Q & A

JUST PLAIN Q&A: Of Partitioned Matrices and Doublespacing.
Alan Hoenig

This column serves as a forum in which people can seek answers to TeX questions, with an emphasis on plain TeX. Questions at all levels of difficulty are welcome. We hope to hear from you.

How Do You Typeset Partitioned Matrices?
David Handelman sent a request for aid on the typesetting of partitioned matrices. His question lay on my desk for a shamefully long time, and my apologies to him. I hope he will agree that this response is worth the wait, as a superior set of tools has appeared to make his life much easier.

David seeks to typeset partitioned matrices, which are things of this ilk:

\[
\begin{pmatrix}
A_1 & Z \\
A_2 & A_3 \\
A_4
\end{pmatrix}
\]

that is, rectangular arrays of expressions (matrices) pierced through by solid or dashed vertical and horizontal line segments.

Since David posed his query, Mike Spivak has created the \texttt{IAMS-TEX} macro package which handles this type of thing. At the July 1991 TUG meeting in Dedham, Mike announced the placing of this extensive package in the public domain. It can be downloaded from any of several archives.

Here's how you can typeset matrix (A). First, load in the appropriate macro and style files at the beginning of the document.

\input /lamstex/amstexl
\input /lamstex/lamstex
\input /lamstex/ptmatrix

File \texttt{amstexl} is subset of normal \texttt{IAMS-TEX}, and \texttt{lamstex} is the core set of \texttt{IAMS-TEX} macros. The partitioned matrix macros are in \texttt{ptmatrix}. In math display mode, enter

\begin{verbatim}
\left[
\begin{pmatrix}
A_1 & Z \\
A_2 & A_3 \\
A_4
\end{pmatrix}
\end{verbatim}

where the syntax resembles that of \texttt{AMS-TEX} with a few obvious additions. The \texttt{\vsolid} command specifies a vertical solid rule after column one extending from the top of row 0 (bottom of the matrix) to the top of row 2. \texttt{\hsolid} directs the creation of a solid horizontal line after row 1 which extends from the zeroth column (left of the matrix) to the second column.

More complicated examples are possible, such as

\begin{verbatim}
\left[
\begin{array}
\lambda_1 \\
\vdots \\
\lambda_r
\end{array}
\end{verbatim}

which comes from

\begin{verbatim}
\left[
\begin{array}
\lambda_1 \\
\vdots \\
\lambda_r
\end{array}
\end{verbatim}

Many additional options exist for twiddling with the position and appearance of the rules.

The \texttt{IAMS-TEX} macro package contains many powerful and useful features. It's an extension of \texttt{AMS-TEX} with the functionality of \texttt{IPEX} (but more concise syntax). Automatic numbering schemes are very flexible, and can easily be modified for special circumstances. Extensive table-making abilities are part of the package. It's possible too to create complicated and professional commutative diagrams along with partitioned matrices. An index program comes with IAMS-TEX, and IAMS-TEX...
now interfaces with \texttt{BibTeX}. No commands are fragile. And much, much more. You may contact the author via e-mail at \texttt{spivak@math.rice.edu}.

Manuals for the package may be purchased from the \texttt{TeXplorators Corporation}, 3701 W. Alabama, Suite 450-273, Houston, TX 77027.

\textbf{Controlling Interline Spacing in \LaTeX{} and \TeX{}}

I'd like to present a limited discussion of double-spacing on behalf of the many people over the years who have wanted to doublespace their \LaTeX{} and \TeX{} documents. At first blush, you might wonder why such an anachronism is needed in this day and age, but copy editors still demand a doublespaced manuscript to ensure enough room for their red pencils. If this is why you need doublespacing, then it's reasonably easy to jury-rig important doublespacing details. If you need doublespacing because of the archaic needs of a thesis style, say, then you can embed all the proper doublespace formatting in your style file. (Or perhaps the style file you use already takes it into account.)

It's not enough simply to reset \texttt{\baselineskip}, because there are plenty of situations where other parameters such as \texttt{\lineskip} control the interline spacing. A better idea is to reset all the relevant parameters by means of the \texttt{\openup} command:

\begin{verbatim}
\openup1pc
\end{verbatim}

or

\begin{verbatim}
\openup \baselineskip
\end{verbatim}

for example.

Even this is not enough, though. There may well be groups within which \texttt{\offinterlineskip} has been set and spacing is controlled by \texttt{\strut}'s. (This is often the case within specialized table macros.) Here is one way to extend a strut after you've \texttt{\openup}'ed the baseline.

\begin{verbatim}
\newbox\newstrutbox
\setbox\newstrutbox=\vbox{\vrule height \baselineskip depth .3\baselineskip width 0pt}
\setbox\strutbox=\box\newstrutbox
\end{verbatim}

It's easy to combine all these details into a single macro.

You may also want to increase the space above and below a display with commands like

\begin{verbatim}
\advance \abovedisplayskip by 6pt
\end{verbatim}

or whatever, and similarly for \texttt{\abovedisplayskippshort}, \texttt{\belowdisplayskip}, and \texttt{\belowdisplayskippshort}.

But of course this will not give exact doublespacing in all circumstances. If you \texttt{\openup1pc}, then a footnote whose original baseline is 8 pt will have a new baseline of 20 pt (remember, 20 = 8 + 12) rather than the 16 pt that you might prefer. If you are running off a quick draft for a copy editor, you probably don't care. If this is your thesis, well, you'll have to work a bit harder with your \texttt{\footnote} macro.

For \LaTeX{} users, the same considerations apply, but you must implement them in proper \LaTeX{} syntax. The details are in two files that are easily \texttt{ftp}able. The first, put together by S. Page and subsequently modified by J.-F. Lamy, is \texttt{doublespace.sty} at \texttt{sun.soe.clarkson.edu} in the directory pub/tex/latex-style. The same file, with additional modifications by S. Rahtz to render it useful when using the new font selection scheme of Mittelbach and Schönig, resides in the \texttt{ymir} archive in [\texttt{anonymous.tex.inputs.latex-contrib}]. Check the TUG resource directory for assistance in retrieving these (or any) files from the archives.

\textbf{Department of Amplification}

Bernd Raichle, Dante coordinator for \texttt{german.sty}, has pointed out that the macros I presented last time for playing around with \texttt{\fontdimen} parameters could be easily defeated by users with a sense of the macabre. (But my apologies nevertheless for not being more rigorous in my own testing.) He has taken the trouble to fortify those macros, for which I thank him, and I present them here with his comments, together with a short test.

\begin{verbatim}
\font\roman=cmr10
\font\specroman=cmr10
\rm
\newdimen\savedvalue\savedvalue=\fontdimen2\roman
\newdimen\specialvalue\specialvalue=19.99pt
\newif\ifspec\ifspec
\% changed global, indicates need to reset
\endverbatim
Elementary Text Processing
and Parsing in \TeX
— the appreciation of tokens —

L. Siebenmann

Background

Token lists make up the material found in the upper digestive tract of \TeX, and token list registers are very useful means to improve \TeX’s digestion. I begin this tutorial by showing how to do elementary ‘text processing’ with token lists. Then I apply this ‘token list processing’ to parsing of classical keyword syntax where the keys come in any order and their fields (or arguments) are terminated by nothing more than the next keyword. This processing and parsing are simple concepts that many \TeXperts, not to mention beginners, have largely neglected. I find that \TeX assimilates them well, and hope they will see wider use in the future.

I originally explored this parsing as a possible method to fix a subtle line-breaking bug in \AMS-\TeX bibliographies that was pointed out by Barbara Beeton in 1990. This remains a convenient example to test methods; but in truth an academic one, since Michael Downes [Do] has successfully fixed the bug (for version 2.1 of July 1991) using a very different \vbox trick proposed by Don Knuth. The general subject of parsing in \TeX language, to which this tutorial contributes two methods called (A) and (B) below, was introduced by W. Appelt in his book [App].

I want to thank Michael Downes, Victor Eijkhout, and Ron Whitney for contributing many helpful comments as this tutorial evolved. My ignorance and uncertainty about what all can or cannot be found in \book was a problem that delayed this tutorial; one remedy I enjoyed using is surely of interest to readers of \TUGboat, namely string searches in an online version of \book. \footnote{The .tex file for \book can for example be obtained by anonymous ftp from the archives}

Alan Hoenig
17 Bay Avenue
Huntington, NY 11743
(516) 385–0736
ahjji@cunyvm

1) The .tex file for \book can for example be obtained by anonymous ftp from the archives

labrea.stanford.edu
rusinfo.rus.uni-stuttgart.de

It fits on a diskette and can conveniently be used on a microcomputer.