## TEXNortheast: Workshops and additional papers

Summaries follow for workshops as well as for papers which have been published elsewhere, or for which no final text was received by the TEXNortheast Program Committee.

### Workshops

#### Moving On: LATEX 2.09 to LATEX 2ε

Anita Z. Hoover

**Prerequisites:** Little or no experience in LATEX 2ε and most familiar with LATEX 2.09 conventions.

**Description:** Learn the basics to convert a document from LATEX 2.09 to LATEX 2ε. The focus was on

1. Discuss the differences between LATEX 2.09 and LATEX 2ε;
2. New features in LATEX 2ε;
3. Standard classess, packages, and options; and
4. Custom packages.

◇ Anita Z. Hoover
University of Delaware
anita@udel.edu

#### More Multiline Equation Environments

Stephanie Hogue

**Prerequisites:** Basic knowledge of standard LATEX math environments, including `eqnarray` and `array`.

**Description:** This workshop was an introduction to the multiline equation environments of the amsmath package for LATEX 2ε, which supersedes the amstex package. The following environments were discussed:

- `gather`, `multline`: environments without alignment across lines;
- `split`, `align`, `flalign`, `alignat`: environments with one or more alignments across lines.

The discussion included guidelines for breaking equations, according to the AMS. Enhancements to equation numbering were also addressed.

This was *not* an exhaustive presentation of the amsmath package. Complementary material on font issues in amsmath was presented in Anita Hoover's workshop "Moving On: LATEX2.09 to LATEX 2ε".

◇ Stephanie Hogue
The TypeWright
shogue@typewright.com

#### Customizing LATEX Lists

Donald W. DeLand

**Prerequisites:** Intermediate LATEX 2ε for Authors workshop, or solid understanding of LATEX fundamentals.

**Description:** The `\list` mechanism is the basic building block of most non-sectioning LATEX environments. This workshop reviewed the generic LATEX environments that use `\list`, and how they are constructed. The following more advanced topics were covered in detail:

1. Changing default indents, labels, and vertical spacing using `\list` parameters and localized definitions of `\makelabel`.
2. Adding an optional argument to `\begin{enumerate}` to "clear for widest label" by using `\@ifnextchar` and linking the `\leftmargin` to the `\labelwidth`.
3. Using `\newcounter` and `\refstepcounter` to write theorem-like environments *without* using `\newtheorem`.
4. Tricks of the trade and aside comments:
   (a) Adding design elements using `\item`
   (b) Marking "optional" list items (e.g., in exercises or sections)
   (c) Boxing a theorem or definition
   (d) Enumerating horizontally rather than vertically
   (e) Why `\hangindent` and `\hangafter` don't work within a `\list`

◇ Donald W. DeLand
Integre Technical Publishing Co.
deland@cs.unm.edu

#### Beyond Tabular

Stephanie Hogue

**Prerequisites:** Basic knowledge of standard LATEX `tabular` environment.

**Description:** This workshop was an overview of several packages which provide enhanced features for tabular material. The following packages were presented:

- array: provides some new preamble options in addition to those found in the `tabular` environment;
- tabularx: automatically calculates *column* widths for a table of specified width;
- longtable: automatically breaks a long table across pages;

- dcolumn: provides a new column type for specifying a decimal-aligned column.

◇ Stephanie Hogue
The TypeWright
shogue@typewright.com

### "WYSIWYG" LaTeX: EXP and Scientific Word/Workplace/Notebook

Donald W. DeLand

Two WYSIWIG applications — Simon Smith's EXP and TCI's Scientific Word/Workplace/Notebook — allow authors to create LaTeX documents without learning LaTeX. This talk reviewed and demonstrated the major features of both programs, and explored some of their limitations with respect to document design, user interface, and LaTeX compatibility.

EXP is a "scientific word processor" whose word-processing features are easy to learn, but EXP documents need to be set up in a particular way to guarantee a smooth transition to LaTeX. Although EXP is easy to use, its automatic numbering mechanisms, lack of macro support, and inability to handle large tables or import non-EXP documents make it cumbersome to work with.

One major strength of TCI's Scientific Workplace is its built-in support for Maple, a popular computer-algebra system. Scientific Workplace also includes a style editor that lets the user customize numerous design elements, then process the document via LaTeX for outputting. There is a great deal of confusion, however, as to what the relationship is between Scientific Workplace and LaTeX. Workplace uses LaTeX as its output (print) engine, but it does not generate a "clean" LaTeX document.

◇ Donald W. DeLand
Integre Technical Publishing Co.
deland@cs.unm.edu

### Making Web Sites using LaTeX2HTML

Ross Moore

LaTeX2HTML is an extremely flexible tool for creating Web pages. Indeed it is best used when requiring technical information to be presented as a 'Web' of linked HTML pages.

One immediately encounters questions like:

- How many HTML pages?
- How much information should go on each page?
- How to link pages for easy access to related pieces of information?
- Indexing, Table-of-Contents and other Navigation aids.

The aim of this workshop was to get some familiarity with the way LaTeX2HTML tackles these issues, using configuration variables and command-line switches.

The LaTeX source provides the information presented, but there are many options available to affect the appearance and arrangement of the resulting Web pages.

◇ Ross Moore
ross@mpce.mq.edu.au

---

### Presentations not included in this issue

### Virtual Fonts

Alan Hoenig

TeX makes special demands on the fonts that it works with. Although this presents no problem for fonts (like Computer Modern) that were created explicitly for use by TeX, what do we TeX users do if we want to use any of the hundreds of beautiful fonts provided by mainstream digital foundries? The concept of virtual fonts provides this mechanism — for this and much more, as this talk will demonstrate. Discussion will center about available virtual font tools and some simple virtual font projects.

Editor's note: This article appeared in *TUGboat* **18** (2) pp. 113–121.

◇ Alan Hoenig
ajhjj@cunyvm.cuny.edu

### Breaking Equations

Michael Downes

Some flaws in the way TeX and LaTeX handle displayed equations are of such long standing that they are scarcely noticed any more except by beginning users — for example, the fact that `\left ... \right` constructs cannot span multiple lines, if an equation must be broken into more than one line. Other flaws that have to do with relatively subtle typographical issues go unnoticed by most users — for example, the fact that in multi-line equations `\abovedisplayshortskip` isn't applied when applicable, and intra-line shrink isn't used when available.

This is a report on a new LaTeX package called "breqn" that substantially eliminates many such problems. One of its main goals is to support automatic linebreaking of displayed equations, to the extent possible within the current limitations of TeX and LaTeX.

Editor's note: This paper appeared in *TUGboat* **18** (3), pp. 182–194.

◇ Michael Downes
mjd@math.ams.org

### Designing Books with TEX in Mind

Donald W. DeLand

This paper presents an overview of TEX's structure and how that structure impacts the implementation of book designs. Most book designs cannot be implemented using only TEX's internal components; rather, design implementation usually involves a combination of TEX and PostScript, and further depends on the specific font encodings and PostScript drivers used by the operating system and TEX implementation being used. The programmability of TEX combined with the flexibility of PostScript can be powerful. On the other hand, TEX predates PostScript, so the two do not always merge gracefully.

Specific design issues covered here include selecting fonts for use with math, using graphics as design elements, limitations in setting multicolumn text, and a discussion of how TEX's paragraph-building and page-breaking mechanisms impact marginal text and color usage. In addition, this paper presents some examples of how TEX's programmability can be used to automate or simplify design elements that could only be handled manually in other typesetting or desktop systems.

◇ Donald W. DeLand
Integre Technical Publishing Co.
deland@cs.unm.edu

### Custom Legal Documents for the Auto Loan Exchange

Douglas Lovell

The Auto Loan Exchange is a project of IBM Research which connects automobile dealerships directly to lenders and credit bureau reporting services for rapid approval and funding of automobile loans. We have used TEX to typeset the loan contract and related documents required to complete the loan and close the automobile purchase.

In many ways, TEX was the perfect choice to satisfy our document needs. We have been able to eliminate the preprinted forms stocked by dealers and instead, print complete contract documents customized for each loan. We will discuss the unique document requirements of this internet commerce application and describe our TEX-based solution.

Editor's note: This article appeared in *TUGboat* **18** (3), pp. 175–181.

◇ Douglas Lovell
dcl@us.ibm.com

### Hops, Skips, and Jumps: White Space

Joe Weening

An important part of the appearance of a document is the proper use of white space. Obeying well-established traditions of typography helps the reader to understand the document better. Failing to follow these rules may cause confusion and draw the reader's attention away from the content of the document.

TEX tries to insert the proper amount of white space wherever it can, but it sometimes gets it wrong. It is then up to the author of the document, or someone else editing the TEX file, to find and correct these errors.

In this talk we will explain TEX's rules for inserting white space, describe cases in which they don't work correctly, and explain how to get TEX to insert the right amount of space. We will include examples from TEX's horizontal mode, vertical mode, and math mode.

◇ Joe Weening
jweening@ccrwest.org

### Introducing METAPOST

John Hobby

METAPOST is a picture-drawing language very much like MetaFont except with PostScript output. I will give a brief overview of the METAPOST language and discuss drawing and filling, dashed lines, using TEX and LATEX output, and the graph-drawing package.

◇ John Hobby
hobby@research.bell-labs.com