Typesetting Bangla in Unicode-aware TEX engines — from user experiences to development insights

Qutub Sajib

Abstract

Typesetting Bangla (also known as Bengali) script in TEX was first introduced more than 15 years ago from now through a transliteration system. The system would require the user to input Bangla texts with fonts from Roman script in a specific ASCII transliteration scheme and then process the input to display the texts using its own METAFONT-generated Bangla fonts. The transliteration-based system falls short in terms of, among others, its harder-to-read source file and its requirement of one particular Bangla typeface family. With the introduction of Unicode-aware TEX engines, XETEX for example, and the emergence of Unicode-compliant free Bangla fonts, new possibilities have evolved. Today both XETEX and LuaTEX support Bangla typesetting allowing the user to input texts directly with Unicode Bangla fonts in the editor; thus making the source file easy-to-read and eliminating the need for transliteration schemes. Although several years have passed since the XETEX system was first seen to support Bangla typesetting, it is still in a state where the *finest* typographic quality is nearly unachievable for this particular script. While working with different Unicode Bangla fonts with both of these engines, several rendering issues were observed. In order to ensure fine typographic quality in Bangla script, these issues, among others, need attention from users and developers.

1 Introduction

The language Bangla, also known as Bengali, is the sixth most-spoken language in the world as reported by *Ethnologue* it its latest edition. Native speakers of this language are primarily from Bangladesh, a rather small country in south Asia. Another good number of native speakers are from West Bengal, a province of India. The Bangla script is one of the thirteen major Indic scripts and has made its way in the Unicode standard. Publishing in this language has a good history of many centuries. Like other Indic scripts, typesetting Bangla in TEX has seen several attempts in last few years but the typographic quality is still to find its prime.

Apart from the beautiful rendering of mathematical contents in TEX, another goal of this typesetting system was the *finest* typographic quality [1]. The same philosophy can be expected while typesetting other scripts including Bangla in TEX. Considering the present-day situation of different attempts to typeset Bangla in TEX, this article tries to cover a few of them and provides some insights for future development.

2 Scope of this article

Before the Unicode standard was born for the purpose of writing most scripts of the world in computer, several attempts were taken to typeset Bangla in TEX using mostly ASCII based transliteration systems. Brief discussion of those systems are presented in Section 3. Following this, typesetting Bangla script in Unicode-aware TEX engines — XETEX and LuaTEX — are discussed with examples in subsequent sections. Some development ideas for this particular script are discussed in Section 11.

In this paper, when we say using XETEX we mean compiling the `.tex` file with the xelatex program; similarly when using LuaTEX we mean compiling the same file with lualatex. The TEX-specific examples presented in this paper were produced using the TEX Live 2019 distribution on a computer running GNU/Linux operating system, the Slackware 14.2 to be specific.

It is predictable that most Bangla documents would contain at least English, math and other scripts; but in this article we will consider typesetting only Bangla using two engines in TEX that supports the Unicode standard. This article does not cover the discussion about font selection techniques for different scripts except for Bangla. In order to learn selecting specific fonts for Roman (English) and maths along with Bangla, one may like to consult the fontspec package documentation.

3 An ASCII based transliteration system made the first attempt

As Anshuman Pandey mentioned [2], it was Avinash Chopde who first introduced Bengali (Bangla) in TEX in his itrans package. The subsequent developments in Bangla typesetting in TEX include the arosgn by Muhammad Ali Masroor (later dropped), "ItxBengali" by Shrikrishna Patil, "Bengali Writer" and bwti by Abhijit Das, and the bengali along with a preprocessor beng by Pandey. Although the transliteration based system works, it has the limitation of its harder-to-read source file (Figure 1). Also, the font used to typeset the Bangla script is limited to one particular font family.

The bengali-omega package by Lakshmi K. Raut was probably the first to introduce typesetting Bangla using either the transliteration system or Unicode based direct input [4]. Although this package has the capability of taking Unicode Bangla as input,

কে লইবে মোর কার্য, কহে সন্ধ্যা রবি
শুনিয়া জগৎ রহে নিরুত্তর ছবি ।
মাটির প্রদীপ ছিল, সে কহিল, স্বামি
আমার যেটুকু সাধ্য করিব তা আমি ।
— রবিন্দ্রনাথ ঠাকুর

```
{\bn ke la{}ibe mor kaarya, kahe sandhyaa rabi
"suni.yaa jagaT rahe niruttar chabi |
maa.tir pradiip chila, se kahila, sbaami
aamaar ye.tuku saadhya kariba taa aami |
                        -- rabindranaath .thaakur}
```

Figure 1: Typesetting Bangla in TEX with a transliteration based system as found in [2].

কে লইবে মোর কার্য, কহে সন্ধ্যারবি।
শুনিয়া জগৎ রহে নিরুত্তর ছবি।
মাটির প্রদীপ ছিল, সে কহিল, স্বামী,
আমার যেটুকু সাধ্য করিব তা আমি।

– রবীন্দ্রনাথ ঠাকুর

```
কে লইবে মোর কার্য, কহে সন্ধ্যারবি।\\
শুনিয়া জগৎ রহে নিরুত্তর ছবি।\\
মাটির প্রদীপ ছিল, সে কহিল, স্বামী,\\
আমার যেটুকু সাধ্য করিব তা আমি।

\□□□□□6□□\□□□□□□□□□□
-- রবীন্দ্রনাথ ঠাকুর
```

Figure 2: Typesetting Bangla in XETEX with Unicode fonts.

it uses the fonts as used in the package bangtex by Anshuman Pandey [3] to typeset the document.

4 Then Unicode-aware TEX engines came into play

With the introduction of XETEX and LuaTEX engines, and the fontspec package for selecting TrueType and OpenType fonts, typesetting Bangla using Unicode fonts came into reality. Today, a good number of Unicode compliant Bangla fonts are freely available that seem to work with these engines.

To start, one needs a keyboard layout that would allow to input Unicode Bangla characters in the editor. In most Linux systems, a keyboard layout called Probhat is available to input Unicode Bangla characters. Another popular alternative is the *Avro Keyboard* (`https://www.omicronlab.com/index.html`) available for installation in Mac OS X, Linux, and Windows operating systems. In emacs, three layouts are available, namely bengali-inscript, bengali-itrans, and bengali-probhat. None of the Unicode Bangla keyboard layouts available today were designed with TEX users in mind; hence one may need to switch the layout frequently in order to type TEX-special characters (e.g., &, % etc.). Using any of these keyboard layouts, one can input Unicode compliant Bangla characters directly in the editor. An appropriate font containing the Bangla script has to be set via the fontspec package (details in Section 6). Then, upon processing the `.tex` file either with xelatex or lualatex one gets the typeset document.

Typesetting with the Unicode based system in TEX has the advantage over the older systems in terms of its easy-to-read source file. As shown in Figure 2, the source file can be read easily compared to the source file in Figure 1 for the same text. It should be mentioned here that to typeset the same text as in Figure 1, we have corrected some spellings following the original source [5]. But this system is not free from shortcomings; as seen in Figure 2 some codes are not readable which should read "`\vskip6pt\raggedleft`" (more on this later).

5 Packages to support Unicode Bangla

The polyglossia package by François Charette is designed to provide support for typesetting Bangla script, along with many other scripts using Unicode fonts and Unicode-aware TEX engines. Although the package comes with language definitions files for different scripts including Bangla, it has many limitations. Even worse, the language definition file for Bangla script (`gloss-bengali.ldf`) that comes with this package has several words incorrectly spelled. It is well-known to the speakers and writers of Bangla language that the spelling of a few words has undergone changes and few are still in debate to agree on the "correct" spelling. But those misspelling in the file `gloss-bengali.ldf` are beyond any debate, as found in most dictionaries. For example, as of TEX Live 2019, the `gloss-bengali.ldf` file contains "সারণি", "খণ্ড", and few other words incorrectly spelled. (The author of this paper is not convinced to write incorrect spelling; so in order to see the *incorrectly* spelled words, one has to look into the above-mentioned file for the corresponding words).

The polyglossia package does come with few good features. It has the option to change the numbering style of LATEX counters; i.e., one can typeset LATEX counters with Bangla numerals. As the author of latexbangla stated, the polyglossia package's current support does not offer any easy way to select fonts with Bangla script. This package tries to solve the problem by introducing control sequences to select several fonts for Bangla script but it has its shortcomings, too. As of our knowledge, there are no Unicode Bangla fonts designed to be used especially with TEX, the package sticks with the limited fonts available freely and tries to make bolded and monospaced texts using different fonts from different designers. This results into using the "AutoFakeBold" and "AutoFakeSlant" features of existing "Regular" style fonts. As a result, the typeset document becomes something passable but not of great aesthetic taste. The fonts used in latexbangla package are not available with TEX Live distribution. Besides, its dependence on the ucharclasses package makes it not usable with LuaTEX.

6 Unicode compliant Bangla fonts to try with TEX

The TEX Live 2019 comes with the gnu-freefont package which contains Unicode fonts in both TTF and

দেশকালের বক্রতার জন্যই অভিকর্ষ—এ-কথা রবীন্দ্রনাথ এমন সময়ে বাঙালি-
পাঠককে বলেছিলেন, যখন সে-সম্বন্ধে খুব বেশি ঔৎসুক্য এ-দেশে ছিল না। আসলে
এ-দেশ হলো কবিতা আর গানের দেশ। রবীন্দ্রনাথ তাঁর অজস্র কবিতায় আর গানে
ছন্দ ও সুরের ঝংকারের প্রতি বাঙালি মনের চিরন্তন আকর্ষণকে রূপ দিয়েছেন; কিন্তু
তাঁর নিজের মনটি যে আশ্চর্যজনকভাবে বিজ্ঞানানুগ ছিল—এ-খবর কজনে রাখেন?

দেশকালের বক্রতার জন্যই অভিকর্ষ—এ-কথা রবীন্দ্রনাথ এমন সময়ে বাঙালি-
পাঠককে বলেছিলেন, যখন সে-সম্বন্ধে খুব বেশি ঔৎসুক্য এ-দেশে ছিল না। আসলে
এ-দেশ হলো কবিতা আর গানের দেশ। রবীন্দ্রনাথ তাঁর অজস্র কবিতায় আর গানে
ছন্দ ও সুরের ঝংকারের প্রতি বাঙালি মনের চিরন্তন আকর্ষণকে রূপ দিয়েছেন; কিন্তু
তাঁর নিজের মনটি যে আশ্চর্যজনকভাবে বিজ্ঞানানুগ ছিল—এ-খবর কজনে রাখেন?

Figure 3: Rendering of Bangla script in XeTeX using Free Serif fonts: top: using MikTeX 2.8; bottom: using TeX Live 2019.

OTF formats and cover a wide range of Unicode character set. Fonts for Bangla script are available in serif and sans-serif version including regular and slanted styles. Unfortunately, no bold or bold italic fonts are available for Bangla script. Figure 3 shows rendering of few lines of Bangla script using Free Serif fonts with xelatex. In this figure, the typeset script on top is from the author's own typesetting that was done sometime in 2009 using MikTeX 2.8 distribution. The same text when compiled with xelatex using TeX Live 2019, however, produces a different result. To be specific, few conjunct characters or ligatures are incorrectly rendered in TeX Live 2019. The reason behind this, whether with the fonts or the xelatex program, needs a fix in upcoming releases.

Besides the Free Serif fonts included in TeX Live 2019, Google noto fonts (`https://www.google.com/get/noto/`) include "Noto Serif Bengali" and "Noto Sans Bengali" in TTF format. The serif version comes with regular and bold styles while the sans-serif version contains other *seven* styles apart from the regular and bold. All these fonts can be used to typeset Unicode Bangla in TeX but they have to be downloaded and setup correctly so TeX finds them. The OTF version of Google noto fonts *are* available in TeX Live 2019 but they do not include the fonts for Bangla script.

There are a good number of Bangla fonts that can be freely downloaded from online. The Avro Keyboard website has a dedicated page to download fonts (`https://www.omicronlab.com/bangla-fonts.html`) and the Ekushey have their own page (`http://ekushey.org/index.php/page/33`). Most Unicode Bangla fonts available today, except the two fonts mentioned above, do not have their slanted/italic, bold etc. styles. Those fonts were not designed with professional publishing in mind, neither TeX users in mind. When using XeTeX, users can use the "AutoFakeBold" and "AutoFakeSlant" features of fontspec package but the result is somewhat acceptable. In LuaTeX, however, these features are not acceptable. So at this stage, users may have to stick with the limited styles available.

Besides the limited number of Unicode Bangla fonts or their availability in limited styles, rendering of Bangla characters in XeTeX and LuaTeX needs deeper attention. In order to experiment with both of these engines, we selected three fonts to render the same texts. First one is the Free Serif fonts available in TeX Live 2019, second the "Noto Serif Bengali" fonts, and third the "Lohit Bengali" fonts. The last one of these three is available in most Linux distributions; in case it is not, can be downloaded from the Ekushey font page mentioned above. The font setup used in producing all examples in this article is given below. The "Noto Sans Bengali" is used in Figure 2 to show the verbatim texts. Considering the x-height of Free Serif fonts as "normal", other fonts were scaled accordingly to get the identical output. This font setup works with both xelatex and lualatex:

```
\usepackage{fontspec}
%
\newfontfamily{\freeserifbn}{FreeSerif.ttf}
  [Script=Bengali, Ligatures=TeX]
\newfontfamily{\notoserifbn}
  {NotoSerifBengali-Regular.ttf}
  [Script=Bengali, Scale=0.85, Ligatures=TeX]
\newfontfamily{\notosansbn}
  {NotoSansBengali-Light.ttf}
  [Script=Bengali, Scale=0.85, Ligatures=TeX]
\ifxetex
\newfontfamily{\lohitbn}{lohit_bn.ttf}
  [Path=/usr/share/fonts/TTF/,
  Script=Bengali, Scale=0.82, Ligatures=TeX]
\else
\newfontfamily{\lohitbn}{lohit_bn.ttf}
  [Script=Bengali, Scale=0.82, Ligatures=TeX]
\fi
```

7 The mystery of DOTTED CIRCLE

The DOTTED CIRCLE belongs to the "Geometric-Shapes" of the Unicode block and holds the character code U+25CC. It can be typeset with the TeX command `\char"25CC`, using the font setup mentioned above and compiling with either xelatex or lualatex producing: ◌ (here with Free Serif font). In many non-Roman scripts the DOTTED CIRCLE is used to indicate the position of various diacritical marks. In Bangla scripts, for example, short form of a vowel (known as *kaar*) replaces the actual vowel when the vowel comes after a consonant. Different *kaars* have their different positions to sit with consonants. The DOTTED CIRCLE is helpful to visualize the position of a *kaar* to sit with a consonant when the short form is typeset independently.

Figure 4: Typesetting short forms of vowels, independently or with consonants, produces different results in `xelatex` and `lualatex` for different fonts.

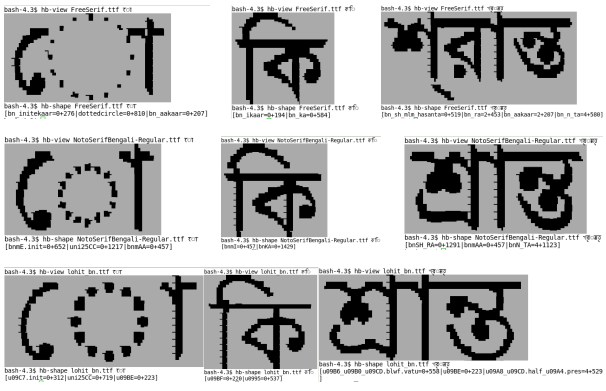

Figure 5: Rendering of Bangla script in HarfBuzz using different fonts in Linux shell.

Figure 6: Rendering of Unicode Bangla script in Word processors.

In Figure 4, all ten vowel short forms are shown independently (first row of a pair) and when they are combined with the consonant *kô* ক (second row of a pair). The example is shown in three pairs — using three fonts: Free Serif, Noto Serif Bengali, and Lohit Bengali — compiling with both xelatex and lualatex.

The problem arises when one wants to typeset short forms of vowels without the DOTTED CIRCLE. It is particularly necessary when children are taught these characters in schools. Ideally, in TEX-engines, one should be able to typeset the short forms of vowels. To our surprise, in its current situation, we cannot typeset them without the DOTTED CIRCLE with xelatex. We say *ideally* because of few reasons: (i) when a consonant takes a short form of vowel, the DOTTED CIRCLE disappears implying the presence of those characters independent of the DOTTED CIRCLE in current font; (ii) as described in the Unicode standard, the purpose of DOTTED CIRCLE is to show the relative position of the short form of a vowel; and (iii) when seen in a font viewer like FontForge, short forms of vowels can be found as independent glyphs. Therefore, rendering of vowel short forms in xelatex with a DOTTED CIRCLE can be considered as a bug.

8 Bangla script in HarfBuzz and Word processors

To understand the mystery of DOTTED CIRCLE in TEX, we can have a look in the output produced by the text rendering stack HarfBuzz as this program works behind the XETEX engine as well as in many Word processors. Two modules hb-view and hb-shape are available as part of HarfBuzz Indic Shaper and can be used in Linux shell to get the rendered output of a Unicode script. As shown in Figure 5, HarfBuzz produces the same result for different fonts as we have seen with xelatex. The same result is also found in LibreOffice, OpenOffice, and MS Word (Figure 6).

From these examples it can be said that the HarfBuzz program is responsible for the unusual rendering of DOTTED CIRCLE in XETEX when typeset with the short forms of vowels. This assumption is also supported by the fact that the LuaTEX engine does not depend on HarfBuzz to render Unicode supported scripts and it produces expected results in terms of DOTTED CIRCLE rendering: (first row of second pair in Figure 4).

9 Dealing with hyphenation

In modern-day Bangla publications hyphenation is hardly seen, either because of technical limitations or lack of interest. Fortunately, polyglossia package supports hyphenation for Bangla script which works with both xelatex and lualatex. But unfortunately, the hyphenation rules in this package are too simple for this script. For example, one may like the word "একখানি" to be hyphenated between এক and খানি instead of what we see in Figure 7 (right).

10 Working with colors

Use of colors in texts can significantly improve the visual as well as readability for particular types of contents. Colorful texts can be essential in books written for children. For Bangla script, sometimes it is desirable to typeset different parts of a conjunct character in different colors to help children learn them separately. A good example can be to flag a

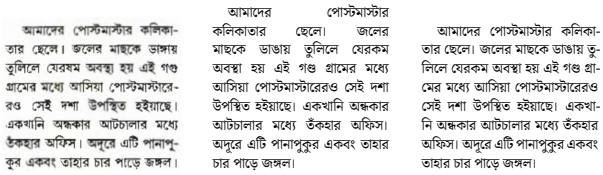

Figure 7: Hyphenation supported by the polyglossia package works with both xelatex and lualatex.

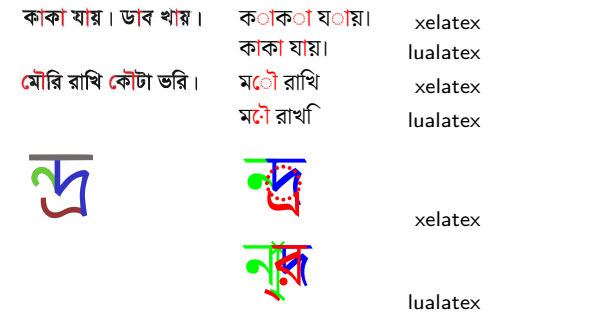

Figure 8: Using different colors for different parts of a conjunct or ligature is a challenge; left: example from online (`http://tiny.cc/4xly9y`) right: TEX-output.

kaar in a different color when it sits with a consonant, as can be seen in textbooks for children (Figure 8, top left). The same result in TEX, either with XETEX or LuaTEX is currently not achievable. The same problem arrives when trying to typeset a ligature flagging its different parts with different colors.

11 What is next?

To address the *finest* typographic quality in Bangla script, several things can be taken into consideration. A font family can be designed with TEX users in mind and a supporting macro package can be developed. Use of only the fontspec package at the primary stage would be a good idea; integration with the polyglossia package may come next.

To solve the rendering issues especially with the DOTTED CIRCLE in XETEX and ligatures in LuaTEX, attention could be made with the HarfBuzz developers. For now, because there are limitations in both XETEX and LuaTEX, it is probably good to experiment with both of these engines. Eventually we may want to set up with one particular engine. This could take us to the LuaTEX as this engine is supposed to be the successor of pdfTEX.

Until a new dedicated font family is designed for the Bangla script, the Noto Bengali (both serif and sans-serif) fonts can be included in the future versions of TEX Live. This would allow the users to try Unicode-aware TEX engines with at least two font families including the already existing Free Serif and Free Sans fonts.

A keyboard layout could be designed for the purpose of Unicode Bangla character input making it emacs- and TEX-friendly. In designing such a layout, the TEX escape character (“\”), percent (“%”), ampersand (“&”) etc. keys can be left as is, so that these keys can be used to format Bangla texts without having to switch the keyboard layout.

12 Conclusion

The present-day Bangla publishing industry is mostly not using the fine typographic power of TEX. The reasons behind this are many of which few are discussed here. Interests in solving those issues have been seen in recent years. Although usage of TEX in Bangla fiction books could be a bigger challenge mostly due to non-TEXnical reasons, a good number of science books could be expected being typeset in TEX. For this to happen, the current limitations of Unicode engines and fonts need to be addressed. The few insights we were able to uncover in this article could lead us to the beginning of the *finest* typographic quality in Bangla publishing.

References

[1] D. E. Knuth. *The TEXbook*, vol. A of *Computers & Typesetting*. Addison–Wesley Publishing Company, Massachusetts, 2012.

[2] A. Pandey. Typesetting Bengali in TEX. *TUGboat* 20(2):119–126, 1999.

[3] A. Pandey. *Bengali for TEX*, 2.0 edition, 2002.

[4] L. K. Raut. *Typesetting Bengali in Ω using Velthuis Transliteration or Unicode Text*, 2006.

[5] R. Tagore. The complete works of Rabindranath Tagore. Retrieved: 15 July 2019. `http://tiny.cc/tagore`

◇ Qutub Sajib
China University of Geosciences, Wuhan
388 Lumo Road, Wuhan 430074
China
`qsajib71@gmail.com`