Michael Whapples

Supporting Students with Disabilities series

Obtaining Braille mathematical documents

Michael Whapples
School of Physics and Astronomy
University of Nottingham
mwhapples@aim.com

Introduction

I have just completed a physics degree at Nottingham University. As a blind person, I face all the normal challenges of doing a physics degree with the additional challenge of trying to find suitable written physics materials which I can read. I had expected there may be some difficulties in finding Braille physics texts, but I had not expected the number of difficulties encountered and that I would be working on my own Braille translation software by the end of my course.

Background reading

Within this article, I will make some references to Braille and solutions which have been used by me in completing my degree. I would also suggest reading the article in the previous edition of *MSOR Connections* by Steve Maddox to get a background of these topics [1].

My history

I have always had visual impairment and learnt Braille from a young age. I went to RNIB New College Worcester (NCW), which is a specialist school for the visually impaired. At RNIB NCW I had no difficulties in accessing any of the usual subjects, as the teachers had knowledge of teaching the visually impaired, skills such as being able to read and write Braille, and had access to equipment to allow students to work independently. This meant I could work directly in Braille, handing answers to questions in Braille and the teachers could give feedback in Braille. Also, the teachers were able to produce handouts in Braille if there were no suitable books. This meant I could just get on with learning the subject without concern on how to access it. Due to this, no subject was more difficult than any other to access and meant I was able to do the subjects I enjoyed when it came to A-level choices.

Moving to university

When I came to university it was a big change. Now the staff had no experience in teaching the visually impaired and lacked skills such as reading and writing Braille. This meant I needed to alter the way I would work. Writing has been less of an issue than reading. I had used computers when I was at school and can touch type, so it seemed sensible for me to produce work on a computer. The WinTriangle software [6] was initially used, but as this could only provide speech output, which was difficult to use when dealing with complicated equations, I felt another solution was needed. The LaTeX language was suggested, which was difficult to learn initially, but once it

was learnt it proved to be the most suitable. Its advantages were that it could be written in any text editor, so any access software could be used, allowing me to use speech output and a Braille display. Additionally, as it is widely used, when I had difficulties with the LaTeX code someone at the university could help me. Unfortunately, reading hasn't been quite as easy as this.

The initial situation for reading

Initially reading looked like it would be fairly simple, but it soon proved to be much more difficult than I expected. As there was one book recommended as being suitable for all the modules in the first year, enquiries were made to find out if this could be obtained in Braille. Searches of libraries in this country and the Braille libraries in America revealed that there wasn't even a book similar to the recommended book in Braille. So if there isn't a book in Braille, would there be any other way for me to access books? As I am a confident user of computers, there was the thought of trying to get the book in an electronic format. The only electronic copy the university could get was in postscript format, which unfortunately is not accessible by any of the access software I am aware of. Due to this lack of text books, the lecture notes were going to be of greater importance to me as they would be the only source of information.

Most lecture notes were already in an electronic form, but if not it was possible to have a note taker making notes in lectures. As none of the staff were able to translate documents into Braille manually and most of the notes were already in an electronic form, it seemed sensible to look to computers for a solution. There were two systems considered, either using a Braille translator to produce standard British Braille or the DotsPlus system [6]. The DotsPlus system claims to give very good access to maths by removing the issues with translation as it breaks from the standard six dot Braille cell and uses graphical tactile symbols for more complicated mathematical symbols. Due to this increased number of symbols, it may be applied as a font (using direct substitution), so simplifying the production technique. It also keeps the layout of the print document, meaning that the Braille now is in a two dimensional format. While these advantages seem to be very appealing, I was unfamiliar with the code, and was uncertain about the ease of learning it, and difficulties with the code was the last thing I wanted whilst trying to study a physics degree. This meant I chose to use Braille translation software. As there are many different Braille codes used around the world, and I am only familiar with the British code, it meant that the choice of software was limited to the Duxbury Braille Translator (DBT) [4].

Systems tried

As discussed in Steve Maddox's article, Braille is more complicated than just making a substitution of Braille characters for the print ones, as it uses contractions, or abbreviations for common combinations of letters. Braille also has symbols for formatting. As Braille is a linear format, when performing a translation the print needs to be converted to this linear form. Generally, Braille translation of standard text is very reliable, but when faced with mathematical notation, the problem becomes much more complicated. We found the output of Duxbury became unreliable when the document contained notation more complicated than the very basic maths symbols. When nobody at the university could proof read the output of Duxbury, this made things very difficult. Over time, we started to find that it seemed to be a very limited subset of LaTeX that Duxbury could handle. As problems were found, these were noted down and added to a perl script, which would fix LaTeX files ready for use with Duxbury. For more details on this, refer to the article by Steve Maddox [1]. This still proved to be difficult for me, as whenever I discovered a new command that Duxbury could not handle, I had to wait for a corrected version of that set of notes to be produced. This meant that notes were normally arriving later than the lecture, and there was no guarantee that they were even correct.

Feeling that this was substandard, I kept thinking of trying to write my own Braille translation software. Originally, the thought was to use the script for fixing LaTeX for use with Duxbury, and modifying the output of the script to be Braille rather than other LaTeX commands. As I started work on this, it proved to be more difficult than I expected. This was due to factors such as, Braille being context dependent, LaTeX being designed for visual layout rather than content and there just not always being a one to one translation. As this was proving to need so many rules to be defined and complicated matching procedures for finding the meaning of the LaTeX, I decided this was going to be too much work for me on my own, considering that I still had to do my degree as well. This meant I abandoned this approach early on.

The approach used

In the summer of 2006, I came across the PlasTeX project [3] (see Fig 1). This sparked my enthusiasm for writing a Braille translator of my own again. PlasTeX is a python package designed for LaTeX processing, which allows people to write their own output modules. This seemed to reduce the work required, as I would now only need to concentrate on the Braille output and could leave the LaTeX processing to PlasTeX. So with the summer vacation ahead of me, I started work on it and the BrlTex project [2] came into existence (see Fig 2).

While PlasTeX did greatly simplify the problem, as time went on, various difficulties were found to be still present. Some of these are due to: the content of LaTeX being designed for visual layout, the design of Braille and PlasTeX. One problem I needed to solve was the conversion of the text parts of the documents. I felt that to write my own text to Braille translator would be a large task and distract from

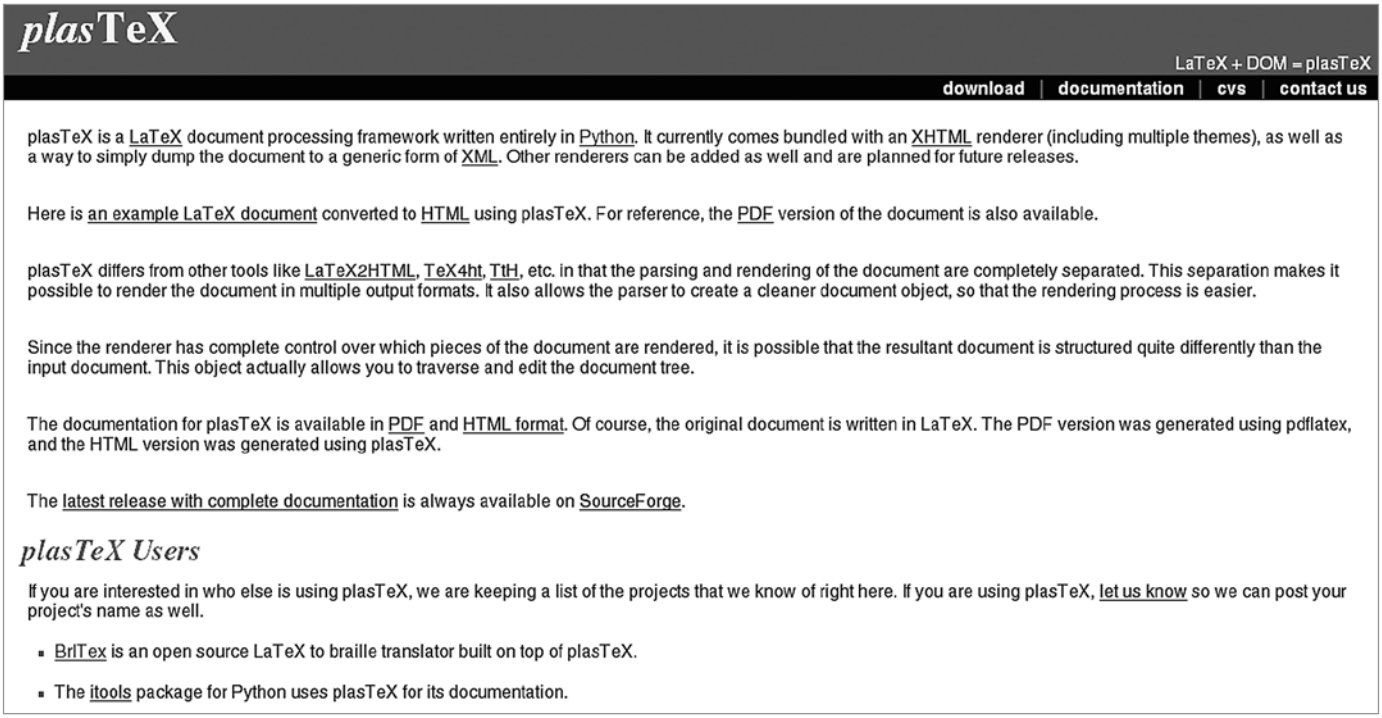

Fig 1 – PlasTeX project website

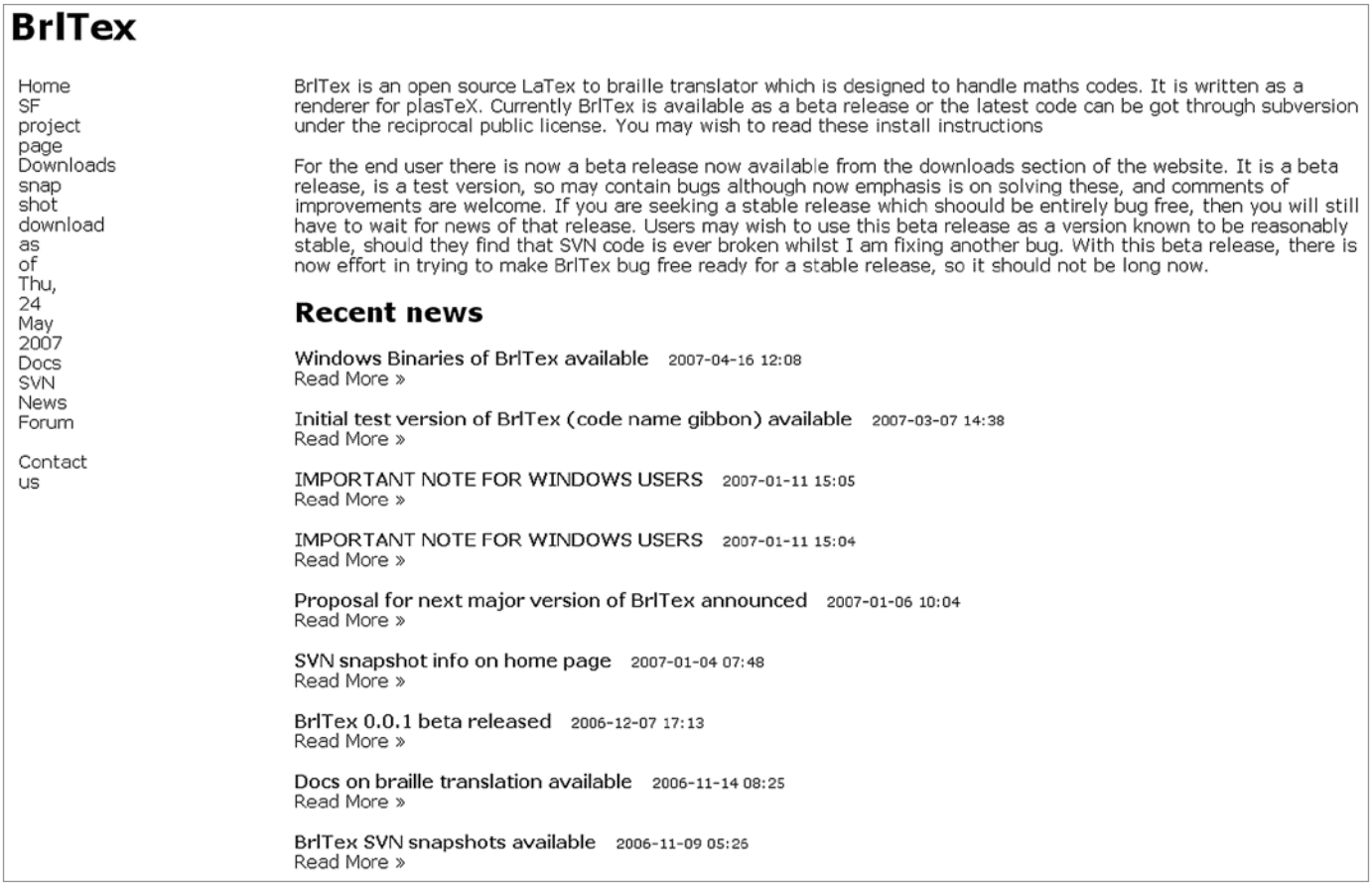

Fig 2 – BrlTex project website

the task at hand, and with the accuracy of other translators for plain text, I thought it would be best to use a third party translator. However, this did require writing some translation tables for simple maths for the text to Braille translator, as maths uses a different Braille code and simple maths (e.g. + - / etc.) are considered text by PlasTeX.

While PlasTeX does present the document information in a well structured python object, allowing specific details to be extracted from the document easily, sometimes the information required for Braille translation just isn't there. This is normally due to the LaTeX source. This can be in situations where print characters look very similar, so LaTeX only implements the one (e.g. omicron and o), but the two

symbols may be very distinct in Braille. This meant that I had to write some rules to detect the cases where the symbol would be different to the one specified. This wasn't very satisfactory, as it meant only certain cases were being dealt with and exceptions were always likely to show up.

Another problem with getting accurate Braille output is that, some LaTeX is coded poorly; while the print output looks fine, the meaning in the LaTeX is slightly different, so leading to unusual Braille output. The most common of this type of problem is the use of math environments, either not using it for simple maths, or having a word in an equation (e.g. $$x^2 + y^2 = constant$$), which leads to unusual output because of the different translation rules for math and standard text.

PlasTeX has fairly good LaTeX processing, but there were some cases where it didn't always do what I expected, either due to not implementing a command, or just not handling it correctly. This meant I had to write some code to correct this LaTeX into PlasTeX LaTeX, which proved to be a fairly simple task and was much less work than creating my own LaTeX processor.

The current situation

BrlTex has not been the system used by the university for Braille translation, but I have used it on my own computer with a Braille display to get Braille access to course notes when paper Braille versions have been unavailable. The advantages for me of using BrlTex is that: it produces standard British Braille, works on both Microsoft Windows and Linux, and with its free and open source form allows it to be improved as required. When used with a Braille display, this meant that if there was a problem with the translation of a document, I could fix the error and have a correct Braille version within a very short time.

I will admit that BrlTex is not perfect in its translation, it doesn't support all LaTeX commands, isn't always able to find the meaning of some LaTeX and only supports the British Braille code. Despite currently having these limitations, I feel that for a years work whilst doing a degree as well, it shows promise and proves to me that fairly accurate Braille translation is possible. I hope that with the work I will be doing as part of the Maths, Stats & OR Network sponsored project "Accessibility in MSOR: LaTeX and Braille", I will be able to make some major design changes to BrlTex, which would hopefully simplify or solve some of the issues which exist, while adding greater opportunity for supporting other Braille codes, input formats and developing other improvements.

For all the time I have been doing my degree, getting accurate Braille texts for it has been difficult, but some progress is being made. Unfortunately this progress is typically slow due to many factors which have been discussed in this article. There is the issue of having technology which is capable of the task, which still needs some more development to get that to a good standard. Sometimes the technology struggles because the document is in an unsuitable format, such as postscript which doesn't contain enough information, or sometimes the document is written incorrectly and as a result the software completely misinterprets the meaning. Hopefully, more texts will become accessible as standards, such as MathML, with accessibility built in are adopted more widely, allowing the technology to do its work. The situation is improving and, with the right work, it should be possible to make further significant improvements.

"For all the time I have been doing my degree, getting accurate Braille texts for it has been difficult, but some progress is being made."

References

1. Maddox, S. 2007 Mathematical equations in Braille, *MSOR Connections* May 2007, Vol 7 No 2, p45-48: http://mathstore.ac.uk/newsletter/may2007/pdf/45_maddox_s_braille.pdf [Accessed 16 July 2007].
2. BrlTex LaTeX Braille translator: http://brltex.sf.net [Accessed 9 July 2007].
3. PlasTeX python LaTeX processing package: http://plastex.sf.net [Accessed 9 July 2007].
4. Duxbury Braille Translator (DBT) [Braille creation software]: http://www.duxburysystems.com [Accessed 9 July 2007].
5. The Science Access Project [A research group aiming to develop methods for making science, math, and engineering information accessible to people with print disabilities.]: http://dots.physics.orst.edu [Accessed 9 July 2007].
6. DotsPlus [an extension of standard Braille for science based documents]: http://dots.physics.orst.edu/dotsplus.html [Accessed 16 July 2007].
7. WinTriangle – Math and physics for the blind [A scientific word processor to help blind people.]: http://www.wintriangle.com [Accessed 9 July 2007].