
Interview: Michael Sharpe

David Walden



Michael Sharpe has been using \TeX since the mid-1980s. In more recent years he has been active in the \TeX fonts world.

Dave Walden, interviewer: Please tell me a bit about yourself.

Michael Sharpe, interviewee: I was born in Sydney, Australia in 1941. After 1945, my father joined the Commonwealth Public Service, which corresponds in the US to the federal civil service, and moved frequently in order to advance in the system. I had a disjointed schooling in various suburbs of Sydney, Melbourne and Hobart (Tasmania), completing high school and university in Tasmania. I began as a student in Electrical Engineering but found Physics and Mathematics more appealing, thanks in no small part to some inspiring faculty in those areas, and eventually graduated with a degree in Mathematics, after which I completed a Ph.D. in Mathematics at Yale specializing in Analysis and Probability. The interests I shared with some faculty members at the newly formed Mathematics Department at UCSD (University of California at San Diego) led to a position there which continued, except for a year at the University of Paris VI, until my retirement in 2004.

I fantasize sometimes about how different my life might have been had Electrical Engineering at UTAS encompassed what we now know as Computer Science at the time I was a beginning undergraduate in 1959. I also fantasize about what might have been had I continued in physics, though it seemed I had limited capabilities at lab work. (When I asked my friends working in experimental physics about their backgrounds, an unexpectedly high proportion responded that they were the sons of farmers, who understood and could repair all farm machinery.) My youth was misspent on sport, not on understanding

complex machinery, though I did spend a couple of years working as an assistant to a projectionist in the local movie theater during my high school years.

DW: When you say “misspent on sport”, what are you thinking of?

MS: Because we moved regularly, I was motivated to focus on making new friends as quickly as possible, and sport was a good way to do it in that environment. I played cricket, Australian Rules football and tennis. It was fortunate for my later career that I was not really good at any of them.

DW: Were you already doing electronics things as a hobby and enjoying high school math and science before university?

MS: I was not into electronics as a hobby, finding the analog radio of those days not very interesting. I did do well in sciences and math in high school. If there had been computers available in those days, it may have been a different story.

DW: What took you away from Australia and to Yale for your Ph.D. work?

MS: Just previous to my generation of college graduates in Australia, most students wanting to pursue an advanced degree in sciences and engineering went to Great Britain if they could manage it. In the early 1960s the US graduate programs in those areas expanded greatly due in no small part to the post-Sputnik flood of funding. From my limited knowledge of advanced mathematics as an undergraduate, Yale seemed to have a number of major figures in areas of mathematics I enjoyed and thought important. I planned to return to Australia after my Ph.D. and a post-doc, but I married an American while I was at graduate school. My wife did not want to live so far from her family, so I decided to stay in the US (semi-)permanently.

DW: How did you first become involved with \TeX ?

MS: From the mid-1970s, I was working on a manuscript on Markov processes that I hoped could be turned into a book. I was using TROFF and spending much of my research funding to process the source. In the early 1980s, I heard of \TeX and decided to convert to that system, largely because of cost. I made much use of UNIX sed scripts to transform the original source to \TeX . The book was eventually published under the title *General Theory of Markov Processes*, Academic Press (1988).

DW: Are you saying that \TeX was less expensive in terms of computer time? That’s a bit of history I have not heard of before.

MS: It was more a matter that TROFF had to be run on the school's mainframe, for which there was a cost associated with each run. Even at the cheapest overnight rate, my research grant took a serious hit. I learned about Textures in the mid-1980s, and could purchase a license for about \$200 due to our site license. The idea of almost instant feedback was very appealing. (In fact, it took close to 40 minutes per page to process using my early generation Mac.)

DW: Do I gather correctly that you were using T_EX and not L^AT_EX at that early date? (By the way, it appears to me that your book is available on-line for the cost of registering on a website¹.)

MS: I used plain T_EX with the AMST_EX additions for almost my entire academic career, as my experiments with early L^AT_EX made it seem very slow and harder to modify than plain T_EX, which I understood well. Mine was one of the first books accepted in T_EX by Academic Press. They were happy to not have to reprocess it, but asked me to make the chapter headings and such conform to their standards using a Times font and a layout they prescribed. That was my first font job. I understand that they distributed my macros to their other authors whose manuscripts were still in progress.

DW: During your years of teaching at UCSD, what sorts of courses did you teach, and were they mostly to students of math or also students from other departments? Also, given your considerable knowledge of T_EX, etc., did you end up being a resource on T_EX for your department?

MS: I taught courses of many descriptions. Much calculus and advanced calculus of course, as the bulk of our workload was engineering and science majors. I also taught upper division courses in real and complex analysis, probability, mathematical statistics and the mathematics of financial models, mostly taken by majors in math, engineering and economics. At the graduate level, I taught the basic foundation courses in real and complex analysis, probability, as well as advanced topics designed to attract students to one's research area. Many of the students who took my probability courses were electrical engineering graduate students working in communication theory, and as a result, I ended up serving on the dissertation committees of many of them. (That area attracted a steady flow of very capable students thanks to the presence of some first rate researchers and the generous support of QUALCOMM. It was the first time I ever posed a question to a student that elicited the response: "I'm sorry, but that information is covered by a non-disclosure agreement.")

I did end up as a T_EX resource for my department, trying to bring the staff and graduate students up to speed with T_EX. In the years of the severe budgets of the early 1990s, the department had inadequate computer support and lacked skilled staff to do serious document preparation. It took some time for new staff to learn to use T_EX productively, and they found that processing graphics in T_EX documents was too time consuming, so I ended up trying to help them use PSTricks. This was not really a success as PSTricks required regular use to maintain proficiency. I also set up a number of automated processes to handle departmental information efficiently, much of it using T_EX for output processing.

My passage to L^AT_EX was fairly sudden. The department was contacted by the Dean of Natural Sciences (this was about 2002) saying that one of his major donors was trying to write a math textbook for his grand-children, and was having serious problems getting his chosen fonts to work on his Mac using L^AT_EX, and could we do something to help. As no one else stepped forward, I was volunteered. It turned out he was trying to follow one of the recipes in the Alan Hoenig book, *T_EX Unbound*, wanting to set up Adobe Garamond with math from MathTime using a fontinst script. (He was a very smart man in his late 80s who had a Ph.D. in Chemistry, and had set up a very successful corporation that developed rocket parts and fuels. He said he had always loved mathematics but felt it was never properly explained to pre-college students. I visited him in his retirement home, and he declared that I might be the first mathematician to make house calls since Bertrand Russell called on Lady Ottoline.) So, that was my first real font effort, getting things set up for proper fontinst run, and moving the output to the non-standard places expected by Textures. That meant I had to understand L^AT_EX, and I found it much more compelling than I had earlier.

After I retired in 2004, I used L^AT_EX for the first time to write a math paper, having the luxury of time to learn its fine points. I also became very interested in PSTricks for a period and wrote some packages for it, some of which worked its way into the basic PSTricks packages. That gave me my first real experience working with Bezier curves, which are the foundation for work with outline fonts. It was also a good learning experience with complex T_EX packages, as there is some fiendishly clever code in PSTricks, mixed with non-trivial PostScript code. Its long-time maintainer, Herbert Voß, is very knowledgeable and helpful in all (L^A)T_EX matters.

After I made some baby font packages, I saw an opportunity with the TXfonts and PXfonts families,

which had been developed but left in an unfinished (or, at least, unpolished) state. “How hard could it be to finish them?”, I thought. The answer is that it took about four months to be ready for an initial release, but work has continued for the last five years through today, fixing bugs, improving metrics, adding and making changes to symbols, and so forth. Once I learned what was necessary to produce suitable text and math fonts for L^AT_EX, I continued to look for opportunities to bring new fonts to the T_EX world. I feel strongly that the future of Unicode T_EX is not in making use of system fonts, which are not the same for everyone, and can be suddenly changed or withdrawn, but with free fonts that have been enhanced enough to meet the needs of the demanding academic user.

As a side note, I add that I see many journals abandoning proprietary fonts in favor of free packages. With submissions using their specified free-font packages, the editorial work is reduced and the submitter gains by a reduction in the number of errors caused by reworking the submission to make use of a different, sometimes not completely compatible, package.

DW: Since your “first real font effort, getting things set up for proper fontinst run, and moving the output to the non-standard places expected by Textures”, your CTAN entry² suggests you have done a *lot* more with fonts and a good bit with L^AT_EX. What led you to get so deeply involved with fonts and can you talk about the different things that were involved with two or three of your font efforts?

MS: I never got over the thrill of seeing a page of mathematical content pop up on the screen with relatively little effort, especially compared with what had to be done to prepare manuscripts in the years prior to T_EX. For day to day use, I found that Computer Modern had some drawbacks, not copying well on machines available in the late 1980s, so homework and exams had to be made at very large sizes to be guaranteed to be readable. I started looking for other font systems, and purchased MathTime and Lucida, solving the copying problem. The Garamond+MathTime project after that reawakened my curiosity about T_EX and other fonts.

A second factor was that, after my cleanup period following retirement, I needed a serious interest to occupy my time and my mind, and T_EX seemed to qualify. I studied Hoenig’s book at length and spent several weeks trying to modify Hoenig’s mathinst Perl script so that it would actually work as expected. Though I eventually dropped the project, I learned in the process a great deal about the use

of fontinst to produce L^AT_EX math font support files, all of which has influenced my subsequent work on math fonts for T_EX.

Creating a font, and especially a math font, from scratch is a project of several years duration. I don’t normally create fonts, I try to rescue them. There are a number of font projects that have free licenses but were abandoned, in many cases because the author eventually needed to make a living. I’ve picked up some of those projects that seemed promising and worked to turn them into more polished products that can, I hope, be used for professional work.

My biggest project has been to rescue the text/math families TXfonts and PXfonts by Young Ryu, renaming them to newtx/newpx. They are based on the URW clones of Times and Palatino and mostly share a common math core, modulo scaling. Initially, the main issue was the metrics, and that took several months to correct. Later, I turned to a complete reworking of the math extension fonts, where the extensible delimiters were not matched well. This required about six weeks of effort. There followed a complete revision of all the delimiters, redrawing the smaller ones so that they worked better with the larger ones. After that came the conversion of formerly 7-bit math fonts to 8-bit, in an effort to economize on math families. Following that, I dropped the TXfonts and PXfonts text fonts and based the text fonts on the T_EX Gyre fonts, which had much more to offer, extending them with larger small caps which, in the case of newtx, were metrically equivalent to Adobe Times Small Cap fonts. I would estimate that I’ve spent over a year working on the project so far, and more is to come as it has become a headache to administer, and I’m trying to rework the entire package so that it is organized more rationally and can be more easily taken over by another person when I’m no longer capable. At last count, these packages are used, sometimes in conjunction with Libertine, to typeset at least a hundred journals, so this is quite an important issue.

I would say that LibertinusT1Math has to be considered my next most important math font package, based on Khaled Hosny’s LibertinusMath, a Unicode math font to match Libertinus, his fork of Libertine. Going backwards, so to speak, from a Unicode math font to a L^AT_EX math font family, involved adding several dozens of glyphs, mostly slanted integrals and extensible delimiters. I used STIX as my model for constructing 8-bit math fonts and a sty file, spending nearly four months on the project from the beginning to first release. Recently, at the urging of Claudio Beccari, I added a parallel sans serif branch to the math fonts so that (a) Liberti-

nusT1Math could be used without compromise as an ISO-compliant math font, and (b) the font could be used as a math font in which all alphanumerics (Roman and Greek) are sans serif. This took close to two weeks of additional work.

DW: What kind of feedback have you gotten from the community regarding your font work, and have you created a big maintenance task for yourself by your development work?

MS: The positive responses I've had from people in the T_EX world for whom I have great respect — Karl Berry, Boris Veytsman and Claudio Beccari in particular — have been very significant to me, helping me to sustain my energies. I've had a steady stream (one every two to four weeks) of email from users about bugs, to which I try to respond as quickly as I can. This is less of a burden than I was expecting. I also receive requests from users who would like some new features added. A few of these ask me to take on some very large scale new projects, and, while there are some interesting proposals to consider, my personal interest would have to be very high to agree to such an effort.

DW: Please talk about the tools and methods, both computer and anything non-computer you use in your font work? For instance, do you print out big proofs of changes and view them off a monitor, study books and font specimens, do glyph-by-glyph design and spacing, or does FontForge make it practical to be more efficient?

MS: I use mainly FontForge, but I turn constantly to Python to create special scripts to analyze and gather information about glyphs in a font. (The `sfd` file format used by FontForge is plain text and not hard to parse.) The free `ttx` program (part of `fonttools`) has also been useful, but recent versions have in some cases required much digging to repair problems caused by the installer. The version of FontForge I use is the binary with all required libraries built-in. Unfortunately, it does not allow external python scripts — i.e., FontForge was not built as an extension of python. I've tried to get this functionality when needed by building FontForge on a Linux machine as an extension of python. I don't use it frequently and find it highly bothersome, as it seems that I often have to reinstall a new version of Ubuntu and FontForge just to run a couple of scripts.

DW: Do you have esthetic preferences among fonts? Which was the most actual fun to work on, and the least?

MS: My esthetic preferences are not constant in time. A few years ago, I was most enthusiastic about

old-style fonts, which I take to mean a font either designed prior to the early eighteenth century, or a later revival of such a font. I liked their playful qualities and their close relationship to pre-sixteenth century formal handwriting. While in this frame of mind, I found fonts like Utopia and Charter austere and rather dull, seeming to strive for the opposite of old-style. My attitude is no longer so rigid, and for most of my current typesetting purposes, I find myself leaning toward plainer text fonts. It may be that I find them much easier to modify.

I enjoyed working on Garamondx at the time I did it, as the payoff was very pleasing, turning a rather limited font family into one that many people seem to find useful for academic writing, especially in the humanities. The least fun I've had was with Cochineal. It's not that I didn't really like Crimson, the font family it extends, but I agonized a great deal about whether the many months spent to rework glyphs and metrics outside the basic Latin area (e.g., extending Greek and Cyrillic in italic, bold and bold italic) would matter to anyone except me. In the end though, I was happy with Cochineal, and even happier when I learned that the `suftesi` package used by scholars in Italian humanities had taken this as its default font package.

DW: You have also created many pieces of T_EX utility software³. Please speak to your general approach/philosophy of creating tools to help you, versus pushing ahead brute force with a primary task, perhaps using a couple of examples.

MS: With font work, much of what I have done is indeed best characterized as brute force, but in a number of cases it was useful to turn the brute force approach into a script. Most of the software I have on github is either earlier versions of font packages or Python scripts, shell scripts or AppleScripts designed to automate some common workflows in T_EX on the Mac. Many are specific to TeXShop, Dick Koch's very fine front end to T_EX Live and some Mac-specific T_EX binaries. TeXShop provides a menu interface to AppleScripts that can be used to modify the T_EX source or provide access to some of its features. Some T_EX luminaries (e.g., Will Robertson of `fontspec` fame) have contributed scripts. I find the available scripts indispensable in my daily T_EX life.

DW: What is your view of how T_EX and its derivatives fit into the world today, e.g., within the evolution of the T_EX user community and its place (if any) in the larger world of typesetting, type design, and fonts?

MS: You may be asking something here that is beyond my pay grade. I interact mostly with academic

types, whether my mathematical colleagues who, if they write their own papers, do it with some version of \TeX , or with people who use my font packages. So far, the latter have all been academics or graduate students writing dissertations. In the case of my more decorative text fonts (Garamondx, newpx, Cochineal, BaskervilleF) the users are almost always from the humanities, and with a good number of them, matching Greek and Cyrillic is important, as are such supposedly deprecated encodings as OT2 and LGR. This has encouraged me to add support for them in font families that support those encodings, and to add the glyphs required for polytonic rather than monotonic Greek, where possible. I would be overjoyed if \TeX became important to a larger world in publishing, but I see no interest in this except for publishers of mathematical content, save for some Indian companies who have been very innovative with automating their output streams using \TeX .

I think there are people out there who will find \TeX more attractive with a wider selection of fonts that allow the full range of typeset options expected in academic writing — choices of figure styles, superscript and subscript styles, small caps, Greek and Cyrillic alphabets, and an accompanying math font. It has been my goal to provide more options for that group, which I think is critical to \TeX 's future.

DW: Has your work with \TeX , etc., led to broader interest in book design, etc.? And, with your considerable efforts with fonts and other aspects of \TeX , does that leave you time for non-typesetting enjoyment?

MS: I have not left myself with much time for other activities outside what I already do in \TeX , fonts and programs to support TeXShop, but my wife and I are opera fans and spend much of our spare time attending performances or watching/listening to electronic versions of them. There is now an extraordinary collection of historic opera on youtube and other Internet sites, the exploration of which is so absorbing as to concern me about my font future.

DW: Thank you very much for taking the time to participate in our interview series.

[Interview completed 2017-09-11]

Links

¹ <http://ebooksdownloads.xyz/search/general-theory-of-markov-processes>

² <https://ctan.org/author/id/sharpe>

³ <https://mjsharpe.github.io/tex-software/>

◇ David Walden
<http://tug.org/interviews>

DW: An addendum: is there something you would like to show from your recent packages or recent additions to your packages? In addition, could you give us some idea of any new packages you are thinking about?

MS: (The previous paragraph used Cochineal, similar to Minion Pro. This paragraph uses XCharter.) I think the newish changes to XCharter are notable: (a) the addition of Cyrillic alphabets, including small caps, in T2A encoding; (b) the addition of Serbian Cyrillic italic glyphs. These can be used with pdf \LaTeX as well as x \LaTeX and lua \LaTeX . To my knowledge, this is currently the only package which offers correct rendering of Serbian Cyrillic italic under pdf \LaTeX . You can see from this paragraph set in XCharter that it needs to be scaled down a bit, and has a substantially larger x-height than Computer Modern and Times.

The font in this paragraph is AlgolRevived, designed for typesetting coding, but marginally suitable for brief sections of text. (Unlike most coding fonts, it is not monospaced, though its figures are. Don't use this for Fortran.)

(Back to Cochineal.) Here's a sample of some mathematical text using sans math available in the libertinusmath package with preamble

```
\usepackage{libertine}
\usepackage[sansmath]{libertinus1math}
```

The text font in this paragraph is Libertine and math is sans serif (derived from STIX math sans) with symbols from libertinus1math.

$$\tilde{f}(\alpha) := \frac{1}{\sqrt{2\pi}} \int_0^t e^{-\alpha x} f(x) dx$$

My plans for the immediate future are to complete my reworking of newtxmath and newpxmath to a form where it is not hard to provide options that will allow the user to select from a list of alphabets for calligraphic, script, double-struck and fraktur alphabets without needing additional math groups. I would hope to do the same with libertinusmath. With the latter package, I'm not fond of the existing very small math binary relation symbols, and would try to add alternatives more like those in CM and Times.

With text fonts that have a much larger x-height than Times and CM, such as XCharter, Utopia, PTSerif, Erewhon and Century Schoolbook, a matching math font should be heavier and of larger x-height than Times math, and as far as I know, the Fourier package, which works currently only with Utopia and Century Schoolbook, is the only free choice. Given the time, I would like to work to extend Fourier math to have the same interchangeable features as newtxmath.