

Representing linguistic pitch in (X_Y)L^AT_EX

Kevin Donnelly

Abstract

Linguists, especially those working with tone languages, may need to depict pitch levels in the language examples they use. This article looks at some options in L^AT_EX and X_YT_EX for representing pitch and tone. The emphasis is on Africanist linguistics, since that is the area with which I am most familiar.

1 Introduction

For extended linguistic work X_YT_EX is preferable, since it uses UTF-8 natively, and allows fonts to be chosen that will represent all aspects of the language's orthography (whether that is already standardised, or is one of the things that you are working on). However, it is still possible to represent pitch and tone effectively even if you need to use L^AT_EX, though a different set of tools is required. Notes for both T_EX flavours are given here, and sample documents for each are attached — *pitch_Llatex.tex* and *pitch_{XETEX}.tex*. Everything has been tested on Ubuntu 12.04 running T_EX Live 2012 (note that older T_EX Live versions may not produce the same output).

2 Setting up the document

Both sample documents assume that you are writing a book, and additional packages or code to enable pitch and tone to be represented are then added.

For L^AT_EX, the package *tipa* [8] is used, with additional code to allow pitch marks to be printed without sidebars (this is common for East Asian languages, but not for African languages):

```
\makeatletter
\renewcommand\@tonestembar{%
  \setbox0\hbox{\tipaencoding\char'277}%
  \hbox{\vrule height \ht0 depth \dp0
        width 0pt}}
\makeatother
```

Times is set as the default font:

```
\usepackage{mathptmx}
```

For X_YT_EX, the package *fontspec* [9] is used. Most (but apparently not all) of the features in *tipa* are now part of *xunicode*, so there is no need to activate *tipa* (doing so produces warnings such as: *Command \sup_s already defined*). Unfortunately, without *tipa*, commands like `\textupstep` in X_YT_EX do not work properly (they appear after the letter they refer to rather than before it), so some workaround code is needed — see Section 3. Moreover, the L^AT_EX code above to print pitch marks without sidebars will not work (thanks to Alan Munn for pointing this out), since the equivalent glyphs in Unicode can-

not be segmented — an alternative will be presented in Section 4.

Charis SIL [10] is selected as the main font. This is a normal text font, like Times, but includes a vast range of glyphs to meet a range of phonetic and orthographic requirements. (This article is typeset in Charis SIL instead of the normal *TUGboat* font in order to enable easier display of text being discussed.)

```
\defaultfontfeatures{Mapping=tex-text,
  Scale=MatchLowercase}
\setmainfont{Charis SIL}
```

The `\defaultfontfeatures` command must precede the actual font selection, and tells `fontspec` to use common T_EX aliases (e.g. `--` will produce `—`, and `|` will produce `|` instead of `—`), and to scale all fonts to match the main font's lower-case size. This ensures that any additional fonts selected (see next paragraph) will better match the weight of the main font.

`fontspec` also allows fonts to be set up for particular purposes. For instance, if we are annotating the syntactic relationships of particular words, and wish to use a separate font for that, in the preamble we could add

```
\newfontfamily{\syntaxfont}{Liberation Sans}
\newcommand \syn[1]{\{\syntaxfont #1}}
```

Here, the first line specifies Liberation Sans as the font for syntax annotations in example glosses, and the second line sets up a new command so that

```
\syn{annotation}
```

can be used instead of the more cumbersome

```
{\syntaxfont annotation}
```

For both files, the package *expex* [2] is optional, but recommended because it offers good typesetting of linguistic examples, and allows word-by-word glossing.

3 Tone-marking individual words

The first priority in representing a tone-language is the availability of sufficient glyphs (mostly diacritics) for the tone-marking. It is worth noting that the readability of diacritics (or even whether they are displayed at all) depends crucially on the font — not all are capable of showing all diacritics, or placing them in the right location.

Diacritics commonly used in representing tone include: `á à â ã ä å ã à á 'a 'a 'a 'a`. This list is not exhaustive — many other diacritics are available.

Many of these diacritics may already be available via your keyboard (for instance, on the default UK keyboard in Ubuntu, `á` is produced using `AltGr + ;`; then `a`; `à` by `AltGr + #` then `a`, and `â` by `AltGr + '` then `a`). If diacritics are to be used frequently, it is worth setting up a keyboard layout to allow this.

In L^AT_EX, commands can be used to access diacritics — the *Comprehensive L^AT_EX Symbol List* [6] and Appen-

dix A.3 of the *TIPA Manual* [8] are valuable here. Most of the other diacritics above can be accessed as follows:

```

ā \v{a}
ā \={a}
á \H{a}
â \H*{a}
á \textvbaraccent{a}
'á \textupstep{a}
`á \textdownstep{a}

```

The exceptions are the superscript exclamation marks [!] and [˘] (used in African-language linguistics to represent upstep and downstep respectively).

X_YTeX offers more scope for defining diacritics, since UTF-8 allows diacritics to be combined (rather than having to depend on precomposed glyphs). This means that writing `a\char"030D` will produce `á`, because the number for the UTF-8 combining diacritic for the vertical diacritic is 030D — the relevant numbers for each glyph can be found by using a utility like `KCharSelect`. [3]

However, since it is difficult to remember these character numbers, and they are tedious to type, it is best to set up commands to produce them. Setting up the following commands in the preamble:

```

\newcommand \aup[1]{\char"F19E#1}
\newcommand \adown[1]{\char"F19F#1}

```

will allow us to get the Africanist up/downstep characters by using `\aup{a}` and `\adown{a}`: `'a`, ``a`.

In those cases where the L^AT_EX `tipa` commands do not produce the desired results in X_YTeX, you can override them — this may also be useful if you have an older text which you are converting, where you wish to minimise the number of textual changes that need to be made:

```

\renewcommand \textupstep[1]{\char"A71B#1}
\renewcommand \textdownstep[1]{\char"A71C#1}

```

This allows `\textupstep{a}` and `\textdownstep{a}` to continue to be used to give `'a` and ``a` respectively.

The greater flexibility offered by combining diacritics can also be used to produce new diacritics. For instance, adding an acute and then a macron:

```

\newcommand \hbend[1]{#1\char"0301\char"0304}

```

can yield a diacritic to represent the end of a high-tone bridge: `\hbend{a}` gives `ā̄`.

However, not all combining diacritics may be visible to X_YTeX — in Charis SIL, for instance, glyphs in the ranges 07nn, 1Dnn and 20nn do not give the expected output.

4 Pitch representation for individual words

Inline pitch representation of individual words has been used in the past in African-language linguistics to draw attention to the pitch contours of individual words, so that, for example, the pitches of the kiKongo word `ibuuna` [_ —] (*so*) can be shown without drawing premature con-

clusions as to how these pitches should be represented in the tone-marking.

The `\tone` command can be used for this in L^AT_EX: `ibuuna [-\tone{11}\, \tone{5555}\, \tone{11}~]` Here, the individual pitches are represented by the number of the tone letter (see A.2.1 of the *TIPA Manual* [8]) — five levels are available. Longer pitches can be depicted by repeating the tone numbers, as in `\tone{5555}`. Glides can be represented by sequences of numbers, and some variation is possible here; as shown in the accompanying file `pitch_latex.tex`, a rising pitch on the last syllable of a variant pre-pausal form of `ibuuna` can be represented using any of `\tone{13}`, `\tone{113}`, `\tone{1133}` and `\tone{133}`.

The most reliable way of handling spacing between the pitchmarks is to use `\hspace{<dimen>}`, giving measurements in em, mm or pt. For ease of use, this can be set up in a command such as:

```

\newcommand \gap[1]{\hspace{#1mm}}

```

so that a gap of 2mm length can then be inserted by using `\gap{2}`, and so on.

In X_YTeX, Charis SIL provides five tone letters similar to those in TIPA, in each of four variants: line or dot with tonestembar on the right or the left (see section 1.4 of *Marking Tone* [7]). Issuing the command

```

\fontspec[Renderer=Graphite, RawFeature=
  {Special=Hide tone contour staves}]
  {Charis SIL}

```

after `\begin{document}` will remove the tonestembars on glides, but not on level pitches, so these glyphs are inappropriate for African-language linguistics.

However, Charis SIL also provides a set of nine pitch marks (glyphs F1F1–F1F9) specifically aimed at addressing this issue (see section 2 of *Marking Tone* [7]); these allow pitches to be shown inline. To simplify usage, the glyph numbers can be replaced by a `\pitch` command:

```

\newcommand \pitch[1]{\char"F1F#1}

```

and `ibuuna` [_ —] can then be produced by writing:

```

ibuuna [-\pitch1\gap{1}\pitch9\pitch9
  \gap{1}\pitch1~]

```

repeating the glyph number if longer pitches need to be shown. Inserting the command:

```

\fontspec[Renderer=Graphite]{Charis SIL}

```

after `\begin{document}` will again allow glides to be represented in a variety of ways: [_ ^] [_ ^] [_ ^] [_ ^].

Since these pitch-glyphs are in the Private Use Area of the font (in other words, they are not yet official Unicode glyphs) Charis SIL is the only font that they can be used with.

Mark Wibrow has presented a solution [1, 13] using TikZ [11] that works with both L^AT_EX and X_YTeX, is

font-independent, and allows a greater variety of pitchmarks. Using this, marking such as **ibuuna** [- — -] can be produced by writing:

```
\tikz[baseline={0,0.25ex}]%
\contour[contour only, contour scale=2ex/6,
  contour marks={1.55.1}]
  {ibuuna};]
```

The positioning and size of the inline marking is handled by the `baseline` and `contour scale` directives. The marking itself is input using numbers 1–5 to represent the levels, locating them above each letter that requires a pitchmark; letters not requiring pitchmarks must be designated by a full stop (.)—that is, the number of items in the pitchmark line must be the same as the number of letters, spaces, etc. in the text line. (To do this easily, your editor should be switched temporarily to a mono-space font if it is not already using one.)

A variety of glides can be represented: [- — ↘] [- — ↗] [- — ↙] [- — ↕], along with the sort of shape which would be difficult to represent with either of the earlier options, e.g. [^] [∪]

Since only the pitchmarks are shown, they can be lengthened by duplicating letters in the text. For instance, to get an extra-long mark for the first syllable, use:

```
contour marks={1111.55.,}
  {iiiiibuuna};]
```

giving [— — — ↘]—see also the end of section 7.

5 Representing pitchlevels over word-sequences

Depicting pitchlevels over word-sequences rather than individual words is usually best done using a tiered format. The same general principles as for inline marking apply.

For \LaTeX , the `tipa` pitchmarks look best when used in conjunction with `expex`, as shown in the accompanying file `pitch.tex`, where the pitchmark line takes the place of a gloss. Note that sequences of more than one word in the text line need to be enclosed in braces, and items in the text line which do not need pitchmarks aligned to them should be marked by empty braces in the gloss (pitchmark) line.

The main drawback to this approach is that trial-and-error is required to place the pitchmarks, experimenting with different `\gaps` until the marks are aligned with the relevant syllable. The pitchmark line is also hard to edit, since it is unclear to which part of the text line it refers.

For $X_{\text{L}}\TeX$, the Charis SIL pitchmarks can be used in conjunction with `expex`—as with `tipa`, this requires the same trial-and-error to space the pitchmarks (in the following examples, a raised dot (·) denotes a short pause):

```
(1)  ibuuna · basiidi kilumbu
     - - - - - - - - - - /
     so · they set aside a day ...
```

An approach that avoids this uses `pstricks` code by John Frampton, which sets up six pitchmarks (glides can also be specified). The marks are then applied simply by prefacing each vowel with the relevant level, so that writing:

```
\ib\5u\5un\1a · b\1as\5i\5id\4i
  k\3il\2umb\pitchup u
```

will give:

```
(2)  ibuuna · basiidi kilumbu
     - - - - - - - - - - /
     so · they set aside a day
```

Mark Wibrow’s TikZ solution [13][1] works on both \LaTeX and $X_{\text{L}}\TeX$. The correspondence between letters and pitchmarks makes it easy to place and edit the marks while keeping the text easy to read:

```
contour marks={0.55.0...0.55.4..3.2..?}]
  {ibuuna · basiidi kilumbu};
```

The pitchmarks can be placed above or below the text, depending on the values used for `contour raise`:

```
(3)  ibuuna · basiidi kilumbu
     - - - - - - - - - - /
     so · they set aside a day ...
```

```
(4)  ibuuna · basiidi kilumbu
     - - - - - - - - - - /
     so · they set aside a day ...
```

Note that the TikZ solution will not work in \LaTeX if the text contains characters such as raised dots (·), so this is another argument for using $X_{\text{L}}\TeX$. Note also that `expex` places its example number at the baseline of the TikZ graphic—there is currently no means of adjusting the vertical placement of the example number.

6 Representing pitch contours over phrases

In tone languages, the pitch domain is the syllable, and sections 3–5 have shown various means of representing pitch on the syllable. For pitch-accent and intonation languages, the pitch domain is the word, phrase, or sentence, and this section sets out a method for depicting pitch contours over these longer stretches.

Mark Wibrow’s TikZ code [1, 12] allows contours to be represented in a variety of ways in both \LaTeX and $X_{\text{L}}\TeX$. In the following examples, an English interrogative sentence (“Where are you going?”) will be used, on which two pitch contours will be represented. The first of these (a) is the neutral or default contour, in answer to “I’m going out now”—this has high pitch on *where* and *go*. The second contour (b) is a focussed one, in answer to “I’m going out now, but not to the shops”, and has high pitch on *are*.

One option is to place the words themselves on different levels, using the tokens `follow` `contour` directive.

- (5) a. Where
are you go
ing?
- b. are
Where you going?

An inline notation is used, where a defined character (by default, the pipe character `|`, but this can be changed to other characters, e.g. an asterisk) is used to specify the anchor points for the contour, and the height of the contour is specified by a following digit from 0–10 in square brackets:

```
\contour[tokens follow contour]
  {| [10]_Where_ | [3]are you | [6]_go_ | [0]ing?};
```


The stress/high-pitch composite in example (5) has been marked by underlining (generated by a preceding and following underscore). It is also possible to place the words in boxes (`box` tokens), and expand the spaces between them (`space` token `width`), as in this example:

- (6) a. Where are you go
ing?

A more familiar option is to show the text on a line as usual, with a contour line above it. Example (7) shows this, and also different styles for the contour line (`contour / .style`)—*thick* and *gray* in (a), and *ultra thick* and *red* in (b):

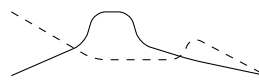
- (7) a. Where are you going?
- b. Where are you going?

A contour can be displayed by itself, without showing the text line, by using `contour` only:

- (8) a. 

In some cases it may be useful to compare two contours directly. This can be done by drawing both contours, but setting one of them to `contour` only, as in example (9). In this case, the focussed contour is compared to the neutral contour, with the latter styled so that it is drawn with a dashed line. This example also shows how

the gap between the text line and the contour can be specified by using `contour raise`—the gap here is twice that in the previous examples.

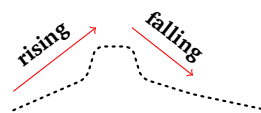


- (9) **Where are you going?**

The contour can also be annotated by drawing additional `\paths` such as the following, part of example (10):

```
\path [draw=red, ->] ([yshift=0.25cm]
  mycontour-1) -- ([yshift=0.25cm]
  mycontour-3) node [midway, sloped, above]
  {\small rising};
```

`contour` mark prefix is used to give the contour a name (here `mycontour`), and then a red arrow (grayscaled for the printed version) is drawn between the relevant anchor points of `mycontour`, counted sequentially (in this case, between anchor points 1 and 3). The `yshift` directive specifies how far above the contour the arrow should be, and the `node` clause specifies the location and size of the annotation “*rising*”. The contour is also styled to appear as a dotted line.



- (10) b. **Where are you going?**

7 Replicating early typesetting of tone

Nowadays the conventions of describing tone and intonation are well-established, but at the beginning of the study of the role of pitch in languages even concepts such as “absolute pitch” and “relative pitch” were not well-understood. One of the earliest researchers in this field, Karl Laman, a Swedish missionary to the Kongo, produced the first dictionary of an African language to be tone-marked throughout [5]. His 1922 book, *The Musical Accent or Intonation in the Kongo Language* [4], was one of the first modern attempts to systematise and describe pitch linguistically.

When referring to early research, it may be useful to quote it in its original format, and this may include diacritics and pitchmarks that are no longer widely used, or were even invented solely for that particular piece of research. L^AT_EX and X_YL_AT_EX offer possible ways to replicate this type of material.

Laman’s 1922 marking system for the six main pitch-levels he posits for kiKongo (there are a few others) are set out in Table 1. His marking is emulated here by using

stress diacritics (`\textprimstress` and `\textsecstress` from the `tipa` package for \LaTeX , and Unicode characters 02C8 and 02CC for $X_{\text{L}}\TeX$, though the `tipa` characters will work in $X_{\text{L}}\TeX$ too), and a super/subscript + sign.

Gradation	Designation	Diacritic
high	very high	a ^{''}
	high	a [']
mid	semi-high	a ⁺
	semi-low	a ₊
low	low	a _i
	very low	a _{ii}

Table 1: Laman's pitch-marking system

For ease of use, commands can be set up to reproduce the marks (which can appear before or after the relevant vowel):

```
\newcommand{\hi}{\textprimstress}
\newcommand{\vlo}{\char"02CC\char"02CC}
\newcommand{\shi}{\scriptscriptstyle +}$}
```

and so on. This allows Laman's examples to be replicated by inserting the mark shortcut at the appropriate point in the word, as the following examples (from p. 12 of [4], but with inline pitchmarking added) show:

mfu⁺mu [- -] (*king*)
muk_a'nda [- / -] (*book*)
zo'bongo₊ [- - -] (*antelope*)
ka₊ba [- -] (*to share*)
ba₊ka₊la⁺ [- - -] (*man*)
tū⁺la [- -] (*to put*)
dī₊nsu₊su₊ [- - -] (*herbal plant*)

Similar approaches can be used for other markings in [4], and for works by other early researchers, enabling such material, of intrinsic interest to the history of linguistics as well as to descriptive linguistics, to be distributed without great expense.

These examples incidentally demonstrate that more aesthetic placement of the inline pitchmarks discussed in section 4 can be achieved by manipulation of the text line in the `contour` marks directive. For instance, writing out **mukanda** in full

```
contour marks={.1.>.1}
{mukanda};]
```

gives **muk_a'nda** [- / -], which contains unsightly gaps in the inline marking. To deal with this, simply remove excess letters, using:

```
contour marks={1.>.1}
{ukana};]
```

to give the more attractive **muk_a'nda** [- / -].

8 Conclusions

\TeX is a powerful and versatile system for typesetting academic work dealing with pitch, whether relating to tonal

or intonational languages. A variety of diacritics is available, particularly if $X_{\text{L}}\TeX$ is used to give access to a comprehensive font like Charis SIL. Add-on packages such as TikZ/PGF allow diagrammatic representation of pitch levels and contours. Intricate marking systems from early works on pitch can be replicated simply and inexpensively.

9 Acknowledgements

I am grateful to Christina Thiele for comments on earlier versions of this paper.

References

- [1] Kevin Donnelly and Mark Wibrow. TikZ pitch contour, 2013. <http://gitorious.org/tikz-pitch-contour>.
- [2] John Frampton. ExPex, 2012. <http://www.math.neu.edu/ling/tex/expex>.
- [3] KDE. KCharSelect, 2013. <http://utils.kde.org/projects/kcharselect>.
- [4] Karl E. Laman. *The Musical Accent or Intonation in the Kongo Language*. Svenska Missionsförbundet, Stockholm, 1922. Available from the Internet Archive at <http://archive.org/details/musicalaccentori00lamarich>.
- [5] Karl E. Laman. *Dictionnaire Kikongo-Français*. IRCB, Brussels, 1936.
- [6] Scott Pakin. The Comprehensive \LaTeX Symbol List, 2009. <http://www.ctan.org/pkg/comprehensive>.
- [7] Lorna Priest. Marking tone, 2007. http://scripts.sil.org/cms/scripts/render_download.php?format=file&media_id=PitchContours.v1.pdf&filename=PitchContours.v1.pdf via <http://scripts.sil.org/ipahome>.
- [8] Fukui Rei. `tipa`—Fonts and macros for IPA phonetics characters, 2002. <http://www.ctan.org/pkg/tipa>.
- [9] Will Robertson. `fontspec`—Advanced font selection in $X_{\text{L}}\TeX$ and $\text{Lua}\TeX$, 2013. <http://www.ctan.org/pkg/fontspec>.
- [10] SIL. Charis SIL 4.112, 2011. <http://scripts.sil.org/CharisSIL>.
- [11] Till Tantau. TikZ and PGF, 2010. <http://www.ctan.org/pkg/pgf>.
- [12] Mark Wibrow. Using TikZ to depict intonation, 2013. <http://tex.stackexchange.com/questions/107941/using-tikz-to-depict-intonation>.
- [13] Mark Wibrow. Using TikZ to depict pitchlevel, 2013. <http://tex.stackexchange.com/questions/108530/using-tikz-to-depict-pitchlevel>.

◇ Kevin Donnelly
 Llanfairpwllgwyngyllgogerychwyrndro
 bwlllantysiliogogoch
 Wales
 kevin (at) dotmon dot com
<http://kevindonnelly.org.uk>