MathJax: A Platform for Mathematics on the Web

Davide Cervone

Over the past three years, a project has been quietly evolving that has important implications for those interested in using mathematical notation within webpages in a way that not only displays that mathematics beautifully but allows it to interact with other applications and environments. That project is MathJax [1], and it is an attempt to provide a universal, industrial-strength, math-on-the-web solution that is standards-based and applicable to a diverse range of audiences. Current users include publishers of large-scale scientific websites, bloggers and social networkers, users of course-management systems, and individual faculty members who just want to post their homework assignments easily online. Under the sponsorship of the American Mathematical Society (AMS), the Society for Industrial and Applied Mathematics (SIAM), and Design Science, Inc. (DSI), with the support of StackExchange, the American Physical Society (APS), Elsevier, the Optical Society of America (OSA), Project Euclid, WebAssign, and others, MathJax is an open-source project, drawing on the talents of a variety of individuals. MathJax builds on the techniques developed by the author as part of his earlier jsMath package [7] and can be considered the "next generation" of that software.

Anyone who has tried to include mathematical notation in a webpage knows that this is not an easy task. The traditional solution is to use images of the equations and link those into the page to represent the mathematics. This is a cumbersome approach that has a number of drawbacks (it is hard to get the images to match the surrounding text, they don't scale or print well, they cannot be easily copied, and so on). The Mathematical Markup Language (MathML) was intended to solve this problem (see [19] and [20]), but for a variety of reasons, more than a decade after its specification was released, most of the major browsers still don't support it.[1] The MathJax project plugs the gap left by a lack of browser support for MathML, making it possible for mathematicians—and the scientific community at large—finally to take advantage of the MathML standard and all it implies.

In the past, images were the only reliable cross-platform way to present mathematical equations within webpages, despite their faults. Recently, however, several web technologies have come together that make it possible to use a different approach that can resolve a number of these issues. The increased speed of JavaScript engines, the standardization of Cascading Style Sheets (CSS) implementations across browsers, the support for unicode fonts, and the ability to obtain fonts over the Web when they are not installed on the user's computer can be harnessed to provide a means of including mathematics in webpages that overcomes many of the deficiencies inherent with the use of images.

MathJax is being developed as a platform for mathematics in webpages that works across all the major browsers (including mobile devices such as the iPad, iPhone, and Android phones). It allows authors to write their equations using several formats, including MathML and TeX, and displays the results using MathML in those browsers that support it or HTML-with-CSS in those that don't. A Scalable Vector Graphics (SVG) output mode is nearing completion, and other input and output formats are possible. For example, a user-written input processor for the ASCIIMathML format [8] is scheduled for inclusion in the next release of MathJax.

Davide Cervone is professor of mathematics at Union College. His email address is dpvc@union.edu.
DOI: http://dx.doi.org/10.1090/noti794

[1] *Historically, only Firefox has had native support for MathML, and Internet Explorer has a plug-in that handles it, but these have required users to install extra components before being able to view the mathematics. Opera has limited MathML capability through CSS styling. Apple's WebKit recently has added support for MathML (though it is not as full-featured an implementation as Firefox's). It is included in Safari 5.1, and eventually we should see it in Chrome and other WebKit-based browsers.*

Publications results for "Contents of: Geometry & Topology [13 (2009), no. 1]" MathJax is on

MR2469512 (2010m:14073)
Lai, Hsin-Hong(1-BRND)
Gromov-Witten invariants of blow-ups along submanifolds with convex normal bundles. (English summary)
Geom. Topol. 13 (2009), no. 1, 1–48.
14N35 (14E05)
PDF | Clipboard | Journal | Article | Make Link

Citations
From References: 1
From Reviews: 0

In this paper the author compares the Gromov-Witten invariants of a projective manifold X with those of the blow-up $\pi\colon \tilde{X} \to X$ of X along a submanifold Z. A connected submanifold $Z \subset X$ is said to be of type I if the normal bundle $N_{Z/X}$ is convex (i.e. $H^1(\mathbb{P}^1, f^*N_{Z/X}) = 0$ for any holomorphic map $f\colon \mathbb{P}^1 \to Z$) and there exists a subbundle $\mathcal{F}$ in $N_{Z/X}$ of rank $\mathrm{rk}(\mathcal{F}) \geq 2$ which is generated by global sections. A connected submanifold $Z \subset X$ is said to be of type II if every holomorphic map $f\colon \mathbb{P}^1 \to Z$ is constant. The first result of the paper (Theorem 1.4) states that if $Z \subset X$ is a submanifold such that each connected component is of type I or of type II then the following equality between the twisted n-pointed genus zero Gromov-Witten invariants of X and of $\tilde{X}$ holds:

$$\langle \alpha_1, \ldots, \alpha_n \rangle^{X,\mathbf{c},V}_{0,n,\beta} = \langle \pi^*\alpha_1, \ldots, \pi^*\alpha_n \rangle^{\tilde{X},\mathbf{c},\pi^*V}_{0,n,\pi^!\beta},$$

where $\alpha_i \in H^*(X)$, $\beta \in H_2(X,\mathbb{Z})$, V is a vector bundle on X and $\mathbf{c}$ is an invertible multiplicative characteristic class. The previous formula is also generalized to include descendant insertions (Corollary 4.8).

Figure 1. MathJax output as it appears in an entry in MathSciNet.

MathJax does not require the viewer to download any software (though it will take advantage of certain locally installed fonts when they are present), and since it uses actual fonts, its output scales and prints better than math presented as images. Because the pages include the original TeX or MathML markup, search engines can index the mathematics within them. Since there are no images to create, the mathematics on the page can be dynamically generated and can include links and other interactive content. The internal format for MathJax is essentially MathML, and the user can obtain either a MathML or a TeX version of the equations on the page, which can be copied and pasted into other documents or programs.

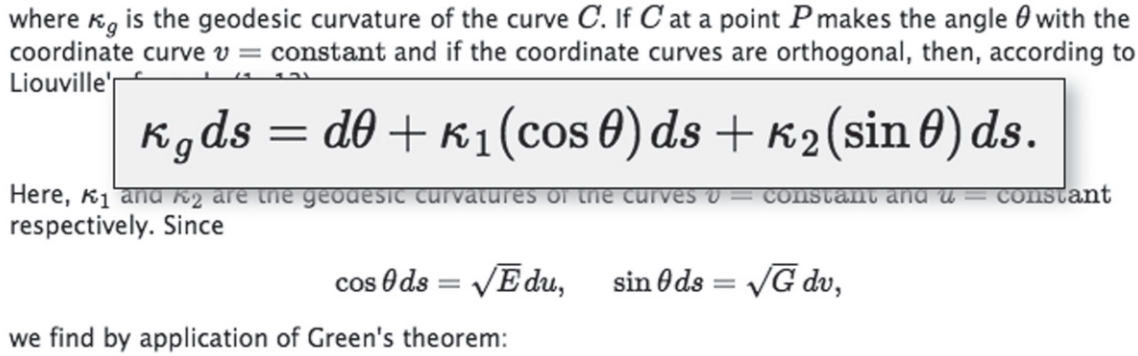
For a user viewing a page that uses MathJax, nothing special needs to be done (provided JavaScript and page-specific fonts are enabled). After the document is opened, MathJax will look through the page for mathematics to typeset. Users can control some settings for MathJax by using the MathJax contextual menu, which can be obtained by right-clicking (in Windows and Linux) or control-clicking (in MacOS) on any typeset equation.[2] The menu allows the user to view the MathML or TeX version of the mathematics, so that it can be copied and pasted into another document, for example. It also allows users to enable a zooming feature for the visually impaired or to select whether to use the browser's native MathML support (if any) for displaying mathematics, rather than MathJax's HTML-based rendering.

It is possible for users to take advantage of MathJax even on pages where the author has not included MathJax explicitly. Using Greasemonkey [9] (a plug-in for Firefox) or one of its work-alikes for other browsers, one can arrange for MathJax to be inserted into existing webpages to render the mathematics. This can be used, for example, to make pages with MathML work in browsers that don't have native support for it, or in some cases, to replace math displayed as images by font-based mathematics generated by MathJax. This has implications particularly for visually impaired users who need assistance reading the mathematics on the screen. With MathJax's zoom and scaling features, these users can get larger versions of the mathematics easily (without the artifacts of enlarged images).

Those who use screen readers to read their webpages can also take advantage of MathJax, as MathJax works together with programs like MathPlayer [10] to turn the mathematics on the page into a form that can be passed to screen readers. When combined with Greasemonkey, this can make a wide range of existing pages accessible to the visually challenged. For example, this process can be used to make much of the mathematics in Wikipedia accessible (see [9]).

If you are a page author and want to use MathJax to typeset the mathematics in your webpages,

[2]*Note that sites can disable the menu if they choose to. This is the case with MathSciNet, for example, so the contextual menu is not available there.*



where κ_g is the geodesic curvature of the curve C. If C at a point P makes the angle θ with the coordinate curve $v =$ constant and if the coordinate curves are orthogonal, then, according to Liouville'

$$\kappa_g ds = d\theta + \kappa_1(\cos\theta)\,ds + \kappa_2(\sin\theta)\,ds.$$

Here, κ_1 and κ_2 are the geodesic curvatures of the curves $v =$ constant and $u =$ constant respectively. Since

$$\cos\theta ds = \sqrt{E}\,du, \qquad \sin\theta ds = \sqrt{G}\,dv,$$

we find by application of Green's theorem:

Figure 2. An equation zoomed by MathJax for greater readability.

```
<!DOCTYPE html>
<html>
  <head>
    <title>MathJax LaTeX Example Page</title>
    <script type="text/javascript"
      src="http://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML">
    </script>
</head>
<body>
   <p>
     When \(a \ne 0\), there are two solutions
     to \(ax^2 + bx + c = 0\) and they are
     \[x = {-b \pm \sqrt{b^2-4ac} \over 2a}.\]
   </p>
</body>
</html>
```

Figure 3. A complete example that uses MathJax to typeset LaTeX mathematics within a webpage. The output can be seen at [3].

you need only include a single tag in your HTML file that loads MathJax into the page. MathJax can be configured either within the HTML file itself or through the use of a configuration file on the server, and many of its features are customizable. Since early 2011 MathJax has been available as a web service through a "cloud" distributed network. That means you don't even have to install MathJax on your own server; simply link your page to the MathJax web service and you can start including mathematics in your pages right away. Because the service resides on a collection of servers around the world, access to MathJax should be fast wherever your readers are located, and they may already have MathJax cached in their browser from viewing previous pages that use it.

You can enter equations into your pages either using TeX and LaTeX notation or as MathML (other input formats may be available in the future). For TeX notation, MathJax will look through the page for math delimiters like \(…\) and $$…$$ to identify the mathematics to be typeset. Note that MathJax implements only LaTeX's math macros and environments, not all the text-mode commands, so you can't simply place a complete TeX file on the Web and have MathJax render it. MathJax expects that the textual formatting will be performed through HTML, not TeX, though there are preprocessors that can convert a TeX file to HTML with the mathematics handled by MathJax (e.g., [11]). On the other hand, MathJax does implement a large subset of the LaTeX commands and environments, including the amsmath symbols and environments. See the MathJax homepage [1] for more details on using it in your own pages and [2] for a list of the LaTeX macros and environments supported by MathJax.

MathJax already has been adopted by a number of important mathematics sites. You may have noticed that it is now being used in MathSciNet [12] and in recent AMS Feature Columns [13], but it is also in use by the *Annals of Mathematics* [14], the MathDL library [15], Project Euclid [16], and in online forums such as MathOverflow [17] and StackExchange [18]. There are plug-ins for a number of popular content-management systems to allow them to use MathJax for their mathematics, and MathJax is ideally suited for use in blogs and wikis, where the content is dynamically generated by the users. A more complete list is available on the MathJax website [5].

As MathJax is an open-source project, you can lend your expertise to it whether you are a programmer, a writer, a tester, or just a user who can help others learn about MathJax; see the community page [6] for details. MathJax continues to develop and improve, and as more groups and individuals support and adopt it, it should provide a rich platform for handling math on the Web, both now and into the future.

References

[1] MathJax homepage, http://www.mathjax.org.

[2] MathJax support for TEX and LATEX, http://www.mathjax.org/docs/1.1/tex.html.

[3] MathJax TEX output example, http://cdn.mathjax.org/mathjax/latest/test/sample-tex.html.

[4] Using MathJax with Greasemonkey, http://www.mathjax.org/docs/1.1/dynamic.html#mathjax-and-greasemonkey.

[5] MathJax in use, http://www.mathjax.org/community/mathjax-in-use/.

[6] MathJax community page, http://www.mathjax.org/community/.

[7] jsMath homepage, http://www.math.union.edu/locate/jsMath.

[8] ASCIIMathML homepage, http://www1.chapman.edu/~jipsen/mathml/asciimath.html.

[9] Greasemonkey homepage, http://www.greasespot.net/.

[10] MathPlayer homepage, http://www.dessci.com/en/products/mathplayer/.

[11] tex4ht homepage, http://www.tug.org/applications/tex4ht/mn.html. (There is a list of other converts at the bottom of this page.)

[12] MathSciNet, http://www.ams.org/mathscinet/.

[13] Feature Columns from the AMS, http://www.ams.org/samplings/feature-column/fcarc-index.

[14] *Annals of Mathematics*, http://annals.math.princeton.edu/.

[15] MAA's MathDL digital library, http://mathdl.maa.org/mathDL/.

[16] Project Euclid, http://projecteuclid.org/.

[17] MathOverflow, http://mathoverflow.net/.

[18] StackExchange, http://stackexchange.com/.

[19] Robert Miner, The importance of MathML to mathematics communication, *Notices of the AMS* **52**, no. 5, 2005.

[20] Ralph Youngen, Toward a mathematical markup language, *Notices of the AMS* **44**, no. 9, 1997.

Notices Interview with Davide Cervone, Chief Developer of MathJax

Notices: *In your article you mention that MathML on its own never took off. Can you say more about what went wrong?*

Cervone: Actually, I think I said that it did not have wide support in Web browsers, which is somewhat different. It turns out that MathML is used quite heavily in publishing workflows and as an interchange format for mathematical content. Even with image-based websites, the data behind the scenes might have been MathML originally. It would not make it directly into webpages, however, because you could not rely on your viewers having MathML access in their browsers and because until recently you had to use XHTML (which is far less forgiving than HTML) in order for it to work. Making matters worse, the two major MathML viewers (Firefox and MathPlayer in Internet Explorer) required the content to be delivered in incompatible ways, so servers had to be set up to detect the browsers and customize the pages to suit them. This was cumbersome and unreliable and required too much technical expertise for general use.

It appears that the future looks much brighter for MathML than the past, however. The recent incorporation of native MathML support in WebKit (available in Safari 5.1 and above) suggests that MathML will become much more widely implemented, as WebKit underlies many popular Web browsers including those used on mobile devices. Moreover, MathML has been incorporated into HTML5 (making it much easier to use it in webpages) and in the latest EPUB specification for electronic books. This has caused renewed interest in MathML for display purposes. Apple's iBook application can process eBooks that include MathJax to render mathematics, as can Calibre, which is available on Mac, Windows, and Linux. I expect to see others in the near future that can do the same. In fact, I know of one new eBook reader that includes MathJax within its own code in order to implement the MathML portion of the EPUB3 specification.

After fifteen years of waiting, it seems that the scene is finally set for MathML to begin to fulfill its potential as the means of representing and viewing mathematics in a wide range of settings. I like to think that MathJax contributes to that, as it makes MathML available today, in all modern browsers, allowing mathematical content in MathML to be developed now so that we can take advantage of the new document formats and applications as they become available.

Notices: *You say also that certain technical innovations made MathJax feasible. Can you say more about what went right?*

Cervone: It may sound trite, but one of the most important steps was the development of standards that the browser manufacturers actually implemented (largely the work of the W3C and its working groups). In the early days of Web development, there was a proliferation of innovations and ideas about how the Web should evolve, and each browser had its own set of extensions, incompatible with other browsers. Anyone doing Web development in those days will tell you how difficult that made things. In recent years, there has been a coalescing of the best of these ideas, and while there are still differences to contend with, you can now rely on a uniformity of results that was not possible in the past.

Great strides have been made by the browser manufacturers in standardizing their implementations of Cascading Style Sheets (CSS) and in the Document Object Model (DOM) and the methods for manipulating it via JavaScript programs. These are two key ingredients that make MathJax possible. Browser vendors have also been hard at work optimizing their JavaScript engines, and that makes them fast enough and efficient enough to handle the heavy load that MathJax puts on them.

Another critical piece is font support. Mathematics involves a wide range of specialized symbols, and traditionally there was no uniform way of specifying such symbols. The development of Unicode, which provides a standardized means of identifying an enormous range of characters (from musical notation to Canadian aboriginal syllabics), including hundreds of mathematical and technical symbols, makes it possible to specify mathematical notation in a reliable, font-independent way. The adoption of this standard by all modern browsers means MathJax can refer unambiguously to the mathematical symbols it needs.

Being able to indicate a symbol is not quite enough, however. No font includes all the hundreds of thousands of characters in the Unicode specification, and it turns out that very few include the mathematical symbols, so there is no guarantee that an appropriate font will be available to MathJax on a reader's computer. In the past, this would have required users to download and install fonts, and that would have made MathJax impractical. In the last two years, however, one final technical advance has become available widely enough to solve this problem. Browsers now have the ability to download a font over the Web when it is not available on the user's computer already, and MathJax takes advantage of this by providing a set of mathematical fonts (based on TEX's Computer Modern fonts) that are loaded on demand when required. This makes it possible for MathJax to work without any installation on the part of the user, as all its resources can be obtained via the Web as they are needed. This last step is probably what made it possible for MathJax to be as successful as it is.

Notices: *How was the development of MathJax apportioned among various people and/or groups? Along what time line?*

Cervone: So far, MathJax has been nearly four years in the making, and it is based on my experience with an earlier program called jsMath that I developed initially in 2004; but MathJax is a complete rewrite from the ground up. As far as the programming goes, that is pretty much entirely mine, but as an open source project, anyone has the potential to contribute.

MathJax has had major support from groups like the AMS, SIAM, the APS, and the other sponsors mentioned in the article. MathJax would never have happened without their contributions and their readiness to incorporate MathJax into their own websites. Of course, special recognition is due to Design Science, Inc., who administers the MathJax project. Robert Miner has been indispensable in his role as both project leader and a collaborator in the design decisions and the vision for MathJax. Sean Hogan has done important work in performance analysis, and Frédéric Wang has developed the test suite for MathJax (now comprising thousands of individual tests), which has been crucial to the reliability and quality of MathJax. Hylke Koers has been tireless in his promotion of MathJax within the publication industry and in recruiting sponsors for the project.

There are a number of groups that are building applications around MathJax (like equation editors, eBook readers, and e-learning sites), taking advantage of MathJax's programmer's interface. As MathJax gains wider use, we hope to receive contributions from those types of users back to the MathJax project. Indeed, we are beginning to see this happening with several major extensions being developed by the user community and being made available for others to use. The beauty of an open source application is that it grows and matures based on the experiences of its users.

Robert Miner, the director of the MathJax project at Design Science, Inc., passed away on December 6th from liver cancer. He is survived by his wife, Emily, and his son, Bill. Robert was one of the guiding lights for mathematics on the Web. His contributions to the field began in the mid 1990s with his WebEq software, originally developed at the Geometry Center in Minneapolis, MN, and later acquired by Design Science, Inc. At Design Science he oversaw the MathType and MathPlayer products, and most recently the MathJax initiative. Robert was co-chair of the World Wide Web Consortium (W3C) working group that developed the MathML standard, and was tireless in promoting its use in mathematical publications, both electronic and in print. He spoke often at conferences, and wrote extensively on the use of MathML to enhance mathematical communication.

Without Robert Miner, MathJax likely would not exist. He helped put together the consortium that funded the initial development of MathJax, and his vision and design insight were critical to the success of the project. Robert was one of those who could take all the arguments surrounding a contentious issue and synthesize them into a coherent whole that focused everyone on the right direction. He had a keen sense of humor, a sharp intellect, a deep sense of honor, and a personal philosophy that took great joy in living. His leadership, his energy, and his camaraderie will be missed; we are impoverished by his loss.

—Davide Cervone