AlQalam for typesetting traditional Arabic texts*

Hossam A. H. Fahmy
Electronics and Communications Department,
Faculty of Engineering, Cairo University, Egypt
hfahmy (at) arith dot stanford dot edu

Abstract

AlQalam ("the pen" in Arabic) is our freely available system intended for typesetting the Qur'an, other traditional texts, and any publications in the languages using the Arabic script. From a typographical point of view, the Qur'an is one of the most demanding texts. However, there is a long historical record of excellent quality materials (manuscripts and recent printings) to guide the work on a system to typeset it. Such a system, once complete, can easily typeset any work using the Arabic script, including those with mixed languages.

1 Characteristics of Arabic typography

The Arabic alphabet has been adopted for use by many languages in Africa and Asia, including Arabic, Dari, Farsi, Jawi, Kashmiri, Pashto, Punjabi, Sindhi, Urdu, and Uyghur. The Arabic script is also used for a number of other languages either to present how the language used to be written historically (as for Turkish) or how some write it in an unofficial manner (as for Hausa in western Africa).

Similar to the Latin alphabet, with its adoption by several languages, the Arabic alphabet has acquired new symbols to represent the sounds that do not exist in Arabic. In contrast to the Latin alphabet that has dots only on the 'i' and 'j', the Arabic alphabet uses dots extensively, both above and below the letter shapes, to distinguish the different characters. This explicit distinction between the different letters in written text using dots was in itself an addition to the original script, which had no dots. In the original Arabic script, the distinction between بيت (house) and بنت (girl) when represented as ﺑﺖ was understood from the context. In general, the symbols developed for other languages follow the same idea as Arabic and use more dots (up to four) and special marks on the original shapes of the Arabic letters.

Arabic being a semitic language, usually only the consonants are written in a word. The equivalent of short vowel sounds are written as additional marks on top of the letters. Obviously, the different languages have different vowels and need different symbols to mark them. In addition to that, since the geographical area covered by the Arabic script historically is quite vast, different regions of the world developed different symbols. The result that we see today is a plethora of additional marks developed historically.

A beginner learns that Arabic is written from *right to left* and must practice writing each letter and its connection rules to other letters. Because of this cursive nature, any letter may connect to the previous and following letters. Hence, a beginner learns the general *four basic forms* of a letter: at the start of a word, at the middle, at the end, and isolated. The simplest example of this rule is the equivalent of 'b' in Arabic: ﺑـ ـﺒـ ـﺐ ب. A reader with a sensitive eye will notice that the four shapes of the same letter differ in their width, height above the line, and depth below it. The same structural shape is used for the equivalent of 't' (ﺗـ ـﺘـ ـﺖ ت), and 'th' (ﺛـ ـﺜـ ـﺚ ث). The equivalent of 'n' and 'y' share the same shapes as 'b' in the initial and medial forms (ﻧـ ـﻨـ and ﻳـ ـﻴـ) but not in the final or the isolated forms (ﻦ ن and ﻲ ي). In many writing styles, both traditional calligraphic scripts and typographical fonts, the final and isolated forms of 'y' are written without dots as (ﻰ ى).

In traditional Arabic writing styles (but with the exception of thuluth, riqā', and tawqīʿ [12]), the letters ﺍ ﺩ ﺭ ﻭ and their siblings with dots or marks do not connect to the following letter but only to the preceding one.

The cursive nature of the Arabic script adds another characteristic: many letters combine together to produce new shapes as in الحج becoming ﷲ. In the Latin script this phenomena occurs infrequently and when it happens, a *ligature* is used to improve the appearance of the problematic letter combinations such as 'ff' and 'ffl'. In Arabic typography, on the other hand, the presence of combined letters is abundant *but optional* in many cases.

* A project under the author's supervision. This paper combines the material presented at two conferences: EuroTEX 2006 and TUG 2006.

Haralambous [1] gives a long list (still not exhaustive) of possible 'ligatures' in Arabic. While speaking about the history of Arabic typography, Milo [8] explains that "each letter can have a different appearance in *any* combination, something that can only be crudely imitated with ligatures". According to Milo [8], most modern books present the connected letter groups "as 'ligatures' and 'artistic expressions' without so much as a hint at traditional morphographic rules". In 1990, MacKay [7] discussed the range of context evaluation in Arabic and concluded that clusters of four, five, six, and sometimes more letters may combine into a unique shape. MacKay then proposed the use of virtual fonts with TEX as an adequate solution.

Another feature in the Arabic script is its reliance on subtle changes to the letter shape to aid the reader in identifying the beginning and end of each letter within a combination. The letter سـ has three "teeth" (vertical pen strokes) similar to the teeth in بـ and تـ. When سـ is connected to بـ, the tail of the سـ may be elongated to alert the reader to the correct grouping of the teeth. Furthermore, the two words سبع and تسع present different heights for the teeth of the بـ and تـ to help the reader as well. This difference in width and height is a type of encoding to prevent a misreading of the word and to aid the trained eye in quickly catching the letter combination.

That encoding helps in other cases as well. If for any reason the dots fade away, a reader faced with بسع can guess its correct origin. This encoding to emphasize the different letters by raising some teeth is essential in words such as تبينت and تسببت. However, the raising of the teeth is only possible by investigating the group of letters. So the height of the tooth for نـ in تبينت and تبنت is not a feature of the individual letter but of the whole combination. The same goes for the height of the dot on top of that same letter as in سننجح or سنن .

In traditional (manual) writing, the "skeleton" of the letter combination is written first, then the dots and the other marks are provided. So, a writer probably progresses from the skeleton to the dotted to the vocalized form as سس → سنن → سُنَّ .

In the Arabic script, a good calligrapher justifies the lines mainly by using optional ligatures or wider forms of some letters and not by stretching the spaces between the words, as is done with the Latin script. The use of optional ligatures and wider forms is the preferred method in high quality works. Another method is to add an elongation to the tail of some letters by using the taṭwīl or kashīdah symbol '-' such as ســبب instead of سبب. This second method has been widely abused in newspapers and low quality materials using mechanical typewriters.

The Arabic script has a large number of writing styles that were developed historically to accommodate the different languages and different purposes. Latin scripts use bold, italic, or larger fonts for section headings and for emphasis. Traditional Arabic writings vary the typeface instead. The printings of the Qur'an as well as of most books almost always use the naskh typeface for the main body. The headings, the introductory materials, and the back materials frequently use other typefaces such as the thuluth, ta'līq, and ruq'ah.

To summarize, the Arabic script has its own unique requirements for typesetting:

1. Arabic is written from right to left.
2. Characters in general have four different forms (initial, medial, final, and isolated).
3. These forms are of different width, height, and depth.
4. The shape of a specific form depends on its context. For example, the height of the teeth (the vertical stroke at the start of the character) changes to help the reader to distinguish this character from its neighbors.
5. There are additional marks (mostly for short vowels) that are put on top or below the character.
6. The horizontal and vertical location of the dots and marks on the characters is not always at the same position but depends on the character *and* its context.
7. Almost any letter of the script may enter into a ligature and those ligatures may be up to six letters long.
8. Ligatures and variable width forms of the letters are used to justify the lines.
9. Several typefaces are needed for special materials in a work of good quality.

With all of these issues, to find a suitable position for the dots and marks on the letter combinations is sometimes a real challenge even for a human, let alone a machine.

2 Automated typesetting of Arabic

The Arabic script does not enjoy the same luxury that Latin script has when it comes to automated typesetting on computers. Milo correctly asserts [8] that the use of individual letters as the building

block is not suitable for Arabic. Both MacKay [7] and Milo [9] argue that a layered approach is a better solution. In such a layered approach, some basic elements are provided in the font to represent the skeleton of some letter combinations, an individual letter's skeleton, or even a part of a letter. These elements are combined first to give the correct skeleton of the word with all the needed shaping for the teeth or other style requirements. On top of that skeleton, a second layer for the dots is added. Then, the vowel marks and any other marks come in subsequent layers. Milo [10] developed a system with a layered approach for his company, DecoType. It is a proprietary system used by a number of commercial software tools. Due to its proprietary nature, the full details and the extent of the capabilities of this system are not widely known.

Our goal is to provide a freely available system capable of typesetting the Qur'an, other traditional texts, and any publications in the languages using the Arabic script. From a typographical point of view, the Qur'an is one of the most demanding texts. However, there is a long historical record of excellent quality materials (manuscripts and recent printings) to guide the work on a system to typeset it. Such a system, once complete, can easily typeset any work using the Arabic script including those with mixed languages.

Knuth and MacKay [4] were the first to present a working solution for including right-to-left text (for Arabic and Hebrew) in the TEX family. Their proposed TEX-XET system is an extension of TEX that produces an extended DVI file. The enhanced mode of ε-TEX allows bidirectional text processing and produces regular DVI files, but ε-TEX does not provide any Arabic fonts or any specific functionalities that ease the typesetting of Arabic books. Within the TEX extensions, both Ω [3] and ArabTEX [5, 6] have been used for Arabic and have met some of the basic requirements to varying degrees.

With the historical trend to extend TEX, Ω evolved as an implementation allowing multilingual text processing. As an offshoot of the work on Ω, the Al-Amal system [1] was an early attempt to typeset the Qur'an specifically. Unfortunately, it is not freely available and its output (as shown in the example published in the paper describing it) falls short of the desires of a native reader.

Due to its various attractive features, Ω was the first choice to achieve our goal. However, the result of our early experiments with the available Arabic font provided with Ω and with the system itself were not satisfactory. Ω in its current state does not easily lend itself to the layered approach described earlier. The modification of Ω is not an easy task since it is a very large system and such a modification means the creation of a new system that is not compatible with the existing base of TEX. The newer developments to Ω [2]—once they are stable, widely available, and documented—should be a great help in implementing the layered approach necessary for typesetting high quality texts in Arabic.

Lagally in ArabTEX [6] preferred to stay within the stable TEX standard and perform all the necessary processing with TEX macros. That decision allowed ArabTEX to be portable to any TEX implementation. However, ArabTEX had to compromise on the issue of line breaking. Although not a simple program, ArabTEX is confined to a number of style files each performing a specific task. ArabTEX implements a layered approach where each character is represented by a skeleton and additional modifiers (dots and vowels). For high quality work, the font provided with ArabTEX still needs improvements but it is an acceptable start. ArabTEX uses the LATEX license and hence we changed the name of our work to AlQalam ('the pen' in Arabic).

3 Implementation

As a start, AlQalam grows out of modifications to ArabTEX [5, 6]. Hence, it inherits ArabTEX's good features:

- All the necessary processing is done with TEX macros which allows it to be portable to any TEX implementation.
- Although still in need of many improvements, the font provided with ArabTEX is the best available within the TEX family.
- The shapes of the letters change with their context (teeth are raised and automatic detection of many ligatures).
- A layered approach is used.

Although the use of TEX macros brings the virtue of portability it also brings severe limitations:

- ArabTEX had to compromise on the issue of line breaking and justification. For right-to-left text, ArabTEX is forced to handle the line breaking itself, using a slow and complicated algorithm, thus bypassing one of the best parts of TEX for the Latin script.
- ArabTEX analyzes the character combinations to decide on ligatures using TEX macros as well. This analysis using macros is
  - less efficient than the ligature tables used for the Latin script in METAFONT; and

– limits the extent of the search for alternative letter combinations. In Arabic, four, five, six, and sometimes more letters may combine into a unique shape [7].

4 Specific needs of the Qur'an

Our goal of typesetting the Qur'an and traditional texts implies a few more challenging requirements in addition to those for general Arabic typesetting. To assist the reader in recitation, several indicators for vowels, joints, text structure, and pausing locations have historically been added to the text of the Qur'an. We present here a few symbols.

Signs of pause (علامات الوقف) :

- The ۖ sign: the reader may continue but it is better to pause.
- The ۗ sign: it is allowed to pause but it is better to continue.
- The waqf jā'iz sign ۚ: equally good to pause or continue.

Additional diacritics :

- Ra's khā' ـۡـ corresponds to sukūn.
- The madda 'ـٓـ' appears in the Qur'an on many letters such as in كٓهيعٓصٓ.
- The small 'ـٓ' in ﴿أَن يُشۡرِكَ بِهِۦ﴾ and 'ـٗ' in ﴿وَٱللَّهُ عِندَهُۥ حُسۡنُ﴾.

Furthermore, there are different "narrations" of the Qur'an that differ in the pronunciation in some locations and hence lead to a plethora of additional marks needed. The vast majority of the printed copies of the Qur'an are in the narration known as Hafs. Only three other narrations (with their special marks for the special pronunciations) are printed in the whole Muslim world. The remaining narrations (sixteen remaining for a total of twenty) are still in manuscript form.

Fig. 1 shows an example of the four narrations that exist in print. To make the comparison easier, we present the same two lines from the four narrations written by the same calligrapher. A simple look at the first word (top right in each example) reveals some of the different symbols needed. Those additional symbols fit well in a layered approach but would be quite difficult to accommodate otherwise. The ج symbol appearing on the first word of the topmost narration is a pausing sign.

The use of transliteration for a purely Arabic document that is several hundred pages long is obviously neither practical nor desirable. Hence, the

Figure 1: The first two lines of surat al-ra'd: an example from the four printed narrations.

default assumption for AlQalam is an input file with the characters coded in Unicode. Bi-directional editors such as emacs and gedit (we used both) are good options. Editors have their own limitations though. If a symbol has a Unicode point associated with it but there is no key combination mapped to that symbol or no glyph in the editor's font to represent it, another facility must be used. In a case such as ۜ the user might supply the Unicode value as `^^db^^96`. AlQalam interprets this code as belonging to the "signs of pause" category, then raises it to its correct position such as in ٱلۡحَقُّ ۜ. The improvement of the input method is one of the major steps in future developments.

Another feature of typesetting the Qur'an is the use of colors. In some printings, certain letters, marks, or sometimes complete words take a different color usually to remind the reader of a pronunciation rule. Educational texts for young children also often use color encoding schemes to stress new reading concepts and to help train their eyes in picking up the distinctive features of the script. A complete system for dealing with the Arabic text should be able to color a piece of a letter combination or some specific marks.

5 New features in AlQalam

The first version of AlQalam, which became available by the end of 2005 and was presented at EuroTEX 2006, introduced many additional symbols to the font to enable the typesetting of quotations from the Qur'an. It also added an additional layer in typesetting Arabic text for the pausing marks of the Qur'an. Three features must exist in this layer:

- the correct vertical and horizontal positioning of those marks on the underlying word;
- the ability to stack some marks on top of each other; and
- the scalability of those marks when the size of the underlying text is scaled.

The first version of AlQalam implemented the concept of the additional layer but was deficient in regards to the three features just mentioned.

5.1 Positioning the marks

The different pausing marks vary in their sizes and shapes. They are raised on top of words that vary in their heights as well. The second sample from the top in Fig. 1 has four words followed by the same pausing sign ص. Its vertical position in

ٱلْكِتَٰبِۖ and ٱلْحَقُّۖ

is different because of the underlying text.

Two primitive algorithms existed in the initial version of AlQalam.

1. Raise the pausing sign at a predefined height from the baseline regardless of the height of the underlying text. The worst case is when a sign with a descender such as ج comes on top of a diacritic mark on a high letter. With a fixed height we get:

   الٓمٓرۚ صَبَرُواْۖ تَرَوْنَهَاۖ الْقَمَرَۖ → الٓمٓر صَبَرُواْ تَرَوْنَهَا الْقَمَرَ

   using this first method.

2. Raise the pausing sign by a fixed height *above* the diacritics on top of the character. This second method results in

   بِرَبِّهِمْۖ تَرَوْنَهَاۖ الْقَمَرَۖ ٱلْحَقُّۖ → بِرَبِّهِمْ تَرَوْنَهَا الْقَمَرَ ٱلْحَقُّ

   where ص has a varying vertical position.

Human calligraphers, in contrast, use neither of these methods. In the current implementation, we attempt to come closer to what is done in the best of the art. Calligraphers never lower the pausing marks below certain limits. Hence, AlQalam now starts by raising any pausing sign a minimum height depending on the current font size. If the underlying text is high enough so that an overlap occurs, the pausing mark is raised further. It is important to note that AlQalam now allows more than one mark to appear on top of a word. The new algorithm is not linked to the diacritic mark, as in the second method mentioned above, but can handle *any* underlying text, be it a diacritic mark or another pausing mark. The new algorithm thus yields:

بِرَبِّهِمْۖ تَرَوْنَهَاۖ الْقَمَرَۖ الٓمٓرۚ ٱلْكِتَٰبِۗ ٱلْحَقُّۖ صَبَرُواْۖ

for the case of a single mark and

أَعْمَىٰۚ مَّرْقَدِنَاۜۗ

for the case of multiple marks.

As for the third new feature of scaling, if the user writes

```
\RL{\vsmaller الْقَمَرَ^^db^^96 \larger
الْقَمَرَ^^db^^96 \larger الْقَمَرَ^^db^^96 \larger
الْقَمَرَ^^db^^96 \larger الْقَمَرَ^^db^^96 \larger
الْقَمَرَ^^db^^96 \larger الْقَمَرَ^^db^^96 \larger
الْقَمَرَ}
```

the output is

الْقَمَرَۖ الْقَمَرَۖ الْقَمَرَۖ الْقَمَرَۖ الْقَمَرَۖ الْقَمَرَۖ الْقَمَرَۖ الْقَمَرَۖ

which is easily achieved, since the height at which the pausing mark is positioned depends on the font size.

It is obviously not desirable to type sequences such as `^^db^^96` throughout the input file. An editor capable of understanding user-defined shortcuts may be used to ease this task. The user can just type the shortcut key and the editor puts the correct UTF-8 code into the file. Furthermore, to help all users, we also assigned some shortcuts for marks that appear frequently in the Qur'an, such as the dagger alif ـٰ for which the user types '`!`' instead of '`^^d9^^b0`'. Our system now translates the '`!`' and the other shortcuts on the fly to the corresponding UTF-8 code before processing the file. Here are the shortcuts currently enabled in AlQalam:

type	←	mark	type	←	mark	type	←	mark
3	←	ـۚ	2	←	ـۗ	1	←	ـۖ
6	←	ـ۠	5	←	ـۙ	4	←	ـۘ
9	←	ـۜ	8	←	ـۛ	7	←	ـۚ
!	←	ـٰ	.	←	ـۭ	0	←	ـ۬
^	←	۩	*	←	۞	+	←	۝

الٓمٓرٰ تِلۡكَ ءَايَٰتُ ٱلۡكِتَٰبِۗ وَٱلَّذِيٓ أُنزِلَ إِلَيۡكَ مِن رَّبِّكَ ٱلۡحَقُّ وَلَٰكِنَّ
أَكۡثَرَ ٱلنَّاسِ لَا يُؤۡمِنُونَ ١ ٱللَّهُ ٱلَّذِي رَفَعَ ٱلسَّمَٰوَٰتِ بِغَيۡرِ

روایة حفص عن عاصم Hafs

الٓمٓرٰ تِلۡكَ ءَايَٰتُ ٱلۡكِتَٰبِ وَٱلَّذِيٓ أُنزِلَ إِلَيۡكَ مِن رَّبِّكَ ٱلۡحَقُّ وَلَٰكِنَّ
أَكۡثَرَ ٱلنَّاسِ لَا يُؤۡمِنُونَ ١ ٱللَّهُ ٱلَّذِي رَفَعَ ٱلسَّمَٰوَٰتِ بِغَيۡرِ

روایة ورش عن نافع Warsh

الٓمٓرٰ تِلۡكَ ءَايَٰتُ ٱلۡكِتَٰبِ وَٱلَّذِيٓ أُنزِلَ إِلَيۡكَ مِن رَّبِّكَ ٱلۡحَقُّ وَلَٰكِنَّ
أَكۡثَرَ ٱلنَّاسِ لَا يُؤۡمِنُونَ ١ ٱللَّهُ ٱلَّذِي رَفَعَ ٱلسَّمَٰوَٰتِ بِغَيۡرِ

روایة قالون عن نافع Qālūn

الٓمٓرٰ تِلۡكَ ءَايَٰتُ ٱلۡكِتَٰبِۗ وَٱلَّذِيٓ أُنزِلَ إِلَيۡكَ مِن رَّبِّكَ ٱلۡحَقُّ وَلَٰكِنَّ
أَكۡثَرَ ٱلنَّاسِ لَا يُؤۡمِنُونَ ١ ٱللَّهُ ٱلَّذِي رَفَعَ ٱلسَّمَٰوَٰتِ بِغَيۡرِ

روایة الدوری عن أبی عمرو Al-dūrī

Figure 2: The same first two lines of surat al-ra'd as in Fig. 1, typeset with AlQalam.

Fig. 2 shows the same quotes presented in Fig. 1 as typeset by the current version of AlQalam. This example reveals a few more new features. The interword spacing in high quality Arabic typography is much smaller than in Latin-based typography. In some cases it is even completely absent. The reader relies on the fact that letters at the end of a word have a different shape in order to separate the words. A minimal spacing within a right-to-left environment is the default now, compare:

هذا مثال للمسافات المتروكة بين الكلمات العربية

to

هذا مثال للمسافات المتروكة بين الكلمات العربية

The command `\newspacefalse` was used in the second case to retain a large spacing as in ArabTEX and regular Latin script.

The earlier version of AlQalam allows hamzat-alwasl (ٱ) to appear only at the start of a word. However, in the Qur'anic text, it may appear in a medial form as in

وَٱلَّذِى فَٱدَّرَءۡتُمۡ وَٱتَّقُواْ بِٱسۡمِ

which is now possible in the current version.

Another new feature concerns the positioning of diacritic marks on top of the letters. Compare أَسۡمَاء to أَسۡمَاء. The first is the default, while the second with the raised mark is achieved by the command `\hightrue` within the Arabic script environment.

The current algorithm handles the small dagger alif according to its context. It is considered a separate character that appears on its own in cases such as ٱلسَّمَٰوَٰت ءَايَٰتُ رَوۡسِىَ. On the other hand, it is considered a mark on top of the underlying character in cases such as ٱسۡتَوَىٰ ٱلصَّلَوٰةَ ٱلۡحَيَوٰة. If the dagger alif is a mark, its positioning on the character is similar to that of the short vowels.

The contextual analysis for the dagger alif is conceptually simple, although a bit elaborate to program using TEX macros. The dagger alif is a mark modifying the underlying character if it is

- on the 'final' ى as in ٱسۡتَوَىٰ, even if it is followed by a connected pronoun as in سَوَّٰىٰهُنَّ تَقۡوَىٰهَا زَكَّىٰهَا, or
- on a و which is pronounced as an ا as in ٱلرِّبَوٰاْ or ٱلصَّلَوٰةَ.

5.2 Coloring

The coloring scheme has been improved as well. In Fig. 2, the color of every م or ن that has a shaddah on top of it is different, to indicate its special pronunciation in the Qur'an. The user chooses the color by `\colmnshadd{`*colorname*`}` (a few other commands for various rules were also added). The issue of coloring while maintaining the contextual analysis to decide on the correct form (initial, medial, final, or isolated) of the letter is not easily done.

In the Latin script, it is easy to to get 'text' via `'te\textcolor{blue}{x}t'` — as long as no special ligatures or kerning are needed between the 'x' and the 'e' or 't'. However, in Arabic script the command sequence in the middle of the word breaks the contextual analysis and the ligature formation.

In our case, since the coloring rules are known a priori, we code them in the system. Thus, after all the contextual analysis to decide on the appropriate letter form and the ligatures is done, but before the complete word is fed to the output, we intervene and check for the existence of the requested letter sequence. If a part of the word matches the pattern, we color it. Such a hard-coded "programmatic" way is not suitable for arbitary coloring that the user may wish to introduce into regular text outside of Qur'anic quotations.

5.3 Other improvements

A large number of glyphs in the current font are improvements of what ArabTEX or the first version of AlQalam provided. ArabTEX uses a circular pen for the diacritic marks. We found that a rotated

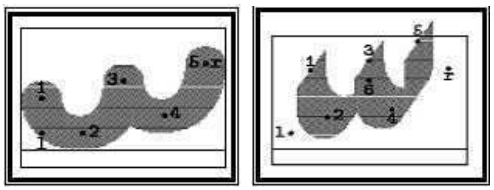


Figure 3: Effect of changing the pen and improving the shape on the "shaddah".

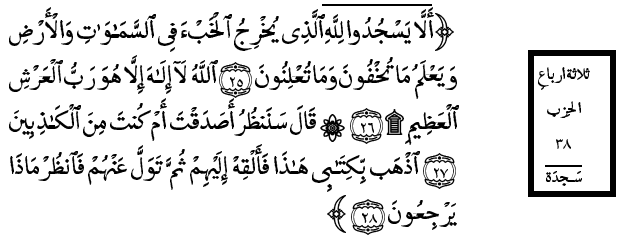
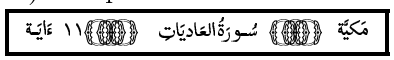


Figure 4: Marginal notes indicating partitions and prostrations.

square pen gives a much more satisfying output, as shown on the right side of Fig. 3.

To mimic the printed versions of the Qur'an, we define a `\sura` command ("sura" is a chapter of the Qur'an) that produces:



and

when given the number of the sura (100 in the first case and 112 in the second). AlQalam provides default values for the remaining information to be displayed (number of verses and whether the sura was revealed in Makkah or Madinah) if they are not supplied by the user.

Fig. 4 shows verses with marginal notes indicating the partitioning and the location of a prostration, as is customary in printings of the Qur'an. The counter of the partition and the corresponding note are produced automatically when the user writes * in the file. The indicator and the note for the prostration are similarly produced by ^ in the input file.

6 Future work

Much more work is needed on the fonts to produce new typefaces and to enhance the current one. The production of multi-letter ligatures with a layered approach where the dots, vowels, and additional marks are stacked on the basic structure is still an open issue. It might even require changes to the way TEX and METAFONT (or other font generation tools) handle ligature tables.

The use of TEX macros for programming has its merits and problems as discussed earlier. We think that we have brought AlQalam quite close to the limits of such an approach. To handle line breaking and justification correctly, a much more fundamental change in TEX itself is needed. After describing the line breaking algorithm of TEX [11], Plass and Knuth propose a refinement where the badness function for the lines depends on the number of varying-width letters in the paragraph. Neither TEX nor its descendants have implemented this refinement. In the case of the Arabic script, it will not be just the varying-width letters but also the optional ligatures that may be formed or broken to change the length of the text on the line. We hope that one of the current projects to extend TEX (ε-TEX, Ω, XHTEX, Oriental TEX, ...) will include this change to the badness calculation. Much experimentation using several different languages will be needed to come up with the most suitable algorithm.

Another issue that requires more work is the contextual analysis to decide on the glyphs used. This analysis is not only within a word; it must sometimes look at two consecutive words. The following example shows why such an inter-word analysis is needed. In general in Arabic, a silent 'n' sound is pronounced normally before

ء ه ع ح غ خ

and goes through some form of vocal assimilation into the sound of the following letter otherwise. If the silent 'n' sound is at the end of a word in the form of a tanwin (for example), and the following word starts by a letter into which the 'n' assimilates, the tanwin will be changed from ـٍ to ـٌ if the following letter is ب or to ـٍّ otherwise. The first letter of that following word gets a shaddah on it in the case of a full assimilation and does not get the shaddah for the incomplete assimilation.

For a program, these rules mean that we must do our analysis across any intervening spaces or command sequences, including counters for the verses and indicators of prostration or partitions that might come between consecutive words. The inter-word analysis is needed in many cases, not just for the 'n' sound, and it has some implications on the coloring rules as well.

As might be expected, our first attempts to achieve that endeavor using TEX macros proved to be quite laborious and are not yet fruitful. Currently, the user chooses the appropriate shape from the font manually. Once more we hope that the future TEX extensions can come to our help by providing easier means to program such an analysis.

7 Conclusions

In this article, we provided a summary of the traditional Arabic typesetting requirements as well as the first steps of a system to fulfill them. From these requirements, it appears that a layered approach is mandatory for high quality typography. We hope that future changes to TEX and METAFONT will enable us to achieve better ligature formation, line justification, and contextual analysis. To the best of our knowledge, there is no other software system available to serve the requirements of the different Qur'anic narrations.

References

[1] Yannis Haralambous. Typesetting the holy Qur'an with TEX. In *Multi-lingual computing: Arabic and Roman Script: 3rd International conference*, Durham, UK, December 1992.

[2] Yannis Haralambous and Gábor Bella. Omega becomes a sign processor. In *EuroTEX 2005: Proceedings of the 15th Annual Meeting of the European TEX Users, Pont-à-Mousson, France*, pages 8–19, March 2005.

[3] Yannis Haralambous and John Plaice. Multilingual typesetting with Ω, a case study: Arabic. In *Proceedings of the International Symposium on Multilingual Information Processing, Tsukuba*, pages 63–80, March 1997.

[4] Donald E. Knuth and Pierre A. MacKay. Mixing right-to-left texts with left-to-right texts. *TUGboat*, 8(1):14–25, 1987.

[5] Klaus Lagally. ArabTEX: A system for typesetting Arabic. In *Multi-lingual computing: Arabic and Roman Script: 3rd International conference*, page 9.4.1, Durham, UK, December 1992.

[6] Klaus Lagally. ArabTEX — Typesetting Arabic with vowels and ligatures. In Jiří Zlatuška, editor, *EuroTEX 92: Proceedings of the 7th European TEX Conference*, pages 153–172, Brno, Czechoslovakia, September 1992. Masarykova Universita.

[7] Pierre A. MacKay. The internationalization of TEX with special reference to Arabic. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 481–484, November 1990. IEEE catalog number 90CH2930-6.

[8] Thomas Milo. Arabic script and typography: A brief historical overview. In John D. Berry, editor, *Language Culture Type: International Type Design in the Age of Unicode*, pages 112–127. Graphis, November 2002.

[9] Thomas Milo. Authentic Arabic: A case study. right-to-left font structure, font design, and typography. *Manuscripta Orientalia*, 8(1):49–61, March 2002.

[10] Thomas Milo. ALI-BABA and the 4.0 Unicode characters. *TUGboat*, 24(3):502–511, 2003.

[11] Michael F. Plass and Donald E. Knuth. Breaking paragraphs into lines. In Donald E. Knuth, editor, *Digital Typography*, pages 67–155. CSLI Publications, Stanford, California.

[12] Mohamed Zakariya. أنماط الحرف العربي. *Al-Computer, Communications and Electronics Magazine* الكمبيوتر والاتصالات والإلكترونيات, 22(8):48–53, October 2005.