Creating Matching PDF and HTML Documentation Using tex4ht

By: Björn Windén

September 30, 2022
Source : http://ocean-cfd.engr.tamu.edu/doc/latex2web

Attachment : document_template.zip

Report No.: **OCEN CFD Technical Report #220001**

1 Introduction

This document gives an introduction on how to use LaTeX-style document preparation for creating documentation on this website. LaTeX (stylized as LaTeX) is a convenient tool for creating professional looking layouts of documents such as reports, articles, books and more. It is especially convenient for academic use and with documents that contain equations. Furthermore, when writing a LaTeX document, the author only works in a plain text file with no formatting so it is very portable and shareable. Platforms exist for collaborative writing online such as Overleaf.

In it's basic form, LaTeX converts text in a plain text file to a PDF document based on various formatting settings. These settings are given directly in the plain text document. For this website, a script exists to translate the LaTeX output to HTML for displaying in a web browser.

The base LaTeX → HTML translation is done using tex4ht. However, the script heavily modifies both the input and output to the base translator in order to produce the appearance shown on this website. The website is meant to be an easy-to read "Wiki"-style environment.

The script also creates a PDF using the standard *pdflatex* command. The PDF and the web page (bearing exactly the same content) automatically reference each other with hyperlinks.

As a result, at the top of this document, there will be a link to the corresponding PDF(when viewing the website) or URL(when viewing the PDF). Click it to compare!

2 Preparing a custom LaTeX document for this website

Creating documentation for this website is done, for the most part, with regular LaTeX without modification. However, in order to work with the conversion script, a few custom additions and rules are needed. Those relating to the general document are detailed below. For the document content, see subsequent sections relating to adding content.

2.1 Creating and editing a LaTeX document

Two methods/platforms are currently available for creating and editing articles for this website.

- Manually edit and compile the document on your local machine. A template document is attached to this article for this purpose. This method requires that you have a LaTeX compiler installed in order to be able to view the resulting article in PDF-form.

- Use the web-based Overleaf platform to edit and view the document. This does not require any locally installed software. Please use the provided Overleaf document template, found at: https://www.overleaf.com/latex/templates/ocen-cfd-article-template/mtpnnssgzjzw. If you have an Overleaf account, this link should bring you directly to an editable document. When finished, the article can be saved as a zip-file, downloaded and submitted.

2.2 Document linking and author profile

Certain commands shall precede the `\begin{document}`-marker in order to properly link it to the website and the author's profile.

```
\title{<Title of your document>}
\author{<Your name>}
\authorID{<Your member ID>}
\documentID{<A unique identifier>}
\fileInclude{<Files to attach>}
\date{\today}
```

The `\documentID` is used to identify this document on the website. It can be set to anything (alphanumeric) but it may be useful to set it to relate to the contents somehow. This document is named ”latex2web” for example. The `\authorID` should match the author's ID on the website members page since the document will automatically be given a link to the author's profile there. The `\date{\today}` command should be left as it is, it will automatically show the date the document was created (compiled) as the publication date. If a different publication date is desired, it can be modified accordingly.

The `\fileInclude` command can be used to attach files to the document. The attachment should be placed in the `attachments`-folder as shown in Section 2.3.

2.3 File/folder structure

The LaTeX source file, attachments and figure images shall be placed in a certain folder structure as shown in Fig. 1. A template document with the correct structure is attached to this document. The root folder and LaTeX source file can be called anything but it may be useful to name them according to the `\documentID` discussed above.

3 Displaying code

Code (such as C++, bash, python etc.) can be displayed in using the LaTeX *lstlisting* environment. This is used in the LaTeX code as :

```
\begin{lstlisting}[language=C++]
        <Paste some code here>
\end{lstlisting}
```

For example, if one pastes some C++ code as shown above, it produces the following in the document:

```

tmp<fvVectorMatrix> tUEqn
(
        fvm::div(phi, U)
        + MRF.DDt(U)
        + turbulence->divDevReff(U)
        ==
        fvOptions(U)
);
fvVectorMatrix& UEqn = tUEqn.ref();

UEqn.relax();
fvOptions.constrain(UEqn);

if (simple.momentumPredictor())
{
        solve(UEqn == -fvc::grad(p));
        fvOptions.correct(U);
```

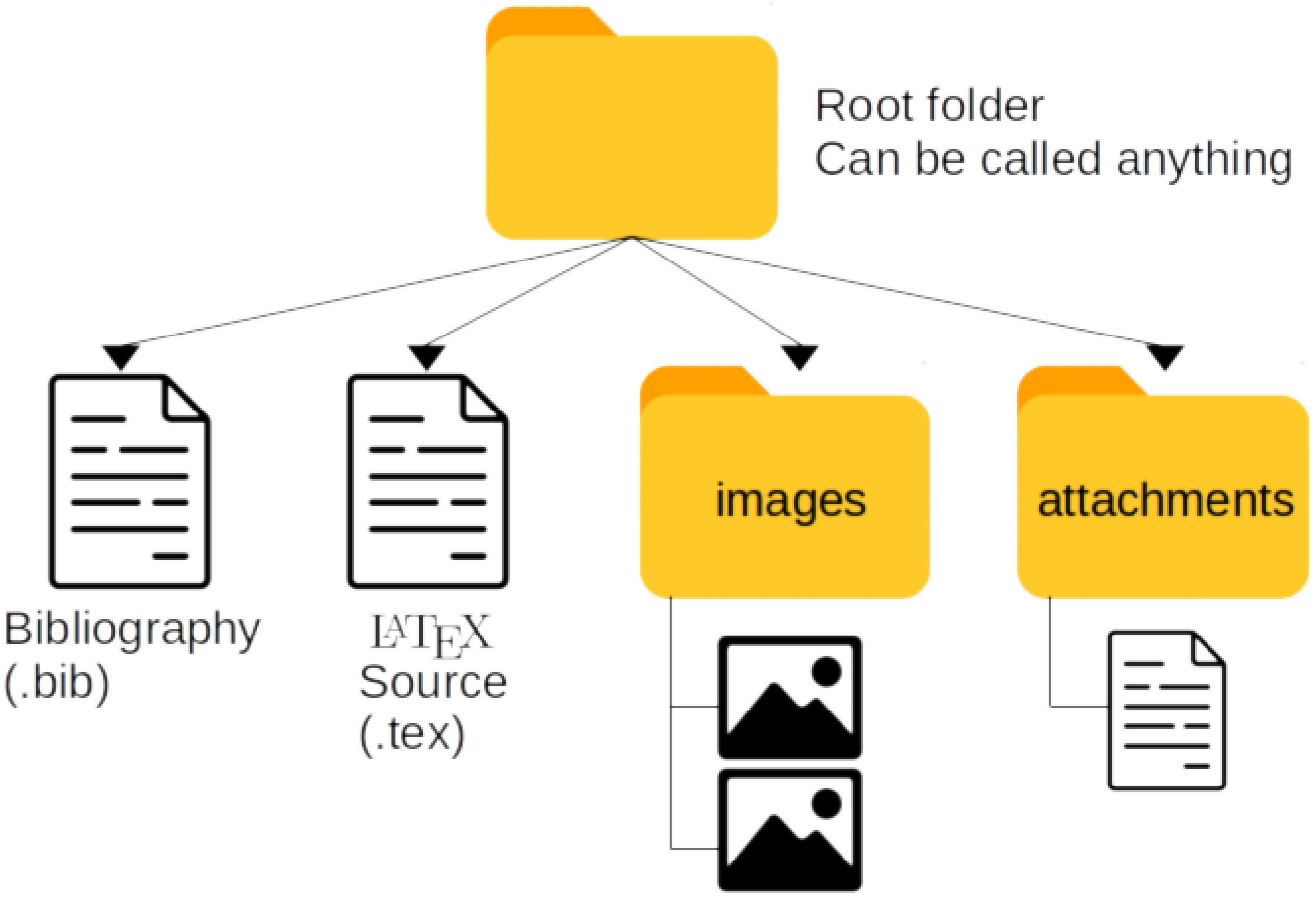

Figure 1: Folder structure to be used with the documentation generation script.

```
}
```

4 Adding tables and figures

Tables and figures are created with the standard `\begin{table}`...`\end{table}` and `\begin{figure}`...`\end{figure}` environments respectively. Add the *[h!]* option to prevent them from free floating in the PDF. There is no floating in the HTML by default. Adding a `\label{<some name>}` tag inside these environments allows for cross-referencing. For example, to create a table with a caption and a hyperlink to reference it :

```
\begin{table}[h!]
        \caption{An example table.}
        \label{tabletest}
        \begin{center}
                \begin{tabularx}{\textwidth}{ X X X }
                        \toprule
                        Heading 1 & Heading 2 & Heading 3 \\
                        \midrule
                        1 & 2 & 3 \\
                        4 & 5 & 6 \\
                        7 & 8 & 9 \\
                        \bottomrule
                \end{tabularx}
        \end{center}
```

```
\end{table}
```

which produces Table 1.

Table 1: An example table.

Heading 1	Heading 2	Heading 3
1	2	3
4	5	6
7	8	9

Figures are created similarly as:

```
\begin{figure}[h!]
        \includegraphics[width=\linewidth]{images/example.png}
        \caption{An example image.}
        \label{figtest}
\end{figure}
```

which produces Fig. 2. The images should be placed in the `images`-folder as shown in Section 2.3.

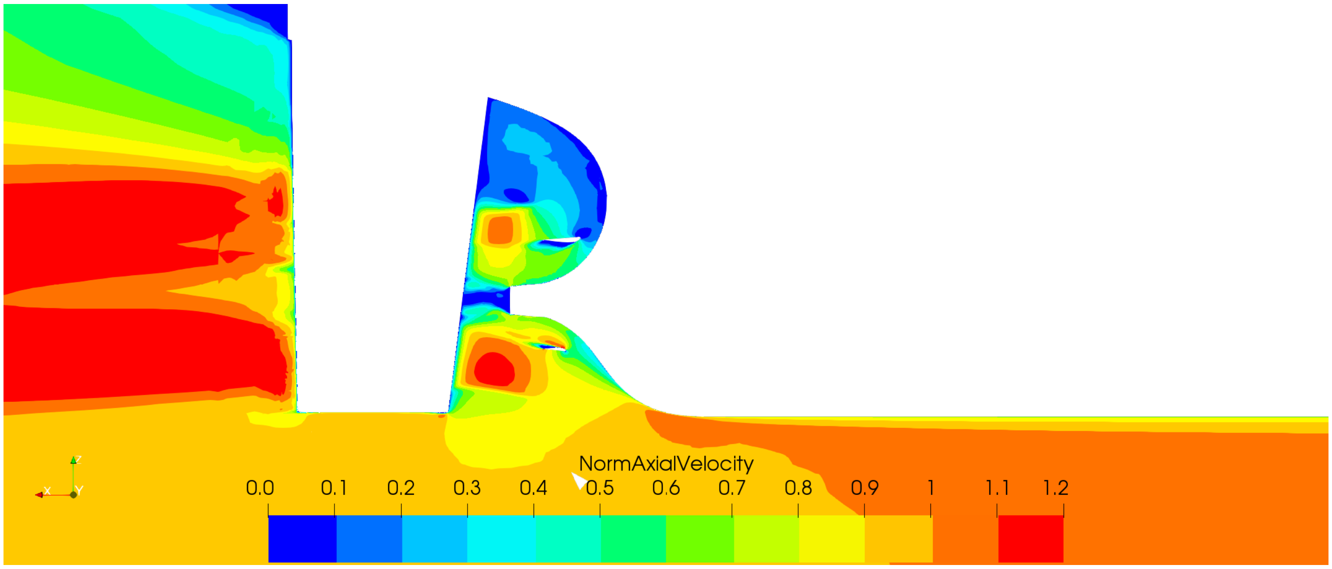

Figure 2: An example image.

5 Adding equations

LaTeX is a useful program for writing documents containing equations. This document supports two types of equations, inline and block.

5.1 Inline equations

Inline equations appear directly in the text. For example: ”the velocity in the x-direction is defined as $U_x = \partial x_i/\partial t$”. To create an inline equation, please use tag `\ieq`. Using the custom tag `\ieq` over any other ways to define a math environment ensures that the equation is given proper alignment in the HTML. Example definitions of inline equations are given in Table 2.

Table 2: Inline equation syntax examples

LaTeX code	Output
`\ieq {x}`	x
`\ieq {U_x=\partial x_i / \partial t}`	$U_x = \partial x_i / \partial t$

5.2 Block equations

Block equations are put in their own separate space and given an equation number. Use the standard `\begin{equation}...\end{ec` environment to define block equations. Adding a `\label{<some name>}` tag inside this environment allows for cross-referencing. For example, to create an equation and create a hyperlink to reference it :

```
\begin{equation}
        \label{example_eqn}
        a^2+b^2=c^2
\end{equation}
```

which produces Eq. (1). The equation reference to Eq. (1) is created as: `\eqref{example_eqn}`.

$$a^2 + b^2 = c^2 \tag{1}$$

In the PDF, long equations will automatically be broken up onto multiple lines using the *breqn* package and the *dmath* environment. The *equation* environment will automatically be changed to *dmath* when compiling the PDF.

For the HTML output, using *breqn* is not possible. Therefore, locations of where the equation should break need to be defined by manually. This is done with the custom `\mainflexsplit` and `\secondaryflexsplit` commands. The equation will be broken up onto different lines at the location where these are inserted. For narrower screens, such as tablets, the equation will be broken at `\mainflexsplit`. For even narrower screens, such as smartphones, the equation will be broken at `\secondaryflexsplit` and `\mainflexsplit`. These splits have no effect for the PDF compilation where equations will be broken up automatically using *breqn*.

For example, Eq. (2) is a pretty long equation that is probably fine to display on one line on desktops and tablets, but needs to be broken up onto two lines to display nicely on smartphones. It is created as:

```
\begin{equation}\label{longeqn}
        \kappa_N(r) \secondaryflexsplit
        =\frac{2}{\pi}\cos^{-1}{exp\left(-Z\left(1-\frac{r}{r_0}\right)\frac{\
           ↪ sqrt{r_0^2+h(r)^2}}{h(r)}\right)}
\end{equation}
```

and displays as:

$$\kappa_N(r) = \frac{2}{\pi}\cos^{-1} exp\left(-Z\left(1-\frac{r}{r_0}\right)\frac{\sqrt{r_0^2+h(r)^2}}{h(r)}\right) \tag{2}$$

Eq. (3) is an even longer equation that needs to be split into two lines for tablets and into multiple lines for smartphones. It is created as:

```
\begin{equation}\label{verylongeqn}
        \left(\frac{2\sqrt{r^2+a(r)^2}}{Zk_1c(r)}+\frac{r^2+h(r)^2}{2rh(r)\
           ↪ kappa_N(r)}\right)\Gamma(r,\theta) \secondaryflexsplit
        +\left[\frac{\partial \phi_{Pf\infty}}{\partial x} \right]_P
        -\frac{h(r)}{r}\left[\frac{\partial \phi_{Pf\infty}}{r\partial \theta}\
           ↪ right]_P \mainflexsplit
        =\frac{a(r)}{r}\left({\Omega}r + \left[u_{p\theta}(r,\theta)\right]_P\
           ↪ right)
        -\left[u_{px}(r,\theta)\right]_P
```

```
\end{equation}
```

and displays as:

$$\left(\frac{2\sqrt{r^2+a(r)^2}}{Zk_1c(r)}+\frac{r^2+h(r)^2}{2rh(r)\kappa_N(r)}\right)\Gamma(r,\theta)+\left[\frac{\partial\phi_{Pf\infty}}{\partial x}\right]_P-\frac{h(r)}{r}\left[\frac{\partial\phi_{Pf\infty}}{r\partial\theta}\right]_P=\frac{a(r)}{r}\left(\Omega r+[u_{p\theta}(r,\theta)]_P\right)-[u_{px}(r,\theta)]_P \tag{3}$$

6 Adding references and bibliography

References can be added using the standard `\cite` and `\citep` commands together with a bibliography stored in a `.bib`-file (see Section 2.3). The bibliography is added to the document using `\bibliography{bibfilename}`, where *bibfilename* is the name of the `.bib`-file without the `.bib` extension. The bibliography will appear the location where this command is given in the LaTeX source file.

For example. By adding the following to a file called `latex2web.bib`:

```
@book{ latexsource,
       author = "Leslie Lamport",
       title = "LATEX : a document preparation system",
       year = "1986",
       publisher = "Addison-Wesley Pub. Co.",
       isbn="020115790X"}
```

this book can be cited as:

```
\LaTeX{} was created by \cite{latexsource}.
```

which results in:"LaTeX was created by Lamport (1986)." The reference also shows up in the bibliography (created as: `\bibliography{latex2web}`) automatically.

References

Lamport, L. (1986). *LATEX : a document preparation system.* Addison-Wesley Pub. Co.