Nemeth braille math and LaTeX source as braille

Susan Jolly

Abstract

This article, which is dedicated to the late TeX expert Prof. Eitan Gurari [13], introduces braille math for sighted persons unfamiliar with braille. Braille systems represent print in one of two ways: either by transcription or by transliteration. Most braille systems, including the Nemeth system for mathematics, employ shorthand, markup, meaningful whitespace, context-sensitive semantics, and other strategies to transcribe general printed material to a six-dot braille format that accommodates the special requirements of tactile reading. Transliteration, by contrast, is limited by the small number of braille cells and is typically only used for representing plain text ASCII files such as LaTeX source and computer code.

This article argues that while reading and writing mathematics as LaTeX source transliterated to braille is possible for facile braille readers who have eight-dot refreshable braille displays and are able to learn LaTeX, it is not an appropriate general solution for making mathematics accessible to braille users. Tactile reading of transliterated LaTeX source is no different from visual reading of LaTeX source. On the other hand, tactile reading of math transcribed using the Nemeth system is the tactile analog to visual reading of *rendered* math. Simulated braille is used to illustrate this for the benefit of sighted readers.

1 Braille cells

Most sighted persons reading this article are probably at least somewhat familiar with the punctiform appearance of braille and are aware that the dots need to be physically raised for tactile reading as opposed to visual reading. The individual braille characters, which are usually referred to as braille cells, are officially designated as Unicode Braille Patterns. Technically speaking, the braille cells aren't characters per se but rather *symbols* assigned meanings by a braille system. There are currently more than 150 different braille systems. The majority represent natural languages but there are several systems for mathematics and for other specialized material, including music and chess.

There are two forms of braille: six-dot and eight-dot. The standard six-dot patterns are comprised of three rows of two dot positions each and the eight-dot ones are comprised of four such rows. There are thus 63 different six-dot braille patterns with at least one dot and 255 eight-dot ones. Unfortunately Unicode does not treat the six-dot cells as separate character codes, but simply as the subset of the eight-dot patterns which happen to have both dot positions in their fourth row unfilled. This omission requires the use of additional information to avoid an extra blank line when embossing a braille file encoded as Unicode Braille that is intended as standard six-dot braille and the use of a custom simulated braille font to avoid misalignment when typesetting Unicode Braille intended as six-dot braille.

2 Transliteration and computer braille

Conversion of print to braille is usually done with braille systems using transcription, in order to accommodate the special requirements of tactile reading, including those described later in this article. This section describes conversion done with transliteration.

Transliteration is of limited practical use except for conversion of plain text ASCII files to braille. Since there are 94 ASCII keyboard characters (excluding space) and only 63 six-dot braille cells, one-for-one transliteration of plain text files isn't possible with six-dot braille. Six-dot transliteration of plain text files thus requires some sort of braille system that uses both single-cell and two-cell (prefix-root) symbols. One-for-one transliteration of plain text is of course possible with the use of eight-dot braille.

General eight-dot braille systems aren't widely used since tactile recognition of all 255 eight-dot cells isn't feasible for the majority of tactile readers. Nonetheless, many refreshable braille displays have eight-dot cells and incorporate built-in support for an ad hoc one-for-one eight-dot transliteration of the ASCII characters. This transliteration is known as computer braille because of its usefulness for representing computer code.

Computer braille adds 31 cells with a dot in the left columns of their fourth rows to the 63 standard six-dot cells. Twenty-six of these additional cells simply represent capital letters by adding the extra dot to the six-dot pattern for the corresponding small letter. Several options for the remaining five are in use. Because computer braille is a limited extension to six-dot braille it, in contrast to general eight-dot braille, is feasible for tactile reading.

Since computer braille specifies a one-for-one transliteration of all 94 ASCII keyboard characters it thus provides a method for braille users to directly read and write any ASCII-based plain text including classic LaTeX source. The advantage is that computer braille transliteration doesn't require special software to translate from print to braille or to back-translate from braille to print. The disadvantage is that LaTeX source and many other plain text source files are primarily intended to be rendered for visual reading, not to be read directly.

3 Nemeth braille math

The well-known six-dot Nemeth braille system, "The Nemeth Braille Code for Mathematics and Science Notation", is an outstanding example of a braille system. It was developed by Dr. Abraham Nemeth (1918–2013), a congenitally blind American mathematician who became a facile braille reader as a young child and was a math professor at Wayne State University for some 30 years. It had been in use for a number of years prior to the adoption and publication of the current official version in 1972. A PDF facsimile of the 1972 book is now available for download [1]. This book incorporates a thoughtful guide to numerous issues for converting print math to braille so as to retain all the essential information while avoiding extraneous print-specific details.

Nemeth braille was designed with a thorough understanding of both mathematics and the requirements for efficient tactile reading as outlined in the next section. Nemeth is a complete system for representing text and mathematics and includes formatting specifications. Since it was designed long before the use of digital media, it was originally intended for embossed braille transcriptions of math textbooks and other STEM material as well as for direct entry of math by braille users. The Nemeth system for mathematics has stood the test of time and is used not only in the United States but in numerous other countries, including India [12]. It is used with refreshable braille displays and, like LaTeX, its "math mode" is used for representing individual math expressions. There is also an equivalent form for spoken math called MathSpeak. MathSpeak is a unique method of speaking math since it is the only method of speaking math that supports dictation of math so it can be written correctly in either print or Nemeth braille [5].

Although Nemeth braille is a complete system, the United States, along with other English-speaking countries, recently adopted the somewhat controversial Unified English Braille (UEB) system [6]. In the U.S. it replaces both Nemeth braille and the prior system for English braille. The UEB math specification is awkward for a number of reasons, one being the use of the same braille cells for the decimal digits and for the letters a–j. The negative reaction to UEB math in the United States is significant enough that many U.S. states allow for Nemeth math, tagged with switch indicators, to be used for transcribing math content together with UEB used only for non-math text [2, lesson 3.8].

The next several sections of this article provide background for understanding the pros and cons for braille users of using computer braille to read and write mathematics as transliterated LaTeX source, versus using Nemeth braille for that purpose.

4 Tactile reading

This section describes some of the aspects of tactile reading that are taken into account when developing braille systems such as Nemeth braille. Note that braille systems are linear because tactile readers can't easily sense the relative vertical positions of the braille cells.

The rate of tactile reading, rather like the rate of typing on a standard keyboard, depends more on the number of braille cells than on the number of words. Braille systems that have been designed to minimize the number of braille cells can thus be read more efficiently.

The rate of tactile reading is also affected by particular dot patterns. For example the seven six-dot braille cells that only have dots in their right column are easier to recognize when used as markup or indicators affecting a subsequent braille cell or cells and are typically used for this purpose. The cell with a single dot at the bottom of its right-hand column indicates that the following letter is capitalized. Nemeth braille uses several others of these seven indicators to identify transliterations of Greek and other non-English alphabets.

Certain braille cells or sequences can be hard to distinguish from other possibilities. Distinguishing a lower braille cell, one which has filled dots only in its lower two rows, from the corresponding upper cell, which has the same dot pattern in its upper two rows, is an example. Nemeth uses the lower cells for the decimal digits but ensures that they are easily distinguished from the corresponding upper cells by requiring a preceding dot locator indicator, a cell with dots in all three rows, before a digit that would otherwise be at the start of a line or preceded by a space.

Braille systems can be easier to remember and recognize when they use tactile mnemonics related to the specific patterns of the braille cells. For example, the cell with dots in the upper two positions on its right is the superscript indicator and the one with dots in the lower two positions on its right is the subscript indicator. Another type of tactile mnemonics in Nemeth braille uses related dot patterns for related mathematical symbols. Examples are pairs of mirror-image braille symbols used for parentheses and for the less-than and greater-than symbols.

Finally, Nemeth uses braille-specific constructs to reduce the memory load for tactile comprehension of complex expressions. Nemeth, like LaTeX (or MathML), naturally represents planar layouts, including fractions, in a linear manner. However, when an expression, such as a fraction with another fraction in its numerator, requires nested layout indicators of the same type, Nemeth adds an explicit indication of the nesting by prefacing outer layout indicators with additional markup to represent the order or level of nesting. (This is similar to the usefulness to visual readers of highlighting matching pairs of nested parentheses using different colors for each pair.)

In summary, braille systems like Nemeth braille are carefully designed to accommodate the special requirements of tactile reading. This is in contrast to computer braille transliterations of plain text files originally designed for other purposes.

5 Simulated braille example of Nemeth math

This section uses simulated braille to display the Nemeth braille translation of the well-known equation,

$$e^x = \int_{-\infty}^{x} \sum_{n=0}^{\infty} \frac{\lambda^n}{n!} d\lambda$$

This equation, chosen in part because it employs three different planar layouts, illustrates how Nemeth's elegant use of tactile mnemonics and other tactile considerations enhances the tactile readability and information content of the braille math.

First, here's a transliteration from Nemeth braille to standard ASCII Braille used for six-dot braille:

```
e^x .k $;-,=^x"".,s<n .k #0%,=]?.1^n"/n&#d.1
```

This may look odder than other markup languages, but I've found it helpful and not too difficult to learn. Of course, since braille transcribed according to braille systems isn't one-for-one with print and also because the same braille cells typically have different semantics in different contexts, an ASCII Braille transliteration is simply a print equivalent of the braille, not a backtranslation of the braille to print. Nonetheless, the letters and digits and some special characters can be read directly in transliterated Nemeth math. Also, some of the other print characters in ASCII Braille are used because their glyphs resemble the dot patterns of the corresponding braille cells.

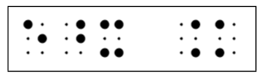
The Nemeth braille for the example uses 40 braille cells and 4 spaces. The number of braille cells is approximately 60 percent of the number of print characters in the corresponding LaTeX source.

5.1 Simulated display of Nemeth braille

The simulated braille for this expression is displayed below in five segments to make the descriptions easier to follow. Note that the standard six-dot simulated braille font used here has shadow dots in the unfilled positions. Shadow dots are intended to make visual reading easier although one may need to take care not to let the shadow dots obscure the dot patterns of the corresponding tactile braille cells.

5.1.1 First segment

Now let's try to see how the Nemeth braille for $e^x =$ from the equation above would be experienced by tactile readers. It looks like this:

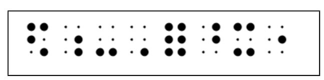


`e^x .k`

The first and third cells are the standard cells for the lower case letters e and x so are nothing new for a braille reader. The second cell, described previously, indicates that the following expression is a superscript. This superscripted expression is terminated by default by the space always required before comparison symbols. The two-cell symbol for an equals sign purposely resembles a print equals sign, as Dr. Nemeth believed that such similarities helped communication between braille readers and their sighted peers and teachers.

5.1.2 Second segment

The second segment represents $\int_{-\infty}^{x}$:

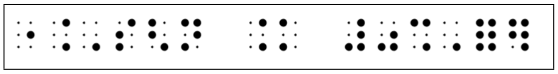


`$;-,=^x"`

The first cell in this segment, which somewhat resembles an integral sign, is the braille symbol for a single integral. The second cell is a subscript indicator with its argument terminated by default by the superscript indicator also used in the first segment. The three cells following the subscript indicator thus represent the subscripted expression. The minus sign is obvious. It is followed by the indicator cell with one dot in its lower right, familiar to braille readers from its use to indicate capital letters in six-dot systems. This indicator is also used in Nemeth math to indicate that it together with the next non-alphabetic cell is a special symbol; here, the braille cell used for infinity, resembling a rotated print infinity symbol. The cell following the superscript indicator is the symbol for x, also used above in the first segment, and the last cell is required to explicitly terminate the superscripted expression since the next item in the expression isn't a space.

5.1.3 Third segment

The third segment is a symbol decorated with, as termed in braille, an underscript and overscript, $\sum\limits_{n=0}^{\infty}$:



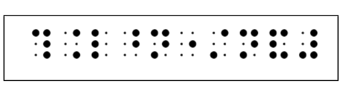
`".,s<n .k #0%,=]`

The first cell in this segment, with just one dot, is the indicator specifying that the next item is decorated. The fourth cell is the braille cell for the letter s, which is here transliterating a capital Greek sigma per its two preceding indicators. This is followed by the Nemeth underscript indicator and then the expression for $n = 0$ which uses the standard cell for the letter n, the same space-delimited symbol for an equals sign used in the first segment; the number sign dot locator described in Section 4, which is required because the following digit would otherwise be preceded by a space; and then the lower cell for the digit zero. The zero is followed by the Nemeth overscript indicator and then the same two-cell symbol for infinity used in the second segment. The last cell is the required Nemeth terminator for any layout using one or more underscripts and/or overscripts.

This is a case where the Nemeth math, which is intended to represent print presentation in a consistent manner, is especially lengthy in comparison with the compact print rendering. It might be desirable to develop more informative print shorthand rather than replicating print presentation for common expressions that use underscripts and overscripts. For example, since summation is essentially a function application, a custom string like "sumnzi" could be added to the function name abbreviations already recognized by Nemeth braille. This would reduce the number of braille cells and spaces for this segment from 17 to 7 counting the extra space required to separate a Nemeth function reference from its argument and would thus be a reduction of about 25% in the number of braille cells in the entire expression.

5.1.4 Fourth segment

The fourth segment of the formula above is the fraction $\frac{\lambda^n}{n!}$, which uses the Nemeth simple fraction layout indicators:

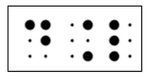


`?.l^n"/n&#`

The first cell in this segment, which resembles an upside-down print L, is a strong tactile shape used as the fraction start indicator. The last cell, which is used as a dot locator in other contexts, is another strong tactile shape that is here used as the fraction end indicator. The cell with two dots that resembles a print forward slash separates the numerator from the denominator. The third cell is the letter l; you shouldn't have too much trouble reading the numerator since the other four cells have already been encountered. The letter n in the denominator is followed by the one-cell Nemeth braille symbol for a factorial sign.

5.1.5 Fifth segment

The fifth and final segment, `d.l`, is simply $d\lambda$:

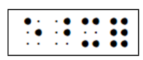


`d.l`

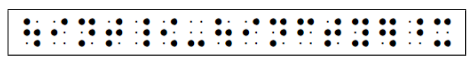
The first cell is the letter d. The remaining two cells are the symbol for lambda also used in the fourth segment.

5.2 LaTeX source as simulated braille

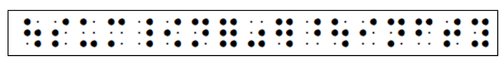
I hope that these descriptions have allowed you to appreciate how both the tactile form and the braille symbols specified by Nemeth braille supply information to tactile readers. Here, for contrast, is the corresponding ASCII Braille transliteration of the LaTeX source for each segment:



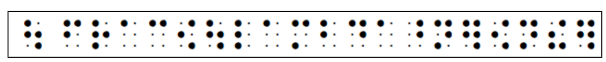
`e^x=`



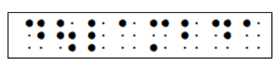
`\int_{-\infty}^x`



`\sum_{n=0}^\infty`



`\frac{\lambda^n}{n!}`



`d\lambda`

6 Conclusion and future work

Here is some good news: two hardware issues for real-time access to braille have recently been addressed. First, new hardware designs have resulted in significantly cheaper single-line braille displays such as the six-dot BrailleMe [11]. Second, and of special importance for braille math, is the newly available six-dot Canute braille display which is not only low in cost but also the first multi-line refreshable braille display [3].

An urgent need for future work is accurate and free automated backtranslation from braille math to print math. Currently available applications, most of which are not free, are problematic and students typically require their often unavailable itinerant braille teachers to interpret their braille work for their classroom teachers. Addressing this issue is a current goal of the Euromath project [4]. High school and college students sometimes resort to learning to read and write LaTeX math as a result of poor support for braille math. In my opinion their time as students would be much better spent on improving their mathematical ability. In any case, LaTeX source is not an especially convenient basis for manipulating math.

A possible starting point for providing automated backtranslation is the beta version of my free and open source BackNem 3.0 app for accurate backtranslation of Nemeth math to MathML, as demonstrated by several samples [9]. This app, which is based on the ANTLR 4 parser generator, is to my knowledge the first use of parsing technology for backtranslation of braille to print [7]. One valuable feature of ANTLR 4, which is especially important in educational contexts, is that its parsers can recognize input errors, provide optional developer-supplied error messages and, unlike other parsers, continue processing despite encountering input errors.

Future work needed to support the claims in this article includes development of a software system for real-time conversion of LaTeX source to six-dot braille mathematics designed for integration with screen readers and other applications. The difficulties of direct conversion of LaTeX to other formats is well-known. A two-step process that first converts LaTeX to MathML with one of the currently available applications and then converts MathML to braille math is a more viable approach. This second step is straightforward for Nemeth math due to similarities between it and MathML, and a beta version of my MML2Nem app is available for consideration of a new approach [8, 10]. The need for real-time translation is especially critical for education due to the recent dramatic increase in the use of electronic information in this context.

Finally, I should point out that I'm not in a position to develop the needed software nor to provide the infrastructure necessary to test, distribute, or maintain software. I am however very glad to volunteer to help other developers of open source braille software as well as to answer questions about braille mathematics.

References

[1] AAWB-AEVH-NBA, Advisory Council to the Braille Authority, Louisville, KY. *The Nemeth Braille Code for Mathematics and Science Notation. 1972 Revision*, 1987. `www.brailleauthority.org/mathscience/nemeth1972.pdf`

[2] American Printing House for the Blind, Inc., 1839 Frankfort Avenue, Louisville, KY. *Nemeth Tutorial*, 2015. `nemeth.aphtech.org`

[3] Bristol Braille Technology CIC. Electronic Braille: reimagined, 2018. `bristolbraille.co.uk`

[4] EuroMath Education Platform. EuroMath project, 2017. `www.euromath.eu/wp-content/uploads/2018/12/EuroMath_Presentation.pdf`

[5] gh, LLC. 2004 MathSpeak initiative, 2004. `www.gh-mathspeak.com`

[6] C. Gray. Call to action: Prevent educational harm to Braille readers, 2014. `all4braille.org/C1002.htm`

[7] S. Jolly. Using ANTLR 4 for braille translation, 2011. `github.com/SusanJ/BasicUEB/wiki`

[8] S. Jolly. Positive impacts of EPUB 3: MathML and Braille mathematics, 2012. `www.dotlessbraille.org/mathmlandbraille.htm`

[9] S. Jolly. Samples of BackNem 3.0 output, 2018. `github.com/SusanJ/Baknem/wiki/Samples`

[10] S. Jolly. Translation of MathML to Nemeth Braille, 2018. `github.com/SusanJ/MML2Nem`

[11] National Braille Press. Braille Me, n. d. `www.nbp.org/ic/nbp/BRAILLE-ME.html`

[12] Overbrook School for the Blind; International Council for Education of People with Visual Impairment; The Nippon Foundation. *Mathematics Made Easy for Children with Visual Impairment*, 2005. `www.hadley.edu/Resources_list/Mathematics_Made_Easy_for_Children_with_Visual_Impairment.pdf`

[13] D. Walden. Profile of Eitan Gurari (1947–2009). *TUGboat* 30(2):159–162, 2009. `tug.org/TUGboat/tb30-2/tb95gurari.pdf`

◇ Susan Jolly
120 Dos Brazos
Los Alamos, NM 87544, USA
`easjolly (at) ix dot netcom dot com`