

Preparing Accessible Math Documents using MathJax v3

Volker Sorge

School of Computer Science
University of Birmingham
cs.bham.ac.uk



Progressive Accessibility
Solutions
Birmingham, UK
progressiveaccess.com



MathJax
Consortium
mathjax.org



Accessing Higher Ground 2019, Westminster, 21 November
2019

Slides and material at: <http://progressiveaccess.com/ahg19>

- 1 Quick Intro
- 2 Use Pandoc to get math documents on the Web
- 3 Use MathJax accessibility extension
- 4 Customise MathJax and the a11y extension

A Quick Introduction

- Learn two important open source tools: Pandoc and MathJax
- Transform documents using PanDoc
- Access Mathematics using MathJax's accessibility extension
- Learning outcomes:
 - Know how to convert your own documents
 - Be familiar with MathJax's a11y extension
 - Understand MathJax options to customise the accessibility experience
 - Be able to teach others!

What is MathJax?

- MathJax is a JavaScript library for rendering Mathematics in all browsers
- Can take \LaTeX , AsciiMath, and MathML as input
- Generates browser output, e.g. HTML/CSS, SVG
- Standard Maths rendering solution for: stackexchange, wordpress blogs, mediawiki, etc.

MathJax is the de facto rendering solution of (nearly) all Mathematics on the web (in 2016: 35 million unique daily rendering requests via CDN)

<http://www.mathjax.org>

What is Pandoc?

- Pandoc is a flexible multi-format transformer between document formats
- It supports the transformations of most common sources for Maths document:
L^AT_EX, Word, Markdown
- It allows for the direct integration of MathJax (v3 soon)

<http://pandoc.org>

- Note
 - There are a number of other document transformers out there
 - Many source dependent, i.e., one-source-to-many
 - Pandoc is really many-to-many, open source and easy to use in a pipeline

Getting a Document on the Web

- Get familiar with PanDoc
- Create or download a math document
- Translate document into HTML format
- Run it locally in a browser

What you should have available

- Pandoc binary
 - Download it from <https://progressiveaccess.com/empower19/pandoc> (for Linux) or <https://progressiveaccess.com/empower19/pandoc.exe> (for Windows)
 - Or alternatively from <https://pandoc.org>. This is an older version, though.
 - You can build directly from <https://github.com/jgm/pandoc>. But this takes a long time!
- Shell
 - On Windows: Powershell is best, Command Line will do.
 - On Linux/MacOSX: any shell or terminal is fine
- Firefox or Chrome
- Screen Reader
- ASCII text editor: NotePad, vi, Emacs, jedit, etc.

Getting familiar with Pandoc

- Open your shell/command line.
- Make sure you know your path to the pandoc executable
- We assume that it is at PATH
- Run the command below, which should display pandoc's options.

```
${PATH}/pandoc —help
```

Getting started with Pandoc

- Let's convert our first file: We by converting \LaTeX into HTML
- Use your favourite \LaTeX file or download <https://progressiveaccess.com/empower19/example1.tex>

```
pandoc --standalone -t html -o example1.html example1.tex
```

- Inspect the options:
 - `-t` is the target format. In our case HTML.
 - `-o` is followed by the output filename
 - `--standalone` ensures that pandoc produces a fully HTML file not just an HTML fragment.

Converting a Word Document

- Similar to \LaTeX we can also convert word documents containing mathematics
- If you have one, try your own document.
- Alternatively download the example file at <https://progressiveaccess.com/empower19/examples/example2.docx>

```
pandoc --standalone -f docx -o example2.html example2.docx
```

- A note on the options:
 - Here we use the `-f` option which specifies the input format.
 - Neither `-f` or `-t` are strictly necessary if the formats can be deduced from the file extensions

Inspecting the Output

- Both conversions will have produced some noise on the console.
- Let's inspect the output:
 - Load `example1.html` or `example2.html` into a browser
 - You will note that some of the Math looks somewhat ugly and some is raw TeX
- pandoc tries to translate Math as much as possible, but often fails

Introducing MathJax

- Let's provide a rendering solution for the math
- We could add it to the html files directly to
- Alternatively we can ask pandoc to do it for us
- Example:

```
pandoc --standalone --mathjax -o example2.html example2.docx
```

```
pandoc --standalone --mathjax -o example1.html example1.tex
```

- Note that we have omitted the from and to options.
- Instead we now have a `--mathjax` option.

Inspecting the Output Again

- pandoc should have produced a lot less noise on the console.
- That's because it has a rendering solution and does not bother to translate the Math itself.
- Let's inspect the output in the browser again.
- All the math should now render nicely
- It's still not accessible, but we will get to that later

Inspecting the Code

- Load the html files into an ASCII editor
- Note, do *not* load them into an office suite like Word, LibreOffice, ...
- Inspecting the code, you can now see the MathJax script tag

```
<script src="https://cdn.jsdelivr.net/npm/mathjax@3/es5/tex-mml-c  
type="text/javascript"></script>
```

- The script tag ensures that MathJax is loaded into the page
- It is loaded directly from its Content Distribution Network
- It is loaded directly from the content distribution network
- It uses a default configuration, for input and output
- Specifically it allows
 - \LaTeX or MathML input
 - HTML/CSS output
- Other potential output is for example SVG
- For other configurations that can be provided as default see:
<http://docs.mathjax.org/en/latest/web/components/combined.html>

Preparing HTML Directly

- Let's prepare our own HTML document with Maths
- Write a HTML document
- Add you favourite maths
- Load it into the browser to see if it renders
- **Please do this as an exercise**

- 1 Build a web document for with your favourite maths.
- 2 If you don't have any then use the quadratic formula:

$$\backslash[x = \frac{-b \pm \sqrt{b^2-4ac}}{2a} \backslash]$$

- 3 Proceed in three steps
 - 1 Create a skeleton HTML document
 - 2 Add the MathJax script tag to the header
 - 3 Add some LaTeX to the body

Hint: Basic web document

```
<html>
  <head>
    ... MathJax script tag goes here ...
  </head>

  <body>
    ... Quadratic equation goes here ...
  </body>
</html>
```

```
<html>
  <head>
    <title>Empower-19 Exercise</title>
    <script src="https://cdn.jsdelivr.net/npm/mathj"
  </head>

  <body>
    <h1>Quadratic Equation</h1>
    \[
    x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}
    \]
  </body>
</html>
```

Using and Customising MathJax

- Short overview of a11y features
- Intro to the MathJax context menu
- Starting and working with the a11y extension
- Customising the experience
 - In the content
 - As a user
- An overview on server side rendering

- Inbuilt and optional accessibility features
- Selectable in context menu since MathJax v2.7
- Much improved in MathJax v3.0

More details at:

<http://docs.mathjax.org/en/latest/basic/a11y-extensions.html>

See also (based on v2.7): <https://www.mathjax.org/mathjax-accessibility-extensions-v1-now-available/>

MathJax's AT Features

- Interactive Exploration
- Magnification
- Highlighting
- Speech Generation in multiple rule sets
- Optional Braille Output
- Responsive Equations and Abstraction

- “Walkers” allow to interactively dive into mathematical expression
- Synchronised highlighting together with aural rendering
- Use `<enter>` to explore expressions
- Simple navigation model using arrow keys
- Different types of walkers: syntactic, semantic
- Interactive collapse and expansion of sub-expressions

Aural Rendering and Highlighting

- Speech strings are computed with Speech Rule Engine (SRE)
- Currently uses the MathSpeak rules: verbose, brief, superbrief
 - special summarisations for collapsed parts
 - Other rule sets and localisations in the future
- WAI-ARIA and CSS to implement interactive exploration
 - Speech output by updating ARIA live regions
 - Colour/contrast changes by rewriting CSS properties
- Speech strings can be precomputed or generated on the fly
- Works with all platforms, browsers, and screen readers supporting live regions

Version 3: Larger Choice of Rule Sets

- MathSpeak (already in v2)
 - Very strict disambiguation with three verbosity levels
- ClearSpeak (sponsored by Mozilla foundation)
 - Less disambiguation, more geared towards highschool and undergraduate level Math
- Nemeth Braille (sponsored by Big Ten Academic Alliance)
 - Can be switched on optionally
- Domain specific rules for Chemistry, Physics, Logic

Magnification

- Zoom feature for single math expression
- Magnification up to 500%
- Option selectable in context menu
- Simple triggered by mouse and keyboard
- Goal directed zoom during interactive exploration

Responsive Equations

- Responsive design enhances reflow and readability of math documents
- Automatic reflow for simplifying layout, adapting to form factor of display and magnification
- Intelligent linebreaking by exploiting semantic enrichment
 - Don't break in the middle of an expression
- Chunking: Abstracting over large elements
 - collapsing mathematically meaningful sub-expressions

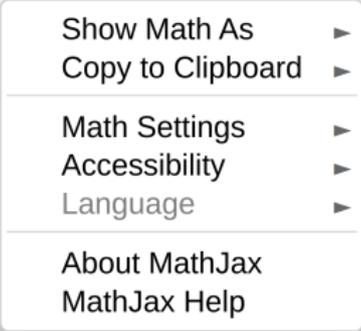
The Context Menu

- The A11y extension is switched on using the context menu
- Each math expression in the page will have a context menu attached
- The context menu is fully keyboard accessible
- You can <TAB> between math expression in a page. This allows you to easily get focus.

Opening the Context Menu

- Focus a math expression on your page (use <TAB>) if necessary.
- It can be opened with
 - Right-clicking on the expression
 - Pressing <SPACE> when focus is on the expression

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- 
- Show Math As ▶
 - Copy to Clipboard ▶

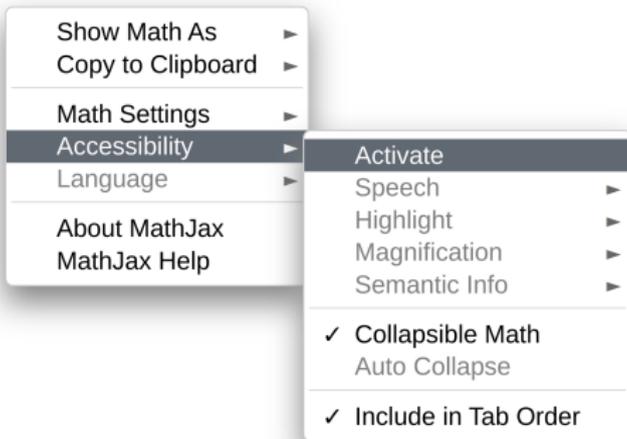
 - Math Settings ▶
 - Accessibility ▶
 - Language ▶

 - About MathJax
 - MathJax Help

Switching on Accessibility

- Navigate down to the *Accessibility* submenu entry
- Open the submenu entry
- Hit the *Activate* command

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



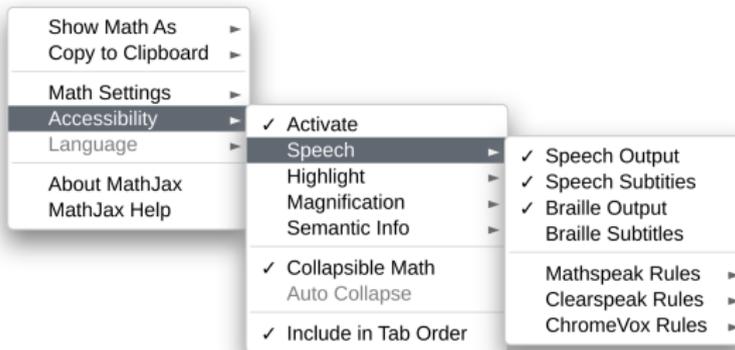
Switching on Accessibility

- Once activated the a11y extension is loaded.
- Depending on the size of the page and the number of expressions this can take a couple of seconds.
- The extension initially loads a default setting that allows for
 - Interaction
 - Speech output (with subtitles)
 - Braille output (hidden from view)
 - Basic highlighting
- More options are available in the accessibility submenu

Investigating the Accessibility Menu

- To investigate the Accessibility submenu open the context menu again.
- Go to Accessibility and open the submenu
- There you can see submenus for the main features: Speech, Highlight, Magnification, and Semantic Info
- Each of the submenus has entries to control these options
For example, the Speech submenu contains also submenus to change the rules to customise how the Math is spoken.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



Starting the Explorer

- Navigate to a math expression
- Press <ENTER> to start the explorer

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

x equals StartFraction negative b plus-or-minus StartRoot b squared minus 4 a c EndRoot Over 2 a EndFraction

- You should now hear the expression spoken by your screenreader
- Alternatively, you can read the subtitles
- If you have a Braille display attached you should get Nemeth Braille feedback.

Starting the Explorer

NOTE:

Depending on the implementation quality of the particular browser/screenreader/OS combination (especially Chrome and IE), users might have to disable screenreader reading modes (e.g., “browse mode” in NVDA, “virtual cursor” in JAWS) before being able to launch the MathJax explorer application.

Exploring the Expression

- You can now explore the expression with the arrow keys
 - <Down> Explore next lower level in the sub-expression tree.
 - <Up> Move up the sub-expression tree.
 - <Right> Navigate the expression horizontally right on the level.
 - <Left> Navigate the expression horizontally left on the level.
- <Escape> Lets you leave exploration mode.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

x equals the fraction with numerator negative b plus or minus the square root of b squared minus 4 a c and denominator 2 a

Keyboard Navigation

- There are a number of other keyboard commands that let you explore an expression.
- For example:
 - `<Space>` Gives you positional information.
 - `<X>` Summarise the expression under cursor.
 - `<Tab>` Repeats previous speech-text.
 - `<Home>` Navigate directly to top-most level of expression.
- For a comprehensive list of commands see <http://docs.mathjax.org/en/latest/basic/explorer-commands.html>

Choice of Rule Sets

- SRE implements a number of rule sets
- Many are exposed in MathJax. Primarily:
 - MathSpeak: A very precise but verbose set
 - ClearSpeak: More natural language, but less disambiguating
- In addition each rule set has a number of styles or preferences (in case of ClearSpeak).

Switching Rule Sets by Keyboard

- `<>>` Switch rule sets between MathSpeak and ClearSpeak, if both are available for the current locale.
- `<<>` Cycle styles or preferences for the currently active rule set.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

StartRoot b squared minus 4 a c EndRoot

Switching Rule Sets in the Page

- Initially MathJax returns MathSpeak for an expression
- While you can change this by keyboard, whenever you reload the page and start the explorer on an expression
- We now want to customise MathJax to always initially return ClearSpeak
- For this we need to first learn how to customise MathJax in the page

The MathJax Configuration Object

- MathJax can be customised in the page content via JSON object
- It has to be initialised in a `script` tag **before** the CDN call
- It allows to customise the various MathJax modules
- You can also set a large number of options. These are divided again by the module they belong to.

Exercise: Set Default Speech Rules

- Open you HTML file in an editor
- Add the following script tag

```
<script type="text/javascript">
  MathJax = {
    options: {
      a11y: {
        speechRules: 'clearspeak-default'
      }
    }
  };
</script>
```

Exercise: Set Default Speech Rules (Ctd.)

- Effectively this passes an option to the a11y module of MathJax
- For more options see the MathJax documentation at: <http://docs.mathjax.org/en/latest/options/accessibility.html#accessibility-options>
- Now reload the document
- Start the explorer
- You should now hear ClearSpeak
“x equals the fraction with numerator. . .”

- Users can still overwrite these options with their personal preferences using the context menu.
- Open the context menu
- Go to: Accessibility > Speech > Mathspeak Rules > Verbose
- Now reload the page
- Start the explorer
- You should now hear MathSpeak
“x equals StartFraction negative b . . .”

- MathJax uses a cookie to remember menu settings
- Anything explicitly set in the menu will always overwrite settings made by the content/page author.
- Thus a user can always retain their personal setting
- They are retained until cookies for a page are deleted

Abstracting Subexpressions

- Go to: Accessibility > Speech > Collapsible Math
- This allows you to abstract sub-expressions
 - With the mouse by left-clicking on it
 - With the keyboard by hiding <->
- You can see which subexpression can be abstracted
 - as the mouse pointer changes on hover
 - when pressing <SPACE> during exploration it is announced
- In the quadratic equation the fraction and the square root can be abstracted
- Speech is adapted to a summary, accordingly.

$$x = \left\langle \left| \right. \right\rangle$$

$$x = \frac{-b \pm \left\langle \sqrt{} \right\rangle}{2a}$$

Note, speech output currently only works reliably in Mathspeak.

We now want to customise MathJax to do the following:

- Enter an inline Math expression of the form:

$$ax^2 + bx + c = 0$$

- Generate SVG rather than CSS/HTML
- Change the highlighting to:
 - Background black and fully opaque
 - Textcolor yellow
- Or alternatively the speech to Mathspeak with superbrief style.

Hint: What needs changing?

- Load `tex-mml-svg.js` instead of `tex-mml-cthtml.js`
- Tell the `tex` module to use

```
inlineMath: [['$', '$']]
```

- Pass the following options to the `a11y`:

```
foregroundColor: 'yellow',  
backgroundColor: 'black',  
backgroundOpacity: 1
```

Solution Simple Exercise

```
<html>
  <head>
    <script type="text/javascript">
      MathJax = {
        options: {
          ally: {
            speechRules: 'clearspeak-default',
          }
        }
      };
    </script>
    <title>Empower-19 Exercise</title>
    <script src="https://cdn.jsdelivr.net/npm/mathjax@3/es5/tex-mml-autoload.js" type="text/javascript"></script>
  </head>

  <body>
    <h1>Quadratic Equation</h1>
    \[
    x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}
    \]
  </body>
</html>
```

Solution Advanced Exercise

```
<html>
  <head>
    <script type="text/javascript">
      MathJax = {
        tex: {
          inlineMath: [['$', '$']]
        },
        options: {
          ally: {
            foregroundColor: 'yellow',
            backgroundColor: 'black',
            backgroundOpacity: 1
          }
        }
      };
    </script>
    <title>Empower-19 Exercise</title>
    <script src="https://cdn.jsdelivr.net/npm/mathjax@3/es5/tex-mml-svg.js" type="text/
  </head>

  <body>
    <h1>Quadratic Equation</h1>
    \[
    x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}
    \]
  </body>
</html>
```

Serverside Rendering and Speech

- MathJax comes also as a node module
- Likewise Speech-Rule-Engine (SRE) comes as a node module
- This allows you serverside rendering of Math or speech output
- Inclusion into your own projects
- Extension and customisations
- Particularly useful for static pages (e.g. with SVG), ePubs etc. containing speech annotations