The Case for LaTeX

Henry Ginn

October 5, 2025

1 Introduction

This document gives a brief overview of what LaTeX (pronounced "Lay-tek") is and highlights some of the advantages it has over Microsoft Word. LaTeX is a well-established piece of typesetting software for producing technical documents where the input is given in plain text which is then compiled into a PDF. It is understandable why one may be wary of any potential barriers to those less comfortable with technology, but the payoff is more than worth it, and LaTeX is far simpler than it may seem.

The main advantages are as follows:

- Less buggy than Word
- No need to worry about formatting - template is followed automatically
- Equations, cross-references, and citations are far easier than in Word
- Vector image compatible
- Images are given by reference
- The LaTeX compiler knows how to typeset better than you so documents look professional

In section 2 we elaborate on why LaTeX formatting is easier than in Word, followed by the advantages when writing technical documents in section 3. We finish with a discussion on the accessibility of LaTeX in section 4 before concluding, with an appendix demonstrating why Word can be frustrating to work.

2 Formatting

We begin by outlining the problems with formatting in Word, even by a trained practitioner who knows how to use Word properly. Throughout this section I will assume that an inconsistent document makes WSP look unprofessional and that fixing issues around inconsistent or poor formatting is trivial nonsense that wastes our time and our client's money.

First of all, it is easy to break the template in Word. For example, many times I have had section numbering disappear even though the numbering is part of the section header style, where reapplying the style did not fix it. In my experience styles do not work reliably in tables either and properties such as spacing after paragraph need to be manually corrected around tables and figures. Once the template is defined the user should not need to put any extra work into ensuring that it is followed - the software should handle this for us.

Often I find Word does weird things that no one understands and there is no good fix. For example, I have had tables where the boldness of text would not save and the typical solution to a problem like this is to recreate the whole table from scratch or copy and paste everything into a new document. I have seen colleagues input tables as images rather than deal with Word's formatting, a veritable crime against typesetting that looks extremely unprofessional.

In LaTeX, all formatting rules for elements such as section and subsection styles, spacing, font sizes, headers and footers, et cetera are defined at the start of the document. The template applies globally[1] which makes documents look consistent and therefore only the type of formatting needs to be given.

[1] It is possible to deviate from the template locally if you really want to but typically this is awkward to do. This is because you are trying to defy the compiler which almost always means you are attempting to commit crimes against typesetting and should not be doing whatever you are trying to do.

As examples, to tell the compiler you are creating new section or subsection, one would write `\section {Example Section Name}` or `\subsection{Example Subsection Name}` respectively and to tell it to format as a bullet point list, the following code is used.

```
\begin{itemize}
        \item Example of the first item
        \item Example of the second item
\end{itemize}
```

This method separates the formatting from the content, meaning consistent documents and no time wasted on formatting. LaTeX also gives a lot more control than Word. As simple examples, it is possible to adjust how much weight it should give towards avoiding widows, or exactly how much space you want between two elements.

3 Technical Writing

One of the main powers of LaTeX is the ability to format mathematics. In equation (1) below is a simple example showing closure of the $SO(2)$ group, something that would be extremely painful to typeset in Word but is trivial to do in LaTeX.

$$R(\theta)R(\phi) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \tag{1a}$$

$$= \begin{pmatrix} \cos\theta\cos\phi - \sin\theta\sin\phi & -\cos\theta\sin\phi - \sin\theta\cos\phi \\ \sin\theta\cos\phi + \cos\theta\sin\phi & -\sin\theta\sin\phi + \cos\theta\cos\phi \end{pmatrix} \tag{1b}$$

$$= \begin{pmatrix} \cos(\theta+\phi) & -\sin(\theta+\phi) \\ \sin(\theta+\phi) & \cos(\theta+\phi) \end{pmatrix} \tag{1c}$$

$$= R(\theta+\phi) \tag{1d}$$

We note the following that are impossible, awkward, or very manual to do in Word:

- The equations are numbered automatically and we do not need to position them ourselves
- The equations are aligned on the equals sign - I did not need to add spaces to do this
- The whole equation or individual parts such as equation (1b) can be cross-referenced easily
- Much more complicated configurations can be created, such as equations in two columns

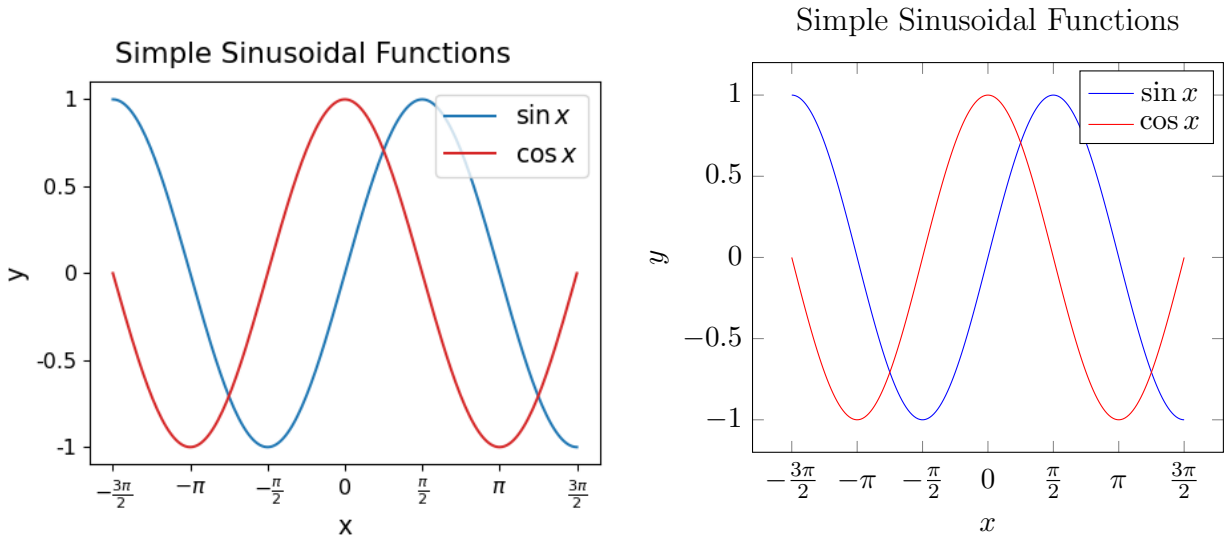
Images are inserted by giving a path to the image. This means when the images are updated, the document also updates when it is recompiled. Copying by reference means there is one version of the truth which is one of the reasons why copying by value in situations like this is poor practice. I have wasted a lot of time reinserting images into documents because it appears Word only copies by value.

Word and other Microsoft Office Suite products are very poor at handling vector graphics. As an engineering consultancy, we feature many images which could be vector graphics yet are forced to rasterise, lowering their quality. The only option I have found that Word is happy with are Enhanced Metafiles (EMF), a format of Microsoft's creation with many problems[2]. On the other hand, LaTeX handles both bitmaps and vector graphics easily.

Word's strange behaviour with images and exporting wastes time. Word will compress images for you by default when pasting an image into the document, and even stranger, the level of compression depends on how zoomed in you are. I have also seen documents where all the figures are unreadable because Word decided to compress them all between revisions. There are many ways to export to PDF in Word, all of which behave differently and compress images. LaTeX will produce a PDF with the exact image it was given.

[2]EMF files do not support transparency. They also do not or antialiasing so all text looks odd as well.

Going further than this, plots and figures can even be created inside LaTeX itself, although this is typically for more advanced users. This means even the font in plot titles and axis labels can be consistent with the rest of the document and even the entire project. I have not found any other way of achieving this conveniently. Figure 1 demonstrates the difference in quality. Even though figure 1a has been well-made in Python, the lower resolution, inconsistent style, and nasty typesetting of π symbols make it look noticibly worse than figure 1b. Figure 1b has infinite resolution and the added benefit of highlightable and searchable text[3].



(a) 96 dpi raster made with matplotlib

(b) Vector graphic made in LaTeX

Figure 1: A comparison between a low quality rasterised image and a vector graphic.

Below is a list of other features of LaTeX that make it better for writing technical documents that Word does not support.

- Subfigures can be given automatically numbered and easily cross-referencable subcaptions.
- Figures and tables can be given an optional short caption which appears in the list of tables and list of figures. This means brief metadata about figures can be kept closely attached to the figures and tables in the caption without cluttering the corresponding lists.
- The exact positioning of objects is handled automatically. For example, the user provides the caption text and the compiler puts it in a sensible place.
- All formatting changes can be tracked and easily copied.
- Smaller file sizes and the ability to handle documents several hundred pages long easily.
- The output is determined exactly from the text input, so it is not possible to reload the output and get different behaviour like in Word.

https://github.com/HenryGinn/Essays/tree/main/TheCaseForLaTeX

LaTeX is also better for version control. As it is all text-based it can be used with version control software like Git, which gives tracability, control over branches, and easy computation of differences between versions. This also tracks any formatting changes that cannot be handled by the template such as the shape of a table.

[3]It is not necessary to make the figure in LaTeX for this functionality, any vector image would have this.

4 Ease of Use

The cost of these improvements is that LaTeX is less simple to use than Word, although I would argue the gap is not that large, especially for basic word processing. Table 1 summarises how complex certain tasks are to do in LaTeX with examples given below. The general principle of simple aims being simple to execute and complex aims being complex to execute holds, as long as it is inline with good typesetting principles, for example sideways text might need some research.

Table 1: The experience and training needs to do certain tasks in LaTeX.

Task	Difficulty
Typing in paragraphs	As easy as in notepad.
Creating headings	Use the `\section{}`, `subsection{}`, and `subsubsection{}` commands instead of applying a style.
Cross-referencing	Use `\label{label name}` in the relevant location and then `\ref{label name}` to reference that location.
Figures and tables	Requires some code which can be copy and pasted. For basic cases the formats are easy to extend from examples.
Single equations	Nearly as easy as Word, for example `\alpha` for α instead of pressing a button from a menu.
Complex equations	Will need some practice and around half an hour of training.
Making a template	Intermediate level of difficulty, but only needs to be done once per project. Regular users do not need to be at this level.
Making diagrams	Intermediate to advanced, but unnecessary for what we do at WSP and can be replaced by a figure produced in other software anyway.

Inserting images is likely the most difficult task most employees would need to do. The below code produces two subfigures side by side, although could easily be modified to have three. I posit that almost all WSP employees would be able to figure out that they would need to add another subfigure block and change 0.49 to 0.33.

```
\begin{figure}[H]
    \begin{subfigure}{0.49\textwidth}
        \includegraphics[width=\textwidth]{example-image-a}
        \caption{Example A}
        \label{Example Subfigure 1}
    \end{subfigure}
    \hfill
    \begin{subfigure}{0.49\textwidth}
        \includegraphics[width=\textwidth]{example-image-b}
        \caption{Example B}
        \label{Example Subfigure 2}
    \end{subfigure}
    \caption[Short caption that appears in list of figures]{Overall figure caption}
    \label{Overall Figure Caption}
\end{figure}
```

LaTeX has been around for around 40 years and is the standard in most technical fields in academia. Common packages are well-documented and there is a large repository of questions and answers on the dedicated TeX Stack Exchange. When an error occurs, a benefit of LaTeX over Word is that the issue is almost always due to the user doing something wrong rather than strange behaviour of LaTeX. This means the user has control over the problem and is able to fix it.

5 Conclusion

In summary, it is far faster and easier to produce technical documents in LaTeX once familiarity has been attained, and the results are of higher quality and appear more professional. Table 2 below gives a comparison of the key differences between Word and LaTeX. The source code for this essay is available on my GitHub.

Table 2: Comparing Word and LaTeX.

Property	Word	LaTeX
Editor format	What You See Is What You Get	Needs to compile
Bugfixing	Poor support and confusing behaviour	Good support with searchable errors
Software	Proprietary	Free and Open Source
Formatting control	Poor	High
Technical features	Lacking	Rich
Microsoft compatibility	High	Medium
Simplicity	Extremely simple	$\sim$ 1 hour of training

A Failures of Word

In case it is not clear how unstable, buggy, and annoying Word is, I have included a few oddities and frustrations of Word below:

- Tables split across page breaks by default.
- Captions are not attached to the corresponding object by default.
- Sometimes invisible, unselectable, and undeletable formatting characters can appear in your document that break your formatting. The only fix for this is manoeuvering the character somewhere it does no harm, or copying everything into a new document.
- Clicking the bullet point button can change the font size and font style.
- Space after paragraph should be used to ensure space between paragraphs, but the space is still added for the last paragraph on the page. As the space is attached to the paragraph, it puts the line on the next page even if that line would fit above. To avoid widows, space after paragraph needs to be manually handled.
- The gap between "Figure" or "Table" and the number in the caption is not a non-breaking space by default. When cross-referencing this can result in the number being on a new line.
- Vertically centred text in a cell of a table does not appear vertically centred, even if space before and after paragraph and margins are symmetrical.
- Page breaks take up space. This means if the last line that fits on the page is long then the page break mark is moved to the next page, meaning a blank page. This means needing to manually handle the page break.
- Changing the width of a column in a table where the last cell highlighted is merged will only change the width of the first row of the merged cell. This means needing to change the width of those cells separately and remerging afterwards.
- The number of words that fit on a line depends on their justification - this should only affect the spacing between words (unless rivers are avoided, which Word cannot do).