My notes on how to use Tex4ht+MathJax

Victor Kozyakin

January 4, 2024

1 Call Command

My problem was to make a HTML file with plenty of mathematics from a LaTeX one. As a beginner, I immediately faced the following problems:

- How to run Tex4ht to get HTML file with mathematics?
- What to do to make proper referencing of mathematical formulas?
- How to cope with TikZ figures?

After a while I discovered that the most suitable way for me is to join Tex4ht with MathJax. And the simplest way was to run the following command to process `test.tex` file:

```
make4ht -s test.tex "myconfig" " -cunihtf -utf8"
```

and after that, if one wants to embed into resulted `test.html` the css-file `test.css` generated during previous command, one should issue one the following command

```
htlatex test.tex "myconfig" " -cunihtf -utf8"
```

or once more

```
make4ht -s test.tex "myconfig" " -cunihtf -utf8"
```

where the config file `myconfig.cfg` is as follows

```
\Preamble{xhtml,html5,0,mathjax,p-indent,charset=utf-8,css-in,fn-in}
\textwidth=2160pt

\Css{body { margin:5\% 5\%; max-width:72em; font-size:large; padding:0 40px;}}
\Css{p.indent {text-indent:1.5em;}}
\Css{.columns-3 p.indent {text-indent:0em;}}
\Css{p.bibitem-p { text-indent: 1.5em; margin-left: 2em; margin-top:0em; margin-bottom:0em;
    background:\#F0F0F0; color:\#000000;}}

\Configure{@HEAD}{\HCode{
<script>
window.MathJax = {
  tex: {
    tags: "ams",
    processEscapes: true,
    processEnvironments: true,
    packages: ['base', 'color', 'ams', 'boldsymbol', 'newcommand', 'verb']
  },
  loader: {
    load: ['[tex]/color', '[tex]/ams', '[tex]/boldsymbol', '[tex]/newcommand', '[tex]/verb']
  }
```

```
};
</script>\Hnewline
}}

\def\eqref#1{$\mathrm{(\ref{#1})}$}

\begin{document}

\EndPreamble
```

2 How to reference equations in TeX4ht+MathJax

Unfortunately, it turned out that reference in conjunction TeX4ht+MathJax works well when they referenced sections, subsections and other structure element that are in **text mode**, but when you are trying to reference the label of equation you are getting ???.

The problem is turned out to be rather easily solvable: to reference labels of equations, align or other things in **math mode** you should put the calling `\eqref` or `\ref` in a **math environment**, e.g. by surrounding them by `$`'s or `\(...\)`. Another way is to redefine the command `\eqref` in order that it will be invoked in math mode automatically (see the appropriate string in the config file `myconfig.cfg`.

So, for the LaTeX code below

```
\begin{equation}
\boldsymbol{f}(x)=1\label{eq}
\end{equation}

\[
1\neq1. \tag{OneIsNotOne Condition}\label{E:mycond}
\]

Here, the reference to \tag{OneIsNotOne Condition} in previous
equation is as follows: \eqref{E:mycond}

\begin{align}
a&=1\label{A}\\
b&=0\label{B}
\end{align}
Example of references: we have equation \eqref{eq} from Sec. \ref{S1}. Or $\ref{A}$-\eqref{B}.
```

we obtain the following output:

$$\boldsymbol{f}(x) = 1 \tag{1}$$

$$1 \neq 1. \tag{OneIsNotOne Condition}$$

Here, the reference to `\tag{OneIsNotOne Condition}` in previous equation is as follows: (OneIsNotOne Condition)

$$a = 1 \tag{2}$$
$$b = 0 \tag{3}$$

Example of references: we have equation (1) from Sec. 1. Or 2-(3).

3 How to cope with TikZ figures

Tex4th supports TikZ, however, for correct displaying text and math symbols in TikZ picture, it is needed to put in the preamble of tex file the following lines, before TikZ package loading:

```
\ifdefined\HCode
   \def\pgfsysdriver{pgfsys-dvisvgm4ht.def}
\fi
```

Below is an example of using TikZ.

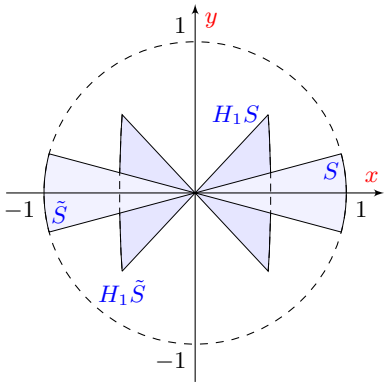

Figure 1: Example of Stanford

Citation example: [1].

Remark With recent versions of `GhostScript` (>=10.02.0) in your TEX system and `dvisvgm` (<3.1.1), `tex4th` creates a corrupted version of the `svg` file, by including the following uncommented string (from `GhostScript` output) in its first lines: "*The old, written in PostScript, PDF interpreter has been removed entirely. You should cease using -dNEWDPF as it has no effect n*". This should be recovered when `dvisvgm` version >=3.1.1 will be installed in your LATEX system.

One way to recover: If it is your case and you will see no normal picture in the created html file, then **this string must be removed manually in the generated svg file using any text editor so that this file to be displayed normally in the html output!**

Another way to recover (until it will be recovered by creators of `GhostScript` or `dvisvgm`). Find the file `pdf_main.ps` in your TEX system. For example, in TeXLive it is placed in the directory `...\texlive\2023\tlpkg\tlgs\Resource\Init`. After that, replace in this file the following fragment of code:

```
systemdict /NEWPDF known {
  systemdict /NEWPDF get not {
    (The old, written in PostScript, PDF interpreter has been removed entirely.\n) print
    (You should cease using -dNEWDPF as it has no effect now.\n) print
    (Continuing to process PDF file using the new, written in C, PDF interpreter.\n) print
  } if
} if
```

with the following one:

```
systemdict /NEWPDF known {
  systemdict /NEWPDF get not {} if
} if
```

After that, the spurious string in the generated `svg` file disappears and this file will be happily accepted by the resulting `html` file! □

References

[1] Al-Nayef A., Diamond P., Kloeden P. et al. Bi-shadowing and delay equations // Dynam. Stability Systems. 1996. Vol. 11, no. 2. P. 121–134.