

MathML Improvements in WebKit

Jul 13, 2016

In a [previous blog post](#), I explained the work made by [Igalia's](#) web platform team to refactor WebKit's MathML layout classes. I stated that although some rendering improvements were a nice side effect, the main goal of [the first phase](#) was really to clean the code up so that it is easier for developers to work on MathML in the future. Indeed this *really* made things easier to review: Quite unexpectedly to me, [the second phase](#) only took 4 days to be upstreamed... Kudos to [Brent Fulgham](#) for having reviewed so many patches in such a short period of time!

In this blog post, I am going to give an overview of the improvements made during these two first phases taking [changeset r203109](#) as a reference. The changes will be available in [WebKitGTK+ 2.14](#) in September and are likely to be included this month in the next [Safari Technology Preview](#). It definitely remains more work to do such as [the third phase](#) or other rendering improvements, but I believe we have already made a big step forward!

Mathematical Fonts

Two years ago, basic support for operator stretching via the OpenType MATH table was added to WebKit. During the refactoring, we improved that support and also made use of more parameters to improve the math layout (see section about OpenType MATH parameters below). While [Latin Modern Math](#) will be used in most screenshots, the following one shows that you can use any [math fonts](#). By default WebKit will try and use one of these fonts but if none are available or if you force a non-math `font-family` then the rendering quality may not be good.

The following screenshot gives the rendering for various fonts. For the last one we used the value `sans-serif` to illustrate issues with non-math fonts (`displaystyle` integral too small, mathvariant italic glyphs taken from another font, missing italic correction etc).

$$\int_{-\infty}^{+\infty} \frac{16\sin(2x + \pi)}{5(x^2 + 2x + 5)^2} dx = \frac{\pi\sin(2)}{e^4}$$
$$\int_{-\infty}^{+\infty} \frac{16\sin(2x + \pi)}{5(x^2 + 2x + 5)^2} dx = \frac{\pi\sin(2)}{e^4}$$
$$\int_{-\infty}^{+\infty} \frac{16\sin(2x + \pi)}{5(x^2 + 2x + 5)^2} dx = \frac{\pi\sin(2)}{e^4}$$
$$\int_{-\infty}^{+\infty} \frac{16\sin(2x + \pi)}{5(x^2 + 2x + 5)^2} dx = \frac{\pi\sin(2)}{e^4}$$

$$\int_{-\infty}^{+\infty} \frac{16\sin(2x + \pi)}{5(x^2 + 2x + 5)^2} dx = \frac{\pi \sin(2)}{e^4}$$

Integral obtained by [contour integration](#)

WebKitGTK+ r203109 with various font-family values.

From top to bottom: [TeX Gyre Schola](#), [Latin Modern](#), [DejaVu Math](#), [STIX](#), and `sans-serif`.

The href Attribute

This new feature is obvious: You can now create a hyperlink for any part of a mathematical formula! Here is a screenshot of the MathML Torture Test 21 with additional links, as displayed in WebKit r203109. Click the image to load the test case in your browser and test it.

$$\sum_{p \text{ prime}} f(p) = \int_{t > 1} f(t) d\pi(t)$$

The mathvariant Attribute

Unicode contains [Mathematical Alphanumeric Symbols](#) to convey special meaning such as [double-struck](#) or specific [Arabic styles](#). Characters for these symbols are generally provided by [math fonts](#). In MathML, mathematical variables are automatically rendered using the italic characters from this Unicode block. One can also access these characters via the `mathvariant` attribute and that attribute is actually used by many LaTeX-to-MathML converters.

In the following screenshot, you can see that the letters f , x and y are now drawn with this special mathematical italic glyphs and that WebKit uses the conventional fraktur style for the Lie algebra \mathfrak{g} . Note that the prime is still too small because WebKit does not make use of the [ssty feature](#) yet.

$$f: \mathfrak{g} \rightarrow \mathfrak{g}', \quad f([x, y]) = [f(x), f(y)],$$

$$f: \mathfrak{g} \rightarrow \mathfrak{g}', \quad f([x, y]) = [f(x), f(y)],$$

Homomorphism of [Lie algebra](#)

Top: Safari 9.1.1. Bottom: Safari r203109.

Operators and Spacing

As said in my previous blog post, the rendering of large and stretchy operators have been rewritten a lot and as a consequence the rendering has improved. Also, I mentioned that [the width of operators may depend on their height](#). This may cause accumulated approximations during the computation of preferred widths. The old flexbox-based

implementation [incorrectly forced layout during preferred computation](#) to avoid that but a quick workaround for that security concern caused the approximate preferred widths to be used for the logical widths. With our new implementation, the logical width is now correctly calculated. Finally, we added partial support for the `mpadded` element which is often used to tweak spacing in mathematical formulas.

The screenshot below illustrates the fix for a [serious regression with large operator](#) (summation symbol) as well as improvements in the rendering of stretchy operators (horizontal braces). Note that the formula has a hack with a zero-width `mpadded` element which used to cause improper spacing (large gap between the group of a's and the group of b's).

$$\sum_{p \text{ prime}} f(p) = \int_{t > 1} f(t) d\pi(t) \quad \sum_{p \text{ prime}} f(p) = \int_{t > 1} f(t) d\pi(t)$$

$$\left\{ \overbrace{a, \dots, a}^{k \text{ a's}}, \overbrace{b, \dots, b}^{\ell \text{ b's}} \right\}_{k + \ell \text{ elements}} \quad \left\{ \overbrace{a, \dots, a}^{k \text{ a's}}, \overbrace{b, \dots, b}^{\ell \text{ b's}} \right\}_{k + \ell \text{ elements}}$$

Tests 21 and 22 from the [MathML torture test](#)

Left: Safari 9.1.1. Right: Safari r203109.

The following screenshot shows how incorrect width computations used to cause excessive spacing after the stretchy radical and slash symbols:

$$\sin 1^\circ = \frac{\sqrt[3]{z} - 1 / \sqrt[3]{z}}{2i},$$

$$\sin 1^\circ = \frac{\sqrt[3]{z} - 1 / \sqrt[3]{z}}{2i},$$

[Sine of 1 degree](#)

Top: Safari 9.1.1. Bottom: Safari r203109.

The `displaystyle` Property

Mathematical formulas can be integrated inside a paragraph of text (inline math in TeX terminology) or displayed in its own horizontally centered paragraph (display math in TeX terminology). In the latter case, the formula is in `displaystyle` and does not have any restrictions on vertical spacing. In the former case, the layout of the mathematical formula is modified a bit to optimize this vertical spacing and to better integrate within the surrounding text. The `displaystyle` property can also be set using the corresponding attribute or can change automatically in subformulas (e.g. in fractions).

In the following screenshot the fix for the large operator regression is obvious but you can also notice that the summation is now slightly different for the definition of a Bézier curve (top) and for the one of a rational Bézier curve (bottom). For example, to save some vertical space in the fractions, the sigma symbol is drawn smaller and the scripts attached

to it are moved on its right. However, the script size could still be improved when we implement the [scriptlevel](#) property.

$$\mathbf{B}(t) = \sum_{i=0}^n b_{i,n}(t) \mathbf{P}_i, \quad \mathbf{B}(t) = \sum_{i=0}^n b_{i,n}(t) \mathbf{P}_i,$$

$$\mathbf{B}(t) = \frac{\sum_{i=0}^n b_{i,n}(t) \mathbf{P}_i w_i}{\sum_{i=0}^n b_{i,n}(t) w_i}, \quad \mathbf{B}(t) = \frac{\sum_{i=0}^n b_{i,n}(t) \mathbf{P}_i w_i}{\sum_{i=0}^n b_{i,n}(t) w_i}$$

Definitions of Bézier Curve

Left: Safari 9.1.1. Right: Safari r203109.

OpenType MATH Parameters

Many new [OpenType MATH features](#) have been implemented following the [MathML in HTML5 Implementation Note](#):

- Use of the `AxisHeight` parameter to set vertical position of fractions, tables and symmetric operators.
- Use of layout constants for radicals (some of them were already used), scripts and fractions. This improves math spacing and positioning and allows to adjust them according to value of the `displaystyle` property discussed in the previous section.
- Use of the italic correction of large operator glyph to set the position of subscripts.

The screenshots below illustrate some of these improvements. For the first one, the use of `AxisHeight` allows to better align the fraction bar with the plus sign. For the second one, the use of layout constants for scripts as well as the italic correction of the integral improves the placement of limits.

$$x = b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2 + \frac{a_3}{b_3 + \frac{a_4}{b_4 + \ddots}}}}$$

$$x = b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2 + \frac{a_3}{b_3 + \frac{a_4}{b_4 + \ddots}}}}$$

Generalized Continued Fraction

Top: Safari 9.1.1. Bottom: Safari r203109.

$$\Gamma(t) = \int_0^{\infty} x^{t-1} e^{-x} dx.$$

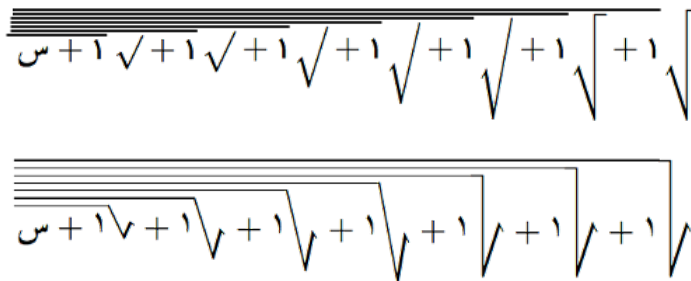
$$\Gamma(t) = \int_0^{\infty} x^{t-1} e^{-x} dx.$$

Definition of the Gamma Function

Top: Safari 9.1.1. Bottom: Safari r203109.

Right-to-left radicals

One of the advantage of the old flexbox-based implementation is that right-to-left layout was available for free. This support has of course been preserved in the new implementation. We also added a simple workaround to mirror radicals using a scale transform as shown in the screenshot below. However, general [glyph-level mirroring](#) is still missing.



Test 13 from the [MathML torture test \(Machrek Style\)](#)

Top: Safari 9.1.1. Bottom: Safari r203109.

Conclusion

Igalia's web platform team has been able to follow the [MathML in HTML5 Implementation Note](#) in order to significantly improve the rendering of mathematical expressions in WebKit. More work remains to do but we will definitely appreciate any feedback that can help improving native MathML support in web engines. We are also excited to continue work and collaboration at the next [Web Engines Hackfest!](#)