
TUG 2020 abstracts

Editor's note: Videos are or will be available for most presentations; links and other information at tug.org/tug2020.

— * —

TeX and L^AT_EX: The user experience

Jonathan Fine

Where are we? How did we get here? What's the future? I'll try to answer these questions, by looking outward in both space and time.

Don Knuth started TeX in 1977. The present version is a direct descendant of the 1982 version. Only at the end of the 1980s did hard disc drives cost less than \$10 a megabyte. Today, for about \$100 I can buy a pocket computer that connects to a ubiquitous network. It has gigabytes of solid state storage. It fits in my pocket, more easily than a book.

Today software that implements Don Knuth's wonderful mathematical typesetting algorithm and fonts can be downloaded for free. And this software installs itself, and is interactive. In real time it previews L^AT_EX-encoded mathematics, as I type it.

My monitor is 40 inch, 3840x2160 with 24 bit color. It's no more expensive than a TV. (In fact, it is a TV.) It cost about \$450. It's not my virtual desktop. It's my vertical desktop, about the same size as my horizontal desktop.

As often as not, when I read beautifully typeset mathematics, it's on my vertical desktop, as part of a web page. The only time I actually need a PDF is to send a file to be printed. So that I can put it on my horizontal desktop, and write on it with a pen.

And the interactive mathematical typesetting software. It's built on HTML5, and it's called MathJax. And the future of TeX and L^AT_EX and our community. To succeed, we have to change, and also keep things the same. Linus Torvalds did something much the same with Unix, to create Linux.

By the way, my wonderful pocket computer that typesets mathematics. It's also known as a mobile phone, and it runs the Linux kernel.

Learning L^AT_EX (and other languages) online

Jonathan Fine

Online education is suddenly more important, from primary schools to research. This talk focuses on beginners learning L^AT_EX. We learn from what's already been done for other computer languages.

First I survey how to provide online L^AT_EX typesetting via a web browser. They are: L^AT_EX as a cloud service; L^AT_EX running in the browser; MathJax running in the browser.

Next, what to teach the student, and how. Teaching is not the same as writing a reference manual. It requires identifying core concepts, and presenting them in a helpful manner and order. Students need to test their understanding, and perhaps explore, before moving on. We consider L^AT_EX from this point of view.

At the school level, the Raspberry Pi Foundation (RPF) (raspberrypi.org) is one of the leaders. They offer many projects in HTML, CSS, JavaScript and Python. They have strong connections with students and teachers. They partner with Code Club (CodeClub.org) and Future Learn (FutureLearn.com). They provide many free resources. They run education research seminars. And they design and produce the Raspberry Pi.

Finally, we blend the experience of the RPF with the task of learning and teaching L^AT_EX, and conclude with some problems, opportunities and challenges. More information: jfine2358.github.io.

Teaching with L^AT_EX and Overleaf

Paul Gessler

When universities and other schools closed campuses to help reduce the spread of coronavirus, many professors and teachers quickly adapted to online teaching by necessity. Likewise, students adapted to online learning and found ways to collaborate with peers while following social distancing guidelines. Overleaf, an online writing platform for TeX, has proven helpful in many of these scenarios. This talk will provide an overview of how Overleaf can be used most effectively in an education context. Topics include: how to effectively organize projects; suggested workflows for sharing assignment templates and receiving completed assignments; using Overleaf's reviewing tools to collaborate and provide feedback on assignments.

The creative evolution of type specimens

Amelia Hugill-Fontanel

Type specimens have been produced from the earliest days of Western printing history as commercial documents designed to sell printing type. Over the course of five centuries, this publication genre has kept pace with the needs of its consumers through innovative change. This time-traveling tour will navigate through the vast holdings of the RIT Cary Graphic Arts Collection to explore the formats and features that unfurl through the history of type specimens.

Bio: Amelia Fontanel is a curator at the RIT Cary Graphic Arts Collection, a renowned library that collects on design, typography, and the book arts. As manager of the Cary technology collection, she is responsible for teaching and maintaining over 30 different presses and thousands of fonts of metal

and wood type. She is actively involved in the international printing community, holding executive board positions with the American Printing History Association and the Hamilton Wood Type and Printing Museum.

T_EX and global mathematics

Patrick Ion

T_EX was developed as a way of communicating mathematics. It has been very successful for that and much more. But T_EX did not completely dominate publishing, though it much expanded the community able to write mathematics directly. MathML (Mathematics Markup Language) was specified as a markup for mathematics in the W3C (World Wide Web Consortium) context; it is both officially part of the web's basic HTML and an ISO standard. The idea that there should be a Global Digital Mathematics Library (GDML) is an obvious one. There's an International Mathematical Knowledge Trust (IMKT) devoted to eventually realizing a GDML, growing out of efforts by the International Mathematical Union. Some of how the present situation came to be and what's evolving now will be examined. For a historical marker see my article *MathML: A key to math on the web*, *TUGboat* 20:3 (1999), pp.167–175, tug.org/TUGboat/tb20-3/tb64ion.pdf.

HarfBuzz in LuaL_AT_EX

Marcel Krüger

Starting with T_EX Live 2020, LuaL_AT_EX uses the `luahtex` engine instead of `luatex`, and therefore allows the use of HarfBuzz instead of the ConT_EXt-derived font shaper. This presentation tries to answer some of the most important questions about this change, e.g.,

- How does this affect existing documents?
- How can this system be used?
- How does this help typesetting of scripts with which LuaL_AT_EX has always had problems?
- Why would I want to use this even for documents in scripts which are well supported by the existing shaper?
- In which cases would you not want to use HarfBuzz?
- How does this compare to X_YL_AT_EX's font handling?

MetaPost-based, dynamic extensible delimiters for LuaT_EX

Marcel Krüger

T_EX's math mode has always had support for extensible delimiters which can grow according to the content delimited by them, but these have been based on vertically stacking repeated parts. While this can

often provide decent results, it leaves a lot to be desired, especially for round or angle brackets. Given that LuaT_EX includes, with `luamplib`, a MetaFont-derived system, it should be possible to dynamically instantiate meta fonts with exactly the right delimiter size, without any restrictions regarding their composition. This presentation shows one implementation of this, which builds on MetaType1/AlgoType in order to output high quality, and potentially even fully hinted, vector glyphs which properly integrate into a modern LuaT_EX document.

Pandoc for T_EXnicians — TUG 2020 keynote address

John MacFarlane

I will give an overview of the document conversion program `pandoc` (pandoc.org), with an emphasis on how it might be useful to people who are already comfortable using L_AT_EX to prepare documents. In the first part, I'll discuss the use of `pandoc` to convert between L_AT_EX and other common formats, including Microsoft Word `docx` and HTML. In the second part, I'll give some reasons why even a seasoned T_EXnician might want to consider writing documents in `pandoc`'s extended Markdown instead of L_AT_EX, and I'll teach some tricks that can be used to recover the tremendous power and flexibility of L_AT_EX in this simpler idiom.

CMaps, Virtual fonts, ActualText for reliable text extraction and accessibility

Ross Moore

The single most critical factor for document content to be accessible is that text can be extracted reliably and accurately. For a PDF file, the CMap structure gives a mapping of each character in a font to a corresponding Unicode code point. The pdfT_EX and dvipdfmx engines have different ways to attach a CMap resource to font instances within PDF files. While it is a vital piece, the CMap is not the whole story; since the same character from the same font can be used in different ways. This is most apparent in a “fake” small-caps font, where uppercase glyphs are drawn from the same base font, but at reduced size using the virtual font mechanism. By defining a second virtual font instance, and attaching a customised CMap file and map file entry, the lowercase letters of a faked small-caps font can be correctly extracted as lowercase Latin letters.

Accented Latin letters are often constructed within virtual fonts by placing the accent first, then the base. This is counter to Unicode where the combining accent character comes after the base. By rearranging the virtual font description this order can be changed, allowing text-extraction of correctly

accented characters. This fixes many difficulties with a \LaTeX T1-encoded font, but some Extended Latin characters still need further consideration. Using a little-known trick of inserting DVI special commands into the virtual font description, ActualText replacement tagging can be encoded inside the virtual font, allowing constructed characters to be mapped to their proper Unicode point.

Making a new \TeX Live release available on Overleaf

Eric Mc Sween

Overleaf is an online collaborative editor for \LaTeX . It produces PDF documents using a full \TeX Live installation to compile projects authored by its users. Every year, when a new \TeX Live version is released, it needs to be integrated in Overleaf without breaking existing projects that worked with previous \TeX Live versions. This talk will explain how this is done. We will also take the opportunity to look at how the compilation service works.

\TeX Live 2020 news; `texlive.info` services

Norbert Preining

\TeX Live 2020 has seen the usual bunch of fixes and new version, but also one more significant change we have been working on for a long time: the renaming of containers to include the revision. The main aim of this change was to make life of distributors who rely on unique names easier. We will report a bit on the necessary changes and its implications.

The second part of the talk will briefly introduce the \TeX -related services at `texlive.info`.

Authoring accessible documents, including with TikZ diagrams

Thomas Price, Ross Moore

There are two parts to this presentation. Firstly, Tom Price will describe a bundle of \LaTeX files designed to build PDF/UA accessible documents from \LaTeX sources using a `pdf \LaTeX` engine. The bundle takes full advantage of the capabilities of the `pdfx.sty` and `tpdf.sty` packages while requiring minimal effort on the part of document authors.

Next, Ross Moore, author of `pdfx.sty` and `tpdf.sty`, will discuss how tagging can be achieved within a diagram created using `tikz.sty` package methods. It is becoming increasingly common to encounter images built this way, so it will be necessary to tag the information in these, so that it becomes accessible to readers with visual disabilities. Ross will demonstrate some promising first steps in this direction; in particular for a ‘SWOT analysis’ diagram.

Typesetting with Python

Brandon Rhodes

What would an algorithm look like that improves on the \TeX typesetting system’s scheme for breaking a book’s text into pages? This talk explores a new Python library for high quality typesetting that I been crafting, and that I have already used to format and print a short-run hardback book.

TopTeX, a new Q&A site for \TeX

samcarter

TopTeX (`TopAnswers.xyz/tex`) is a new site for questions and answers about \TeX and friends. It is part of the `TopAnswers.xyz` network, an open source and not for profit project. Its development is focused on the needs of the users and provides a friendly environment for building a high quality repository of knowledge.

The newest changes to `newtx` and its relatives and codependents

Michael Sharpe

The `newtx` package has undergone very substantial changes over the last couple of years, while striving to remain backwardly compatible with earlier versions. In this presentation, I’ll try to outline the motivations behind the changes and make comparisons with other general purpose \LaTeX math packages. Among the codependency issues are problems in adapting a math package to fit a text font package, and, to a lesser extent, vice versa.

`dePSFrag`, the final nail in the coffin

Paulo Ney de Souza, Vadim Ponomarev

Annotation of graphics in \TeX has always been a difficult subject. Solutions starting with `WARMReader` and passing by `PSFrag` and `pinlabel` did not have a path to more modern \TeX engines (`pdf-`, `xe-` and `lua-`). In this talk we present a framework for moving the source files of the most common of all these packages — `PSFrag` — over to any other labelling desired (`pinLabel`, `Overpic`, `XYOverPic` and `TikZ`), providing a path for processing legacy content and allowing more choices in production environments.