

Transforming \LaTeX documents: software and processes

Emma Cliffe

May 2012

Introduction

This work was carried out as part of a HESTEM Mathematics Curriculum Innovation project. We

have explored methods to produce flexible and accessible learning resources for mathematics with a focus on producing a guide for staff in HEI to enable them as individuals to create flexible resources efficiently and robustly.

This document does not describe methods by which to produce inputs for these processes. Please view the other outputs from the project at <http://www.bath.ac.uk/study/mash/maths-access>.

Contents

1	Linux transformations	1
1.1	Requirements	1
1.2	TeX4ht	2
1.2.1	Installation	4
1.2.2	Flowfix: project output, proof of concept	4
1.2.3	Usage	6
1.3	PlasTeX	9
1.3.1	Installation	11
1.3.2	Usage	13

2	Windows transformations	14
2.1	LibreOffice	14
2.1.1	Installation	15
2.1.2	Usage	15
2.2	MathType	18
2.2.1	Installation	19
2.2.2	Usage	19
2.3	Word2TeX	21
2.3.1	Installation	22
2.3.2	Usage	22
3	Using formats	27

3.1	Standard PDF: screen and print	27
3.2	Large print PDF: screen, print and E-book readers	28
3.3	XHTML+MathML: IE, MathPlayer and Speech	29
3.4	HTML + MathJax: multi-browser with reflow	31
3.5	\LaTeX + BaKoMa: editable	33
3.6	Word + MathType: flexible, editable	36
3.6.1	Installation	37
3.7	\LaTeX : Speech and Braille	38

1 Linux environment for transformations

- The Linux environment is currently required to transform \LaTeX documents.
- The software packages required are available without fees.
- There are Windows versions of all of the software but the methods have not yet been tested in this environment. Future work is to ensure that the below instructions can also be used in Windows 7.
- The environment has been tested on Ubuntu 10.04 (Long Term Stable).

1.1 Requirements

All of the below should be readily available via your package management system:

- TeXLive <http://www.tug.org/texlive/acquire.html>,
- ImageMagick <http://www.imagemagick.org>
- C, Python and Flex <http://flex.sourceforge.net/>.

It is recommended that you also have:

- Stix fonts <http://www.stixfonts.org/> or other fonts to enable a visual check of MathML output in Linux. These may be available via package management.
- A PDF viewer and web browser.

1.2 TeX4ht

TeX4ht transforms documents written in \LaTeX to other mark-up formats. TeX4ht uses styles and

post-processors to manipulate output of the native \LaTeX compiler. In theory this allows TeX4ht to handle the widest range of \LaTeX features. However, the particular transformations we will use the system for are complex. TeX4ht is highly configurable. In addition to documenting features of \LaTeX documents which impede the transformation (see guide to writing documents) we also specify a particular configuration. This is the end point of careful iterative testing and improvement using assistive technology to assess the outputs.

We will be using TeX4ht to produce an XHTML and MathML format tested and suitable for use with text-to-speech and screenreading technology. We will also be using it to produce input for LibreOffice which can then be converted to a format for use with Microsoft Word.

- Main website: <http://tug.org/tex4ht/>
- Main information page: <http://tug.org/applications/tex4ht/mn.html>

1.2.1 Installation

If possible, it is advised that TeX4ht is installed via your package management system. TeX4ht is available for common varieties of Linux such as Ubuntu, Fedora and openSUSE. Ideally you should check that the installed version has already been upgraded as per <http://tug.org/applications/tex4ht/mn-upgrade.html>. For information about updates and the current status of TeX4ht see the main website.

1.2.2 Flowfix: project output, proof of concept

One output of this project is proof of concept code which manipulates standard \LaTeX files produced as per our instructions to create a large print/small screen version of the same document. This method uses the \LaTeX package `breqn.sty`

to line break equations. This cannot be used as the base \LaTeX format as the `breqn` equation environments cannot currently be transformed to the other required formats. The same code also produces a 'fix' of the \LaTeX required for input to TeX4ht in order that a format LibreOffice can import is output. This is a short-term fix to overcome known bugs in the LibreOffice import of MathML.

The flowfix proof of concept code is written in Flex and requires both Flex and C to compile.

Installation: Download the file <http://www.bath.ac.uk/~cspehj/maths-access/flowfix.zip>.

```
> unzip flowfix.zip
> cd flowfix
> make
```

Usage: To use this code your \LaTeX file (but none of the included images) needs to be copied to the flowfix directory. Edit the `settings` as per the

instructions in `settings-instructions`. Different settings may be required depending on the purpose. If a hard copy print size of greater than 20pt is required this can be produced by reducing the page size but then printing to A4 paper.

```
./run.sh file.tex
```

This produces:

1. `LPfile.tex`: compile using `pdflatex` to create the large print/small screen PDF version. For more information see section 3.2.
2. `ODTfile.tex`: for use with `TeX4ht`.

1.2.3 Usage

To create the XHTML + MathML format:

1. Create directory `XHTML`.

2. Copy the standard \LaTeX file `file.tex` to XHTML.
3. Copy any included images to XHTML.
4. Download `http://www.bath.ac.uk/~cspehj/maths-access/xht/unicode.4hf` to the XHTML directory. Do **not** change the name of this file or attempt to use it from another directory.
5. Download `http://www.bath.ac.uk/~cspehj/maths-access/xht/XHT.cfg` to XHTML.
6. In XHTML run the command:

```
> mk4ht xhtmlatex file.tex  
"XHT,mathplayer"
```

This command may take some time to complete.

7. The files `file.xht` and `file.css` will be created. `file.xht` can be viewed with any browser capable of rendering MathML, assuming the fonts are present. See section 3.3 for further information.

To create the input format for LibreOffice (to be converted for Microsoft Word):

1. Create directory ODT.
2. Copy the transformed file `ODTfile.tex` (see section 1.2.2) to ODT.
3. Copy any included images to ODT. The images **must** be in the same directory as the \LaTeX file. If you are using PDF images you will need to convert these using:

```
> convert image.pdf image.png
```

4. Download <http://www.bath.ac.uk/~cspehj/maths-access/oo/unicode.4hf> to the ODT directory. Do **not** change the name of this file or attempt to use it from another directory.
5. Download <http://www.bath.ac.uk/~cspehj/maths-access/oo/ODT.cfg> to ODT
6. In ODT run the command:

```
> mk4ht oolatex ODTfile.tex  
"ODT,bib-,url-"
```

This command may take some time to complete.

7. Output: The file `ODTfile.odt` will be created (as will many temporary files which can be deleted). This file will need to be copied to a Windows machine for further processing (see section 2.1).

1.3 PlasTeX

PlasTeX is a second method of converting \LaTeX to other formats. PlasTeX differs from TeX4ht in that it builds a document model and supplies a collection of renderers. A document can therefore be rendered to multiple formats. However, PlasTeX is currently not able to produce a suitable format for a screenreader.

We will use PlasTeX to produce a HTML and MathJax format which, unlike the XHTML and MathML produced by TeX4ht can be viewed by any modern browser including Firefox, Chrome, Safari and many smartphone browsers. Again, unlike current TeX4ht output, MathJax permits re-flow of equations and hence can be viewed on small screens or in large fonts. We do not rely on this method for large print as the hard-copy output is currently too poor. When using IE this format can also be used for speech output but the speech may not be fully correct hence it is recommended that anyone reliant on speech output does not use this format.

- PlasTeX: <http://plastex.sourceforge.net/>
- MathJax: <http://www.mathjax.org/>

1.3.1 Installation

PlasTeX is likely to be available via your package management system however, during this project we have made additions to PlasTeX in order to produce a browser-independent output format which can be reflowed. This has been accomplished by providing a proof of concept MathJax renderer for PlasTeX. This has required some fixes to PlasTeX itself some of which are again, proof of concept only. Hence, we currently recommend that PlasTeX is installed and compiled from source. Our alterations can then be applied and PlasTeX recompiled.

Download PlasTeX from <http://plastex.sourceforge.net/> then:

```
> tar -xvjf plastex-0.9.2.tgz
> cd plastex
> python setup.py install
```

To update your version of PlasTeX:

1. **Download** <http://www.bath.ac.uk/~cspehj/maths-access/plastex-renderers.zip>

```
> unzip plastex-renderers.zip
> cd plastex-renderers
> mv MathJax plastex/plasTeX/Renderers/
> mv MathJaxS5 plastex/plasTeX/Renderers/
```

2. **Backup files:**

```
> cd plastex
> mv setup.py setup.py-bak
> cd plasTeX/Base/LaTeX
> mv Definitions.py Definitions.py-bak
```

3. **Download** <http://www.bath.ac.uk/~cspehj/maths-access/plastex-changes.zip>

```
> unzip plastex-changes.zip
> cd plastex-changes
> mv setup.py plastex/
> mv Definitions.py
plastex/plasTeX/Base/LaTeX/
```

4. In the directory `plastex` execute

```
python setup.py install
```

1.3.2 Usage

- Use of PlasTeX requires no special directory structure or setup. The command is simply run in the directory in which the \LaTeX file `file.tex` resides.

```
> plastex --renderer=MathJax  
--imager=gspdfpng file.tex
```

- A subdirectory `file` will be created. The document is broken into parts in this format so most of the files and subdirectories are required. To view the output open the file `index.html` in any modern browser. You will need an Internet connection for the equations to load. See section 3.4 for information on how to get the most out of MathJax.

2 Windows environment for transformations

- The Windows environment is required to transform Word documents and to complete the transformation from \LaTeX to Word.
- Some of the software required is available without fees; other software has an associated cost.
- The environment has been tested on Windows 7, running Office 2010 and IE 8. It is unlikely to function on older versions of Windows or Office.

2.1 LibreOffice

LibreOffice is not considered an end point as it does not provide any specific access benefits over other formats. For users requiring a standard font

editable format, who are able to use mouse and keyboard and to select standard sized icons in the display BaKoMa (using the standard \LaTeX file) is recommended, see section 3.5. For users who require an editable format which is more flexible Microsoft Word paired with MathType is recommended. LibreOffice is required to import the output of TeX4ht and then export a docx format.

2.1.1 Installation

LibreOffice should be downloaded from:

- <http://www.libreoffice.org/>

2.1.2 Usage

Given a standard \LaTeX file `file.tex` one output of the processed described in section 1.2 is a file

ODTfile.odt in LibreOffice format. Microsoft Word can open this format but the equations and structure would be lost. Instead we use LibreOffice to transform the file. If the input file has a substantial number of equations **each** step below may take a long time to complete and it may appear that LibreOffice is not responding. It will eventually finish if left!

1. Open ODTfile.odt with LibreOffice.
2. Correct headings styles:
 - File → Find and replace
 - Click on “More options”
 - Check the box “Search for styles”
 - In the first input box find “Heading-2”, in the second “Heading 2”; click “Replace all” and then “Yes”. Repeat this step for heading levels 3 and 4. Then click “Close”.

3. Update contents page: Locate the contents page, right click and select “Update field”. If `file.tex` used the `description` environment it may appear that the contents page now contains incorrect entries. Word will correct this.
4. File → save.
5. File → save as. Select file type OpenOffice XML format (docx). Wait until saved then close.
6. If your original contained images they will not have been correctly copied into the new format. Open the odt file again and repeat the “save as” step. This will now include the images.

The final processing of the docx file is described in the MathType section 2.2.

2.2 MathType

MathType is an equation editor which works with Microsoft Word. It provides functionality to style equations to the requirements of the reader, can provide a larger print interface than the standard Word equation editor, can be used with keyboard alone, mouse alone and can provide larger target click areas for users who have mobility difficulties.

We use Word and MathType to transform the output from LibreOffice to a Word document which is flexible for the end user and can be edited in these modes.

We will also use MathType to enhance the accessibility of documents written directly in Word and to produce XHTML and MathML output from this format suitable for use with text-to-speech and screenreading technology.

2.2.1 Installation

MathType is available from <http://www.dessci.com/en/products/mathtype/>. Information regarding accessibility features can be found at <http://www.dessci.com/en/support/mathplayer/tsn/tsn112.htm>.

For Word 2010 the most recent version of MathType (6.8) is **required** for output of XHTML and MathML.

2.2.2 Usage

To import carry out the final step in the conversion from \LaTeX to Word and MathType format (see sections 1.2 and 2.1 for earlier steps in this process):

1. Open the output `file.docx` from LibreOffice.
2. Click on the MathType ribbon tab.

3. Click on Convert Equations, a pop-up window opens:

- Equation types to convert: tick all boxes..
- Range: Whole document.
- Convert equations to: MathType equations (OLE objects).
- Click Convert: If the file contains a substantial number of equations this step may take a long time to complete and it may appear that Word is not responding or displaying correctly. It will eventually finish if left!

The same process can be used to convert equations in documents written using the Word equation editor.

To export an XHTML and MathML version of a document written in Word:

1. Click on the MathType ribbon tab.
2. Click on Publish to MathPage. In the pop up window:
 - Title: enter the title of the document.
 - Equations: Select MathML using (drop down box): XHTML+MathML.
 - Click OK: If the file contains a substantial number of equations this step may take a long time to complete and it may appear that Word is not responding or displaying correctly. It will eventually finish if left!

2.3 Word2TeX

Word2TeX is a converter which enables Microsoft Word to save files in \LaTeX format. We will use this to produce \LaTeX output for direct use by students (see 3.5 and 3.7).

2.3.1 Installation

Word2TeX is available from <http://www.chikrii.com/products/word2tex/>. Installation and usage guidance can be found in the manual <http://www.chikrii.com/dl/word2tex/word2tex.pdf>.

2.3.2 Usage

1. Before you save as \LaTeX any currently displayed Word auto-generated contents page must be removed or hidden. We want \LaTeX to auto-generate the contents for itself, not to include a hard-coded version of the Word table. Since the table of contents is auto-generated it can safely be deleted and reinstated. Should you prefer you can also turn on track changes and then delete it. Replacement is a simple case of rejecting the change. This method can be used to hide any content you do not wish to be present in

the \LaTeX .

2. You also need to convert any auto-generated bibliography to standard text (again, a new bibliography is easily inserted or track changes can be used).
3. Click on File → Save as.
4. Select .tex from the drop-down list of file formats.
5. Two pop-up windows will open with warnings. The first warns you that not all content may transform. Accept this. The second warns you to only use this method with files you trust (i.e. that you have written). This is at your own risk.
6. In the pop-up window select the following settings:
 - General: $\text{\LaTeX}2\text{e}$; AMS; ANSI; LF; Don't wrap paragraphs; Recognise in-line equations in text; Use Longtable package; Translate font size.

- Equations: Leave factory settings, however, by this mechanism you are able to configure the transformation of equation structures.
- Preamble: change

```
\documentclass{amsart}
```

to

```
\documentclass[12pt,a4paper]{article}
```

and add in

```
\usepackage[a4paper]{geometry};
```

finally, at the base of the preamble add

```
\newif\ifplastex %Required
```

```
\plastexfalse %Required
```

```
\newif\iftht %Required
```

```
\thtfalse %Required
```

```
\newif\ifLP %Required
```

```
\LPfalse %Required
```

```
\ifLP %Required
```

```
%Reflow cannot occur within a root  
symbol in large print PDF
```

%so the root symbol is transformed
to other notation

```
\renewcommand{\sqrt}[2][2]
{\left( #2 \right)^{\frac{1}{#1}}}
\fi
```

```
\allowdisplaybreaks %Required
```

```
\setlength{\parindent}{0.0pt}
```

```
\setlength{\parskip}{1.0\baselineskip}
```

(See the instructions on producing
L^AT_EX files if interested).

- Styles: Leave as default except in AMS-L^AT_EX2e figure inclusion template change the default from:

```
\centerline{
  \includegraphics [width=%width,
height=%height] {%filename}}
```

to:

```
\includegraphics [width=\textwidth,
angle=0] {%filename}
```

- Figures: Select Convert figures; Convert to PDF; leave rest as defaults.
- Title: Click on maketitle, leave what is there as default but add

```
\ifplastex
\else \iftht
\else
\renewcommand{\familydefault}{phv}
\fontfamily{phv}\selectfont
\fi \fi
```

before \maketitle.

- Add the \tableofcontents command at the end.
- Function translation: Tick translate mathematical functions.

7. These settings should save in between use. Press OK.

8. The \LaTeX file will be saved and then **opened in Word in some sense**, that is, you will continue to see the docx but would save over the

tex file. Close it and open the tex file with something more useful!

The resultant output should be suitable for use with BaKoMa (section 3.5), speech and Braille (section 3.7) and for transformation to large print (section 1.2.2).

3 Using the transformed formats

Each section describes the installation of any software required to use the format.

3.1 Standard PDF: screen and print

This format can be produced from \LaTeX in the usual manner and from \LaTeX files produced by

Word2TeX (section 2.3).

The standard print/screen size PDF can be opened with any PDF viewer and printed. It may also be sent to some e-readers including the Kindle and Sony readers. However, the print size is likely to be too small for many readers on a small screen and for some on a standard screen or in print. Although some PDF viewers are able to resize and reflow the English text this destroys equation structure, this format also cannot be edited or annotated with ease by the end user, read aloud or used to produce Braille output. See below for formats more suitable for these contexts.

3.2 Large print PDF: screen, print and E-book readers

This format can be produced from \LaTeX using the proof of concept flowfix (section 1.2.2). It may also

be produced from \LaTeX produced by Word2TeX (section 2.3) assuming the correct output configuration has been selected.

The large print/small screen size PDF can be opened with any PDF viewer and printed. It may also be sent to some e-readers including the Kindle and Sony readers. Different settings in flowfix (section 1.2.2) may be required depending on the purpose. If a hard copy print size of greater than 20pt is required this can be produced by reducing the page size but then printing to A4 paper.

3.3 XHTML+MathML: IE, MathPlayer and Speech

This format can be produced from \LaTeX via TeX4ht (section 1.2) and from Word via MathType (section 2.2). While it **may** display correctly on any given browser this is dependent on the browser

MathML capability and the available fonts. This format is therefore not recommended for use with browsers other than IE. It is a recommended non-editable format for students who use a screen-reader or text-to-speech assistive technology.

MathPlayer, a plugin to Internet Explorer (IE) is required for both visual display and speech mathematics. It is freely available and can be downloaded from <http://www.dessci.com/en/products/mathplayer/download.htm>. Basic tests with the preview release of MathPlayer 3.0 confirmed that there are problems with this version, IE9 and assistive technology hence that combination should not be used. There is a note to this effect on the MathPlayer page.

We tested resources using IE9 and MathPlayer 2.2. Testing was completed using JAWS 12.0 screenreader <http://www.freedomscientific.com/products/fs/jaws-product-page.asp> and TextHelp Read and Write Gold (10) literacy

support software <http://www.texthelp.com/UK/our-products/readwrite>.

Information about using MathPlayer is available at <http://www.dessci.com/en/products/mathplayer/tech/accessibility.htm> and a full list of assistive technology compatible with MathPlayer is available from http://www.dessci.com/en/solutions/access/at-support.htm#Reader_Tools.

3.4 HTML + MathJax: multi-browser with reflow

This format can be produced by the version of PlasTeX described in this document (section 1.3). MathJax is an open source Javascript display engine for mathematics which works in **all** modern browsers including those on some smart phones and tablets. Mathematics can be rendered in a for-

mat appropriate for each browser and hence does not require the installation of fonts or additional software. A connection to the Internet is required.

Information about MathJax:

- **Main website:** <http://www.mathjax.org/>,
- **Accessibility:** <http://www.mathjax.org/resources/articles-and-presentations/accessible-pages-with-mathjax/>.

The MathJax user menu is accessed by right clicking on an equation, this allows users to control zoom and rendering. The mathematics in MathJax pages will scale with the font size of the surrounding text without user intervention but it may be zoomed further as required. If using the HTML-CSS renderer (this can be controlled by the user) equations will reflow as needed.

If the using IE and Mathplayer is available MathJax can use MathPlayer as the renderer and

hence equations can be read aloud. The format produced by PlasTeX is likely to be suitable for use with text-to-speech but not for screenreader users as alt tags are not available for images and some mathematics symbols, while visually rendered correctly are not spoken with enough precision.

3.5 \LaTeX + BaKoMa: editable

BaKoMa TeX is a \LaTeX implementation with a complete visual “what you see is what you get” editor. It is available from <http://www.bakoma-tex.com/>. A copy of the base \LaTeX source and any files to which it refers would need to be provided to the user. It is important to recall that the user will have access to the \LaTeX source and hence comments in the \LaTeX will be available to them.

This format is a flexible, editable format suitable for users who do not require large print or a specialist

editing environment.

Steps to undertake for a better user experience:

1. Open BaKoMa and click on Open an Existing File; select the \LaTeX file to open.
2. Select TeX Format: $\text{\LaTeX}2\text{e}$; leave other parameters as defaults and click OK. A visual display of the document will appear.
3. This is equivalent to running \LaTeX **once**. \LaTeX documents typically require 1-3 runs of \LaTeX before they are visually stable. **Refresh the document twice** using the top, left most icon on the toolbar.
4. To view document navigation click on Options \rightarrow Document Properties In the pop-up window click on the Outline tab. Select Outline Source: TOC and click OK.

BaKoMa includes a version of TeX4ht which is configured via a graphical interface. As a con-

sequence it is possible for the user to produce speech enabled XHTML and MathML documents for use with text-to-speech in literacy support software. However, much of the configuration is lost so this format will not provide clear enough speech information to rely on speech alone.

BaKoMa is particularly suited to students who use text-to-speech software such as those with specific learning disabilities or conditions which affect concentration. It is suitable for users who need to edit or annotate documents and for those for whom producing handwritten mathematics is a difficulty. However, BaKoMa requires fluent keyboard and mouse use and text size cannot be increased substantially without difficulty. For more flexible editable formats see sections 3.6 and 3.7.

3.6 Word + MathType: flexible, editable

MathType is an equation editor which works with Microsoft Word. It provides functionality to style equations to the requirements of the reader, can provide a larger print interface than the standard Word equation editor, can be used with keyboard alone, mouse alone and can provide larger target click areas for users who have mobility difficulties. MathType can therefore enhance the accessibility of Word documents for students who wish to annotate an editable format but require large print or a use alternative input modes.

MathType also provides the option to output edited documents as XHTML+MathML (see section 2.2) and so students using text-to-speech will be able to listen to the edited resources.

Word and MathType not recommended for all users above BaKoMa (section 3.5) as the output

format from \LaTeX is less pleasant to view than that in BaKoMa and may prove distracting.

MathType or MathType Lite is required to view all Word documents which have been created with MathType.

3.6.1 Installation

MathType is available from <http://www.dessci.com/en/products/mathtype/>. Information regarding accessibility features can be found at <http://www.dessci.com/en/support/mathplayer/tsn/tsn112.htm>. Information regarding MathType Lite is available from <http://www.dessci.com/en/products/mathtype/faqs.htm#read>.

3.7 \LaTeX : Speech and Braille

None of the Braille translation methods tested could completely transform documents of any level of complexity. We therefore provide information on those which do not lose information, retaining the underlying \LaTeX . A copy of the base \LaTeX source and any files to which it refers would need to be provided to the user. It is important to recall that the user will have access to the \LaTeX source and hence comments in the \LaTeX will be available to them.

The \LaTeX -access project provides real time translation of a line of \LaTeX to Nemeth Braille (to be read on a refreshable Braille display) and speech. A matrix browser enables reading of larger matrices. Please note that only a limited set of symbols and structures are defined by the \LaTeX -access scripts however, there is support for user defined \LaTeX commands and users are able to add to the symbol and command lists more easily than

in other software we tested. The \LaTeX -access scripts are available from <http://latex-access.sourceforge.net/>.

Some students may prefer to use a screenreader but to read the \LaTeX source directly in Braille rather than a Braille mathematics code. They may find the following useful:

- LaTeXLex cleans up \LaTeX source to produce a 'human readable' \LaTeX which is easier to listen to and read in Braille. Available from: <http://mesarhameed.info/projects/latexlex>;
- ChattyInfty: is an editor for mathematical documents with voice output, where equations are completely transformed they can be navigated and are easier to listen to than the raw \LaTeX . Braille of the underlying \LaTeX is available on a Braille display. ChattyInfty is available from <http://www.sciaccess.net/en/ChattyInfty/>.

Both of these pieces of software transformed all test files retaining the raw \LaTeX where no useful transformation was available. Thus, assuming the student can read \LaTeX none of the document is lost.