## Typesetting the Bangla script in Unicode TEX engines — experiences and insights

Md Qutub Uddin Sajib

**Abstract**

The typesetting of Bangla (also known as Bengali) script in TEX was first introduced more than 15 years ago through transliteration-based systems. These systems have shortcomings: among others, the source files are harder to read and they require one or two particular Bangla typeface families for typesetting. With the introduction of Unicode-aware TEX engines, such as XETEX, and the emergence of Unicode-compliant free Bangla fonts, new possibilities have evolved. Today both XETEX and LuaTEX, as available in TEX Live 2019, support Bangla typesetting allowing the user to input the text directly with Unicode Bangla fonts in the editor. Although several years have passed since the XETEX system was first seen to work, it is still in a state where the *finest* typographic quality is nearly unachievable for this particular script. Several rendering issues were observed while working with Unicode Bangla fonts in four Unicode-aware TEX engines. Precision typesetting of the Bangla script in Unicode TEX requires attention in terms of fonts, rendering, hyphenation, use of colors, and more.

### 1 Introduction

The language Bangla, also known as Bengali, is one of the ten most-spoken languages in the world, as reported by *Ethnologue* in its 2019 edition. Native speakers of this language are mainly from Bangladesh, a small but populous country in south Asia. Another good number of native speakers are from the West Bengal state of India. The Bangla script, the written form of the Bangla language, is one of the thirteen major Indic scripts and has made its way into the Unicode Standard. Publishing in this script has a history of many centuries. Like other Indic scripts, typesetting of the Bangla script in TEX has seen several attempts in the last few years but typographic quality has yet to reach a peak.

Apart from beautiful rendering of mathematical contents in TEX, another goal of this typesetting system is the *finest* typographic quality [5]. The same philosophy can be expected in typesetting other scripts, including Bangla. Considering the present-day support of the Bangla script in TEX, this article discusses a few rendering issues, mostly gathered from the author's day-to-day typesetting experiences; it also provides some insights for future development.

### 2 Scope of this article

Before the Unicode Standard was created to enable the writing of most scripts of the world on computers, the attempts to typeset Bangla script in TEX were confined to ASCII-based transliteration systems. Brief discussions of ASCII- and Unicode-based typesetting of this script are presented in sections 3 and 4. The TEX packages and fonts available today that support Unicode Bangla typesetting are discussed in sections 5 and 6.

It is predictable that most Bangla documents contain at least English, math, and possibly other scripts. In this article, however, we have considered typesetting of the Bangla script only, using the four TEX engines that support the Unicode Standard. This article does not cover the discussion on font selection techniques for different scripts except Bangla. For information on selecting specific fonts for Roman (English) and math along with Bangla, the fontspec package [11] can be consulted.

TEX engines known to support the Unicode Standard are XETEX, LuaTEX, HarfTEX, and LuaHBTEX. The first two are available in TEX Live 2019; the last two via tlcontrib or Akira Kakuto's w32tex and w64tex distributions (`http://w32tex.org/`). We used all four engines to typeset some text of Bangla script to observe the rendering with different fonts (Section 7), hyphenation (Section 9), and use of colors (Section 10). Some development ideas for this particular script are discussed in Section 12.

In this paper, using XETEX means compiling the `.tex` file with xelatex; using LuaTEX, HarfTEX, and LuaHBTEX means compiling the same file with lualatex, harflatex, and luahblatex, respectively. The TEX-specific examples presented here were produced using the TEX Live 2019 distribution on a computer running the GNU/Linux operating system (Slackware 14.2). The HarfTEX and LuaHBTEX engines were installed via tlcontrib, following the instructions at `https://contrib.texlive.info/`.

### 3 ASCII-based transliteration systems

ASCII-based systems to typeset the Bangla script in TEX were first seen to work more than 15 years ago. The two transliteration schemes known to support the Bangla script are `ITRANS` (**I**ndian languages **TRANS**literation) by Avinash Chopde and the Velthuis system by Frans Velthuis. Both of these schemes were primarily developed for the Devanagari script. Later, the schemes were adapted to typeset the Bangla script in TEX.

The typeface families that work with `ITRANS` include the "SonarGaon" (`sgaon`) fonts by Anisur Rahman [7] and "AroSgaon" fonts by Muhammad

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 of Bangla script in TEX with a transliteration system using the METAFONT-generated "Bengali" fonts (source: [7]).

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।

\vskip6pt\raggedleft
-- রবীন্দ্রনাথ ঠাকুর
```

**Figure 2**: Typesetting of Bangla script in XETEX with Unicode fonts (spelling of a few words, as they appeared in Figure 1, were corrected following [14]).

Masroor Ali [1]. The latter was available with its METAFONT sources and was replaced by the Type 1 "ITXBengali" fonts of Shrikrishna Patil in ITRANS. The bwti (Bengali Writer TEX Interface) package by Abhijit Das included METAFONT-generated "Bengali" fonts and worked in TEX through a special interface.

The bengali package [8] by Anshuman Pandey uses the Velthuis transliteration scheme instead of ITRANS. It uses the latest version of Das's "Bengali" fonts for typesetting. The bangtex package [6] by Palash Baran Pal includes class files and METAFONT sources for its "Bangla" fonts. The Type 1 fonts for this package were created by Ananda Kumar Samaddar [6] and are included in the bengali-omega package [10] of Lakshmi K. Raut. The latter uses the Velthuis transliteration scheme. It also supports Unicode-based input but would convert the Unicode text into the transliteration scheme for typesetting.

The transliteration-based systems require the user to input Bangla text in a specific scheme with fonts from the Roman script. Then the text would be processed with preprocessors for typesetting in TEX. Although these systems work, the source file is harder to read (Figure 1). They seem to use the "Bengali" fonts (from bwti) or "Bangla" fonts (from bangtex) to typeset the document. The typographic quality of these fonts may not be comparable with fonts we see in modern Bangla publications.

## 4 Unicode-aware TEX engines

With the introduction of XETEX and LuaTEX around 2007, and the fontspec package for selecting TrueType and OpenType fonts, typesetting of Bangla in TEX using Unicode fonts became a reality. Today, a good number of Unicode-compliant Bangla fonts are freely available that work with these engines.

To start, one needs a keyboard layout that supports the input of Unicode Bangla characters in an editor. In most GNU/Linux systems, a keyboard layout called Probhat is available for this purpose. A popular alternative is the *Avro Keyboard*, available for free (https://www.omicronlab.com/index.html), which can be installed in GNU/Linux, Mac OS X, and Windows systems. In the emacs editor, as of version 26.2, 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 special TEX characters (\, %, &, etc.). 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 with xelatex, lualatex, harflatex, or luahblatex one gets the typeset document.

The Unicode-based systems in TEX for this script have many advantages over the older systems. For example, the source file is now easy to read (Figure 2, right versus Figure 1, right). In addition, any font that contains the glyphs for this script can be used for typesetting. However, the current situation is not free from shortcomings. The verbatim text in Figure 2 (right), which should read "\vskip6pt\raggedleft", is unreadable because the font used there contains glyphs only from the Bangla script. Other shortcomings that we have observed are discussed in the following sections.

## 5 LATEX packages for Unicode Bangla

The polyglossia package by François Charette [2] is designed to provide support for typesetting Bangla script, along with other scripts, using suitable Unicode fonts and TEX engines. It provides a style file (begalidigits.sty) for this script that translates the Arabic numerals into Bangla numerals. The language definition file (gloss-bengali.ldf) implements the Bangla numerals in LATEX counters. It also provides Bangla translation for the names of LATEX sections and counters, and for the Gregorian calendar months.

The latexbangla package by Adib Hasan [3] introduces some control sequences to select Bangla fonts. To our knowledge, there are no Unicode Bangla fonts designed to be used with TEX. As such, the package sticks with the limited fonts available today and makes bold, slanted, and monospaced text using fonts from different designers. It uses the AutoFakeBold and AutoFakeSlant features to produce *fake* styles. As a result, the typeset document looks something passable but not of great aesthetic taste. The fonts used

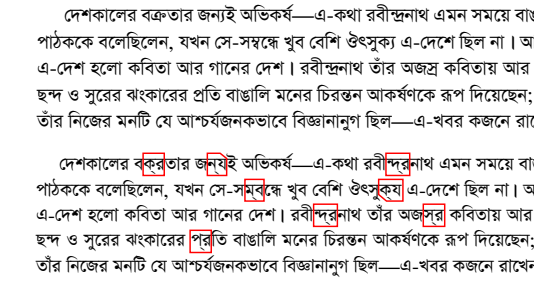


**Figure 3**: Rendering of Bangla script in XeTeX using Free Serif fonts: top: using MiKTeX 2.8; bottom: using TeX Live 2019 (red boxes indicate wrong rendering).

in this package are not available in TeX Live 2019. Besides, its dependence on the ucharclasses package [4] makes it unusable with other engines than XeTeX.

## 6 Unicode-compliant Bangla fonts and TeX

TeX Live 2019 comes with the gnu-freefont package which contains Unicode fonts in both TTF and OTF formats and covers a wide range of the Unicode character set. Fonts for the Bangla script are available in serif and sans-serif versions, in regular and slanted styles. Unfortunately, no bold or bold italic fonts are available for this particular script. Figure 3 shows the rendering of a few lines of Bangla script using Free Serif fonts with xelatex. In this figure, the typeset text on top is from the author's own typesetting for a book [9] that was compiled in 2012 with xelatex using the MiKTeX 2.8 distribution. The same piece of text was found producing a bit different output when compiled with xelatex using TeX Live 2019; to be specific, a few conjunct characters and ligatures are incorrectly rendered. This rendering problem, whether it concerns the Free Serif fonts or the xelatex program, needs to be fixed in the future.

Besides the Free Serif fonts in TeX Live 2019, the noto font family from Google (`https://www.google.com/get/noto/`) includes Noto Serif Bengali and Noto Sans Bengali fonts in TTF format. The serif version includes the regular and bold styles while the sans-serif version contains *seven* other styles. All these fonts can be used to typeset Unicode Bangla in TeX but they must be downloaded and set up correctly so that TeX finds them. The OTF version of this font family *is* available in TeX Live 2019 but it does not include the fonts for Bangla script.

A good number of Unicode-compliant Bangla fonts are available today and can be downloaded for free. The *Avro Keyboard* website has a dedicated page for such fonts (`https://www.omicronlab.com/bangla-fonts.html`); the *Ekushey* project also has a page (`http://ekushey.org/index.php/page/33`) for this purpose. Most Unicode Bangla fonts available today, except the two mentioned above, are not available in slanted, italic, bold, etc., styles. This is probably due to the fact that those fonts were not designed with professional publication in mind; also, not with TeX users in mind. When using XeTeX, the AutoFakeBold and AutoFakeSlant features can be used and the result is somewhat acceptable. In LuaTeX, HarfTeX, and LuaHBTeX, however, these features are not supported.

Besides the lack of publication-quality Unicode Bangla fonts, the rendering of Bangla script in Unicode TeX engines needs deeper attention. To experiment with the four engines available today, we selected three fonts to typeset the same piece of texts. The first one is the Free Serif font available in TeX Live 2019, second one is the Noto Serif Bengali, and third one is the Lohit Bengali font. This last is available in most GNU/Linux distributions; otherwise, it can be downloaded from the *Ekushey* font page mentioned above. The font setup used for all examples in this article is given below. The Noto Sans Bengali font was used in Figure 2 to typeset the verbatim text. Considering the x-height of Free Serif fonts as "normal", other fonts were scaled accordingly to get the identical typeset output. This font setup works with xelatex, lualatex, harflatex, and luahblatex:

```
\usepackage{fontspec}
\usepackage{ifxetex}
%
\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=1, 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 DOTTED CIRCLE in rendering the Bangla script

The glyph named DOTTED CIRCLE is in the Geometric-Shapes Unicode block, assigned the character code U+25CC. Thus it can be typeset with the TeX command `\char"25CC`, using the font setup above

and compiling the input file with xelatex, lualatex, harflatex or luahblatex to get: ◌ (here with the Free Serif font). In non-Roman scripts, the DOTTED CIRCLE can be used as the base character to typeset a combining mark [15] which would otherwise be combined with a real base character of a script.

In the Bangla script, a *vowel sign* or the short form of a vowel (known as *kaar*) replaces the glyph of that vowel when the vowel is followed by (or modifies) a consonant (base character). For example, the vowel "আ" (BENGALI LETTER AA) has the short form "া" (BENGALI VOWEL SIGN AA). When the consonant "ক" (BENGALI LETTER KA) is modified by the vowel আ, the vowel sign (া) replaces the actual vowel and the consonant-vowel conjunct is typeset as "কা". Similarly, several consonants have short forms (known as *phalas*); they replace their actual glyphs (bases) when two or more consonants are combined to produce a conjunct character.

The *kaars* and *phalas* in Bangla script can collectively be called combining marks. Since different *kaars* and *phalas* have different positions to stand with a base character, the DOTTED CIRCLE can be helpful to visualize the actual position of a combining mark when a *kaar* or *phala* is typeset independently.

In Figure 4, the most common vowel signs are shown as independently typeset combining marks (first row of a pair) and combined with the consonant ক (second row of a pair). The example is shown using three fonts, Free Serif, Noto Serif Bengali, and Lohit Bengali, compiling each with our four Unicode TEX engines. As seen in this figure, a DOTTED CIRCLE unexpectedly appears when a *vowel sign* is typeset with xelatex; this is not the case with lualatex, harflatex or luahblatex. On the other hand, when vowels are combined with a consonant, xelatex seems to render the script correctly; the other three engines fail except for few consonant-vowel combinations with the Lohit Bengali font. In the case of consonant-consonant combinations, xelatex failed for specific combinations with the Free Serif font (Figure 4, middle column, top), although it worked for other combinations (not shown in figure).

Problem arises when one intends to typeset a vowel sign or other combining marks *without* the DOTTED CIRCLE. It is particularly necessary when these signs are taught to children. Ideally, in TEX, one should be able to typeset the combining marks (*kaars* or *phalas*) independent of the DOTTED CIRCLE. We say *ideally* because: (i) when a consonant combines with a vowel sign, the DOTTED CIRCLE disappears implying the presence of these signs as independent glyphs in the current font; (ii) as described in the Unicode Standard and mentioned previously, the DOTTED CIRCLE *can be* used as a base character to a combining mark; and (iii) in a font viewer like FontForge, the vowel signs are indeed found as independent glyphs. Therefore, rendering of the combining marks, especially the vowel signs, in xelatex with a DOTTED CIRCLE even when it is undesirable can be considered as a bug (more in Section 11).


**Figure 4**: Typesetting of the vowel signs, independently and as a conjunct with a consonant (left column), and a consonant-consonant conjunct character (middle column) in four engines using the Free Serif (top), Noto Serif Bengali (middle), and Lohit Bengali (bottom) fonts.

## 8 Rendering of Bangla script in HarfBuzz, word processors, and elsewhere

In order to understand the appearance of DOTTED CIRCLE in typesetting the combining marks, it is logical to look into the output produced by the text rendering stack HarfBuzz (`https://www.freedesktop.org/wiki/Software/HarfBuzz/`). This software is known to work behind the XETEX engine as well as in many word processors, text editors, web browsers, and probably elsewhere. Two modules hb-view and hb-shape are available in the HarfBuzz Indic Shaper and can be used on Unix systems 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 in Figure 4.

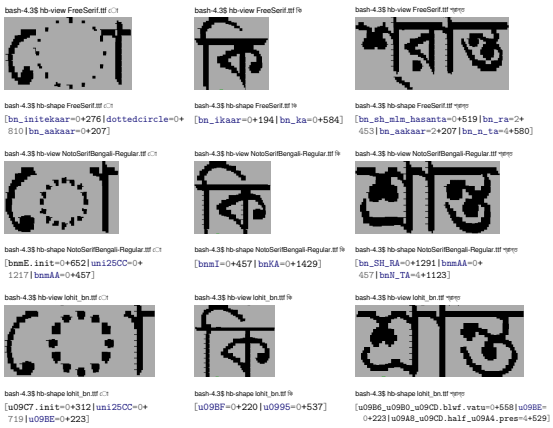

**Figure 5**: Rendering of Bangla script in HarfBuzz using different fonts in a GNU/Linux shell.

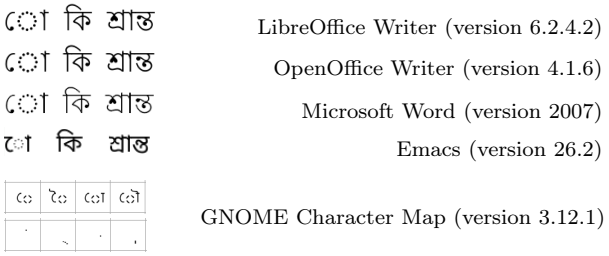


**Figure 6**: Rendering of the Unicode Bangla in word processors and emacs (top); Bengali (Bangla) and Hebrew scripts in GNOME Character Map (bottom).

When Unicode Bangla characters are input in word processors or the emacs editor using a compatible keyboard layout, the rendering (Figure 6) can be seen to be the same as the HarfBuzz-generated output (Figure 5). In the address bars of popular web browsers, as tested in Firefox and Chromium, the same kind of rendering was also observed (not shown in figure). The GNOME Character Map, however, displays the combining marks of Bangla and Hebrew scripts differently in terms of the DOTTED CIRCLE (Figure 6).

The examples in figures 5 and 6 imply that HarfBuzz is responsible for the unexpected appearance of DOTTED CIRCLE in XeTeX when the vowel signs are typeset independently. This assumption is also supported by the fact that the LuaTeX engine produces expected results in terms of DOTTED CIRCLE (first row of second pairs in Figure 4), as it does not depend on HarfBuzz for rendering. The HarfTeX and LuaHBTeX rendering (third and fourth pairs in Figure 4), however, seems puzzling as they are known to use HarfBuzz but output LuaTeX-like rendering, although XeTeX-like rendering could be expected.

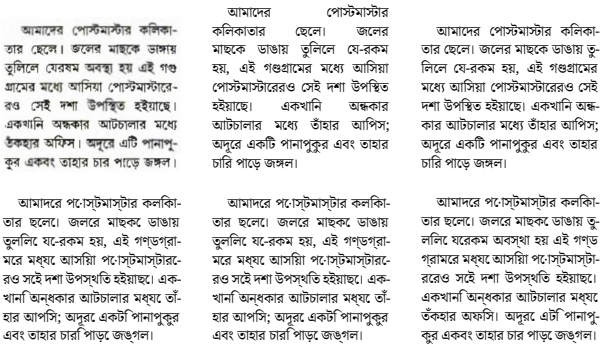

**Figure 7**: Hyphenated text from [12, p. 391] (top left), xelatex with no hyphenation (top middle). polyglossia-generated hyphenation in: xelatex (top right), lualatex (bottom left), harflatex (bottom middle), and luahblatex (bottom right). (The spelling of a few words was corrected following [13]).

## 9 Dealing with hyphenation

In modern-day Bangla publications, hyphenation is hardly seen, either because of technical limitations or lack of interest. Fortunately, the polyglossia package supports hyphenation for Bangla script which seems to work well (Figure 7). All four Unicode TeX engines were found to be working with hyphenation, although the lualatex, harflatex, and luahblatex have rendering issues as discussed previously. However, the hyphenation rule `hyphenmins={2,2}` as found in the `gloss-bengali.ldf` file is probably too low for this script. It was also found that any changes made in this file, e.g., `hyphenmins={3,3}` has the desired effect but using the same in a `.tex` file has no effect in the hyphenation pattern.

## 10 Typesetting with color

Use of colors in text can significantly improve the visual as well as readability for particular types of contents. Colorful text can be essential in books written for children. For Bangla, sometimes it is desirable to typeset different parts of a conjunct character in different colors to help children learn and recognize them with ease. A good example is to flag a *kaar* in a different color than its consonant base, as can be seen in textbooks for children (Figure 8, first column, first and second row). Such use of color was not found to be working with xelatex; the lualatex, harflatex and luahblatex were found to be working in a few combinations (Figure 8, second column, first and second row) but failing in other cases. Similarly, typesetting a ligature with its different parts flagged in different colors was found to be not possible (Figure 8, third row). The code to typeset the colorful

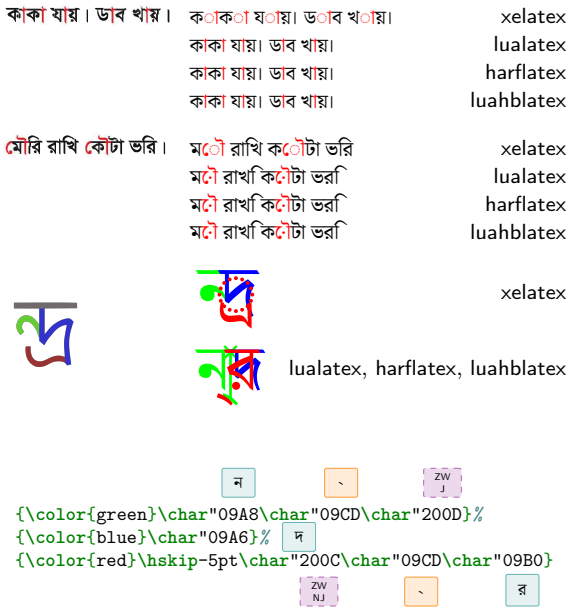


**Figure 8**: Use of different colors for different parts of a conjunct or ligature is a challenge; left: example from `http://tiny.cc/bdnctb-classone` (top and middle rows) and `http://tiny.cc/4xly9y` (bottom row); middle column: TEX output.

ligature (Figure 8, fourth row) is somewhat complex and may not be very useful in real life typesetting.

## 11 The DOTTED CIRCLE mystery, revisited

To put it simply, one should be able to typeset any *glyph* of a font independently, that is, without the DOTTED CIRCLE as a base when expected. As seen in previous examples, a few glyphs cannot be typeset independently. To understand this behavior in TEX, we try to deconstruct the rendering of DOTTED CIRCLE using the TEX primitive `\char"` and typesetting some Unicode character codes from Bangla script (Figure 9). In this example, the appearance of DOTTED CIRCLE (`\char"25CC`) can be considered as *unusual* because a single call of `\char"09BE` is seen to typeset both "◌া" (`\char"25CC`) and "া" (`\char"09BE`). Also, both `\char"09BE` and `\char"25CC\char"09BE` commands are seen to produce the same typeset output. Other examples in this figure further suggest that the rendering may not be called satisfactory.

The apparent problem that DOTTED CIRCLE *cannot* be separated even when it is undesired in typesetting Indic scripts was previously reported as a bug in LibreOffice (`http://tiny.cc/bsebez`), and also mentioned on the xetex mailing list (`http://tiny.cc/atgbez`). The LibreOffice page has declared this issue as not a bug while no conclusion was found on the list. However, Khaled Hosny gave

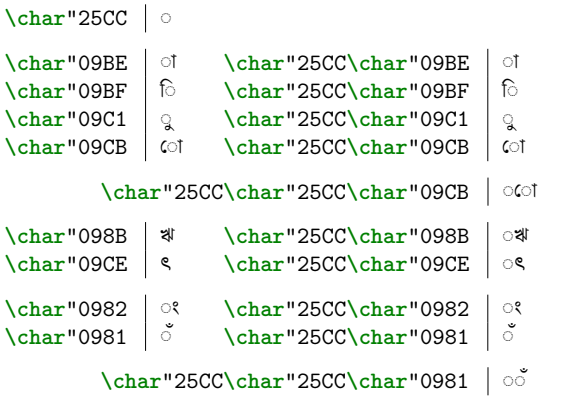


**Figure 9**: Deconstructing the DOTTED CIRCLE in XETEX with base characters from Bangla script.

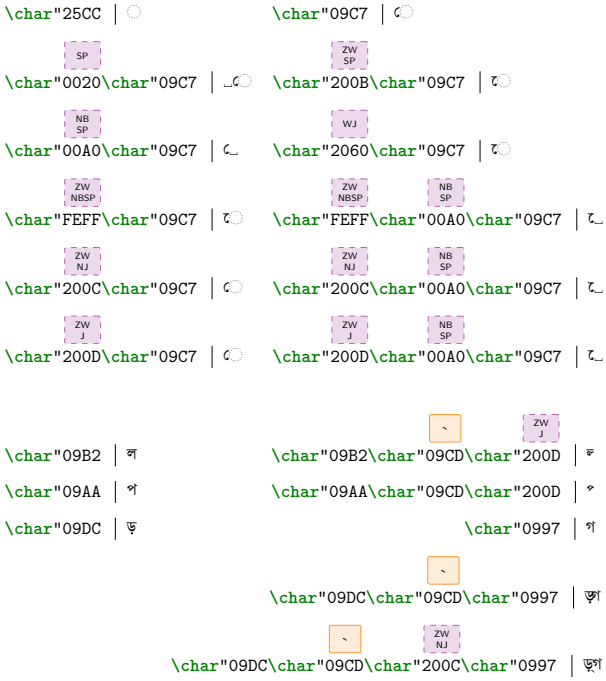


**Figure 10**: Deconstructing the DOTTED CIRCLE to typeset combining marks and special conjuncts in XETEX with Bangla script.

advice on the LibreOffice page to use a SPACE or NO-BREAK SPACE before the given glyph when DOTTED CIRCLE is undesired.

The NO-BREAK SPACE (NBSP) was found to work to "remove" the DOTTED CIRCLE when used before a combining mark while use of a SPACE (SP), as predicted, did not work (Figure 10, top). However, this trick for removing DOTTED CIRCLE may not be acceptable because it actually *replaces* the DOTTED CIRCLE with a space (shown with a "␣" in figure).

Other combinations were tested, using ZERO WIDTH SPACE (ZWSP), WORD JOINER (WJ), ZERO

WIDTH NO-BREAK SPACE (ZWNBSP), ZERO WIDTH NON-JOINER (ZWNJ), and ZERO WIDTH JOINER (ZWJ). The result is interesting as we get ৎ and ৎ (notice the horizontal stroke on top, known as *maatra* in Bangla script) using different combinations but the same character code (U+09C7). The ZWNJ and ZWJ characters were found useful in typesetting short forms of consonants and special conjunct characters (Figure 10, bottom).

## 12 What next?

In order to achieve the *finest* typographic quality in Bangla script, several things can be taken into consideration. The rendering issues especially with the DOTTED CIRCLE in XeTeX, and conjunct characters and ligatures in other engines may take priority. Contact can be made with the HarfBuzz developers for this purpose. For now, because rendering issues are observed in XeTeX, LuaTeX, HarfTeX, and LuaHBTeX, it is probably necessary to experiment with all these engines. Eventually we may want to settle on one particular engine.

The Noto Bengali fonts, both serif and sans-serif, 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. Eventually, a dedicated font family for the Bangla script should be designed especially with TeX users in mind and a supporting macro package developed. Supporting only the fontspec package at the primary stage would be fine; integration with the polyglossia package may come next.

A keyboard layout can be designed for the purpose of Unicode Bangla character input making it emacs- and TeX-friendly. In this design, keys for the special TeX characters (\, %, &, etc.) can be retained, so that these keys can be used to format Bangla text without having to switch keyboard layouts. Keys for the Unicode characters NO-BREAK SPACE, ZERO WIDTH NON-JOINER, ZERO WIDTH JOINER, ZERO WIDTH NO-BREAK SPACE, etc., should also be in the layout design since these characters can be helpful in typesetting combining marks and special conjuncts. As these characters are not visible in the editor when input directly, they can be even better input with newly-defined TeX macros.

## 13 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 a few are discussed here. Interest in solving those issues has been seen in recent years. Although use of TeX in typesetting Bangla fiction books might be a bigger challenge, mostly due to non-TeXnical reasons, a good number of science books can be expected in the future. For this to happen, the current limitations of Unicode TeX engines and fonts need to be addressed. The few insights we were able to bring into light in this article may lead us to the beginning of the *finest* typographic quality in Bangla publishing.

## References


[1] M. M. Ali. *AroSgaon (More SGAON) 2.1*, 1996.

[2] F. Charette. *Polyglossia: An Alternative to Babel for XeLaTeX and LuaLaTeX*, 1.44 edition, 2019.

[3] A. Hasan. *The LaTeXbangla Package: Enhanced LaTeX integration for Bangla*, 0.2 edition, 2016.

[4] M. P. Kamermans. *ucharclasses*, 2017.

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

[6] P. B. Pal. Bangtex: A package for typesetting documents in Bangla using the TeX/LaTeX systems, 2002. http://www.saha.ac.in/theory/palashbaran.pal/bangtex/bangtex.html

[7] A. Pandey. Typesetting Bengali in TeX. *TUGboat* 20(2):119–126, 1999. https://tug.org/TUGboat/tb20-2/tb63pand.pdf

[8] A. Pandey. *Bengali for TeX*, 2.0 edition, 2002.

[9] A. M. H. Rashid. *Baanglaay Renesnaar Pothikrit: Rabindranath O Chaar Bangali Bigynani (Trailblazer of Renaissance in Bengal: Rabindranath and Four Bengali Scientists)*. Nabajuga Prokashani, Dhaka, 2015. in Bangla.

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

[11] W. Robertson. *The fontspec package: Font selection for XeLaTeX and LuaLaTeX*, 2.7c edition, 2019.

[12] F. G. E. Ross. *The Evolution of the Printed Bengali Character from 1778 to 1978*. PhD thesis, School of Oriental and African Studies, University of London, 1988. https://eprints.soas.ac.uk/29311/1/10731406.pdf

[13] R. Tagore. The complete works of Rabindranath Tagore: Stories. Retrieved: 2 August 2019. http://tiny.cc/tagore-postmaster

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

[15] The Unicode Consortium. *The Unicode Standard, Version 12.1.0*. The Unicode Consortium, Mountain View, CA, 2019. https://unicode.org/versions/Unicode12.1.0/



◇ Md Qutub Uddin Sajib
China University of Geosciences, Wuhan
388 Lumo Road, Wuhan 430074, China
qsajib71 (at) gmail dot com
ORCID 0000-0002-7090-7981