has defined formats for fonts and other necessary files to deal with the demands arising from Unicode input, and upgraded versions of METAFONT, the virtual font utilities, and so on, have been written.  $\Omega$  itself is based on the normal web2C distribution that is at the base of most modern Unix implementations, and of at least one of the PC versions that is freely available.

### 3 Why Unicode?

Michel explained to us the sorry history of the development of character sets for use in computing<sup>3</sup>. There are something between 3000 and 6000 languages in use in the world, for which a writing system exists. (The set of languages is shrinking all the time as the deadening effect of cultural intrusion, primarily through the electronic media, overwhelms the desire to support existing cultures to the extent of teaching their language to the young.) The distribution of languages is by no means even throughout the globe (Michel showed us a map), and there are many that have not been and will presumably now never be formally recorded.

When we come to writing systems, we find almost every variation imaginable in use somewhere in the world. The Latin-like system (written left to right with modest numbers of diacritics simply arranged) has very wide penetration, not least because so many languages were first written down by Western European missionaries or other explorers. Languages such as Vietnamese are classified as 'complex Latin-like', with  $\geq 2$  diacritics per character; an artificial example of the same effect is IPA (the International Phonetic Alphabet) which has sub- and super-scripts and joining marks. Languages such as Hebrew and Arabic are written right to left, and constitute another class. Then there are the multiple-ligature writing systems typified by the Indic languages such as Devanagari (of which we had a fascinating

---

<sup>2</sup>Richard Walker, who is currently working in Germany

<sup>3</sup>This is an area where I have some expertise, too, so not all of this comes from Michel



exposition at the 1993 UK T<sub>E</sub>X Users' Group Easter meeting on 'non-American' languages, from Dominik Wujastyk), and finally the syllabic scripts (such as Korean Hangul and Japanese Hiragana and Katakana), and the ideographic scripts (Chinese and Japanese Kanji).

Encodings are needed for computer operations on language of any sort. There are differences between the coded representation and the written (or printed) representation. Everyone who's read about T<sub>E</sub>X at all will know about ligatures (the CM fonts, and most POSTSCRIPT fonts, implement ligatures so that, for example, 'fl' typed appears as 'fl' printed). More significantly, almost all adults in Western cultures write 'joined-up', which is in itself application of a form of ligature. All these ligatures are for presentation, not for information, and so it is unreasonable for them to be represented in a character set. Other ligatures, however, form real characters in some languages (examples are æ in Danish and Norwegian, and œ in French).

In the dark ages (in fact, as recently as the early 1960s, when I started computing), every make of computer system had its own character code, many of them based on the 5-bit teleprinter codes used in telex printers. Eventually, the rather more sophisticated teletypes appeared, which used seven bits of an eight-bit code; this 7-bit codification was standardised as ASCII (the American Standard Code for Information Interchange), which was (in the area of application it was designed for) an excellent code. It had all the properties needed for many of the significant development of computers in the 1960s, but it had one serious flaw: it was not able to encode diacritics, which are used in almost every language (but which your all-American information interchanger would seldom have a need for).

To regularise the resulting mess, ISO adopted the ASCII standard as the basis for an international 7-bit character set, ISO 646. ISO 646 is identical to ASCII in the code points that it specifies; however, some of the characters that ASCII does specify are left "for national variation" in ISO 646; ASCII itself then became the USA national variation of ISO 646. An example of national variation is defined for the UK, which specifies that the code point that holds '#' in ASCII should hold a pounds sign (£). There are versions for various Nordic languages that include characters such as æ or å in place of braces, a version for French with acute, grave and circumflex-accented letters, one for German that offers umlauts and 'sharp s' (ß).

There were various attempts at mechanisms to assign different character sets for use by those who need to use characters from several different sets (for example, someone writing an Swedish-English Dictionary); an example is ISO 2022, which defines escape sequences such switches. These efforts proved impractical (at least they seemed so to me), and 8-bit developments of ISO 646 arose, with the ability (comfortably) to express more than one language.

Thus were born the ISO 8859 character sets. The commonest of these (at least in the ken of most English speakers) is ISO Latin-1 (ISO 8859-1, that is part one of the multi-part standard), which was designed for use by Western Europeans. As well as the 'basic ASCII set' in the first 128 characters, it has diphthongs and vowels appropriate to most Western European languages. Oddly, it omits the œ diphthong that French uses, and (perhaps less surprisingly<sup>4</sup>) it omits some of the accent forms used by Welsh. ISO 8859 didn't stop with part 1, though; there are variants that accomodate Cyrillic (for Russian, Serbian, and several other languages of the old Soviet Union), Arabic, Hebrew, and so on.

This is all well and good, but it doesn't answer the needs of a writer preparing multilingual documents, except in the case that the multiple languages are accomodated in the same part of ISO 8859: it will happen some of the time, but most 'interesting' combinations will require switches of character set whenever the language changes.

So ISO (by this time, jointly with IEC) started development of an all-encompassing character set, to be numbered ISO/IEC 10646 (the difference of 10 000 is no accident). ISO/IEC 10646 was to accomodate every possible language in the world by the simple expedient of allowing 32-bit characters. Of course, no-one can comprehend a 32-bit character set, and so the set was to be structured, as a hypercube of different repertoires; the (0,0,0,0) repertoire would be the same ISO-Latin 1, but all the other sets could be accomodated, too.

Independently, Apple and Microsoft got together to found the Unicode consortium, whose aim was to define 16-bit characters that would cover all the economically important world. This criterion of economic importance could easily have brought down the whole edifice: the (increasingly important) languages of the Far East are at best syllabic (e.g., Korean; Korea claims 11 000 of the code points in Unicode), or even one character per word (e.g., Chinese; a full classical Chinese repertoire would require well in excess of 65 536 characters, thus sinking a 16-bit code single-handedly).

Unicode's sponsors therefore enforced a process called 'Han unification', which aims to put the 'same' character in any of Chinese, Japanese and Korean in the same slot in the table. This unification is a distinctly dubious exercise: the same character may have different significance in the different languages, but they are all represented by the same

---

<sup>4</sup>Given that Wales would have been represented by the BSI in the standardisation process

code point. Contrariwise, the Latin ‘H’, the Russian ‘H’ (which sounds as Latin ‘N’) and the Greek ‘H’ (capital ‘η’) all get different code points despite having the same paper representation. For this reason (among others), there remain doubts as to whether the Japanese, in important particular, will adopt Unicode as a long-term replacement for their own national standards.

In the shorter term, however, there remained the possibility that there would be two conflicting standards for the future of character codes — a *de facto* one (Unicode) and ISO/IEC 10646. The ISO/IEC standard reached its (nominal) final ballot without addressing the relation to Unicode . . . but (fortunately) it failed at that hurdle, and for that reason. Standards people are notorious for ignoring the real world<sup>5</sup>, but this time, they conceded defeat. ISO/IEC 10646 was edited to have the whole of Unicode as its (0, 0, \*, \*) plane, and it has thus passed into the canon of published standards.

So we may now discuss Unicode without running out against the ISO/IEC standard: a splendid example of the behaviour known as “common sense prevailing”.


#### 4 Virtual Metafont and Fonts to Support Unicode

It is known that  $\TeX$  is a general-purpose programming language. In ‘plain’ text, we would type "hello world". For  $\TeX$  output we would type ``hello world'', which would be transparently converted to “hello world”. Thus, the two grave accents and the two single quotes constitute ‘programming’. In the last analysis, you can “do everything with  $\TeX$ ”.

When English is typeset, the convention is that the space, after the full stop is the end of a sentence, is expanded;  $\TeX$  makes provision for this to happen by way of the `\sfcode` mechanism. When French is typeset, the convention is that the space is not expanded; the `\sfcode` mechanism can provide this style of typesetting, as well (cf. the `\frenchspacing` macro of plain  $\TeX$ ).

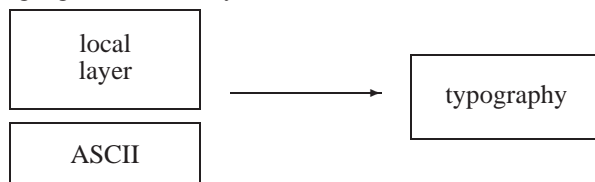
Other features of French typesetting are more difficult to provide in  $\TeX$ . For example, an exclamation mark is separated from the sentence: “en français !”; to program this, the exclamation mark needs to become an ‘active character’, which is always a tricky thing to do.

Setting the French quotation marks (known as guillemets) becomes even more tricky; the guillemets look like little << and >>, and the natural way to program them is by using repeated < or > characters; Bernard Gaulle’s `french.sty` does this (also setting a space between the text quoted and the guillemets), but it’s becoming more and more complicated; even more so when we consider the French rules for quotes within quotes.

More problems arise when we consider the question of diacritics. English rather infrequently has diacritics, so it’s not surprising that  $\TeX$ ’s method of dealing with them isn’t perfect. To typeset an accented character, e.g. ä, one must type `\ "a`; which is typeset as two little boxes stacked on top of one another, rather like . This does work, but these composite glyphs no longer qualify (to  $\TeX$ ) as something that it’s willing to hyphenate— $\TeX$  only hyphenates ‘words’ made up of sequences of letters. A language such as German, with hyphenation suppressed for many words, is hardly a language at all. These observations are what led to the definition of the Cork font encoding, in which a goodly proportion of Western European letters with diacritics appear as single characters; if they are thus represented, words containing them may be hyphenated.

With the Cork encoding, which is in effect an output encoding, we encounter a further problem relating to the nature of communication. The problem arises from the nature of character sets; while there are many well-established character sets, there are seriously different camps into which they fall. For example, the character ‘Ð’ (Thorn), appears in Microsoft Windows’ character set but not in the Macintosh set, while ‘Ω’ appears in the Macintosh set but not in the Windows set; both of these sets are based on ASCII. To solve this problem, of encoding all everything that appears in any character set, there has to be a super-encoding. This can be either a multi-character representation, as in the WWW encoding, html (for example the encoding would for é would be `&eacute;`), or a super-character set, as in Unicode.

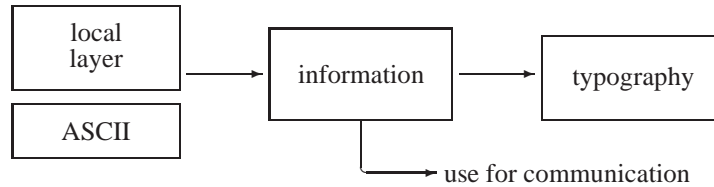
In the present arrangement of typesetting technology, we have the situation where non-English users sit at a computer, and express their own language via a local layer in ASCII or a derivative of it — i.e., we have a picture like:



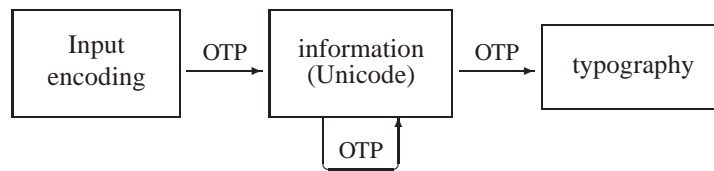
<sup>5</sup>The author has spent an unconscionable long period of his life on these things, and is therefore in a position to know

In this arrangement, the human interface allows the use of local characters, and the display will show what's typed. The typography does the display job again (possibly differently); however, communication of the text to be typeset is difficult, because of the local nature of the interface.

The information to be transmitted needs to be encoded. There is no limit to the number of local encodings that may exist; equally, there is no constraint on the representations used by the typographic system. However, to facilitate the transmission of information, a common schema of its representation in the coded date must exist.



The ultimate mechanism for ensuring that such a schema exists is to require that everything be transmitted in a common encoding scheme;  $\Omega$  employs ISO 10646/Unicode for this. Input text is transformed into  $\Omega$ 's internal 'information' by an Omega Translation Process (OTP); OTPs may also be used to transform the information during its processing within  $\Omega$ , and an OTP is also used to derive the coding of the font, to be used for typesetting, from the Unicode-encoded information within  $\Omega$ :



At this point, we're beginning to trespass on the subject matter of the next article: the internal workings of  $\Omega$ . That article is to appear in the next edition of *Baskerville*.

---

## IX A.N.Other's Gleanings

---

### 1 Framed, or a boojum?

In the latest Nesletter of the BCS eletronic publishing specialist group, my colleague Conrad Taylor points out the following: "it strikes me as odd that people often choose for *electronic* publishing a tool named after the quark, since an electron is, of course, a lepton – just as a muon is – and nary a quark goes into making one up." We can carry this a little further by noting one of the many definitions of Frame: "concoct a false charge or accusation", or even recalling that "the Word was with God". But as someone who has to state that he "works in latex", it is dangerous to bandy these words about. On the other hand, you do get invited to some very interesting parties... What is in a name anyway? I have to admire the sheer chutzpah of naming your product after a zone in an industrial park in Swindon. Trivial Pursuit question: to which vastly under-rated electronic publishing package do I refer?

### 2 Lettres

I continue to be pleasantly surprised by the uniform and high standard of the *Lettres Gutenberg*, the newsletter of our French sister organisation. In fact, you have to admire them even more when you appreciate that they produce not only this excellent newsletter, but a journal which easily stands shoulder to shoulder with *TUGboat*. Our recent joint meeting on Portable Documents was paralleled by a very similar meeting of theirs in Nanterre. Many of the papers presented appeared in *Cahier GUTenberg 19*, making a rather interesting contrast with those published in the last edition of *Baskerville*. The future holds similar excitements: next is a special edition on character encoding (it will be difficult to inject excitement into this dry topic, but if it can be done, it will be done); and in early June, the annual conference takes/took place. This picks up the theme of electronic documents and their distribution once more. Slipping the magic words 'Internet' 'World Wide Web' and 'Acrobat' into your programme is a sure-fire way of packing the halls. I wish them much success: I have many pleasant memories of events organised by GUTenberg: I commend their meetings to you, and urge you to demonstrate your commitment to the progressive forces in Europe by taking the train to their next meeting.

### 3 To be taken daily

A copy of Kopka & Daly's *A Guide to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>* arrived the other day. I have long been an admirer of the first edition of this book. The second edition, which, as its new title implies, reflects the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> standard, maintains the former high level. I think it fairly neatly bridges the gap between Lamport's rather terse account and the uneven whirlwind tour of thousands of packages in the 'three men and a dog' tome which unleashed the new release upon us. I find it intriguing that in a book of over 500 pages, less than 300 are given directly to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, and 200 to appendices. The book is published by Addison-Wesley, ISBN 0-201-42777-X.

### 4 Web

As the World Wide Web continues its inexorable conquest of the world, let's stand back for a moment and see how T<sub>E</sub>X fits in. First, look at its origins. Consider the primitive nature of the early HTML 'standard'. Can you guess why it was so brain dead? If you look at it closely, you'll see that apart from its hypertext anchors, its two main features were tons of headings and various types of list structures. In fact, it was modelled closely on *texinfo*, a variant T<sub>E</sub>X-macro package used chiefly by the Free Software Foundation aficionados. Consider too what stroke of chance it was that led Tim Berners-Lee to sit down with one of CERN's tribe of SGML people for coffee one morning. Sheer luck (or genius) which has done more for the street credibility of SGML than years of conferences in exotic climes and hob-nobbing with the military-industrial complex. Consider also what might have happened if it had been Michel Goossens who had that fateful cup of coffee with Tim. The world would now be beating a path to the door of L<sup>A</sup>T<sub>E</sub>X. Or would it? Nevertheless, L<sup>A</sup>T<sub>E</sub>X is re-appearing within HTML developments. The new 'standard', very much still in draft, includes maths and tables. There is a test-bed available, called *Arena* which will allow you to see what is

*reprinted from Baskerville*

planned. The encoding of maths is essentially that of L<sup>A</sup>T<sub>E</sub>X. And surprisingly, the encoding of tables looks currently just like L<sup>A</sup>T<sub>E</sub>X too – I say surprisingly since `tbl` is more powerful and equally stable.

#### *The Trojan horse option*

I wonder though if we are not missing a real opportunity. While Acrobat, or `pdf`, is now an ‘official’ MIME type (implying that it can be happily included on the Web, and browsers can be configured to handle it in a standardised way), `dvi` has not yet made it to this status. Consider that Acrobat readers are available for Sun machines running Solaris (and Motif), and not for the other myriad of Unix boxes. What is present on almost every Unix box? Why, T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X, and therefore something like `xdvi` which will display `dvi` files. Acrobat is therefore not quite as widely available as we would like (only Solaris, Mac and Windows), while T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X are not as widespread as we would like on Mac and Windows (it’s not that people can’t afford the disk space, since you can hardly buy a machine these days with less than 500Mbyte of disk, but rather that the installation is still such a pig). The Unix piece of the problem is simple, since downloading a `dvi` file over the Web would merely fire up `xdvi` or some similar previewer. With Windows, the situation is more difficult. If we had a nice simple installation of a previewer which could use Multiple Master fonts, a `dvi` MIME type might be a success. All I’m talking about is a previewer, not the whole T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X installation. Using Adobe’s Multiple Master fonts appeals to me as a way of piggy backing off their technology, and also means that in font terms, it can do at least anything that Acrobat can. Of course, we have to change the name of `dvi`, since it is a registered trade mark (of Intel?) for digital video. . . One advantage is that all the archives of scientific papers created using T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X or `groff` are available no matter what platform you use. It would also be a very open standard, with no whiff of proprietary machinations. It’s so obvious that someone is bound to have done it already.

## **5 Deathwish 1996**

Am I alone in thinking that there are some odd decisions being made on the location of TUG conferences? The notion of going to Florida in the summer didn’t appeal to me (although I appreciate that there may have been a sub-text which somehow equated TUG and Disneyworld), but next year’s conference location outdoes this: Russia. At least I think that’s what they said. Maybe it will really be in Moscow (that’s a wee place in Ayrshire). With at least two TUG members in Russia, it promises to be an unforgettable meeting.

## **6 Another famous Belgian**

You will remember the parlour game where you were challenged to name ten famous Belgians (it isn’t really that difficult). There is now an eleventh! Michel Goossens of *Companion* fame will be President of TUG. This makes three non-US Presidents in a row, neatly confounding any claims that it is solely a US organisation. Someone unkindly remarked upon the similarity between Michel and Tin-tin, but apart from the dogs and the nationality I can’t see this. Perhaps we should be told.

## **7 Thank you**

This edition of *Baskerville* carries an announcement of the elevation of M Clark to Honorary Membership of the group. In a rare interview MC told me how deeply moved he was by this singular honour, all the more so, he said, since the committee at that time had been populated by a large number of people who he held in high regard. He went on to say that one of the things he enjoyed most about the T<sub>E</sub>X (and L<sup>A</sup>T<sub>E</sub>X) world was the comradeship and good fellowship which characterised it. Almost without exception, T<sub>E</sub>X people helped one another, and did so in a friendly and concerned way. If anything, they were sometimes too nice, he added with a twinkle in his eye (too many bottles of Cab Sov). He asked your correspondent to convey his heartfelt thanks to all those involved.

---

## X New PostScript font metrics for L<sup>A</sup>T<sub>E</sub>X

Sebastian Rahtz

---

Over the last 9 months, a group of people have been working on a revamp of T<sub>E</sub>X font metrics for PostScript fonts. We are now ready (finally!) to release this, together with a new release of the PSNFSS package for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, and a new version of the Karl Berry font-naming scheme. The maintainers of PSNFSS, *fontinst*, *web2c*, and *dvips* all urge you to start using this new setup, so that all the bothersome variants can be abolished by the end of the year.

The entire distribution is on the CTAN hosts in `fonts/psfonts.beta`. Its big — don't pull all the files on spec! Get what you need. Prepackaged sets for Textures users are also available.

These files will replace both the L<sup>A</sup>T<sub>E</sub>X PSNFSS (currently on CTAN in `fonts/metrics`) and the font metrics distributed with *dvips*. Both of these packages contained virtual fonts which more or less mimicked the original T<sub>E</sub>X text encoding, and (in the case of PSNFSS) the Cork encoding. However, the two packages used different base fonts, thus making previewing painful (and wasting disk space). Now, everything uses a single base font in a new encoding named 8r (see `tools/8r.enc` or `tools/8r.etx`).

This new base encoding is *not* Adobe Standard Encoding, because that does not provide access to all of the 228 characters normally supplied with a Type 1 font. Therefore, you must use a DVI-to-PostScript driver which can perform reencoding, such as *dvips(k)*, Y&Y's drivers, OzTeX, Textures etc.

We are still contemplating whether the base encoding should be the current one (mostly compatible with Windows), the *texnansi* encoding promulgated by Y&Y (see `tools/texnansi.vec`), or something else. Comments are welcome.

Aside from the base font, there are other small changes in the new fonts. For the Cork-encoded fonts, the stretch and shrink of the interword spacing has been changed; this will almost certainly change again for the final release, and we welcome comments. For the *dvips* fonts, the positions of the preaccented characters have been fixed (e.g., Aring is at the same position in every font). Therefore, the font checksums are different. The actual character dimensions, however, remain unchanged.

The new fonts (mostly) have new names! You can get the new fontname distribution from `ftp.cs.umb.edu` in `private/tex/fontname-*.tar.gz` or in the `tools/` subdirectory of this distribution.

This distribution includes support the standard 35 fonts, various freely available fonts (with Type1 sources), and many commercial-only fonts (no Type1s). Smallcaps and obliques versions are available in bold and normal variants, where applicable. All fonts have ligatures and kerning (no "raw" fonts); therefore, even the base fonts can be used for real typesetting. An experimental support is provided in PSNFSS for this.

We built these fonts using both *fontinst* and *afm2tfm* (and other utilities). Both required changes, which will be merged into the next releases. The changes to *fontinst* are extensive, and if you are not confident you understand them, please wait until the author finds time to merge them into a complete new release later this year.

*Primary perpetrators:* Sebastian Rahtz, Alan Jeffrey, Karl Berry.

*Chief Tester and Bugfinder:* Constantin Kahn.

*Aiders and abettors:* Tom Rokicki, Ciarán Ó Duibhín, Pierre MacKay, Rob Hutchings, Berthold Horn, Damian Cugley.

Please send any questions, comments, or suggestions to `tex-fonts@math.utah.edu`. (Email `tex-fonts-request@math.utah.edu` to join the list.)

---

## XI Standard for T<sub>E</sub>X Directory Structures

---

The TUG Working Group on a T<sub>E</sub>X Directory Structure announces that a draft of the proposed T<sub>E</sub>X Directory Structure standard is available for public review.

You can get it by FTP from:

<CTAN host> : /tex-archive/tds/draft-standard

Comments and suggestions are welcome. Please communicate them by email to [twg-tds@shsu.edu](mailto:twg-tds@shsu.edu) or by paper mail to

Norman Walsh  
O'Reilly & Associates, Inc.  
90 Sherman Street  
Cambridge, MA 02140  
USA

The primary purpose of this document is to describe a standard T<sub>E</sub>X Directory Structure (TDS) for macros, fonts, and other such implementation-independent T<sub>E</sub>X files. As a matter of practicality, it also suggests ways to incorporate the rest of the T<sub>E</sub>X files into a single structure. In the not-so-long run a consistent directory structure will make it much easier to install and maintain T<sub>E</sub>X. We hope that administrators and developers of both free and commercial implementations of T<sub>E</sub>X will adopt this standard. It has been designed to work on all modern systems. In particular, this Technical Working Group (TWG) believes it is usable under Unix, MS-DOS, OS/2, MacOS, and VMS.

We hope to publish another draft, or make the final release (depending on the volume of comments and concerns) shortly after TUG 95.

---

## XII The 1995 Annual General Meeting

---

The Annual General Meeting of the UKTUG will almost certainly be held on the morning of Wednesday 18 October 1995. The place is not yet known. Full details will be given in *Baskerville* 5.4. The preliminary announcement appears now to ensure that Clause 17 (below) is fulfilled even if *Baskerville* 5.4 is delivered later than scheduled.

The Constitution contains the following clauses:

17. The annual general meeting of the UKTUG shall be held in the United Kingdom not later than the last day of November in each year on a date and at a time to be fixed by the Committee and notified to members at least 35 days in advance for the following purposes—

(1) to receive from the Committee a report balance sheet and statement of accounts for the preceding financial year

(2) to fill the vacancies in the Committee and to appoint auditors for the ensuing year

(3) to decide on any motion which may be proposed to the meeting in the manner provided below

(4) to fix the entrance fee (if any) and annual subscription

18. Any member desirous of proposing any motion at the annual general meeting shall give notice in writing to the secretary not later than 14 days before the date of such meeting

23. No amendment (other than a motion for adjournment) shall be moved to any motion proposed at any annual or special general meeting unless written notice of the amendment shall have been sent to the secretary prior to the meeting

I think that Clauses 17 (3), 18 and 23 are self-explanatory.

A draft agenda, covering items (3) and (4) will appear in *Baskerville* 5.4. In the meantime, anyone wishing to propose any motion to the AGM must send me, in writing, the text of the proposed motion and their own name and address, to reach me by 4 October 1995. If there are any changes to the draft agenda, the final agenda will be sent to members in the 14 days before the AGM.

The chair's term of office ends at this AGM. The following clause from the constitution describes the procedure which should be followed.

11. (1) The chair of the group shall be the chair of the committee

(2) The chair of the group shall retire after completion of two consecutive years but shall be eligible for re-election

(3) Any two members who are individuals may nominate any member of the group who is an individual to be the chair in writing to the secretary not less than two months before the date of retirement

(4) If there are two or more nominations then the chair will be decided by a postal ballot by single transferable vote in which every member of The UK T<sub>E</sub>X Users' Group who is an individual shall be entitled but not obliged to vote

Nominations for the chair should be made in writing and sent to me, to reach me by 18 August 1995. There is no official nomination form. If you like, you may use the forms provided at the end of this notice. Otherwise, what you send me should contain, clearly and legibly,

—name and address of nominee

—declaration that (s)he is willing to stand for this office, signed by the nominee

—name and address of the nominator

—statement nominating the nominee, signed by the nominator

—name and address of seconder

—statement in support of the nomination, signed by the seconder.

The nominee, nominator and seconder should all be members of the UK T<sub>E</sub>X Users' Group, and the last two should be different from each other.

As for other vacancies on the Committee, Clause 10 of the Constitution implies that at most 3 members of the present Committee may continue without being re-elected. Thus Clause 9 implies that there are at least 8 vacancies to be filled.

The following parts of Clause 10 describe the procedure for filling these vacancies:



*The UK T<sub>E</sub>X Users Group*

(3) Any two members who are individuals may nominate any member or members of the group who are individuals to fill any of the vacancies by giving at least 7 days notice in writing to the secretary

(4) If there are more nominations than vacancies then the membership will be decided by an election by single transferable vote in which every member of the UKTUG who is an individual and is present at the annual general meeting shall be entitled but not obliged to vote

Nominations for committee members in accordance with the above sub-clauses are thus being sought. Written nominations for new committee members should be sent to me, to reach me by 11 October 1995. There is no official nomination form. If you like, you may use the forms provided at the end of this notice. Otherwise, what you send me should contain, clearly and legibly,

- name and address of nominee
- declaration that (s)he is willing to stand for this office, signed by the nominee
- name and address of the nominator
- statement nominating the nominee, signed by the nominator
- name and address of seconder
- statement in support of the nomination, signed by the seconder.

The nominee, nominator and seconder should all be members of the UKTUG and the last two should be different from each other.

Once elected, the Committee appoints officers from its membership.

In all the above, ‘in writing’ means ‘on paper’, not email.

Please put ‘UKTUG’ on the outside of the envelope when you send me any of the above. Otherwise, people may find themselves elected to the wrong committee.

My address is

School of Mathematical Sciences  
Queen Mary and Westfield College  
Mile End Road  
London E1 4NS

R. A. Bailey  
Honorary Secretary

---

## XIII The UK T<sub>E</sub>X Users Group

---

### 1 Group aims and activities

The UKTUG was founded in 1989; its activities and benefits include:

- Regular meetings covering both T<sub>E</sub>Xnical matters and general typographic subjects;
- Visits to publishers, font suppliers and book producers;
- In-depth workshops, with hands-on computer sessions where appropriate;
- Negotiation of discounts on T<sub>E</sub>X relevant books with publishers.
- Distribution of public domain T<sub>E</sub>X implementations (in cooperation with the international T<sub>E</sub>X archives);
- Discount on joint UK T<sub>E</sub>X Users Group/TUG membership;
- Reduction in conference fees for the TUG annual meeting and the annual European meetings;
- The publication and distribution of *Baskerville* (the annals of the UK T<sub>E</sub>X Users Group).

### 2 Programme of meetings for 1995

**July, 1995** (T<sub>E</sub>X Users Group Annual Meeting in Florida)

**September, 1995:** emT<sub>E</sub>X in depth: a training meeting on MSDOS T<sub>E</sub>X, with Sebastian Rahtz Location: Oxford.

**September, 1995** (EuroT<sub>E</sub>X meeting in the Netherlands)

**October 1995:** AGM, at Queen Mary and Westfield College, University of London, with sessions on ‘non-traditional’ T<sub>E</sub>X.

All members of UKTUG will be provided with more details of these meetings as they become available. Reports of the meetings will appear in *Baskerville*, the organ of the group.

### The 1994–95 UKTUG committee

C. A. Rowley    Chair  
P. Abbott      Treasurer and  
                    Membership Secretary  
R. A. Bailey    Committee Secretary  
M. Clark       Meetings Secretary  
S. P. Q. Rahtz   *Baskerville* editor  
D. P. Carlisle; R. Fairbairns; C. Hewlett; A. S. A. Jeffrey.

### Book Discounts for UKTUG members

The UK T<sub>E</sub>X User group has been in discussion with a number of publishers to arrange discounts on publications which may be of interest to members. We have arrangements with Addison-Wesley for their well-known T<sub>E</sub>X-related publications, and with International Thomson Publishing to supply any of the very excellent O’Reilly & Associates Inc. series of books to members.

The agreed list of books, together with the discounted (at least 10%) price, is distributed occasionally with *Baskerville*, but is always available from the Treasurer, Peter Abbott. The quoted price includes the cost of postage and packing.

We would like to extend this service to members. If you have contacts with the sales and marketing parts of any publisher who publishes relevant books, please inform us and we will try to arrange something mutually convenient and beneficial.

We are only allowed to offer this service to **current** members of the UK T<sub>E</sub>X User Group and/or members of TUG. Please send your order and cheque (in UK £) to Peter Abbott (address in *Baskerville* masthead). Make cheques payable to ‘UKTUG’ please. Books from Addison-Wesley are delivered direct but books from O’Reilly will be routed through UKTUG. *In all cases* please notify Peter Abbott by email, phone, fax or letter when books are delivered.

---

## XIV Obtaining T<sub>E</sub>X

edited by Sebastian Rahtz

---

### *From the network*

The UK T<sub>E</sub>X Archive on `ftp.tex.ac.uk` is part of the CTAN (Comprehensive T<sub>E</sub>X Archive Network) collaborating network of archives on the Internet organised by the T<sub>E</sub>X Users Group. The three main archives follow the same structure and have identical files (`ftp.tex.ac.uk`, `ftp.shsu.edu` and `ftp.dante.de`).

The CTAN archives all run an enhanced *ftp* server which supports dynamic compression, uncompression, and archive creation options. Fetch the top-level file `README.archive-features` for information. The server also supports site-defined commands to assist you. Please read `README.site-commands` for a brief overview.

Details of where to find public domain, or shareware, T<sub>E</sub>X packages for different machines and operating systems are given below.

Please report any problems with CTAN archives via email to `ctan@shsu.edu`. The entire archive is available on CDROM as the ‘T<sub>E</sub>Xcetera’ package from Prime Time Freeware, 370 Altair Way, Suite 150, Sunnyvale, CA 94086 USA (`ptf@cfc1.com`). This is also distributed by the TUG office, and in the UK by Lasermoon Ltd (email `info@lasermoon.co.uk`, phone 01329 826444).

### *Unix tapes*

For a complete Unix T<sub>E</sub>X distribution, a 1/4 inch cartridge, QIC-120 or QIC-150 format (DC600A or DC6150) can sent with envelope *and* stamps for return postage to:

David Osborne  
Cripps Computing Centre,  
University of Nottingham,  
Nottingham NG7 2RD

Due to currency exchange, this service is offered only within the UK.

### *PC and Mac disks*

The UKTUG distributes an emT<sub>E</sub>X kit for PCs, and an OzT<sub>E</sub>X kit for Macintosh. The cost covers copying and postage costs, and the shareware fee for OzT<sub>E</sub>X. Each set costs £30, and is available from Peter Abbott, 1 Eymore Close, Selly Oak, Birmingham B29 4LB. Cheques must be payable to ‘UKTUG’. Please note that this service *is available to UKTUG members only*. Each set comes with an installation guide, and (at least) full T<sub>E</sub>X and METAFONT, a previewer, a PostScript driver, and CM fonts. Two update disks a year will be sent out automatically, with the current version of L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, and other goodies. A subscription service will be available for subsequent years. In addition, subscribers can request up to 3 disks a year with any material from the CTAN archives, but this will be supplied ‘as is’, without instructions.

Enquiries for T<sub>E</sub>X for the Atari ST etc. can be directed to: The Fast Club, 7 Musters Road, Nottingham NG2 7PP. Phone 01602 455250, fax 01602 455305. They also supply a variety of T<sub>E</sub>X-related software in Atari format.

*reprinted from Baskerville*

*Volume 5, Number 3*

| Name                   | Environment | CTAN path                              | Notes  |
|------------------------|-------------|--|--|
| emT <sub>E</sub> X     | DOS, OS/2   | systems/msdos/emtex                    | the betatest drivers, and 386-specific versions of T <sub>E</sub> X and MF, are in the betatest subdirectory               |
| sbT <sub>E</sub> X     | DOS         | systems/msdos/sbtex                    | includes latest T <sub>E</sub> X and METAFONT  |
| gT <sub>E</sub> X      | DOS         | systems/msdos/gtex                     | 386-only T <sub>E</sub> X and METAFONT, which replace parts of emT <sub>E</sub> X, and work with Windows memory management |
| OzT <sub>E</sub> X     | Mac         | systems/mac/oztex                      | This package is shareware  |
| CMacT <sub>E</sub> X   | Mac         | systems/mac/cmactex                    | Shareware port of Unix T <sub>E</sub> X  |
| DirectT <sub>E</sub> X | Mac         | systems/mac/directtex                  | This package is shareware  |
| pasT <sub>E</sub> X    | Amiga       | systems/amiga/pastex                   |  |
| —                      | Atari       | systems/atari/lindner-tex-packed-disks |  |
| —                      | Atari       | systems/atari/cs-tex                   |  |
| web2c T <sub>E</sub> X | Unix        | systems/unix/web2c                     | a complete source kit for T <sub>E</sub> X and METAFONT, which should compile on most Unix boxes; needs a C compiler.      |
| —                      | VMS         | systems/vms                            | Alpha and VAX versions available   |

---

## XV Membership Data

Peter Abbott  
Honorary Treasurer

---

As at 20th March 1995 the membership stood at

|                               |           |
|-------------------------------|-----------|
| UKTUG                         | 62        |
| UKTUG(Student)                | 2         |
| Full TUG and UKTUG            | 98        |
| Basic TUG and UKTUG           | 6         |
| Full TUG and UKTUG (Student)  | 4         |
| Basic TUG and UKTUG (Student) | 1         |
| Institutional                 | 4         |
| Full TUG                      | 2         |
| Basic TUG                     | 0         |
| TUG and UKTUG (Student)       |           |
| <hr/> Total                   | <hr/> 182 |

### 1 Membership Services

Agreement has been reached to but site licences for the software listed below. A member of the UK  $\TeX$  User's Group is permitted to use the software (which is usually shareware) without additional payment as long as they are a member of the group. It is provided for personal use only.

- OzTeX version 1.8 – Either obtain a copy from the CTAN archive, or available from me.
- Dvips and METAFONT Mac – Licence code details not yet known, contact me.
- Alpha – a text editor for OzTeX obtainable from CTAN or me.
- Eddi4TeX – a  $\TeX$  shell and text editor for MS DOS obtainable from me (The access code will be mailed direct from Germany).

---

**Election for chair**

Name of Nominee: \_\_\_\_\_

Address: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

I declare that I am a member of UKTUG and am willing to stand for election for the chair of UKTUG for 1995–97.

Signature: \_\_\_\_\_ Date: \_\_\_\_\_

Name of Nominator: \_\_\_\_\_

Address: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

I declare that I am a member of UKTUG and that I nominate the above-named person for the chair of UKTUG for 1995–97.

Signature: \_\_\_\_\_ Date: \_\_\_\_\_

Name of Seconder: \_\_\_\_\_

Address: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

I declare that I am a member of UKTUG and that I second the nomination of the above-named person for the chair of UKTUG for 1995–97.

Signature: \_\_\_\_\_ Date: \_\_\_\_\_

---

**Election for committee member**

Name of Nominee: \_\_\_\_\_

Address: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_