and gives not just a description of the TEXniques which might be used to solve them, but instances of the actual code; in that sense it is, therefore, a recipe book, and accordingly it possesses both the strengths and the weaknesses which typify books of that *genre*. If the problem requiring a solution is covered in the book (i.e. if its recipe is given), then the problem is completely solved; but if the problem is *not* covered (i.e. if no such recipe can be found), then the reader is left little wiser. Fortunately, by virtue of the very comprehensive Glossary/Index with which this book concludes, this deficiency of the recipe-book approach is mitigated, for even if the exact problem is nowhere discussed within the Examples, there is an excellent chance that the elements of the problem are at least alluded to in the Glossary/Index. Yet this approach, too, has its limitations: for example, the text on pages 16/17 reads '*may be written with the aid of a bordered Hessian, as follows:*', but if one turns to the Glossary/Index, knowing full well that one needs to typeset a '*bordered Hessian*', no such entry exists. One is led round the houses, via `\bordermatrix` (which yields a dead-end, at least in the context of this enquiry), via `matrices` (Plain TEX), to `$\matrix$` (Plain) which finally yields the relevant page number (amongst many others). The Glossary/Index could therefore be improved (in the opinion of this reviewer) by containing key phrases which identify specific instances amongst the examples. In terms of accuracy and precision, the text is less than perfect, but is not so severely flawed as to render it useless; I recommend to Arvind that he employ a TEXnically literate proof-reader for any future volume, for whilst his approach is excellent, the inconsistencies and occasional real errors which populate this book do tend to detract from its value. Finally, in terms of the competition, *Mathematical TEX by Example* has one great advantage: it (and its stablemate, *TEX by Example*) are unique in their approach (at least, as far as I am aware; I have encountered no other TEX books which seek to educate solely through example); for those, then, who prefer to learn by osmosis rather than through orismology, this book is to be recommended, although perfectionists might do well to wait for a second edition, in which one hopes the errors and inconsistencies will have been eliminated.

◇ Philip Taylor
The Computer Centre
RHBNC
University of London
Egham, Surrey TW20 0EX, U.K.
P.Taylor@Vax.Rhbnc.Ac.Uk

Macros

The Bag of Tricks

Victor Eijkhout

A 144 point Hello! to you all.

On `comp.text.tex` I see with some regularity questions about actions on a whole page. For instance, how to put a box around a page, or how to overlay a piece of text on each page. Here is a solution based on redefinition of `\shipout`. Doing things this way has the advantage that it does not depend on the format used.

If you have more than one application like this, you could for instance put the following macros in a file `wholepage.tex`:

```
\let\xshipout\shipout
\def\shipout{\futurelet\SomeBox\yshipout}
\def\yshipout
 {\ifx\SomeBox\box
    \let\next\ShipZero
  \else \ifx\SomeBox\copy
          \let\next\ShipZero
        \else
          \let\next\ShipAfterZero
  \fi\fi
  \afterassignment\next\setbox0=
  }
\def\ShipAfterZero{\aftergroup\ShipZero}
```

This saves the old `\shipout`, and defines a new one that first investigates what kind of box is coming up. This box is then put in `\box0`, and a call to `\ShipZero` processes this and really ships it out by a call `\xshipout\box0`.

For every specific application a definition of `\ShipZero` has to be made. Here is a way to do overlays:

```
\newbox\OverlayBox
\def\ShipZero
 {\setbox0\vtop{\kern0pt \box0}
  \setbox0\vtop{\kern0pt
    \vtop to 0pt{\kern0pt
      \copy\OverlayBox\vss}
    \nointerlineskip \box0}
  \xshipout\box0 }
```

If you put these macros into a file `overlay.tex` followed by a line `\input wholepage`, then you can make an overlay by, for instance

```
\input overlay
\setbox\OverlayBox
  \hbox to \hsize{\hfil\tt TEST}
```

```
Text ...
```

(If you use LaTeX, you create a file `overlay.sty` that can be specified on the `\documentstyle` line.

Cropmarks are a bit harder. I arrived at the following after a lot of tinkering. There are dimensions for the length and width of the crop lines, and for how far they are separated from the page.

```
\newdimen\croplength \croplength=20pt
\newdimen\cropsep \cropsep=10pt
\newdimen\cropwidth \cropwidth=2pt
```

Sometimes you may want to add some extra padding on the top, bottom, or sides.

```
\newdimen\croppadtop \croppadtop=0pt
\newdimen\croppadbot \croppadbot=0pt
\newdimen\croppadlr  \croppadlr=0pt
```

Here are the crop lines.

```
\def\crophrule{\vrule
  height\cropwidth depth0pt
  width\croplength}
\def\cropvrule{\vrule
  width\cropwidth depth0pt
  height\croplength}
```

The following macro does the real work: it takes `\box0` and puts the crop lines around it. You see that there are a lot of kerns of size `.5\cropwidth`: these make sure that cropping through the middle of the lines will give you exactly the page.

```
\def\ShipZero
 {\setbox0\vbox
   {\offinterlineskip
    \dimen0\cropsep
    \advance\dimen0\croplength
    \setbox2\hbox to \wd0
       {\kern-\croppadlr\kern-.5\cropwidth
        \cropvrule\hfil\cropvrule
        \kern-.5\cropwidth\kern-\croppadlr}
    \setbox4\hbox to \wd0
       {\llap{\crophrule
              \kern\cropsep\kern\croppadlr}%
        \hfil
        \rlap{\kern\croppadlr\kern\cropsep
              \crophrule}}
    \kern-\dimen0 \copy2
    \kern\cropsep\kern-.5\cropwidth
    \copy4
    \kern-.5\cropwidth
    \kern\croppadtop
    \box0
    \kern\croppadbot
    \kern-.5\cropwidth
    \box4
    \kern-.5\cropwidth\kern\cropsep
    \box2
  }\xshipout\box0 }
```

Finally, here's how to put a box around the whole page:

```
\newdimen\padding \padding=3pt
\def\ShipZero
 {\setbox0\hbox
   {\vrule\kern\padding
    \vbox{\hrule\kern\padding
          \box0
          \kern\padding\hrule}%
         \kern\padding\vrule}
   \xshipout\box0 }
```

And that's all for now.

◇ Victor Eijkhout
Department of Computer Science
University of Tennessee at
Knoxville
Knoxville TN 37996-1301
Internet: `eijkhout@cs.utk.edu`

[Editor's note: The edges of the page are assumed to be at the edges of the printed area, excluding any material that is "lapped" outward. (In plain TeX, this "lapped" area includes the `\headline`, if any.) Readers who are preparing camera-ready copy will want to add sufficient padding to accommodate necessary margins. For two-sided presentation, they may also find that the gap between edge of page and crop mark should be different on the left and right. These macros can be adapted for that situation by replacing `\croppadlr` by `\croppadin` and `\croppadout` to indicate inside and outside margins, and specifying these appropriately on odd or even pages.]