

Thoughts About Publishing Mathematics on the Web

*Timothy W. Cole, Mathematics Librarian
University of Illinois at Urbana-Champaign*

Introduction

The World Wide Web is an invaluable tool for communicating scholarly information, but there remain difficulties and issues when trying to use the Web to communicate certain kinds of information, most notably scholarly papers and journal articles containing complex mathematics. The good news is that there is a variety of encoding and presentation technologies available that can be used to good effect in the Web environment (“a dazzling array” is the way Robert Miner and Paul Topping of Design Science expressed it¹). The bad news is that, for the present at least, there’s no single, universally accepted silver bullet; a comprehensive Digital Mathematics Library will have to accommodate a variety of technologies and approaches. There is no one right answer for all contexts. Here is a necessarily selective list of options available and my personal thoughts on the strengths and weaknesses of each in this context.

Native HTML

HTML is especially ubiquitous and is also very much an open standard. As such it has advantages of economy and essentially universal availability to end-users. However, direct support for mathematics in HTML is minimal (to put it kindly). HTML provides intrinsic encoding for a limited number of special characters, tags to designate pre-defined super and subscript character positioning, a tag to turn off automatic line breaks, and little else. There’s no direct HTML support for kerning, the use of combining diacritics, arbitrary glyphs and fonts, equation structures, or any of the other rendering system features essential for presenting complex mathematics.

The advent of Cascading style sheets and font embedding has extended the ability to depict mathematics in HTML, but such techniques are limited. CSS was not designed with mathematics in mind and CSS positioning mechanics are clumsy and awkward to apply for mathematics. Robust generic solutions are impossible and the process is generally quite labor intensive. Moreover, both CSS and embedded font solutions vary by Web browser type, and often by Web browser version. A CSS stylesheet that works for Internet Explorer 5 most likely does not work for Netscape 4.

The other approach used in HTML is to embed mathematics needed for a Web page in the form of GIF, JPEG, or PNG images. This approach, though ubiquitous also has severe limitations. Reducing mathematics to a binary screen display format means it is of no use for resource discovery purposes. (Embedding images in the midst of sentences can complicate document full-text search and discovery generally.) The number of images required for even a modest paper can be quite high. Our experience in a recent project at Illinois was that a single 20 page technical journal article could have more than 1,000 discrete instances of embedded mathematics. Binary display representations of mathematics don’t scale as screen fonts are changed, and typically don’t match the fonts used on different clients well.

¹ Miner, Robert and Topping, Paul (2001). Math on the Web: A Status Report, Long Beach, California: Design Science, Inc. Online. Available http://www.dessci.com/webmath/status/status_Jan_01.stm.

Page-Oriented Formats: T_EX and PDF

Of course the challenges associated with mathematical typography are nothing new. Mathematicians have been pushing the envelope of print technologies since printing was invented. As a solution to the problem of mathematics on the printed page, T_EX has proven a popular and relatively satisfactory solution since its introduction in 1978. Use of it in a Web environment leverages the extensive work that has been done in print publishing and typesetting environment. It is extensible and there are a wide range of T_EX-aware authoring and publishing tools. Really more a set of instructions for presentation than a self-contained binary version of a mathematical display, T_EX is most often implemented on the Web only after translation to DVI and then another Web ubiquitous format such as Adobe PDF. (There are plug-ins such as IBM's TechExplorer that utilize T_EX directly, but most are not free, and distribution on end-user workstations is limited.)

The approach has its drawbacks and limitations. It is complicated to learn, somewhat difficult for neophytes to use well (there's a lot of poorly written T_EX out there), and comes in a multiplicity of flavors, confusing to the non-specialist. It was developed and optimized for the printed page. T_EX and PDF are particularly good for mimicking the format and layout of a printed journal article, but as we move to more and more "born-digital" content this printed page orientation has drawbacks. Also, semantic content is limited. In its native form, some semantics are implicit in an equation is encoded in T_EX, but it was not designed to describe the meaning of the mathematics, only the way the mathematics should look when printed. Once transformed to DVI and then to PDF, even more of the semantic meaning is lost. Consider this example (taken from an article by O. Caprotti and D. Carlisle²). A human reader viewing the polynomial:

$$ax^4 + bx^3 + cx^2 + dx + e$$

and the equation:

$$e^{ix} = -1$$

will immediately recognize that the entity "e" appearing in both has a semantically different meaning in the two contexts. To a machine-based process however, viewing these expressions embedded inside a PDF document, the difference will not be clear. Page-oriented formats like T_EX and PDF (and PS, etc.) will continue to be especially useful when doing retrospective digitization of print materials, but we can improve on these formats for new, born-digital mathematics.

Advanced Mark Up Languages: SGML / XML / MathML / OpenMath

SGML has been around since the 1970's. It is (to a degree at least) a direct ancestor of HTML, albeit much more advanced and powerful. Mathematics in SGML has always been problematic, however. While an ISO standard exists defining a DTD for mathematics in SGML, it has proven of limited usefulness. Most implementers have found it necessary to extend the ISO DTD fragment by adding additional tags to cover mathematical expressions not well addressed by the

² Caprotti, O. and Carlisle, D. (1999). OpenMath and MathML: Semantic Mark Up for Mathematics, *ACM Crossroads* 6 (2). Online. Available <http://www.acm.org/crossroads/xrds6-2/openmath.html>.

standard. SGML math tags tend to be more presentation oriented than semantic. Rendering engines for SGML are generally stand-alone applications (i.e., not integrated into the Web environment), and the ones capable of rendering mathematics well are quite expensive. In any event such engines usually are customized to a particular implementers version of SGML mathematics.

XML, another, more powerful descendent of SGML, has found much wider acceptance among Web users. Also, a new generation of specialized XML implementations have appeared, including MathML. XML has also proven useful as a transport layer for semantic-based models of mathematical expression such as OpenMath. These approaches try to express, to one degree or another, the meaning of the mathematics as well as it's presentation. They are optimized for computer display presentation rather than for presentation on the printed page. They support robust searching and manipulation of mathematical content, allowing dynamic interaction with mathematics contained in a document. At this time, rendering and presentation engines lag (or at least are not yet ubiquitous), however, a fair amount of effort is now being expended to remedy that situation (e.g., IBM TechExplorer, Design Science Web3EQ, Mozilla support for MathML, etc.). Sci-Tech publishers and scholarly societies have committed to providing the required public-domain font sets and glyphs to support these efforts (e.g., the STIX project). MathML and related XML open standard approaches to publishing mathematics on the Web show great promise, and therefore can't be neglected. (A hope remains that Web browsers will soon be able to render MathML well either directly or through use of a ubiquitous and free plug-in.)

Also of increasing importance are the proprietary formats used by the major mathematical software vendors (e.g., Wolfram, Maple, Design Science). While these vendors are supporting standards like MathML, they continue to develop and support their own specific formats (e.g., Mathematica workbooks) and to provide Web-based services that support the use of these formats in interactive Web applications. While it's not entirely clear that a national Mathematics Digital Library will need to support such proprietary solutions (particularly if the vendors will support translation from such formats to and from open standard formats like MathML), these approaches can't be neglected entirely given their products popularity and the inherent capabilities for sophisticated interactions with the end-user that these formats have.

So, the goal, to paraphrase Donald Knuth when he introduced T_EX in 1979, remains "to communicate mathematics effectively by making it possible to publish mathematical papers and books of high quality, without excessive cost." What's difficult is the details. There are today an abundance of approaches, and it's too early to focus on just one approach. As is common when trying to build a large-scale, multi-purpose digital library application that will include both current and retrospective content, we need to consider and likely support a variety of approaches.