Oriental TEX: optimizing paragraphs

Introduction

One of the objectives of the Oriental TEX project has always been to play with paragraph optimization. The original assumption was that we needed an advanced nonstandard paragraph builder to Arabic done right but in the end we found out that a more straightforward approach is to use a sophisticated OpenType font in combination with a paragraph postprocessor that uses the advanced font capabilities. This solution is somewhat easier to realise than a complex paragraph builder but still involves quite some juggling.

At the June 2012 meeting of the ntg there was a talk about typesetting Devanagari and as fonts are always a nice topic (if only because there is something to show) it made sense to tell a bit more about optimizing Arabic at the same time. In fact, that presentation was already a few years too late because a couple of years back, when the oriental TEX project was presented at tug and Dante meetings, the optimizer was already part of the ConTEXt core code. The main reason for not advocating it was the simple fact that no font other than the (not yet finished) Husayni font provided the relevant feature set.

The lack of advanced fonts does not prevent us from showing what we're dealing with. This is because the ConTEXt mechanisms are generic in the sense that they can also be used with regular Latin fonts, although it does not make that much sense. Anyhow, in the next section we wrap up the current state of typesetting Arabic in ConTEXt. We focus on the rendering, and leave general aspects of bidirectional typesetting and layouts for another time.

This article is written by Idris Samawi Hamid and Hans Hagen and is typeset by ConTEXt MkIV which uses LuaTEX. This program is an extension of TEX that uses Lua to open op the core machinery. The LuaTEX core team consists of Taco Hoekwater, Hartmut Henkel and Hans Hagen.

Manipulating glyphs

When discussing optical optimization of a paragraph, a few alternatives come to mind:

- One can get rid of extensive spaces by adding additional kerns between glyphs. This is often used by poor man's typesetting programs (or routines) and can be applied to non-connecting scripts. It just looks bad. Of course, for connected scripts like Arabic, inter-glyph kerning is not an option, not even in principle.
- Glyphs can be widened a few percent and this is an option that LuaTEX inherits from its predecessor pdfTEX. Normally this goes unnoticed although excessive scaling makes things worse, and yes, one can run into such examples. This strategy goes under the name hz-optimization (the hz refers to Hermann Zapf, who first came up with this solution).[1]
- A real nice solution is to replace glyphs by narrower or wider variants. This is in fact the ideal hz solution —including for Arabic-script as well— but for it to happen

one not only needs needs fonts with alternative shapes, but also a machinery that can deal with them.

- An already old variant is the one first used by Gutenberg, who used alternative cuts for certain combinations of characters. This is comparable with ligatures. However, to make the look and feel optimal, one needs to analyze the text and make decisions on what to replace without losing consistency.

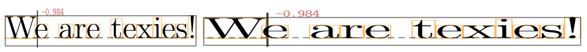
The solution described here does a bit of everything. As it is mostly meant for a connective script, the starting point is how a scribe works when filling up a line nicely. Depending on how well one can see it coming, the writing can be adapted to widen or narrow following words. And it happens that in Arabic-script there are quite some ways to squeeze more characters in a small area and/or expand some to the extreme to fill up the available space. Shapes can be wider or narrower, they can be stacked and they can get replaced by ligatures. Of course there is some interference with the optional marks on top and below but even there we have some freedom. The only condition is that the characters in a word stay connected.[2]

So, given enough alternative glyphs, one can imagine that excessive interword spacing can be avoided. However, it is non-trivial to check all possible combinations. Actually, it is not needed either, as carefully chosen aesthetic rules put some bounds on what can be done. One should more think in terms of alternative strategies or solutions and this is the terminology that we will therefore use.

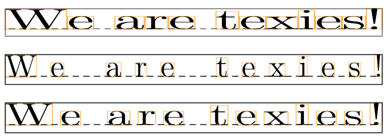
Scaling glyphs horizontally is no problem if we keep the scale factor very small, say percentages. This also means that we should not overestimate the impact. For the Arabic script we can stretch more —using non-scaling methods— but again there are some constraints, which we will discuss later on.

In the next example, we demonstrate some excessive stretching:

In practice, fonts can provide intercharacter kerning, which is demonstrated next:



Some poor man's justification routines mess with additional inter-character kerning. Although this is, within reasonable bounds, ok for special purposes like titles, it looks bad in text. The first line expands glyphs and spaces, the second line expands spaces and adds additional kerns between characters and the third line expands and adds extra kerns.



Unfortunately we see quite often examples of the last method in novels and even scientific texts. There is definitely a down side to advanced manipulation.

Applying features to Latin-script

It is easiest to start out with Latin, if only because it's more intuitive for most of us to see what happens. This is not the place to discuss all the gory details so you have to take some of the configuration options on face value. Once this mechanism is stable and used, the options can be described. For now we stick to presenting the idea.

Let's assume that you know what font features are. The idea is to work with combinations of such features and figure out what combination suits best. In order not to clutter a document style, these sets are defined in so called goodie files. Here is an excerpt of `demo.lfg`:

```
return {
  name = "demo",
  version = "1.01",
  comment = "An example of goodies.",
  author = "Hans Hagen",
  featuresets = {
    simple = {
      mode   = "node",
      script = "latn"
    },
    default = {
      mode   = "node",
      script = "latn",
      kern   = "yes",
    },
    ligatures = {
      mode   = "node",
      script = "latn",
      kern   = "yes",
      liga   = "yes",
    },
    smallcaps = {
      mode   = "node",
      script = "latn",
      kern   = "yes",
      smcp   = "yes",
    },
  },
  solutions = {
    experimental = {
      less = {
        "ligatures", "simple",
      },

      more = {
        "smallcaps",
      },
    },
  },
}
```

We see four sets of features here. You can use these sets in a ConTEXt feature definition, like:

```
\definefontfeature
  [solution-demo]
  [goodies=demo,
   featureset=default]
```

You can use a set as follows:

```
\definefont
  [SomeTestFont]
  [texgyrepagellaregular*solution-demo at 10pt]
```

So far, there is nothing special or new, but we can go a step further.

```
\definefontsolution
  [solution-a]
  [goodies=demo,
   solution=experimental,
   method={normal,preroll},
   criterium=1]

\definefontsolution
```

```
  [solution-b]
  [goodies=demo,
   solution=experimental,
   method={normal,preroll,split},
   criterium=1]
```

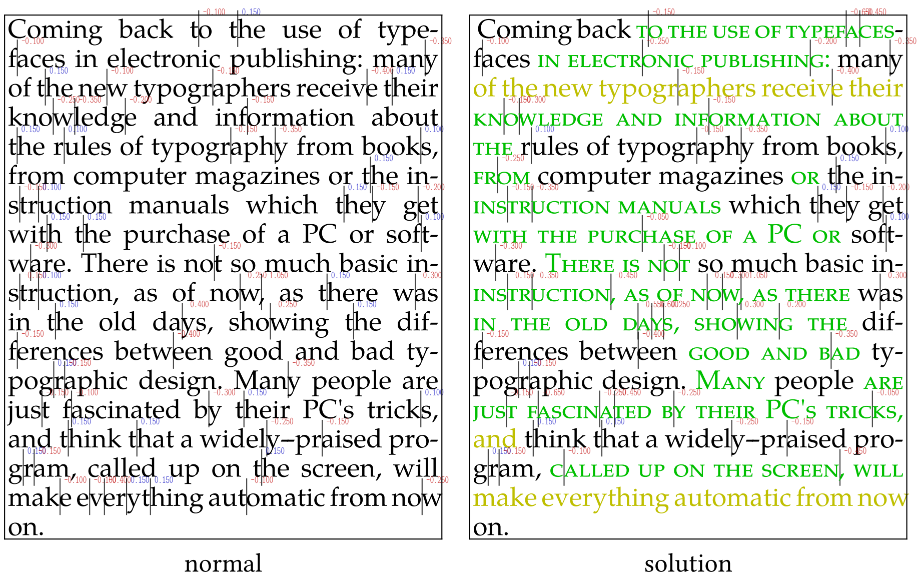
Here we have defined two solutions. They refer to the `experimental` solution in the goodie file `demo.lfg`. A solution has a `less` and a `more` entry. The featuresets mentioned there reflect ways to make a word narrower or wider. There can be more than one way to do that, although it comes at a performance price. Before we see how this works out we turn on a tracing option:

```
\enabletrackers
  [builders.paragraphs.solutions.splitters.colors]
```

This will color the words in the result according to what has happened. When a featureset out of the `more` category has been applied, the words turn green, when `less` is applied, the word becomes yellow. The `preroll` option in the `method` list makes sure that we do a more extensive test beforehand.

```
\SomeTestFont \startfontsolution[solution-a]
\input zapf \par
\stopfontsolution
```

In Figure 1 we see what happens. In each already split line words get wider or narrower until we're satisfied. A criterium of 1 is pretty strict[3]. Keep in mind that we use some arbitrary features here. We try removing kerns to get narrower although there is nothing that guarantees that kerns are positive. On the other hand, using ligatures might help. In order to get wider we use smallcaps. Okay, the result will look somewhat strange but so does much typesetting nowadays.



Coming back to the use of type-
faces in electronic publishing: many
of the new typographers receive their
knowledge and information about
the rules of typography from books,
from computer magazines or the in-
struction manuals which they get
with the purchase of a PC or soft-
ware. There is not so much basic in-
struction, as of now, as there was
in the old days, showing the dif-
ferences between good and bad ty-
pographic design. Many people are
just fascinated by their PC's tricks,
and think that a widely–praised pro-
gram, called up on the screen, will
make everything automatic from now
on.

normal

Coming back TO THE USE OF TYPEFACES-
faces IN ELECTRONIC PUBLISHING: many
of the new typographers receive their
KNOWLEDGE AND INFORMATION ABOUT
THE rules of typography from books,
FROM computer magazines OR the in-
INSTRUCTION MANUALS which they get
WITH THE PURCHASE OF A PC OR soft-
ware. THERE IS NOT so much basic in-
INSTRUCTION, AS OF NOW, AS THERE was
IN THE OLD DAYS, SHOWING THE dif-
ferences between GOOD AND BAD ty-
pographic design. MANY people ARE
JUST FASCINATED BY THEIR PC'S TRICKS,
and think that a widely–praised pro-
gram, CALLED UP ON THE SCREEN, WILL
make everything automatic from now
on.

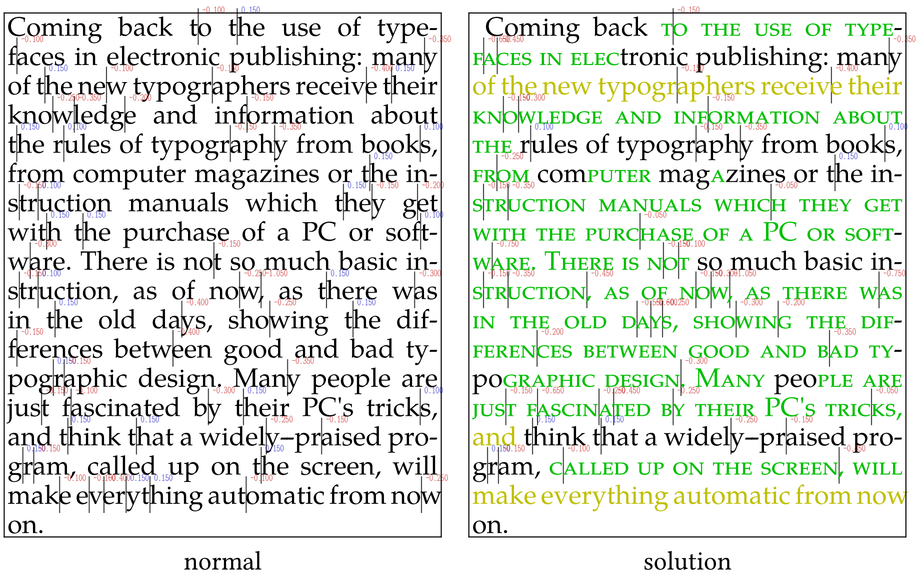
solution

Figure 1. Solution a.

There is one pitfall here. This mechanism is made for a connective script where hyphenation is not used. As a result a word here is actually split up when it has discretionaries and of course this text fragment has. It goes unnoticed in the rendering but is of course far from optimal.

```
\SomeTestFont \startfontsolution[solution-b]
\input zapf \par
\stopfontsolution
```

In this example (Figure 2) we keep words as a whole but as a side effect we skip words that are broken across a line. This is mostly because it makes not much sense to implement it as Latin is not our target. Future versions of ConTEXt might get more sophisticated font machinery so then things might look better.


Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the instruction manuals which they get with the purchase of a PC or software. There is not so much basic instruction, as of now, as there was in the old days, showing the differences between good and bad typographic design. Many people are just fascinated by their PC's tricks, and think that a widely–praised program, called up on the screen, will make everything automatic from now on.

normal

Coming back TO THE USE OF TYPEFACES IN ELECtronic publishing: many of the new typographers receive their KNOWLEDGE AND INFORMATION ABOUT THE rules of typography from books, FROM comPUTER magAzines or the inSTRUCTION MANUALS WHICH THEY GET WITH THE PURCHASE OF A PC OR SOFTWARE. THERE IS NOT so much basic inSTRUCTION, AS OF NOW, AS THERE WAS IN THE OLD DAYS, SHOWING THE DIFFERENCES BETWEEN GOOD AND BAD TYpoGRAPHIC DESIGN. MANY peoPLE ARE JUST FASCINATED BY THEIR PC'S TRICKS, and think that a widely–praised program, CALLED UP ON THE SCREEN, WILL make everything automatic from now on.

solution

Figure 2. Solution b.

We show two more methods:

```
\definefontsolution
  [solution-c]
  [goodies=demo,
   solution=experimental,
   method={reverse,preroll},
   criterium=1]

\definefontsolution
  [solution-d]
  [goodies=demo,
   solution=experimental,
   method={random,preroll,split},
   criterium=1]
```

In Figure 3 we start at the other end of a line. As we sort of mimick a scribe, we can be one who plays safe at the start of corrects at the end.

In Figure 4 we add some randomness but to what extent this works well depends on how many words we need to retypeset before we get the badness of the line within the constraints.

Salient features of Arabic-script

Before applying the above to Arabic-script, let's discuss some salient aspects of the problem. As a cursive script, Arabic is extremely versatile and the scribal calligraphy tradition reflects that. Digital Arabic typography is only beginning to catch up with

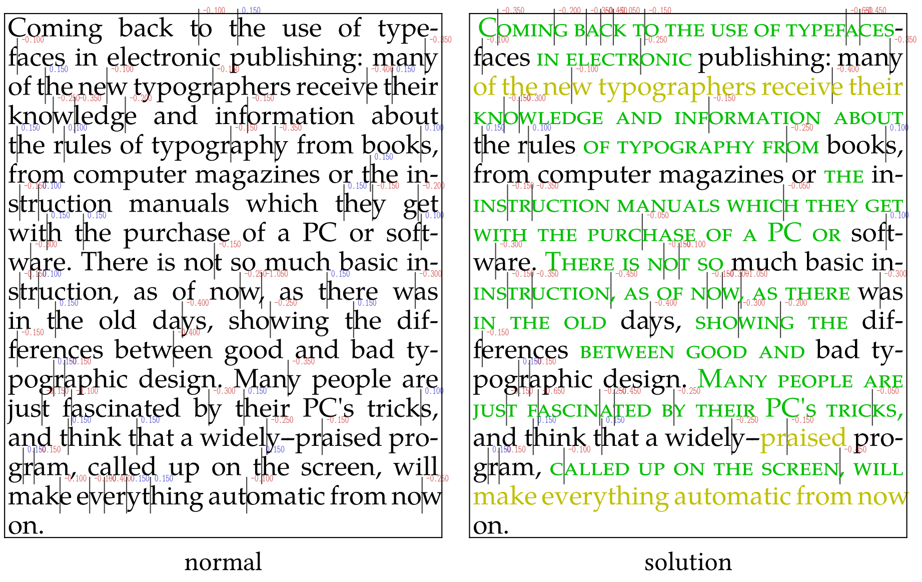


Figure 3. Solution c.

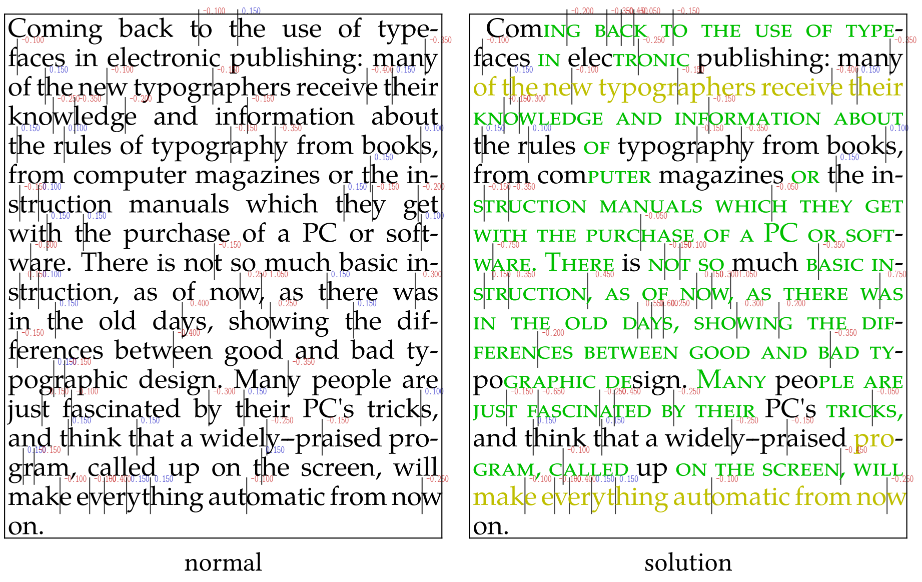


Figure 4. Solution d.

the possibilities afforded by the scribal tradition. Indeed, early lead-punch typography and typesetting of Arabic-script was more advanced than most digital typography even up to this day. In any case, let us begin to organize some of that versatility into a taxonomy for typography purposes.

What's available?

We have to work within the following parameters:

☐ No hyphenation ever (well, almost never)
It is commonly pointed out that there is no hyphenation in Arabic. This is something of a half-truth. In the manuscript tradition one actually does find something akin to hyphenation. In the ancient Kufic script, breaking a word across lines is

actually quite common. But even in the more modern Naskh script, the one most normal Arabic text fonts are based on, it does occur, albeit rarely and presumably when the scribe is out of options for the line he is working on. Indeed, one could regard it as a failure on the part of the scribe once he reaches the end of the line.[4]

But there is still an important rule, regardless of whether we use Naskh, Kufic, or any other Arabic script. Consider the word below:

الفاعل

It is a single word composed of two cursive strings. One could actually hyphenate it, with our rule being to break it at the end of the first cursive string and before the beginning of the second cursive string:

هذا اسم الفا
عل

Again, it's a rare phenomenon and hardly ever occurs in modern typesetting, lead-punch or digital, if at all. On the other hand, it could have some creative uses in future Arabic-script typography.

□ Macrotypography (aesthetic features)
In Arabic there are often numerous aesthetic ways of writing out the exact same semantic string:[5]

الحمد الحمد الحمد

Normally we combine OpenType features into feature sets that are each internally and aesthetically coherent. So in the above example we have used three different sets, reading from right to left. We'll call them simple, default, and dipped.

Just as Latin typography uses separate fonts to mark off different uses of text (bold, italic, etc.), an advanced Arabic font can use aesthetic feature sets to similar effect. This works best on distinguishing long streams of text from one another, since the differences between feature sets are not always noticeable on short strings. That is, two different aesthetic sets may type a given short string, such as a single word, exactly the same way. Consider the above three sets (simple, default, and dipped) once more:

علي علي علي

For the above string the default and dipped aesthetic sets (middle and left) give the exact same result, while the basic one (right) remains, well, quite basic.

Let's go back to our earlier example:

الحمد الحمد الحمد

Note that the simple version is wider than the default, and the dipped version is (slightly) thinner than the default. This relates to another point: An aesthetic feature set can serve two functions:

1. It can serve as the base aesthetic style.
2. It can serve as a resource for glyph substitution for a given string in another base aesthetic style.

This brings us back to our main topic.

□ Microtypography (paragraph optimization features)
Here our job is to optimize the paragraph for even spacing and aesthetic viewing. It turns out that there are a number of ways to look at this issue, and we will begin exploring these in the next subsection.

Two approaches

Let us start off with a couple of samples. Qurʾānic transcription has always been the gold standard of Arabic-script. In Figure 5 we see a nice example of scribal optimization. The scribe here is operating under the constraint that each page ends with the

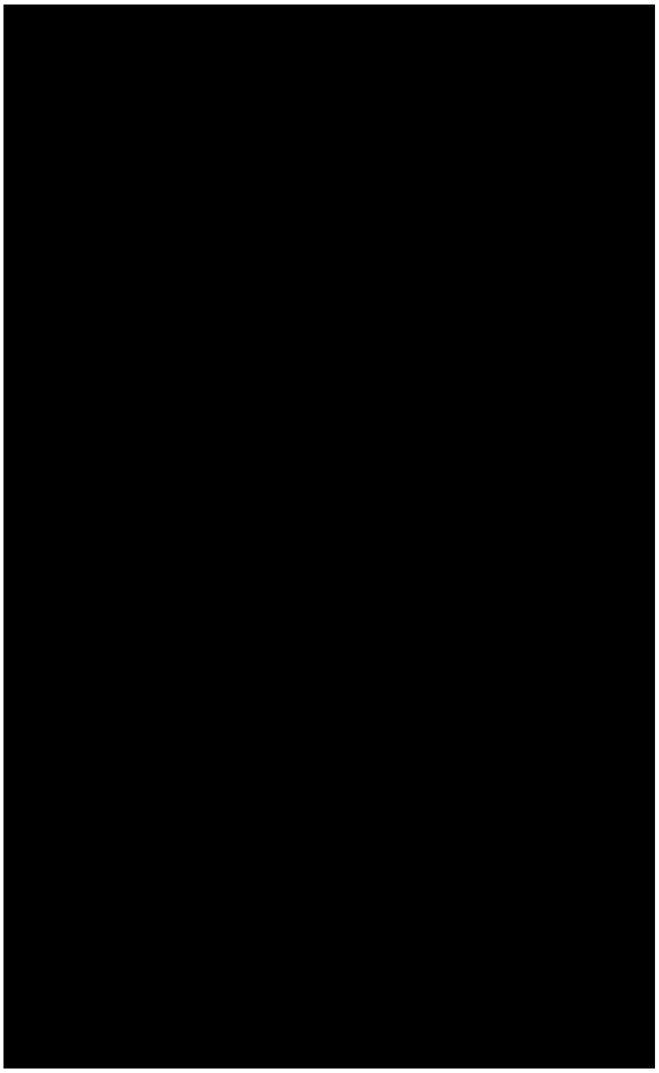


Figure 5. Scribal Optimization. Scribe: *ᶜUthmān Ṭāhā*. Qurʾān, circa 1997.

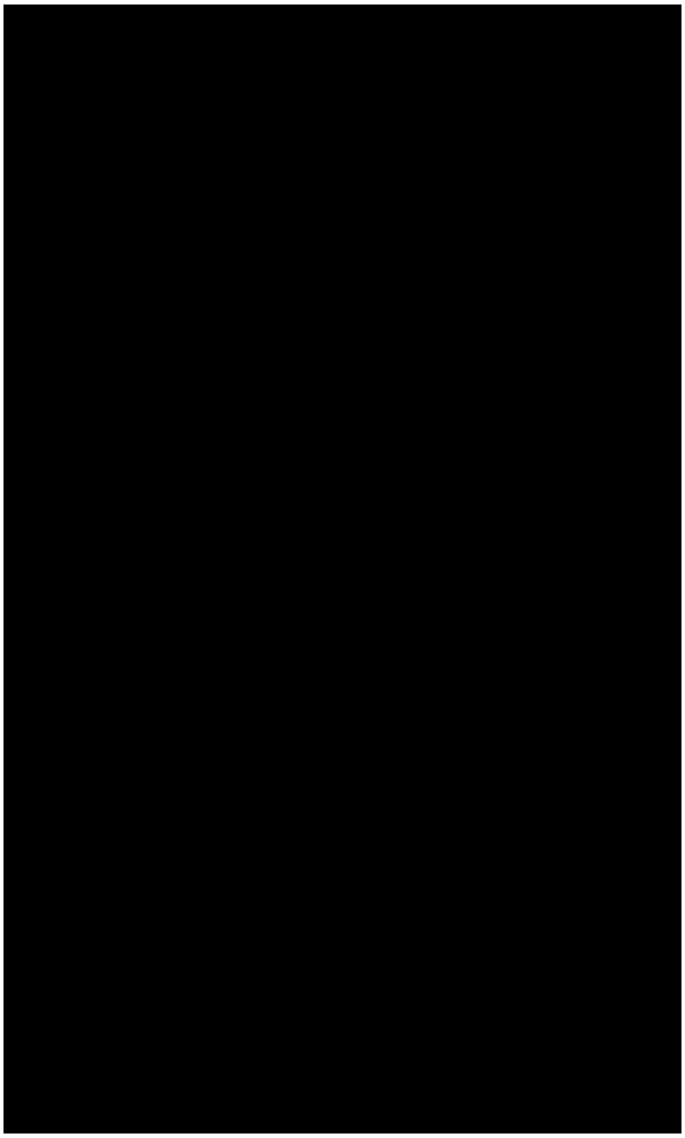


Figure 6. Alternate Fixed Glyphs. From the *al-Husayni Muṣḥaf* of the Qurʾān, 1923.

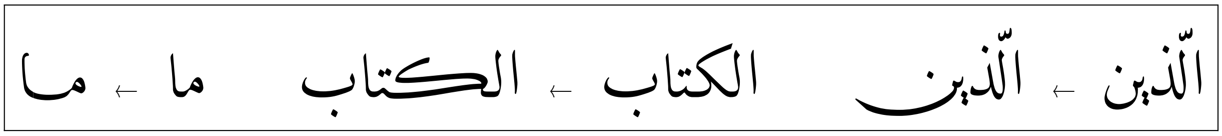
end of a Qurʾānic verse (designated by the symbol `U+06DD` ۝). That is, no verse is broken across pages. That constraint, which is by no means mandatory or universal, gives the scribe lots of space for optimization, even more than normal.

In Figure 6 we have a page of the famous *al-Husayni Muṣḥaf* of 1919–1923, which remains up to this day the only typeset copy of the Qurʾān to attain general acceptance in the Muslim world. Indeed, it remains the standard 'edition' of the Qurʾān and even later scribal copies, such as the one featured in Figure 5 are based on its orthography. Unlike the scribal version, the typesetters of the *al-Husayni Muṣḥaf* did not try to constrain each page to end with the end of a Qurʾānic verse. Again, that is a nice feature to have as it makes recitation somewhat easier but it is by no means a mandatory one.

In any case, both samples share verses 172–176 in commmon, so there is lots to compare and contrast. We will also use these verses as our main textual sample for paragraph optimization.

Using Figure 5 and Figure 6 as benchmarks, we can begin by analyzing the approaches to paragraph optimization in Arabic-script typography into two kinds:

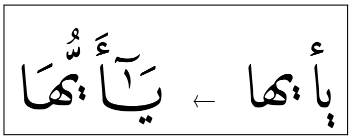
- □ Alternate glyphs
  Much of pre-digital Arabic typography uses this method. Generally, a wide variant of a letter is used to take up the space which would normally get absorbed by hyphenation in Latin. Here are examples of three of the most common substitutions, again, reading from right to left:



Each of the six strings above occurs in Figure 6. Identifying them is an exercise left to the reader. We call these kinds of alternate glyphs *alternate-shaped* glyphs.

The three substitutions above are the most common alternate-glyph substitutions found in pre-digital Arabic-script typography, including some contextual variants (initial, medial, final, and isolated) where appropriate. (The scribal tradition contains a lot more alternate-shaped glyphs. A few lead-punch fonts implement some of them, and we have implemented many of these in our Husayni font.) The results generally look quite nice and much more professional than most digital Arabic typography, which generally dispenses with these alternates.

But one also finds attempts at *extending* individual characters without changing the shape very much. One finds this already in Figure 6. We call these kinds of alternate glyphs *naturally curved widened* glyphs, or just *naturally widened* glyphs for short. Sometimes this is done for the purpose of making enough space for the vowels (which in Arabic take the form of diacritic characters). For example:

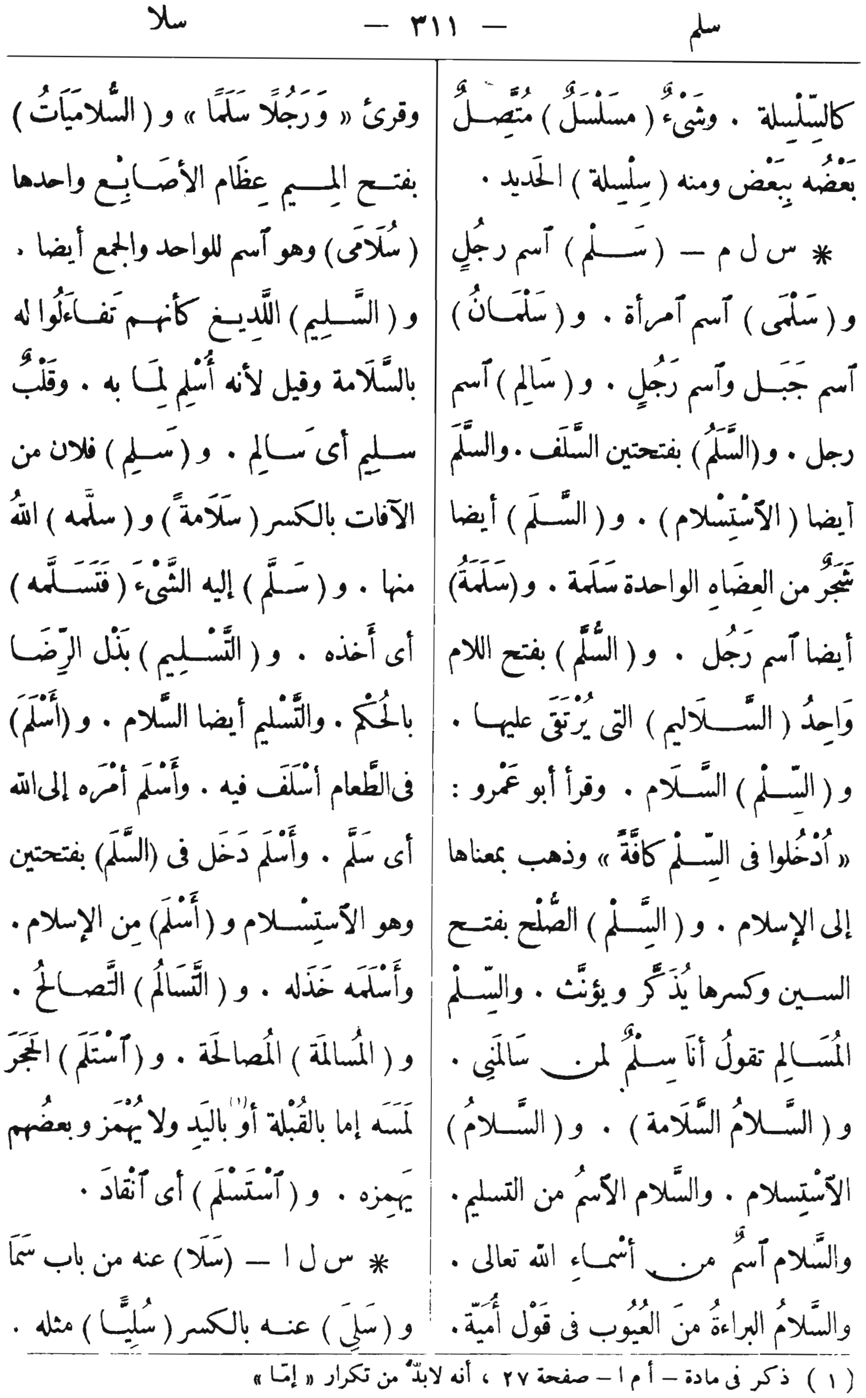


As you can see, there are two letters that have been *widened* for vowel accommodation. In Figure 6 there are some good but near-clumsy attempts at this. We say, 'near-clumsy' because the typographers and typesetters mix natural, curved, *widened* variants of letters with flat, horizontal, *extended* versions. One reason for this is that a full repertoire of naturally curved glyph alternates would be much too unwieldy for even the best lead-punch typesetting machines and their operators. Even with these limitations one can find brave examples of lead-based typesetting that do a good job of sophisticated paragraph optimization via glyph alternates, both widened and alternate-shaped. Figure 7 is a representative example (in the context of columns).

Careful examination of this two-column sample will reveal the tension between naturally *widened* and horizontally *extended* glyphs in the execution of paragraph optimization. On the other hand, there is one apparent 'rule' that one finds in this and other examples of lead-punch Arabic-script typesetting:

> *Generally, there is only one naturally widened character per word or one alternate-shaped character per word.*

In Figure 5 one can see that this 'rule' is not always observed by scribes, see, e.g., the middle word in line 9 from the top, which uses two of the alternate-shaped characters we encountered above (can you identify that word?). But we still need



سلم — ٣١١ — سلا

كالسِّلْسِلة . وشَىْءٌ (مسَلْسَلٌ) متَّصِلٌ
بَعْضُه بِبَعْض ومنه (سِلْسِلة) الحَديد .
* س ل م — (سَـــلْم) ٱسم رجُلٍ
و (سَلْمَى) ٱسم ٱمرأة . و (سَلْمَـانُ)
ٱسم جَبَـل وٱسم رَجُلٍ . و (سَالِم) ٱسم
رجل . و (السَّلَمُ) بفتحتين السَّلَف . والسَّلَمَ
أيضا (الٱستِسْلام) . و (السَّـلَم) أيضا
شَجَرٌ من العِضَاهِ الواحدة سَلَمَة . و (سَلَمَةُ)
أيضا ٱسم رَجُل . و (السُّلَّم) بفتح اللام
وَاحِدُ (السَّـــلَالِيم) التى يُرْتَقَى عليهـا .
و (السِّـلْم) السَّـلَام . وقرأ أبو عَمْرو :
« ٱدْخُلوا فى السِّـلْم كافَّةً » وذهب بمعناها
إلى الإسلام . و (السِّـلْم) الصُّلْح بفتـح
السـين وكسرها يُذَكَّر ويؤنَّث . والسِّـلْم
المُسَـالِم تقولُ أنَا سِـلْمٌ لمن سَالَمَنِي .
و (السَّـلامُ السَّلَامة) . و (السَّـلامُ)
الٱستِسلام . والسَّلام الٱسمُ من التسليم .
والسَّلام ٱسمٌ مِن أسْمـاءِ الله تعالى .
والسَّلامُ البراءةُ مِنَ العُيُوب فى قَوْل أُمَيَّة .

وقرئ « وَرَجُلًا سَلَمًا » و (السُّلامَيَاتُ)
بفتـح المِـــيم عِظَام الأصَـابِـع واحدها
(سُلَامَى) وهو ٱسم للواحد والجمع أيضا .
و (السَّـلِيم) اللَّدِيـغ كأنهـم تَفـاءَلُوا له
بالسَّلَامة وقيل لأنه أُسْلِمَ لِمَـا به . وقَلْبٌ
سـلِيم أى سَـالِم . و (سَـلِم) فلان من
الآفات بالكسر (سَلَامةً) و (سلّمه) اللهُ
منها . و (سَـلَّم) إليه الشَّىْءَ (فتَسَـلَّمه)
أى أَخذه . و (التَّسْـلِيم) بَذْل الرِّضَـا
بالحُكْم . والتَّسْليم أيضا السَّلام . و (أَسْلَمَ)
فى الطَّعام أسْلَفَ فيه . وأَسْلَمَ أمْرَه إلى الله
أى سَلَّمَ . وأَسْلَمَ دَخَل فى (السَّلَم) بفتحتين
وهو الٱستِسْـلام و (أَسْلَمَ) مِن الإسلام .
وأَسْلَمَه خَذَله . و (التَّسَالُم) التَّصالُحُ .
و (المُسالَمَة) المُصالَحَة . و (ٱسْتَلَمَ) الحَجَرَ
لَمَسَه إما بالقُبْلة أو(١) باليَدِ ولا يُهْمَز وبعضُهم
يَهمِزه . و (ٱسْتَسْلَمَ) أى ٱنقاد .
* س ل ا — (سَلَا) عنه من باب سَمَا
و (سَلِيَ) عنـه بالكسر (سُلِيًّا) مثله .

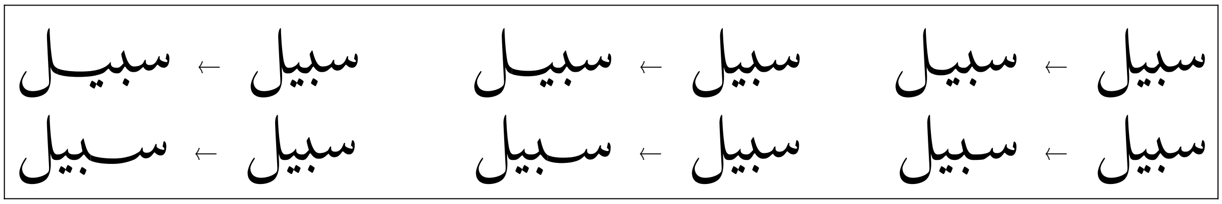
(١) ذكر فى مادة — أ م ا — صفحة ٢٧ ، أنه لابدّ من تكرار « إمّا »

Figure 7. Mixed Alternate Glyphs in Two Columns. From the classical dictionary *Mukhtār al-Ṣiḥāḥ*.

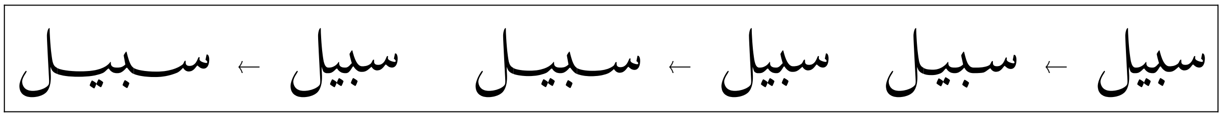
some constraints for decent-looking typesetting, and the above tentative rule is a good place to start the analysis. For widened characters in particular we see that even the scribe (Figure 5) closely approximates this rule. So let's begin improving on our tentative rule somewhat, and expand it into a number of possibilities. Let's look at the naturally-widened-glyph case first:

> *Generally, there is only one naturally widened character allowed per word. However, two extended non-consecutive characters may be allowed.* (The logic of the experimental font Husayni already has contraints that prevent *consecutive* curved widened characters).

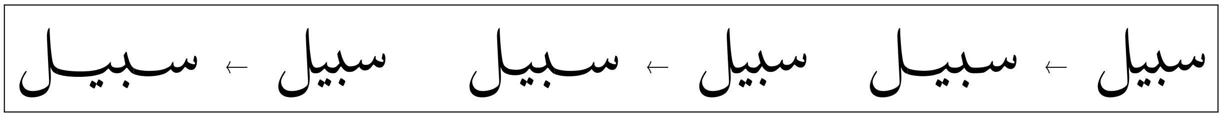
For example, we prefer to get widening like the following:



But as, e.g., a last resort or for stylistic purposes we can also do



Or even better, we mix it up a bit. That is, if there is more than one widened character, one should be longer than the other, e.g.:



One will notice that the middle substitution (where the first widened character is longer than the second) does not look as good as the two outer ones (where the second is longer than the first). These kinds of aesthetic issues can be formalized for future work. In the meantime, here is a working modified version of the rule for naturally-widened-glyphs:

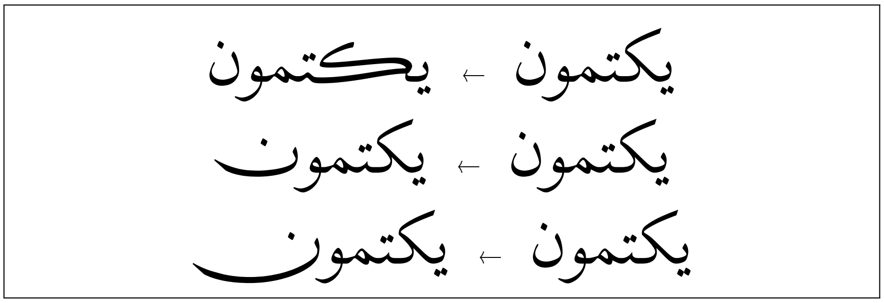
> *Generally, there is only one naturally widened character allowed per word. However, two non-consecutive widened characters may be allowed. In that case, the second widened character should be longer than the first.*

One case where cases of two naturally widened characters will be common is in poetry, which involves wide lines. We'll say more about this in the section on flat extending.

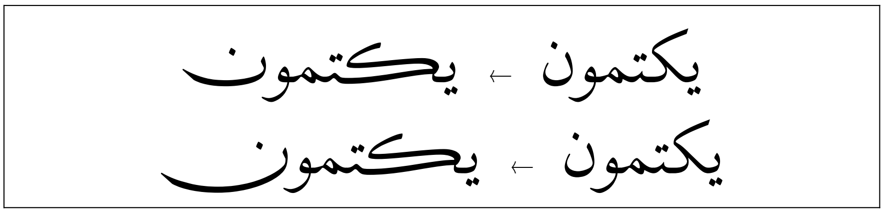
Now let's look at the alternate-shaped case:

> *Generally, there is only one alternate-shaped character allowed per word. However, two non-consecutive alternate-shaped characters may be allowed.*

So we prefer, e.g.,



but we could have, e.g, as a last resort or as a stylistic option,



Again, in poetry this kind of multiple substitution within a single word could occur frequently. A challenge will be to develop a system of parameters where we can almost predict which kinds of substitution will happen under a given set of values of those parameters.

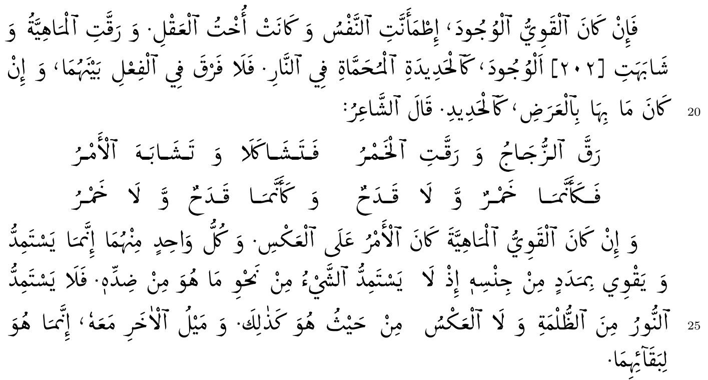
□ Flat extending

In the transition from lead-punch to digital typography, alternate-glyph substitution largely vanished.[6] The problem of spacing remained, and a simple yet inelegant solution was adopted: flat, horizontal extending of characters. Now this solution did have *some* precedent in pre-digital Arabic typography, as you can see in Figure 6 and Figure 7. This solution had the advantage that it required only a single character: a simple horizontal bar called a *taṭwīl* or more commonly a *kashidah* (`U+0640`). This character could then be repeated as often as necessary to fill any extra space.

Now an examination of pre-digital books shows a (rather wise) reticence to using this method too slavishly. That reticence has now been thrown to the winds. This can be seen by looking at the standard implementation of flat extending as provided by Microsoft Word. This program provides three levels of extending that it calls 'justification'. See Figure 9 for examples of all three. The minimum level is actually very close to the default (i.e., no-justification) level. Note that the sample text used in Figure 9 is the same as that used in the earlier samples from the Qur'ān.

Older implementations of Arabic-script within TEX, such as ArabTEX and Omega/Aleph, also provided facilities for flat extending. The most common use was in poetry, which requires a fixed width for each stanza.

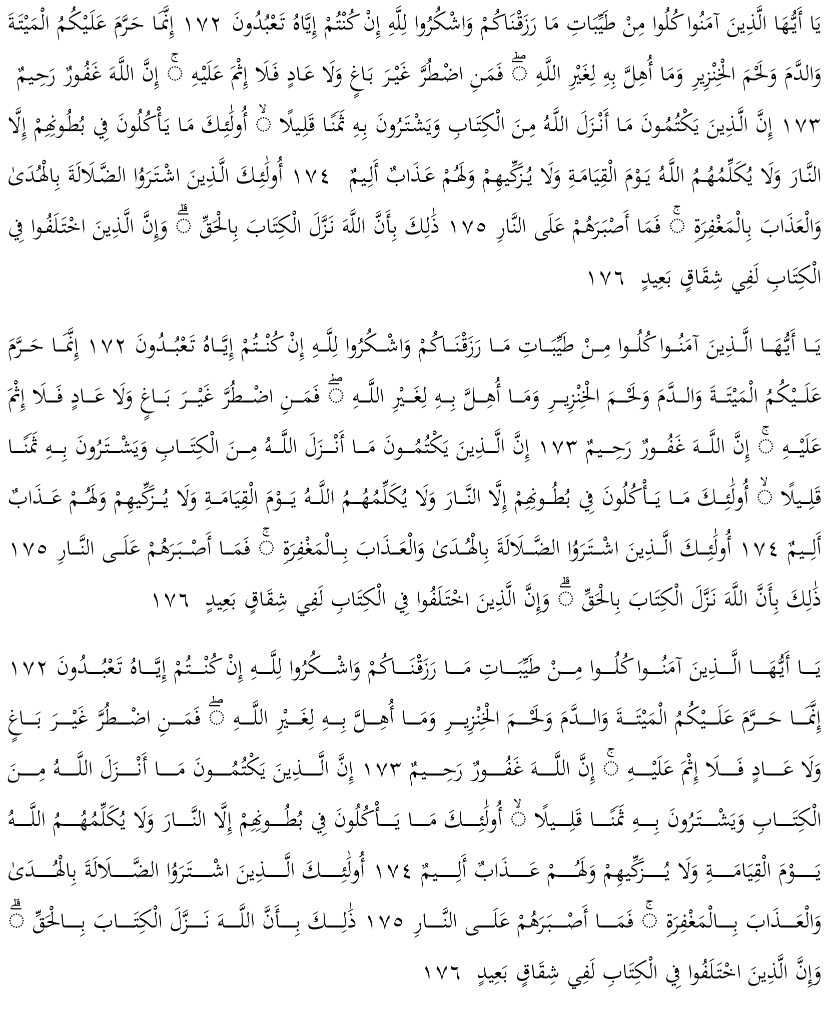
In Omega/Aleph, a method based on `\xleaders` was used, based on a very thin *taṭwīl* glyph (much thinner than `U+0640`) that could be used for very fine extending optimization based on TEX's badness parameter. One nice application is in marginal notes: See Figure 10, where the marginal note on the right is zoomed in. On the other hand, we see that the leaders method creates extending that may be considered too perfectly even: Do we want to impose the rule that only one character should be extended per word (or at most two non-consecutive characters)? I have seen a lot of older digital Arabic typography that does even extending, including the poetry in the ArabTEX sample in figure 8. Compare this with the Microsoft Word method (Figure 9). The method used in Microsoft Word, with only one extension per word, seems to be the current standard for flat-extending justification.


فَإِنْ كَانَ ٱلْقَوِيُّ ٱلْوُجودَ، إِطْمَأَنَّتِ ٱلنَّفْسُ وَ كَانَتْ أُخْتُ ٱلْعَقْلِ. وَ رَقَّتِ ٱلْمَاهِيَّةُ وَ
شَابَهَتِ [٢٠٢] اَلْوُجودَ، كَٱلْحَدِيدَةِ ٱلْمُحَمَّاةِ فِي ٱلنَّارِ. فَلَا فَرْقَ فِي ٱلْفِعْلِ بَيْنَهُمَا، وَ إِنْ
كَانَ مَا بِهَا بِٱلْعَرَضِ، كَٱلْحَدِيدِ. قَالَ ٱلشَّاعِرُ:

رَقَّ ٱلزُّجَاجُ وَ رَقَّتِ ٱلْخَمْرُ   فَتَشَاكَلَا وَ تَشَابَهَ ٱلْأَمْرُ
فَكَأَنَّمَا خَمْرٌ وَّ لَا قَدَحٌ   وَ كَأَنَّمَا قَدَحٌ وَّ لَا خَمْرُ

وَ إِنْ كَانَ ٱلْقَوِيُّ ٱلْمَاهِيَّةَ كَانَ ٱلْأَمْرُ عَلَى ٱلْعَكْسِ. وَ كُلُّ وَاحِدٍ مِنْهُمَا إِنَّمَا يَسْتَمِدُّ
وَ يَقْوِي بِمَدَدٍ مِنْ جِنْسِهِ إِذْ لَا يَسْتَمِدُّ ٱلشَّيْءُ مِنْ نَحْوِ مَا هُوَ مِنْ ضِدِّهِ. فَلَا يَسْتَمِدُّ
ٱلنُّورُ مِنَ ٱلظُّلْمَةِ وَ لَا ٱلْعَكْسُ مِنْ حَيْثُ هُوَ كَذٰلِكَ. وَ مَيْلُ ٱلْآخَرِ مَعَهُ، إِنَّمَا هُوَ
لِبَقَآئِهِمَا.

Figure 8. Poetry Justification in ArabTEX.


يَا أَيُّهَا الَّذِينَ آمَنُوا كُلُوا مِنْ طَيِّبَاتِ مَا رَزَقْنَاكُمْ وَاشْكُرُوا لِلَّهِ إِنْ كُنْتُمْ إِيَّاهُ تَعْبُدُونَ ١٧٢ إِنَّمَا حَرَّمَ عَلَيْكُمُ الْمَيْتَةَ
وَالدَّمَ وَلَحْمَ الْخِنْزِيرِ وَمَا أُهِلَّ بِهِ لِغَيْرِ اللَّهِ ۖ فَمَنِ اضْطُرَّ غَيْرَ بَاغٍ وَلَا عَادٍ فَلَا إِثْمَ عَلَيْهِ ۚ إِنَّ اللَّهَ غَفُورٌ رَحِيمٌ
١٧٣ إِنَّ الَّذِينَ يَكْتُمُونَ مَا أَنْزَلَ اللَّهُ مِنَ الْكِتَابِ وَيَشْتَرُونَ بِهِ ثَمَنًا قَلِيلًا ۙ أُولَٰئِكَ مَا يَأْكُلُونَ فِي بُطُونِهِمْ إِلَّا
النَّارَ وَلَا يُكَلِّمُهُمُ اللَّهُ يَوْمَ الْقِيَامَةِ وَلَا يُزَكِّيهِمْ وَلَهُمْ عَذَابٌ أَلِيمٌ ١٧٤ أُولَٰئِكَ الَّذِينَ اشْتَرَوُا الضَّلَالَةَ بِالْهُدَىٰ
وَالْعَذَابَ بِالْمَغْفِرَةِ ۚ فَمَا أَصْبَرَهُمْ عَلَى النَّارِ ١٧٥ ذَٰلِكَ بِأَنَّ اللَّهَ نَزَّلَ الْكِتَابَ بِالْحَقِّ ۗ وَإِنَّ الَّذِينَ اخْتَلَفُوا فِي
الْكِتَابِ لَفِي شِقَاقٍ بَعِيدٍ ١٧٦

يَا أَيُّهَا الَّذِينَ آمَنُوا كُلُوا مِنْ طَيِّبَاتِ مَا رَزَقْنَاكُمْ وَاشْكُرُوا لِلَّهِ إِنْ كُنْتُمْ إِيَّاهُ تَعْبُدُونَ ١٧٢ إِنَّمَا حَرَّمَ
عَلَيْكُمُ الْمَيْتَةَ وَالدَّمَ وَلَحْمَ الْخِنْزِيرِ وَمَا أُهِلَّ بِهِ لِغَيْرِ اللَّهِ ۖ فَمَنِ اضْطُرَّ غَيْرَ بَاغٍ وَلَا عَادٍ فَلَا إِثْمَ
عَلَيْهِ ۚ إِنَّ اللَّهَ غَفُورٌ رَحِيمٌ ١٧٣ إِنَّ الَّذِينَ يَكْتُمُونَ مَا أَنْزَلَ اللَّهُ مِنَ الْكِتَابِ وَيَشْتَرُونَ بِهِ ثَمَنًا
قَلِيلًا ۙ أُولَٰئِكَ مَا يَأْكُلُونَ فِي بُطُونِهِمْ إِلَّا النَّارَ وَلَا يُكَلِّمُهُمُ اللَّهُ يَوْمَ الْقِيَامَةِ وَلَا يُزَكِّيهِمْ وَلَهُمْ عَذَابٌ
أَلِيمٌ ١٧٤ أُولَٰئِكَ الَّذِينَ اشْتَرَوُا الضَّلَالَةَ بِالْهُدَىٰ وَالْعَذَابَ بِالْمَغْفِرَةِ ۚ فَمَا أَصْبَرَهُمْ عَلَى النَّارِ ١٧٥
ذَٰلِكَ بِأَنَّ اللَّهَ نَزَّلَ الْكِتَابَ بِالْحَقِّ ۗ وَإِنَّ الَّذِينَ اخْتَلَفُوا فِي الْكِتَابِ لَفِي شِقَاقٍ بَعِيدٍ ١٧٦

يَا أَيُّهَا الَّذِينَ آمَنُوا كُلُوا مِنْ طَيِّبَاتِ مَا رَزَقْنَاكُمْ وَاشْكُرُوا لِلَّهِ إِنْ كُنْتُمْ إِيَّاهُ تَعْبُدُونَ ١٧٢
إِنَّمَا حَرَّمَ عَلَيْكُمُ الْمَيْتَةَ وَالدَّمَ وَلَحْمَ الْخِنْزِيرِ وَمَا أُهِلَّ بِهِ لِغَيْرِ اللَّهِ ۖ فَمَنِ اضْطُرَّ غَيْرَ بَاغٍ
وَلَا عَادٍ فَلَا إِثْمَ عَلَيْهِ ۚ إِنَّ اللَّهَ غَفُورٌ رَحِيمٌ ١٧٣ إِنَّ الَّذِينَ يَكْتُمُونَ مَا أَنْزَلَ اللَّهُ مِنَ
الْكِتَابِ وَيَشْتَرُونَ بِهِ ثَمَنًا قَلِيلًا ۙ أُولَٰئِكَ مَا يَأْكُلُونَ فِي بُطُونِهِمْ إِلَّا النَّارَ وَلَا يُكَلِّمُهُمُ اللَّهُ
يَوْمَ الْقِيَامَةِ وَلَا يُزَكِّيهِمْ وَلَهُمْ عَذَابٌ أَلِيمٌ ١٧٤ أُولَٰئِكَ الَّذِينَ اشْتَرَوُا الضَّلَالَةَ بِالْهُدَىٰ
وَالْعَذَابَ بِالْمَغْفِرَةِ ۚ فَمَا أَصْبَرَهُمْ عَلَى النَّارِ ١٧٥ ذَٰلِكَ بِأَنَّ اللَّهَ نَزَّلَ الْكِتَابَ بِالْحَقِّ ۗ
وَإِنَّ الَّذِينَ اخْتَلَفُوا فِي الْكِتَابِ لَفِي شِقَاقٍ بَعِيدٍ ١٧٦

Figure 9. Flat justification from Microsoft Word 2010.

On the other hand, the justification used in Microsoft Word is not particularly aesthetically pleasing. The answer will lie, again, in parameterization of some sort

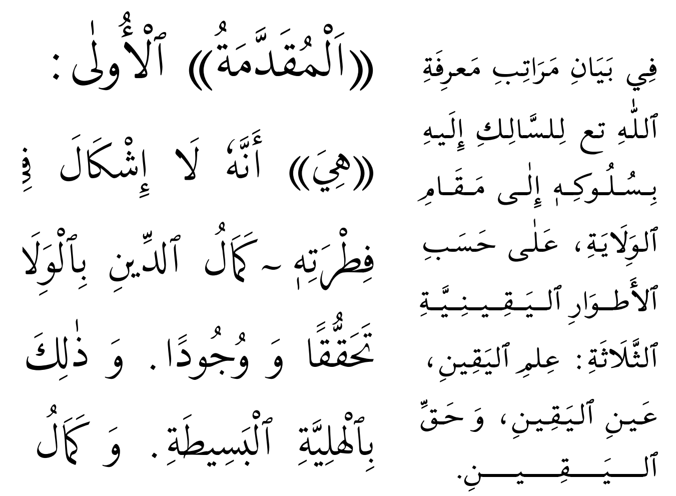
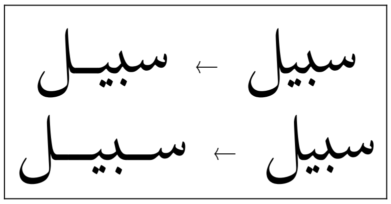


Figure 10. Marginal-note justification in Omega/Aleph.

to be determined. As TEXies, we want to be able to have fine control over this kind of behavior in any case. In the meantime, we mirror the same rule we arrived at for naturally-widened-glyphs:

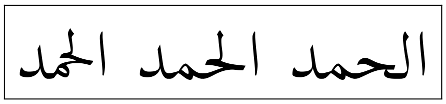
> *Generally, there is only one flat extended character allowed per word. However, two non-consecutive extended characters may be allowed. In that case, the second extended character should be longer than the first.*

For example:

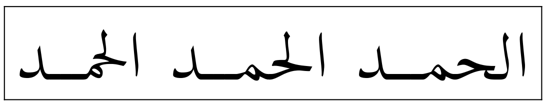


In accordance with our working rule, the top substitution uses only one flat extended character. The bottom uses two, but the second is longer than the first.

In our own estimation, the smaller the type, as in, e.g., footnotes and marginal notes, the less aesthetic variants that are needed. And the less aesthetic variants needed, the better that flat extending will work as a solution. Consider another example of the same word processed in three different variants:

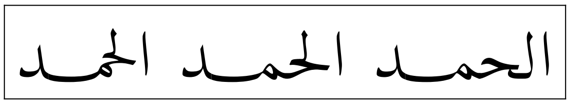


In this case our default is on the left. The variant on the right is about as basic as one can get; the default on the left is a sophisticated aesthetic variant. The middle one is, well, in between. Let's try them with flat extending, using only one extended character per word:



On the left, we have an aesthetic combination of letters followed by a flat *taṭwīl*. This is what Microsoft Word would give us, and the result is aesthetically distasteful. In the word on the right, however, the flat extension fits well with the basic nature of the feature set. As for the middle one, it could go either way and we leave it to the reader to decide what one thinks.

Now let's repeat with more naturally curved widening:



Here, the variant on the left comes out much nicer. The one on the right looks okay with curved widening, although one could arguably do better with flat extending, at least in some contexts. The middle one, again, could go either way, though we think it does somewhat better with curved widening compared to the one on the right. The variant on the left only works well with curved widening.

Towards a ConTEXt solution

In what follows, we will focus on a solution to the problem of paragraph optimization via alternate glyphs (including alternately-shaped and naturally-widened variants). It turns out that the `\xleaders` method used by Omega/Aleph does not work in LuaTEX, so flat extending could not be naively implemented that way. At the moment flat extending is yet to be implemented in ConTEXt.

Since flat extending is so ubiquitous in current Arabic-script typography, and since it does have important applications (poetry and small font sizes where one prefers simpler aesthetic variants), one could ask why this was not implemented first. In part, this is because the immediate priority of the Oriental TEX project has been top-notch, unparalleled aesthetic sophistication of the script. As we noted above, flat extending does not work so well with sophisticated aesthetic variation. So although the flat-extending problem is apparently simpler, it is understandable that we have focused on the more difficult problem first. A clear understanding of the issues and challenges involved with the more general alternate glyph method will help us implement a solution to the the flat-extended problem as a special case. We will come back to this issue towards the end.

Let us now consider the current experimental ConTEXt setup for paragraph optimization for Arabic-script.

Applying Features to Arabic-script

We're now ready for the real thing: Arabic script. The initial setup is not that different from the Latin-script case.

```
\definefontfeature
  [husayni-whatever]
  [goodies=husayni,
   featureset=default]

\definefontsolution
  [FancyHusayni]
  [goodies=husayni,
   solution=experimental]

\definefont
  [FancyHusayni]
  [file:husayni*husayni-whatever at 24pt]
```

But here the definitions in the goodies file look way more complex. Here we have only one shrink set but multiple expansion sets.

```
local yes = "yes"
local basics = {
    analyze  = yes,
    mode     = "node",
    language = "dflt",
    script   = "arab",
}
local analysis = {
    ccmp = yes,
    init = yes, medi = yes, fina = yes,
}
local regular = {
   rlig = yes, calt = yes, salt = yes, anum = yes,
   ss01 = yes, ss03 = yes, ss07 = yes, ss10 = yes, ss12 = yes,
   ss15 = yes, ss16 = yes, ss19 = yes, ss24 = yes, ss25 = yes,
   ss26 = yes, ss27 = yes, ss31 = yes, ss34 = yes, ss35 = yes,
   ss36 = yes, ss37 = yes, ss38 = yes, ss41 = yes, ss42 = yes,
   ss43 = yes, js16 = yes,
}
local positioning = {
   kern = yes, curs = yes, mark = yes, mkmk = yes,
}
local minimal_stretching = {
    js11 = yes, js03 = yes,
}
local medium_stretching = {
    js12=yes, js05=yes,
}
local maximal_stretching= {
    js13 = yes, js05 = yes, js09 = yes,
}
local wide_all = {
    js11 = yes, js12 = yes, js13 = yes, js05 = yes, js09 = yes,
}
local shrink = {
    flts = yes, js17 = yes, ss05 = yes, ss11 = yes, ss06 = yes,
    ss09 = yes,
}
local default = {
    basics, analysis, regular, positioning,
}
```

```
return {
  name = "husayni",
  version = "1.00",
  comment = "Goodies that complement the Husayni font by prof.Hamid.",
  author = "Idris Samawi Hamid and Hans Hagen",
  featuresets = {
    default = {
      default,
    },
    minimal_stretching = {
      default,
      js11 = yes, js03 = yes,
    },
    medium_stretching = {
      default,
      js12=yes, js05=yes,
    },
    maximal_stretching= {
      default,
      js13 = yes, js05 = yes, js09 = yes,
    },
    wide_all = {
      default,
      js11 = yes, js12 = yes, js13 = yes, js05 = yes, js09 = yes,
    },
    shrink = {
      default, flts = yes,
      js17 = yes,
      ss05 = yes, ss11 = yes, ss06 = yes, ss09 = yes,
    },
   },
  solutions = {
    experimental = {
      less = {
        "shrink",
      },
      more = {
        "minimal_stretching", "medium_stretching", "maximal_stretching",
        "wide_all"
      },
    },
  },
  ...
}
```

There are some 55 stylistic and 21 justification features. Not all make sense when optimizing. We predefine some Lua tables to make the sets and solutions easier to understand. The default rendering looks as follows:

```
\FancyHusayni
\righttoleft
\definefontfeature[rasm][script=arab,ss05=yes,js06=no,ss55=yes]
\addff{rasm}
\getbuffer[sample]
\par
```

يَٰٓأَيُّهَا ٱلَّذِينَ ءَامَنُواْ كُلُواْ مِن طَيِّبَٰتِ مَا رَزَقْنَٰكُمْ وَٱشْكُرُواْ لِلَّهِ
إِن كُنتُمْ إِيَّاهُ تَعْبُدُونَ ١٧٢ إِنَّمَا حَرَّمَ عَلَيْكُمُ ٱلْمَيْتَةَ وَٱلدَّمَ وَلَحْمَ
ٱلْخِنزِيرِ وَمَآ أُهِلَّ بِهِۦ لِغَيْرِ ٱللَّهِ ۖ فَمَنِ ٱضْطُرَّ غَيْرَ بَاغٍ وَلَا عَادٍ فَلَآ
إِثْمَ عَلَيْهِ ۚ إِنَّ ٱللَّهَ غَفُورٌ رَّحِيمٌ ١٧٣ إِنَّ ٱلَّذِينَ يَكْتُمُونَ مَآ أَنزَلَ
ٱللَّهُ مِنَ ٱلْكِتَٰبِ وَيَشْتَرُونَ بِهِۦ ثَمَنًا قَلِيلًا ۙ أُوْلَٰٓئِكَ مَا يَأْكُلُونَ فِى
بُطُونِهِمْ إِلَّا ٱلنَّارَ وَلَا يُكَلِّمُهُمُ ٱللَّهُ يَوْمَ ٱلْقِيَٰمَةِ وَلَا يُزَكِّيهِمْ وَلَهُمْ
عَذَابٌ أَلِيمٌ ١٧٤ أُوْلَٰٓئِكَ ٱلَّذِينَ ٱشْتَرَوُاْ ٱلضَّلَٰلَةَ بِٱلْهُدَىٰ وَٱلْعَذَابَ
بِٱلْمَغْفِرَةِ ۚ فَمَآ أَصْبَرَهُمْ عَلَى ٱلنَّارِ ١٧٥ ذَٰلِكَ بِأَنَّ ٱللَّهَ نَزَّلَ ٱلْكِتَٰبَ
بِٱلْحَقِّ ۗ وَإِنَّ ٱلَّذِينَ ٱخْتَلَفُواْ فِى ٱلْكِتَٰبِ لَفِى شِقَاقٍ بَعِيدٍ ١٧٦

Note that we already have a degree of widened substitution in this example. This is all for the accommodation of vowels, and is defined entirely in the OpenType tables of the font. We also added some special orthography (the `rasm` font feature to get the Qur'ānic features just right). You can also do this by adding the feature to the `lfg` file (`local regular =`). There is no paragraph optimization as yet, although the default LuaTEX engine does a good job to start with.

Next we show a more optimized result:

```
\setupfontsolution
  [FancyHusayni]
  [method={preroll,normal},
   criterium=1]

\startfontsolution[FancyHusayni]
  \FancyHusayni
  \righttoleft
\definefontfeature[rasm][script=arab,ss05=yes,js06=no,ss55=yes]
\addff{rasm}
  \getbuffer[sample]
  \par
\stopfontsolution
```

يَـٰٓأَيُّهَا ٱلَّذِينَ ءَامَنُوا۟ كُلُوا۟ مِن طَيِّبَـٰتِ مَا رَزَقْنَـٰكُمْ وَٱشْكُرُوا۟ لِلَّهِ
إِن كُنتُمْ إِيَّاهُ تَعْبُدُونَ ١٧٢ إِنَّمَا حَرَّمَ عَلَيْكُمُ ٱلْمَيْتَةَ وَٱلدَّمَ وَلَحْمَ
ٱلْخِنزِيرِ وَمَآ أُهِلَّ بِهِۦ لِغَيْرِ ٱللَّهِ ۖ فَمَنِ ٱضْطُرَّ غَيْرَ بَاغٍ وَلَا عَادٍ فَلَآ
إِثْمَ عَلَيْهِ ۚ إِنَّ ٱللَّهَ غَفُورٌ رَّحِيمٌ ١٧٣ إِنَّ ٱلَّذِينَ يَكْتُمُونَ مَآ أَنزَلَ
ٱللَّهُ مِنَ ٱلْكِتَـٰبِ وَيَشْتَرُونَ بِهِۦ ثَمَنًا قَلِيلًا ۙ أُو۟لَـٰٓئِكَ مَا يَأْكُلُونَ فِى
بُطُونِهِمْ إِلَّا ٱلنَّارَ وَلَا يُكَلِّمُهُمُ ٱللَّهُ يَوْمَ ٱلْقِيَـٰمَةِ وَلَا يُزَكِّيهِمْ وَلَهُمْ
عَذَابٌ أَلِيمٌ ١٧٤ أُو۟لَـٰٓئِكَ ٱلَّذِينَ ٱشْتَرَوُا۟ ٱلضَّلَـٰلَةَ بِٱلْهُدَىٰ وَٱلْعَذَابَ
بِٱلْمَغْفِرَةِ ۚ فَمَآ أَصْبَرَهُمْ عَلَى ٱلنَّارِ ١٧٥ ذَٰلِكَ بِأَنَّ ٱللَّهَ نَزَّلَ ٱلْكِتَـٰبَ
بِٱلْحَقِّ ۗ وَإِنَّ ٱلَّذِينَ ٱخْتَلَفُوا۟ فِى ٱلْكِتَـٰبِ لَفِى شِقَاقٍۭ بَعِيدٍ ١٧٦

Now let's see what happens when `\parfillskip= 0pt`, i.e., the last line has no extra space after the end of the paragraph. This is important for getting, e.g., the last line of the page to end with the end of a verse as we discussed earlier:

```
\setupfontsolution
  [FancyHusayni]
  [method={preroll,normal},
   criterium=1]

\startfontsolution[FancyHusayni]
  \FancyHusayni
  \righttoleft
\definefontfeature[rasm][script=arab,ss05=yes,js06=no,ss55=yes]
\addff{rasm}
  \parfillskip=0pt
  \getbuffer[sample]
  \par
\stopfontsolution
```

يَـٰٓأَيُّهَا ٱلَّذِينَ ءَامَنُوا۟ كُلُوا۟ مِن طَيِّبَـٰتِ مَا رَزَقْنَـٰكُمْ وَٱشْكُرُوا۟ لِلَّهِ
إِن كُنتُمْ إِيَّاهُ تَعْبُدُونَ ١٧٢ إِنَّمَا حَرَّمَ عَلَيْكُمُ ٱلْمَيْتَةَ وَٱلدَّمَ وَلَحْمَ
ٱلْخِنزِيرِ وَمَآ أُهِلَّ بِهِۦ لِغَيْرِ ٱللَّهِ ۖ فَمَنِ ٱضْطُرَّ غَيْرَ بَاغٍ وَلَا عَادٍ فَلَآ
إِثْمَ عَلَيْهِ ۚ إِنَّ ٱللَّهَ غَفُورٌ رَّحِيمٌ ١٧٣ إِنَّ ٱلَّذِينَ يَكْتُمُونَ مَآ أَنزَلَ
ٱللَّهُ مِنَ ٱلْكِتَـٰبِ وَيَشْتَرُونَ بِهِۦ ثَمَنًا قَلِيلًا ۙ أُو۟لَـٰٓئِكَ مَا يَأْكُلُونَ فِى
بُطُونِهِمْ إِلَّا ٱلنَّارَ وَلَا يُكَلِّمُهُمُ ٱللَّهُ يَوْمَ ٱلْقِيَـٰمَةِ وَلَا يُزَكِّيهِمْ وَلَهُمْ
عَذَابٌ أَلِيمٌ ١٧٤ أُو۟لَـٰٓئِكَ ٱلَّذِينَ ٱشْتَرَوُا۟ ٱلضَّلَـٰلَةَ بِٱلْهُدَىٰ وَٱلْعَذَابَ
بِٱلْمَغْفِرَةِ ۚ فَمَآ أَصْبَرَهُمْ عَلَى ٱلنَّارِ ١٧٥ ذَٰلِكَ بِأَنَّ ٱللَّهَ نَزَّلَ ٱلْكِتَـٰبَ
بِٱلْحَقِّ ۗ وَإِنَّ ٱلَّذِينَ ٱخْتَلَفُوا۟ فِى ٱلْكِتَـٰبِ لَفِى شِقَاقٍ بَعِيدٍ ۘ ١٧٦

Just as the effects are more visible in the `\parfillskip= 0pt` case, the impact is much larger when the available width is less. In figures 11, 12, 13, 14 and 15 we can see the optimizer in action when that happens.

In our estimation, the current experimental solution works best for alternate-shaped glyphs, although there is some success with naturally widened characters. Clearly, some widened substitutions work better than others. A lot of fine tuning is needed, both within the OpenType features as well as the optimization algorithm.

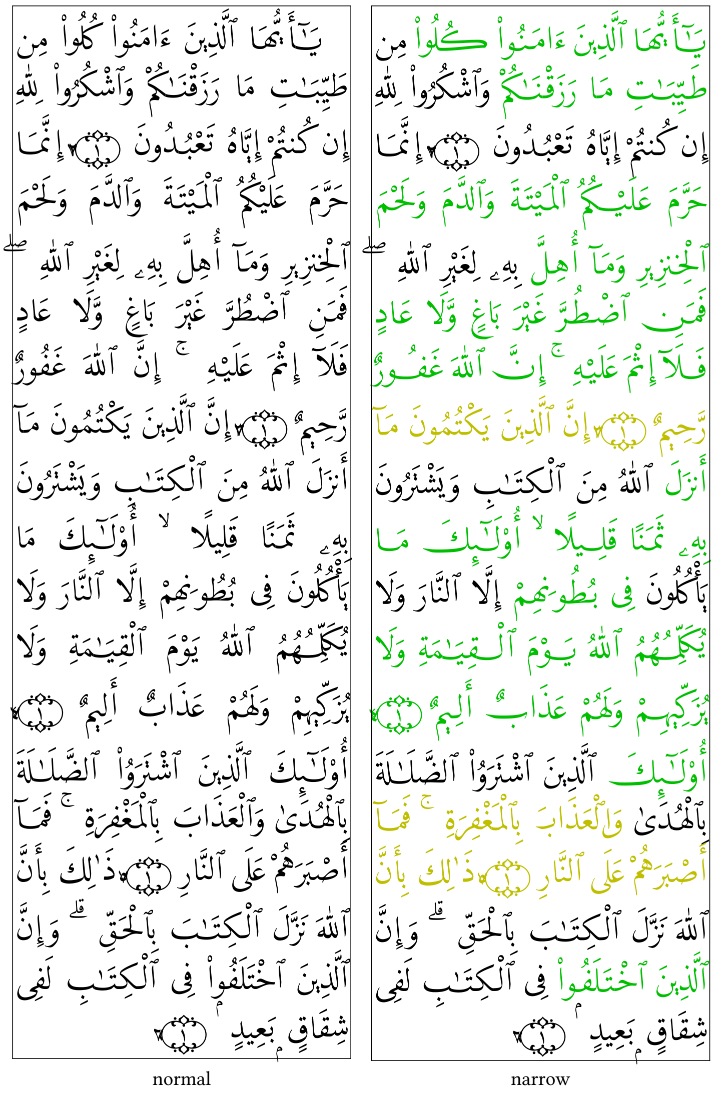
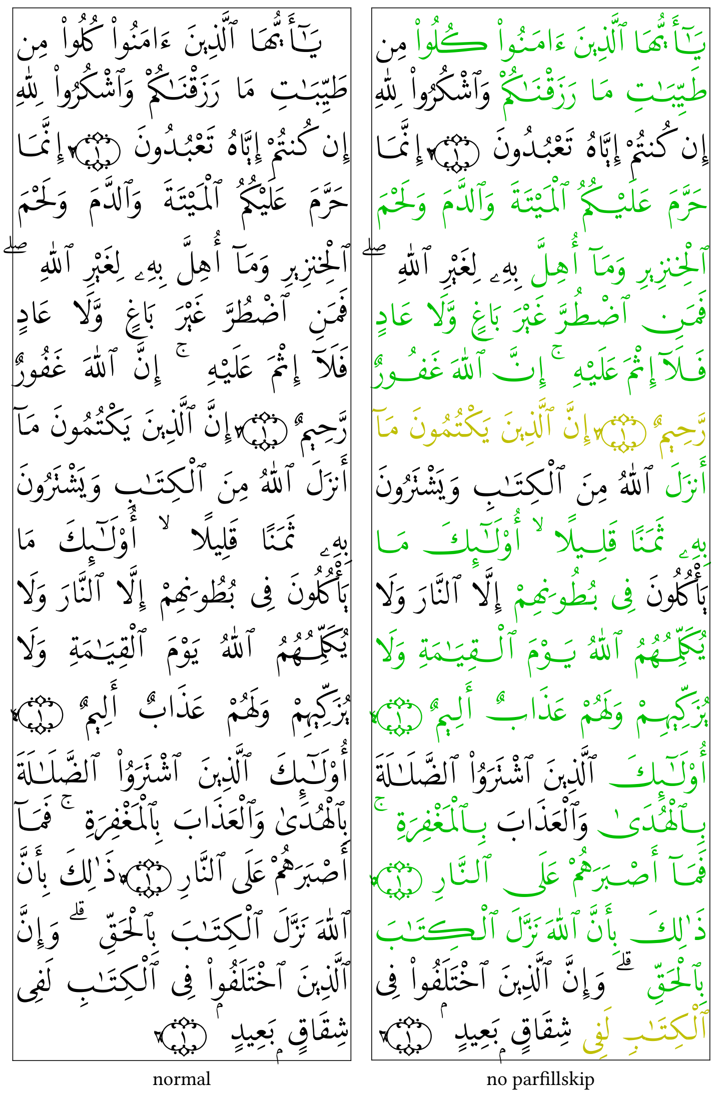


Figure 11. A narrower sample (a).



يَٰٓأَيُّهَا ٱلَّذِينَ ءَامَنُوا۟ كُلُوا۟ مِن
طَيِّبَٰتِ مَا رَزَقْنَٰكُمْ وَٱشْكُرُوا۟ لِلَّهِ
إِن كُنتُمْ إِيَّاهُ تَعْبُدُونَ ﴿١٧٢﴾ إِنَّمَا
حَرَّمَ عَلَيْكُمُ ٱلْمَيْتَةَ وَٱلدَّمَ وَلَحْمَ
ٱلْخِنزِيرِ وَمَآ أُهِلَّ بِهِۦ لِغَيْرِ ٱللَّهِ ۖ
فَمَنِ ٱضْطُرَّ غَيْرَ بَاغٍ وَلَا عَادٍ
فَلَآ إِثْمَ عَلَيْهِ ۚ إِنَّ ٱللَّهَ غَفُورٌ
رَّحِيمٌ ﴿١٧٣﴾ إِنَّ ٱلَّذِينَ يَكْتُمُونَ مَآ
أَنزَلَ ٱللَّهُ مِنَ ٱلْكِتَٰبِ وَيَشْتَرُونَ
بِهِۦ ثَمَنًا قَلِيلًا ۙ أُو۟لَٰٓئِكَ مَا
يَأْكُلُونَ فِى بُطُونِهِمْ إِلَّا ٱلنَّارَ وَلَا
يُكَلِّمُهُمُ ٱللَّهُ يَوْمَ ٱلْقِيَٰمَةِ وَلَا
يُزَكِّيهِمْ وَلَهُمْ عَذَابٌ أَلِيمٌ ﴿١٧٤﴾
أُو۟لَٰٓئِكَ ٱلَّذِينَ ٱشْتَرَوُا۟ ٱلضَّلَٰلَةَ
بِٱلْهُدَىٰ وَٱلْعَذَابَ بِٱلْمَغْفِرَةِ ۚ فَمَآ
أَصْبَرَهُمْ عَلَى ٱلنَّارِ ﴿١٧٥﴾ ذَٰلِكَ بِأَنَّ
ٱللَّهَ نَزَّلَ ٱلْكِتَٰبَ بِٱلْحَقِّ ۗ وَإِنَّ
ٱلَّذِينَ ٱخْتَلَفُوا۟ فِى ٱلْكِتَٰبِ لَفِى
شِقَاقٍۭ بَعِيدٍ ﴿١٧٦﴾

يَٰٓأَيُّهَا ٱلَّذِينَ ءَامَنُوا۟ كُلُوا۟ مِن
طَيِّبَٰتِ مَا رَزَقْنَٰكُمْ وَٱشْكُرُوا۟ لِلَّهِ
إِن كُنتُمْ إِيَّاهُ تَعْبُدُونَ ﴿١٧٢﴾ إِنَّمَا
حَرَّمَ عَلَيْكُمُ ٱلْمَيْتَةَ وَٱلدَّمَ وَلَحْمَ
ٱلْخِنزِيرِ وَمَآ أُهِلَّ بِهِۦ لِغَيْرِ ٱللَّهِ ۖ
فَمَنِ ٱضْطُرَّ غَيْرَ بَاغٍ وَلَا عَادٍ
فَلَآ إِثْمَ عَلَيْهِ ۚ إِنَّ ٱللَّهَ غَفُورٌ
رَّحِيمٌ ﴿١٧٣﴾ إِنَّ ٱلَّذِينَ يَكْتُمُونَ مَآ
أَنزَلَ ٱللَّهُ مِنَ ٱلْكِتَٰبِ وَيَشْتَرُونَ
بِهِۦ ثَمَنًا قَلِيلًا ۙ أُو۟لَٰٓئِكَ مَا
يَأْكُلُونَ فِى بُطُونِهِمْ إِلَّا ٱلنَّارَ وَلَا
يُكَلِّمُهُمُ ٱللَّهُ يَوْمَ ٱلْقِيَٰمَةِ وَلَا
يُزَكِّيهِمْ وَلَهُمْ عَذَابٌ أَلِيمٌ ﴿١٧٤﴾
أُو۟لَٰٓئِكَ ٱلَّذِينَ ٱشْتَرَوُا۟ ٱلضَّلَٰلَةَ
بِٱلْهُدَىٰ وَٱلْعَذَابَ بِٱلْمَغْفِرَةِ ۚ
فَمَآ أَصْبَرَهُمْ عَلَى ٱلنَّارِ ﴿١٧٥﴾
ذَٰلِكَ بِأَنَّ ٱللَّهَ نَزَّلَ ٱلْكِتَٰبَ
بِٱلْحَقِّ ۗ وَإِنَّ ٱلَّذِينَ ٱخْتَلَفُوا۟ فِى
ٱلْكِتَٰبِ لَفِى شِقَاقٍۭ بَعِيدٍ ﴿١٧٦﴾

Figure 12. A narrower sample with no parfillskip (b).

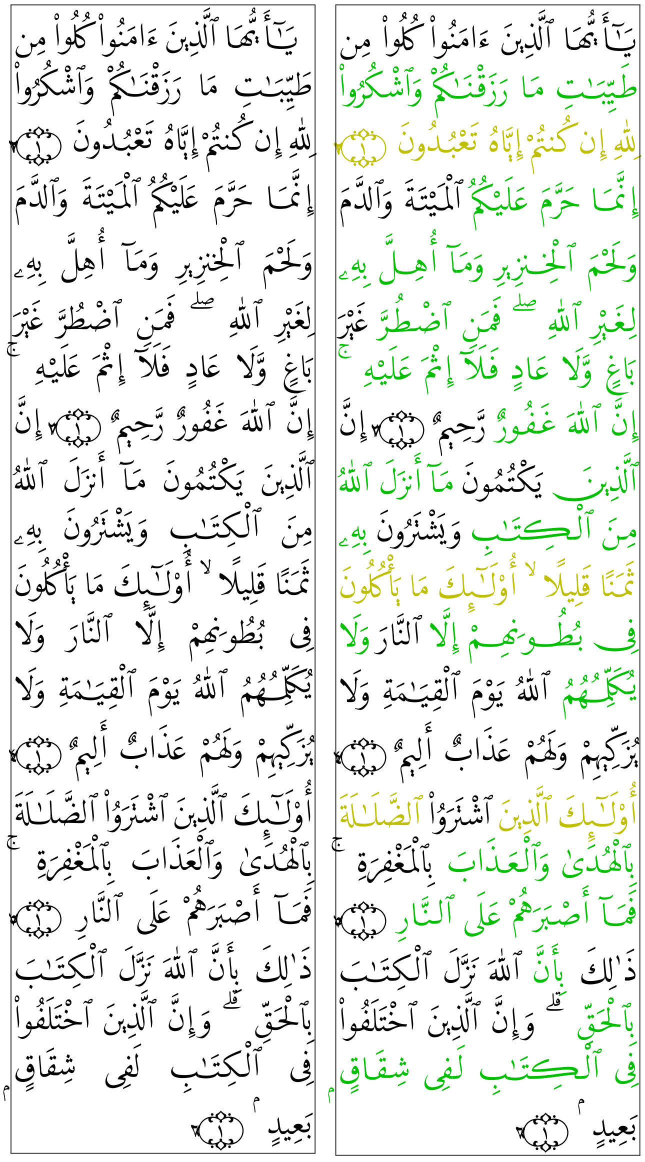
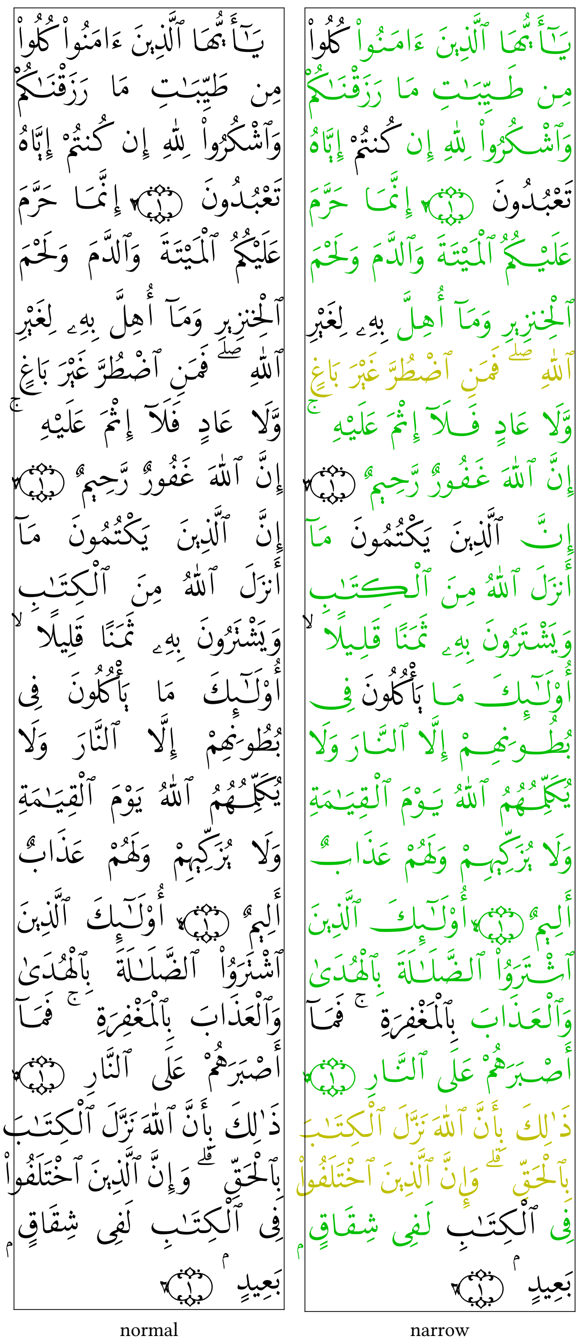


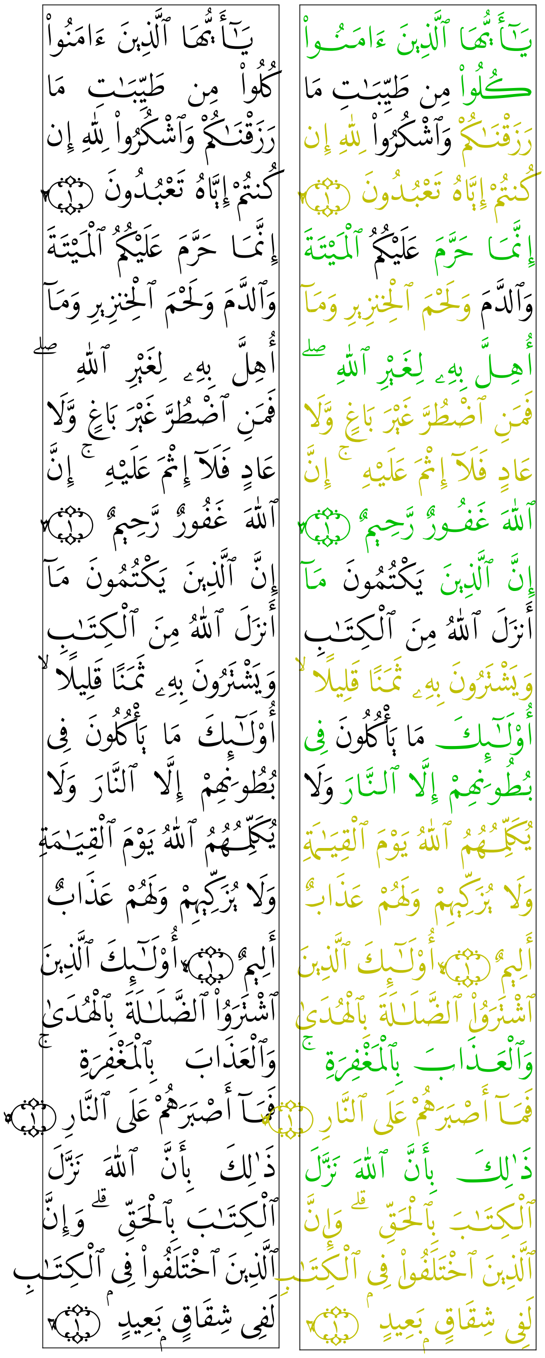
Figure 13. An even narrower sample (c).



يَٰٓأَيُّهَا ٱلَّذِينَ ءَامَنُوا۟ كُلُوا۟
مِن طَيِّبَٰتِ مَا رَزَقْنَٰكُمْ
وَٱشْكُرُوا۟ لِلَّهِ إِن كُنتُمْ إِيَّاهُ
تَعْبُدُونَ ١٧٢ إِنَّمَا حَرَّمَ
عَلَيْكُمُ ٱلْمَيْتَةَ وَٱلدَّمَ وَلَحْمَ
ٱلْخِنزِيرِ وَمَآ أُهِلَّ بِهِۦ لِغَيْرِ
ٱللَّهِ ۖ فَمَنِ ٱضْطُرَّ غَيْرَ بَاغٍ
وَلَا عَادٍ فَلَآ إِثْمَ عَلَيْهِ ۚ
إِنَّ ٱللَّهَ غَفُورٌ رَّحِيمٌ ١٧٣
إِنَّ ٱلَّذِينَ يَكْتُمُونَ مَآ
أَنزَلَ ٱللَّهُ مِنَ ٱلْكِتَٰبِ
وَيَشْتَرُونَ بِهِۦ ثَمَنًا قَلِيلًا ۙ
أُو۟لَٰٓئِكَ مَا يَأْكُلُونَ فِى
بُطُونِهِمْ إِلَّا ٱلنَّارَ وَلَا
يُكَلِّمُهُمُ ٱللَّهُ يَوْمَ ٱلْقِيَٰمَةِ
وَلَا يُزَكِّيهِمْ وَلَهُمْ عَذَابٌ
أَلِيمٌ ١٧٤ أُو۟لَٰٓئِكَ ٱلَّذِينَ
ٱشْتَرَوُا۟ ٱلضَّلَٰلَةَ بِٱلْهُدَىٰ
وَٱلْعَذَابَ بِٱلْمَغْفِرَةِ ۚ فَمَآ
أَصْبَرَهُمْ عَلَى ٱلنَّارِ ١٧٥
ذَٰلِكَ بِأَنَّ ٱللَّهَ نَزَّلَ ٱلْكِتَٰبَ
بِٱلْحَقِّ ۗ وَإِنَّ ٱلَّذِينَ ٱخْتَلَفُوا۟
فِى ٱلْكِتَٰبِ لَفِى شِقَاقٍۭ
بَعِيدٍ ١٧٦

يَٰٓأَيُّهَا ٱلَّذِينَ ءَامَنُوا۟ كُلُوا۟
مِن طَيِّبَٰتِ مَا رَزَقْنَٰكُمْ
وَٱشْكُرُوا۟ لِلَّهِ إِن كُنتُمْ إِيَّاهُ
تَعْبُدُونَ ١٧٢ إِنَّمَا حَرَّمَ
عَلَيْكُمُ ٱلْمَيْتَةَ وَٱلدَّمَ وَلَحْمَ
ٱلْخِنزِيرِ وَمَآ أُهِلَّ بِهِۦ لِغَيْرِ
ٱللَّهِ ۖ فَمَنِ ٱضْطُرَّ غَيْرَ بَاغٍ
وَلَا عَادٍ فَلَآ إِثْمَ عَلَيْهِ ۚ
إِنَّ ٱللَّهَ غَفُورٌ رَّحِيمٌ ١٧٣
إِنَّ ٱلَّذِينَ يَكْتُمُونَ مَآ
أَنزَلَ ٱللَّهُ مِنَ ٱلْكِتَٰبِ
وَيَشْتَرُونَ بِهِۦ ثَمَنًا قَلِيلًا ۙ
أُو۟لَٰٓئِكَ مَا يَأْكُلُونَ فِى
بُطُونِهِمْ إِلَّا ٱلنَّارَ وَلَا
يُكَلِّمُهُمُ ٱللَّهُ يَوْمَ ٱلْقِيَٰمَةِ
وَلَا يُزَكِّيهِمْ وَلَهُمْ عَذَابٌ
أَلِيمٌ ١٧٤ أُو۟لَٰٓئِكَ ٱلَّذِينَ
ٱشْتَرَوُا۟ ٱلضَّلَٰلَةَ بِٱلْهُدَىٰ
وَٱلْعَذَابَ بِٱلْمَغْفِرَةِ ۚ فَمَآ
أَصْبَرَهُمْ عَلَى ٱلنَّارِ ١٧٥
ذَٰلِكَ بِأَنَّ ٱللَّهَ نَزَّلَ ٱلْكِتَٰبَ
بِٱلْحَقِّ ۗ وَإِنَّ ٱلَّذِينَ ٱخْتَلَفُوا۟
فِى ٱلْكِتَٰبِ لَفِى شِقَاقٍۭ
بَعِيدٍ ١٧٦

Figure 14. An even narrower sample (d).



يَٰٓأَيُّهَا ٱلَّذِينَ ءَامَنُوا۟
كُلُوا۟ مِن طَيِّبَٰتِ مَا
رَزَقْنَٰكُمْ وَٱشْكُرُوا۟ لِلَّهِ إِن
كُنتُمْ إِيَّاهُ تَعْبُدُونَ ١٧٢
إِنَّمَا حَرَّمَ عَلَيْكُمُ ٱلْمَيْتَةَ
وَٱلدَّمَ وَلَحْمَ ٱلْخِنزِيرِ وَمَآ
أُهِلَّ بِهِۦ لِغَيْرِ ٱللَّهِ ۖ
فَمَنِ ٱضْطُرَّ غَيْرَ بَاغٍ وَلَا
عَادٍ فَلَآ إِثْمَ عَلَيْهِ ۚ إِنَّ
ٱللَّهَ غَفُورٌ رَّحِيمٌ ١٧٣
إِنَّ ٱلَّذِينَ يَكْتُمُونَ مَآ
أَنزَلَ ٱللَّهُ مِنَ ٱلْكِتَٰبِ
وَيَشْتَرُونَ بِهِۦ ثَمَنًا قَلِيلًا ۙ
أُو۟لَٰٓئِكَ مَا يَأْكُلُونَ فِى
بُطُونِهِمْ إِلَّا ٱلنَّارَ وَلَا
يُكَلِّمُهُمُ ٱللَّهُ يَوْمَ ٱلْقِيَٰمَةِ
وَلَا يُزَكِّيهِمْ وَلَهُمْ عَذَابٌ
أَلِيمٌ ١٧٤ أُو۟لَٰٓئِكَ ٱلَّذِينَ
ٱشْتَرَوُا۟ ٱلضَّلَٰلَةَ بِٱلْهُدَىٰ
وَٱلْعَذَابَ بِٱلْمَغْفِرَةِ ۚ
فَمَآ أَصْبَرَهُمْ عَلَى ٱلنَّارِ ١٧٥
ذَٰلِكَ بِأَنَّ ٱللَّهَ نَزَّلَ
ٱلْكِتَٰبَ بِٱلْحَقِّ ۗ وَإِنَّ
ٱلَّذِينَ ٱخْتَلَفُوا۟ فِى ٱلْكِتَٰبِ
لَفِى شِقَاقٍ بَعِيدٍ ١٧٦

normal

يَٰٓأَيُّهَا ٱلَّذِينَ ءَامَنُوا۟
كُلُوا۟ مِن طَيِّبَٰتِ مَا
رَزَقْنَٰكُمْ وَٱشْكُرُوا۟ لِلَّهِ إِن
كُنتُمْ إِيَّاهُ تَعْبُدُونَ ١٧٢
إِنَّمَا حَرَّمَ عَلَيْكُمُ ٱلْمَيْتَةَ
وَٱلدَّمَ وَلَحْمَ ٱلْخِنزِيرِ وَمَآ
أُهِلَّ بِهِۦ لِغَيْرِ ٱللَّهِ ۖ
فَمَنِ ٱضْطُرَّ غَيْرَ بَاغٍ وَلَا
عَادٍ فَلَآ إِثْمَ عَلَيْهِ ۚ إِنَّ
ٱللَّهَ غَفُورٌ رَّحِيمٌ ١٧٣
إِنَّ ٱلَّذِينَ يَكْتُمُونَ مَآ
أَنزَلَ ٱللَّهُ مِنَ ٱلْكِتَٰبِ
وَيَشْتَرُونَ بِهِۦ ثَمَنًا قَلِيلًا ۙ
أُو۟لَٰٓئِكَ مَا يَأْكُلُونَ فِى
بُطُونِهِمْ إِلَّا ٱلنَّارَ وَلَا
يُكَلِّمُهُمُ ٱللَّهُ يَوْمَ ٱلْقِيَٰمَةِ
وَلَا يُزَكِّيهِمْ وَلَهُمْ عَذَابٌ
أَلِيمٌ ١٧٤ أُو۟لَٰٓئِكَ ٱلَّذِينَ
ٱشْتَرَوُا۟ ٱلضَّلَٰلَةَ بِٱلْهُدَىٰ
وَٱلْعَذَابَ بِٱلْمَغْفِرَةِ ۚ
فَمَآ أَصْبَرَهُمْ عَلَى ٱلنَّارِ ١٧٥
ذَٰلِكَ بِأَنَّ ٱللَّهَ نَزَّلَ
ٱلْكِتَٰبَ بِٱلْحَقِّ ۗ وَإِنَّ
ٱلَّذِينَ ٱخْتَلَفُوا۟ فِى ٱلْكِتَٰبِ
لَفِى شِقَاقٍ بَعِيدٍ ١٧٦

narrow

Figure 15. An even narrower sample (e).

Without going into a detailed analysis at the moment, we restrict ourselves to two critical observations.

First, in our tests one will notice that the glyph substitutions tend to take place on the right side of the line. They should be more evenly distributed throughout each line.

Second, we can say that the current method works better for alternate-shaped glyph substitution than it does for naturally-widened glyph substitution. This leads us to the next step in this research project:

Within the Husayni font there is now a mapping between flat extending via *taṭwīl* and curved widening via alternate glyphs. Consider the following manually typed utf text using the *taṭwīl* character (`U+0640`):

```
فعيل \ARROW\ فعيـــل
```

In flat-extended typography that comes out like this:

فعيل ← فعيـــل

Husayni, through the optional Stylistic Alternates feature (`salt`) will map the flat *taṭwīl*-extended characters to curved widened characters. So with `salt=yes` selected in ConTEXt we get

فعيل ← فعيـــل

This opens up a way to connect a forthcoming solution to the flat *taṭwīl*-extended character method with the curved widened-glyph method. A future version of the optimizer may be able to optimize the paragraph in terms of the *taṭwīl* character and a set of rules along the lines we discussed earlier. Then we can simply convert the result to curves using the *taṭwīl* character. At least this is one possibility.

In any case, the current paragraph optimizer, even in its experimental status at the moment, represents one of the greatest and most important steps in the evolution of digital Arabic-script typography. Its potential impact on for Arabic-script typesetting is immense, and we excitedly look forward to its completion.

Notes

1. Sometimes hz-optimization also goes under the rubric of 'Semitic justification'. See, e.g., Bringhurst in pre-3rd editions of his *Elements of Typographic Style*. This technique does not work well for Arabic script in general because glyphs are connected in two dimensions. On the other hand, a certain basic yet ubiquitous Semitic justification *can* be achieved by using the *taṭwīl* character, commonly called the *kashīdah* (`U+0640`). We will discuss this later in this article.
2. Much of this is handled within the `GPOS` features of the OpenType font itself (e.g., `mark` and `mkmk`)
3. This number reflects the maximum badness and future versions might have a different measure with more granularity.
4. Indeed, even Latin hyphenation, when it occurs, can be considered a 'failure' of sorts.
5. This five character string can be represented in Latin by the five character string '*al-ḥmd*' (not including the '-'). It is pronounced '*al-ḥamdu*'. Note that Arabic script is mainly consonantal: pure vowels are not part of the alphabet and are, instead, represented by diacritics.
6. Indeed, as was the case with Latin typography, Arabic-script typography took a sharp turn for the worse with the advent of digital typography. On the other hand, Latin typography recovered much more quickly, in large part thanks to Knuth's development of TEX.

Hans Hagen & Idris Samawi Hamid