Short Note

Empowering SEP's documents

Sergey Fomel, Matthias Schwab, and Joel Schroeder[1]

keywords: *LaTeX 2ε, LaTeX, TeX, latex2html, Publication, WWW, Web*

INTRODUCTION

The arrival of LaTeX 2ε at SEP enhanced our LaTeX typesetting system and led us to improve SEP's customized enhancements. The revised system enables us to use the script latex2html (Drakos, 1996) to publish our documents routinely on the Internet. Additionally, we improved the communication between a document's makefile and its corresponding LaTeX file. Finally, we replaced a gigantic c-shell script (`texpr`) that governed SEP's entire document processing, by a set of small Perl scripts. These Perl scripts are easier to maintain, because they deal more flexibly with a wide range of processing choices. We hope we can share our customized LaTeX 2ε typesetting system with other communities such as SEG, TRIP (Rice University), or CWP (Colorado School of Mines).

In this paper, we first present a short example of a reproducible SEP document. In the electronic version of the document, this article can be turned into an interactive or standard HTML version, several DVI versions, and, of course, a Postscript version. The second part of the article states the basic author and reader commands (make targets), discusses the most important new features, and lists unresolved issues and goals. In an appendix, Sergey Fomel presents an annotated table of all SEP macros.

Former SEP students, especially Martin Karrenbach, Joe Dellinger, and Kamal Al-Yahya, wrote custom made LaTeX (Lamport, 1994; Goosens et al., 1994) and TeX (Knuth, 1984) macros to enable *Geophysics*-style layout and reproducible figures. Sergey Fomel translated these SEP macros to LaTeX 2ε, the modern LaTeX standard (Kopka and Daly, 1995). Authors at SEP can find a more elaborate document example in `/usr/local/src/our/sephelp/tex`

The article's main topic is SEP's transition from LaTeX to LaTeX 2ε. We usually refer to both systems as LaTeX and use the more explicit name LaTeX 2ε only where we want to distinguish between the two versions explicitly.

[1]**email:** sergey@sep.stanford.edu, matt@sep.stanford.edu, schroeder@alumni.stanford.org

DOCUMENT EXAMPLE

In this section, we imitate a miniature SEP research paper. At SEP, such a research paper not only includes an article with figures, equations, and references, but also a set of accompanying programs, input data, and makefiles. In its electronic form such a research document enables the reader to reproduce the results that the document presents (Schwab et al., 1996).

The research document that we embedded for documentation purposes implements Normal Moveout, a standard geophysical processing step. The equation that governs the Normal Moveout is a mapping from time t to zero offset time t_0

$$t^2 = t_0^2 + \frac{h^2}{V^2} \tag{1}$$

where h is the offset and V the subsurface velocity. We copied the underlying program and the following result from Jon Claerbout (1991).

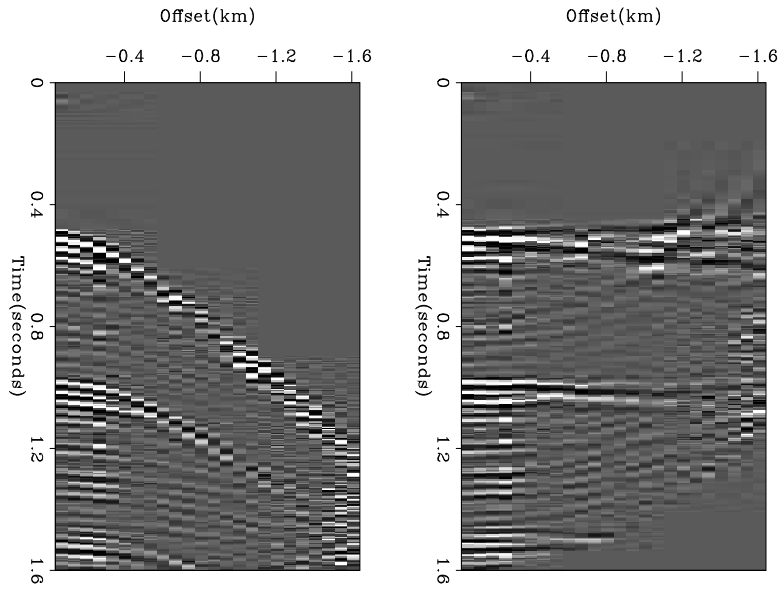

Figure 1: Marine data moved out with water velocity. Input on the left, output on the right. (After Jon Claerbout, 1991) stretch [ER]

If you have access to the electronic version of this article (online at SEP or on a SEP CD-Rom), we suggest you inspect the various output formats of this scientific mini article. Simply execute the commands described in the next section.

DOCUMENT SYSTEM

SEP's make rules provide standard commands to create the various document versions. When within the SEP environment (at an SEP computer or on a SEP CD-Rom), go to any SEP document, type one of the following commands, and inspect the various results:

gmake paper.ps creates a Postscript file.

gmake paper.gs displays a Postscript file with `ghostview`.

gmake paper.dvi creates a standard DVI file.

gmake paper.read creates the DVI file and displays it with `xdvi`.

gmake paper.print creates the DVI file and prints it to the default printer.

gmake paper.ltx converts an SEP style TEX file into a proper LATEX file, paper.ltx. The paper.ltx file is helpful for debugging since the line numbers of TEX's log file refer to it.

gmake paper.slide creates a DVI file using an official slide macro set.

gmake paper.jslide creates a standard SEP DVI file but with an reduced page width of about 40 characters. Jon Claerbout likes to make slides from this DVI file by enlarging the text on a copy machine.

A reader or author may want to try some of these more experimental document commands:

gmake paper.html creates an HTML subdirectory (`paper_html`) that includes an HTML version of the LATEXpaper. The make target will invoke the latex2html script. The rule uses standard X11 programs to generate the necessary gif figures for equations and figures. We still have to define a set of adequate clean rules.

gmake paper.browse updates the `paper_html` directory and starts-up a browser.

gmake paper.idvi creates a DVI version that can be rendered by IDVI, a Java renderer. IDVI is an experimental alternative to latex2html.

gmake paper.xtex displays a DVI file with `xtex`. This command uses an old set of SEP's LATEX macros. You may have to remove some new macros, such as keywords. Additionally, Xtex does not find the activeplot figures correctly. In the near future, we probably have to remove the paper.xtex rule.

The targets above can be applied to text source files with any name as long as the name ends with the suffix *.tex*. For example, to create a Postscript version from a text source file named *mytext.tex*, you execute `make mytext.ps`.

In general, the new LATEX 2_ε macros are backward compatible. You should be able to process any old SEP LATEX file with the new SEP LATEX 2_ε macros. The only known backward incompatibility is that the new activeplot macro always requires a width or height statement.

To ease the maintenance of reproducible documents, we improved the communication between the document's makefile and its LATEX file. Consequently, the author does not need

to define the result directory in the LaTeX file (previously: `\def\figdir{name}`). Instead the new document processing system requests that information from your makefile. Try processing an old SEP LaTeX document after removing the figdir statement.

A SEP LaTeX 2ε document forces an author to supply a keyword list with the help of the macro `\keywords{}`. The keyword list is printed below the list of authors and will help us to index SEP documents in the future. We are planning to supply a set of keywords that an author can choose from (e.g., the SEG keywords for expanded abstracts).

Finally, we added an enhanced clean rule, `jdistclean`, to our standard make rules. The distclean rule creates a minimal set of files: in addition to the intermediate files that are removed by the clean rule, it also removes all reproducible result files and any HTML directories.

Books and reports

The standard LaTeX 2ε book style simplifies the generation of books and reports. The book format solves the TeX memory limitation problem that we frequently experienced in the past when printing our reports. The chapters and their order are defined in a makefile. The table of contents is generated automatically by querying the titles of the included chapters. This organization will free us from storing all included chapters in sister directories.

The system for creating and maintaining books and reports consists of a small set of book-specific make rules (`Doc.rules.book`), empowered by set of simple perl scripts for automatic creation of the book attributes (table of contents, title page, author's biography.)

Installation of SEP's new LaTeX 2ε system

To download SEP's new LaTeX 2ε setup, follow instructions on our web page at `<http://sepwww.stanford.` The page includes instructions as well as URLs to download the corresponding tar files. The web page also includes a Frequently Asked Question (FAQ) table and we invite users to add additional helpful information. Additionally, we distribute our entire document system on our semiannually released CD-Roms.

WHY LaTeX 2ε?

The LaTeX community has adapted the LaTeX 2ε standard. New typesetting tools, such as the script latex2html, the java applet IDVI, or PDF distiller, expect LaTeX 2ε. SEP needs these typesetting tools to publish on the web. Additionally, LaTeX 2ε 's clear package and class organization as well as its ease of installation should reduce SEP's TeX maintenance.

We had to translate the SEP-specific macros from LaTeX to LaTeX 2ε dialect. In our translation, we aimed at two contradictory goals: to ease the portability of SEP documents and to be fully backward compatible.

To ease the portability of SEP documents, we developed a minimal set of SEP macros. SEP's minimal macro set contains only a third (26) of the traditional SEP macro family (78). This reduced macro set mostly deals with the inclusion of Postscript figures (activeplot), the inclusion of source code files (gprogblock), with author information (author, email, keywords), and SEP style definitions[2]. All other commands are standard LaTeX 2_ε commands. We hope that the minimal macro set will enable us to share our LaTeX 2_ε macros easily with other organizations that have similar needs (SEG, TRIP at Rice University, and CWP at Colorado School of Mines).

We recommend this minimal set for future SEP documents. To restrict the processing of his document to the minimal set, an author needs to define `LATOPTS = option=minimal` in his document's makefile. However, the current default ensures backward compatibility since it uses the superset of traditional SEP macros. This backward compatibility will, we hope, simplify the transition to the new document system.

Why do we use LaTeX rather than any other text or word processing system such as Microsoft Word? We have used LaTeX in the past and all students at SEP can write LaTeX documents. We also have a legacy of LaTeX documents. LaTeX still seems to be the method of choice for mathematical typesetting, but we have not seriously tested some attractive-sounding alternatives, such as framemaker (<`http://www.adobe.com/`>). Additionally, since LaTeX2 is freely distributed, we can include it on SEP's interactive CD-Roms.

From one source to many documents

Sergey, we need a graph of the various files and directories. I am afraid the HTML-gif setup will look cluttered. Oh well.

We wrote two relatively short Perl scripts and a few make rules to enable the author to translate his document file, paper.tex[3], to his favorite output format, such as Postscript, HTML, DVI, etc. The two Perl scripts – Actify, Latify, and latex2dvi – are extensible and easily maintained. These scripts replace SEP's former C-shell script, `texpr`.

Actify extracts relevant information, such as `RESDIR` (the result directory) and `ER, CR, NR` (reproducibility status) from the makefile and includes it in the document file. Until now, authors had to state the `RESDIR` in both the document file and the makefile. The reproducibility status was clumsily communicated via the creation of dummy files.

Latify turns the document source file into a complete LaTeX file. Latify adds a document style statement, a begin and end document statement, etc. The script's output is a standard LaTeX file that uses SEP macros.

latex2dvi translates standard LaTeX to a device independent format (DVI). Basically, it runs `latex, bibtex, makeindex` as many times as necessary.

I wish the scripts would echo the commands they run. Think about independent export of make rules and doc system. We still need to add options to process EAEG and SEG abstracts and articles.

This set of Perl scripts currently produces a family of output formats of the original document: The most common formats are DVI versions, that can be displayed with a DVI

[2]These style definitions follow SEG's style definitions for publications in its *Geophysics* journal.

[3]At SEP we call this file `paper.tex` despite the fact that the file is actually a slightly incomplete LaTeX file, not a TeX file.

reader (xdvi) or converted to Postscript (dvips). For the time being, we additionally support an enhanced DVI format based on the xtex DVI reader. Xtex permits us to display DVI files that have buttons in the Figure captions. When pressed, these buttons bring up a graphics user interface that enables the reader to interact with the figure. We hope to replace xtex by an alternative interactive renderer, since xtex is non-standard and hard to maintain.

Maybe we should drop xtex. I had hoped we could keep it until the HTML solution is perfected.

Additionally, we use the freely available **latex2html** script, which translates standard LaTeX to HTML. We enhanced the latex2html Perl script to process certain non-trivial SEP macros into their HTML implementations. In particular, SEP's figure macro `activeplot` is translated by a customized Perl subroutine (latex2html explicitly endorses such enhancements). In the next section, we discuss our attempt to create interactive HTML pages for SEP's reproducible documents.

Additionally, we are currently testing an alternative web solution based on the Java DVI renderer, IDVI (<http://www.geom.umn.edu/java/idvi/>). IDVI allows a reader to interact with a document through embedded Java applets.

Figure buttons in xtex and HTML documents

The button in the caption of Figure 1 (xtex or HTML version) is the entry point to the interactive reproduction of the article's result. The button in the traditional xtex version of this paper will issue a makefile command that in return invokes an xtpanel (Cole and Nichols, 1992) graphical user interface. The GUI offers the four standard commands of SEP's reproducible research: to view, burn, build, and clean the document.

In the HTML version of the paper, the reader can execute these standard commands by pressing buttons beneath the figure. To enable Netscape to issue certain commands (make invocations of the targets view, burn, build, and clean), we introduced a `sepshell` program and added an application mime type to the mailcap file. These `sepshell` programs issue the corresponding GNU-make commands on the local browser client. We code the absolute path of a document into a figure's HTML representation. The HTML figure buttons use this absolute pathname to connect to the document's corresponding makefile. In practise, this mechanism limits interactive HTML versions of reproducible documents to SEP's local machines, unless a reader downloads and installs the SEP environment on his own computer or runs it from a CD-ROM.

In addition to the traditional buttons that relate to the reproduction of a document's results, latex2html adds an edit and reset button. The edit button displays a document's makefile for editing. Other files can be opened and changed with the editor. If the document is maintained under RCS version control (You can find more about RCS at any GNU software site, e.g. <http://www.delorie.com/gnu/>), then the restore button will reset all files to the original state. This feature might be useful for teaching courses, but as a default we plan to disable the edit and restore button to avoid accidental changes of the underlying software.

A simple `nobutton` definition in the paper.tex file will result in a straightforward HTML or xtex document without any interactive buttons.

Issues under discussion

- James Rickett and Matthias Schwab want smaller buttons in the HTML document. What should the default buttons look like? A full set of buttons includes `up, next, previous, up-inactive, next-inactive, previous-inactive` (for navigation) `view, burn, build, edit, restore, print, clean, warning` (for `sepshell` interaction).
- James Rickett points out that the figure reference takes you to the figure caption. Usually, a reader needs to scroll up his page to see the figure. Joel Schroeder tried to fix this but did not succeed.
- We hope we can soon demonstrate a reproducible web document implemented in Java.
- We wish there were a Java version of *make* (preferably called *jake*).
- It would be nice if the HTML version of a document would contain all WWW-references as hyper-links. Do we need the Hyper-TeX package or just the syntax? Can we use the Hyper-TeX extensions in the bibliography files?
- Should we suppress TEX's very verbatim processing messages? Sergey prefers the interactive mode, since it enables him to ignore small errors while processing.
- Were we to give all encapsulated Postscript figures the suffix *eps* instead of *ps*, then we could distinguish between figure.ps and paper.ps files in our clean rule (Tariq Alkhalifah suggested the *eps* suffix for a different reason).

CONCLUSIONS

SEP has successfully switched from LATEX to LATEX 2_ε. The new system permits us to experiment with new Web representation and distribution of our research documents. Additionally, we hope the new system will be easier to maintain and to extend.

ACKNOWLEDGEMENTS

We especially want to thank Martin Karrenbach for his wealth of macros and his continuous effort to move SEP to LATEX 2_ε. Martin also introduced us to IDVI. Thanks to James Rickett, Robert Clapp, and Biondo Biondi for their testing and suggestions.

REFERENCES

Claerbout, J. F., 1991, Earth Soundings Analysis: SEP–**71**, 1–304.

Cole, S., and Nichols, D., 1992, Xtpanel: An interactive panel builder: SEP–**75**, 497–520.

Drakos, N., 1996, All about LaTeX2HTML:
`http://cbl.leeds.ac.uk/nikos/tex2html/doc/latex2html/latex2html.html`.

Goosens, M., Mittelbach, F., and Samarin, A., 1994, The LaTeX companion: Addison-Wesley Publishing Company.

Knuth, D., 1984, TeX book: Addison-Wesley Publishing Company.

Kopka, H., and Daly, P. W., 1995, A guide to LaTeX 2_ε: Document preparation for beginners and advanced users: Addison-Wesley Publishing Company.

Lamport, L., 1994, LaTeX: A document preparation system 2/e: Addison-Wesley Publishing Company.

Schwab, M., Karrenbach, M., and Claerbout, J., 1996, Making scientific computations reproducible: SEP–**92**, 327–342.

APPENDIX A