Math with MathJax 2

> **Note:** This is an alternative method to the one in Math Formulae with MathJax

1) Adapt the page in which you are testing/writing the Math formulae

- Ideally set the default output to SVG. Otherwise the user will need to select this from: Math Settings >> Math Renderer >> SVG
- Add code for a button to send the processed SVG to your PHP script

Example of MathJax page

```
<!DOCTYPE html>
<html>
<head>

<!-- This line adds MathJax to the page with default SVG output -->
<script type="text/javascript"
src="http://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS-MML_SVG"></script>

</head>
<body>

<h3>The Cauchy-Schwarz Inequality (TeX)</h3>
\[ \left( \sum_{k=1}^n a_k b_k \right)^2 \leq \left( \sum_{k=1}^n a_k^2 \right) \left( \sum_{k=1}^n
b_k^2 \right) \]

<h3>Standard Deviation (MathML)</h3>
<math
display="block"><mrow><mi>&#x03c3;</mi><mo>=</mo><msqrt><mrow><mfrac><mrow><mn>1</mn></mrow><mrow><m
i>N</mi></mrow></mfrac><mstyle
displaystyle="true"><mrow><munderover><mrow><mo>&#x2211;</mo></mrow><mrow><mi>i</mi><mo>=</mo><mn>1<
/mn></mrow><mrow><mi>N</mi></mrow></munderover><mrow><msup><mrow><mo
stretchy="false">(</mo><msub><mrow><mi>x</mi></mrow><mrow><mi>i</mi></mrow></msub><mo>&#x2212;</mo><
mi>&#x03bc;</mi><mo
stretchy="false">)</mo></mrow><mrow><mn>2</mn></mrow></msup></mrow></mrow></mstyle></mrow></msqrt><m
o>.</mo></mrow></math>

<h3>Inline equation (TeX)</h3>
<p>Finally, while display equations look good for a page of samples, the ability to mix math and
text in a paragraph is also important. This expression \(\sqrt{3x-1}+(1+x)^2\) is an example of an
inline equation.  As you see, MathJax equations can be used this way as well, without unduly
disturbing the spacing between lines.</p>

<!-- This block of code adds a button to send the processed HTML code to your script:
example_test.php -->
<div id="mpdf-create">
<form autocomplete="off" action="example_test.php" method="POST" id="pdfform"
onSubmit="document.getElementById('bodydata').value=encodeURIComponent(document.body.innerHTML);">
<input type="submit" value="PDF" name="submit"/>
<input type="hidden" value="" id="bodydata" name="bodydata" />
</form>
</div>

</body>
</html>
```

2) Now you need a PHP script (in this example: `example_test.php`) which processes the output code from MathJax so that it is readable by mPDF:

Example of 1st part of example_test.php

```
// You should include a check for unwanted external referrers to prevent
// calls on this script from external websites!
```

```
$mpdf=new mPDF('');

$html = $_POST['bodydata'];
$html = urldecode($html);

preg_match_all('/<svg([^>]*)style="(.*?)"/',$html,$m);
for ($i=0;$i<count($m[0]);$i++) {
    $style=$m[2][$i];
    preg_match('/width: (.*?);/',$style, $wr);
    $w = $mpdf->ConvertSize($wr[1],0,$mpdf->FontSize) * $mpdf->dpi/25.4;
    preg_match('/height: (.*?);/',$style, $hr);
    $h = $mpdf->ConvertSize($hr[1],0,$mpdf->FontSize) * $mpdf->dpi/25.4;
    $replace = '<svg'.$m[1][$i].' width="'.$w.'" height="'.$h.'" style="'.$m[2][$i].'"';
    $html = str_replace($m[0][$i],$replace,$html);
}
preg_match_all('/<path d="(.*?)" stroke-width="(.*?)" id="(.*?)"><\/path>/',$html, $d);
$defs = array();
for ($i=0;$i<count($d[0]);$i++) {
    $defs[$d[3][$i]]['sw'] = 0;
    $defs[$d[3][$i]]['path'] = $d[1][$i];
}
$html = preg_replace('/<defs.*?<\/defs>/','',$html);
$html = preg_replace('/<svg>.*?<\/svg>/','',$html);    // get rid of the <defs> SVG

preg_match_all('/<use xlink:href="#([a-zA-Z0-9\-]+)"([^>]*)><\/use>/',$html,$m);
for ($i=0;$i<count($m[0]);$i++) {
    $replace = '<path d="'.$defs[$m[1][$i]]['path'].'" stroke-
width="'.$defs[$m[1][$i]]['sw'].'"'.$m[2][$i].'></path>';
    $html = str_replace($m[0][$i],$replace,$html);
}

preg_match_all('/<use y="([-]{0,1}[0-9]+)" x="([-]{0,1}[0-9]+)" xlink:href="#([a-zA-Z0-9\-
]+)"([^>]*)><\/use>/',$html,$m);
for ($i=0;$i<count($m[0]);$i++) {
    $replace = '<g'.$m[4][$i].'><g transform="translate('.$m[2][$i].','.$m[1][$i].')"><path
d="'.$defs[$m[3][$i]]['path'].'" stroke-width="'.$defs[$m[3][$i]]['sw'].'"></path></g></g>';
    $html = str_replace($m[0][$i],$replace,$html);
}
```

3a) Finally you can create a PDF document directly based on the MathJax web page submitted:

Example of 2nd part of example_test.php creating a PDF document

```
// ADD a stylesheet
$stylesheet = '
/* This helps alignment for inline equations */
img { vertical-align: middle; }
/* This sets padding for display equations (but not in-line ones) */
.MathJax_SVG_Display { padding: 1em 0; }
/* This prevents the Create PDF button being reproduced in the PDF document */
/* Use this method to suppress other parts of the web-page from displaying */
#mpdf-create { display: none; }
/* Add any other CSS styling here for the rest of the document */
/* The CSS/stylesheet information from the original page is not accessible here */
';

$mpdf->WriteHTML($stylesheet,1);

$mpdf->WriteHTML($html);
$mpdf->Output();
exit;
```

3b) Or you could output the prepared SVG code suitable for including directly in your PDF documents:

Example of 2nd part of example_test.php to output the code to a browser

```
...
// To output SVG files (one for each formula) readable by mPDF as text output
header('Content-type: text/plain');
```

```
preg_match_all('/<svg(.*?)<\/svg>/',$html,$m);
for ($i=0;$i<count($m[0]);$i++) {
    $svg = $m[0][$i];
    $svg = preg_replace('/>/',">\n",$svg);    // Just add some new lines
    echo $svg."\n\n";
}
exit;
```

See an example: http://mpdf1.com/common/mpdf/examples/MathJaxSample.htm

Printed on Wed 05 Aug 2015 12:38:49 GMT +0100 (DST)