AlQalam for typesetting traditional Arabic texts*

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

1 What is AlQalam?

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).

The arabic script has its own specific requirements for typesetting [1]:

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 at the same position always but depends on the character *and* its context.
7. There are too many letters that combine ("ligatures").
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.

A layered approach [4, 5, 1] is probably the best solution to typeset Arabic. 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.

AlQalam (the pen in Arabic) is our freely available system aiming at 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.

As a start, AlQalam grows out of modifications to ArabTEX [2, 3]. Hence, its 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 available with ArabTEX is the best 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 has its 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 by itself in a slow and complicated algorithm bypassing one of the best parts of TEX available 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 alphabet in METAFONT and

* A project under the author's supervision. This paper depends on contributions from the students: Ahmad A. Zareef, AbdelRahman M. M. Ahmed, AbdelNasser I. A. AbdelAziz, Mohamed S. Fayez, and Pradipta Ranjali

  - 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 [4].

2 Specific needs of the Qur'an

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

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.

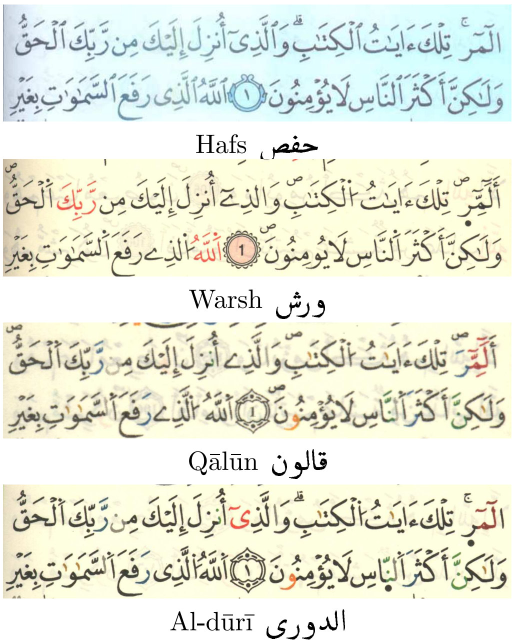
Fig. 1 shows an example of the four narrations that exist in print. To make the comparison easier, we present the same two lines from the four narrations written by the same calligrapher and printed by the same press: King Fahd's complex for printing the Qur'an in Madinah. A simple look at the first 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 top most narration belongs to the pausing signs.

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 often use similar color encoding schemes to stress new reading concepts and to train their little 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.

3 New features in AlQalam

The first version of AlQalam [1] introduced an additional layer in typesetting the 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.



Hafs حفص

Warsh ورش

Qālūn قالون

Al-dūrī الدورى

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

The first version of AlQalam implemented the concept of the additional layer but was deficient in respect of the three features just mentioned. These features are new additions in the current version of AlQalam.

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 ۖ. Its vertical position in

ٱلْكِتَٰبِۖ and ٱلْحَقُّۖ

is different because of the underlying text.

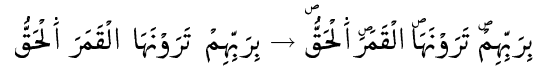
Two primitive algorithms existed in the initial version of AlQalam.

- 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 ۚ comes on top of a diacritic mark on a high letter. With a fixed height we get:

  الٓمٓر صَبَرُوا۟ تَرَوْنَهَا ٱلْقَمَرَ → الٓمٓرۚ صَبَرُوا۟ۖ تَرَوْنَهَاۖ ٱلْقَمَرَۖ

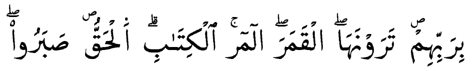
  using this first method.

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

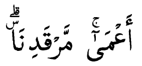


where ص has a varying vertical position.

What human calligraphers do is neither the first nor the second method just described. In the current implementation, we attempt to come closer to what exists in the best of the art. The calligraphers never lower the pausing marks below certain limits. Hence, AlQalam now starts by raising any pausing sign by 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 to *any* underlying text be it a diacritic mark or another pausing mark. The new algorithm thus yields:



for the case of a single mark and

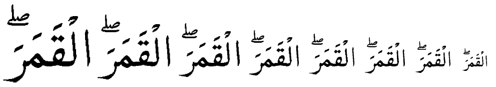


for the case of multiple marks.

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

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

the output is



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

It is obviously not easy to type characters such as `^^db^^96` all over the input file. A smart editor which is 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 utf8 code into the file. To help the general 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 utf8 code before processing the file. The shortcuts:

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

are currently enabled in AlQalam.

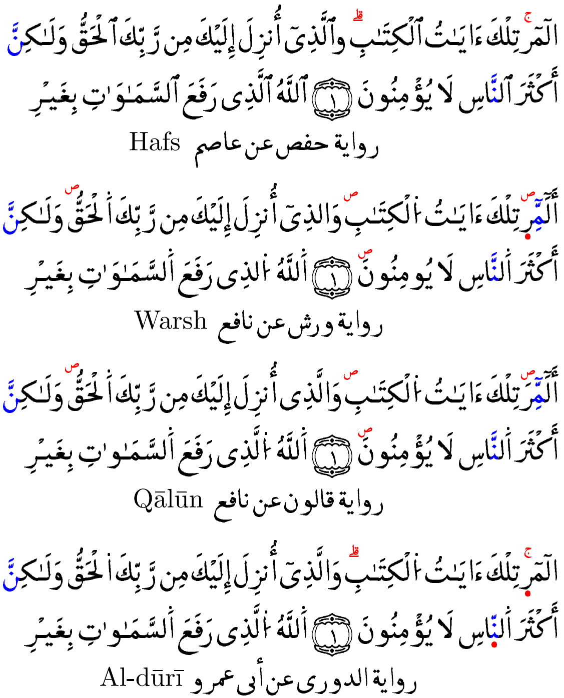

Hafs رواية حفص عن عاصم
Warsh رواية ورش عن نافع
Qālūn رواية قالون عن نافع
Al-dūrī رواية الدورى عن أبي عمرو

Figure 2: The first two lines of surat al-ra'd

Fig. 2 shows the same quotes presented in Fig. 1 written by the current version of AlQalam. This example reveals a few more of the new features. The inter-word spacing in the high quality arabic script typography is much smaller than the latin script. In some cases it is even absent completely. 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

where the command `\newspacefalse` is used in the second case to retain a large spacing as in ArabTEX and in the 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　فَٱدَّٰرَءْتُمْ　وَٱتَّقُواْ　بِٱسْمِ

which is possible in the current version.

Yet another feature concerns the position of diacritic marks on top of the letters. Compare أسْماء to أسْماء. 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 رَوَٰسِىَ. On the other hand, it is considered a mark on top of the underlying character in cases such as ٱسْتَوَىٰ ٱلصَّلَوٰةَ ٱلْحَيَوٰة. 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 ٱسْتَوَىٰ even if it is followed by a connected pronoun as in سَوَّىٰهُ تَقْوَىٰهَا زَكَّىٰهَا,
- on a و which is followed by ا as in ٱلرِّبَوٰاْ, or
- on a و which is followed by ة as in ٱلصَّلَوٰةَ.

The current coloring scheme has 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 latin script, it is easy to to get 'text' via `'te\textcolor{blue}{x}t'` which is fine as long as no special ligatures or kerning is need 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 apriori, we code them in the system. So, 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 existance of the requested letter sequence. If a part of the word matches the pattern, we color it. Such a hard-coded "programatic" way is not suitable for arbitary coloring that the user may wish to introduce into a regular text outside of the Qur'anic quotations.

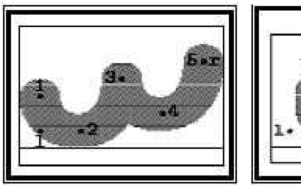


Figure 3: Effect of changing the pen and improving the shape on the "shaddah".

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

مَكيَّة　سُـورَةُ العَاديَاتِ　١١ ءَايَـة

and

مَكيَّة　سُـورَةُ الإِخلاَصِ　٤ ءَايَات

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.

The following figure shows a number of 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 ˆ in the input file.

﴿أَلَّا يَسْجُدُوا لِلَّهِ ٱلَّذِى يُخْرِجُ ٱلْخَبْءَ فِى ٱلسَّمَٰوَٰتِ
وَٱلْأَرْضِ وَيَعْلَمُ مَا تُخْفُونَ وَمَا تُعْلِنُونَ ٢٥ ٱللَّهُ لَآ إِلَٰهَ
إِلَّا هُوَ رَبُّ ٱلْعَرْشِ ٱلْعَظِيمِ ۩ ٢٦ ۞ قَالَ سَنَنظُرُ
أَصَدَقْتَ أَمْ كُنتَ مِنَ ٱلْكَٰذِبِينَ ٢٧ ٱذْهَب بِّكِتَٰبِى هَٰذَا
فَأَلْقِهْ إِلَيْهِمْ ثُمَّ تَوَلَّ عَنْهُمْ فَٱنظُرْ مَاذَا يَرْجِعُونَ ٢٨﴾
سُـورَةُ النَّملِ

ثلاثة ارباعِ
الحزب
٣٨
سَجدَة

4 Future work

A lot of 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. However, we think that we brought AlQalam quite close to the limits of such an approach.

To handle line breaking and justification correctly, a much more fundamental change into TEX itself is needed. After describing the line breaking algorithm of TEX [6], 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 descendents have this refinement implemented. In the case of 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, XeTEX, OrientalTEX, ...) will include this change to the badness function. A lot of experimentation using several different languages will be needed to come up with the most suitable badness function.

Another issue that requires more work is the contextual analysis to decide on the glyphs used. This analysis is not only within a word but also it must sometimes look at two consecutive words. The following example explains 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 ـٌ ـً ـٍ to ـٌۢـًۢـٍۭ if the following letter is ب or to ـࣱ ـࣰ ـࣲ 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 the counters for the verses and the 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. Obviously, 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.

References

[1] Hossam A. H. Fahmy. Typesetting the Qur'an and its specific challenges to the TEX family. In *EuroTEX 2006: Proceedings of the* 16th *Annual Meeting of the European TEX Users, Debrecen, Hungary*, July 2006.

[2] 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.

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

[4] 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.

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

[6] 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.