

The Arabi system ﴿العربي﴾ نظام — T_EX writes in Arabic and Farsi

Youssef Jabri

École Nationale des Sciences Appliquées,

Oujda, Morocco

yjabri (at) ensa dot univ-oujda dot ac dot ma

Abstract

In this paper, we will present a newly arrived package on CTAN that provides Arabic script support for T_EX without the need for an external pre-processor. The Arabi package adds one of the last major multilingual typesetting capabilities to Babel by adding support for the *Arabic* عربي and *Farsi* فارسي languages. Other languages using the Arabic script should also be more or less easily implementable.

Arabi comes with many good quality free fonts, Arabic and Farsi, and may also use commercial fonts. It supports many 8-bit input encodings (namely, CP-1256, ISO-8859-6 and Unicode UTF-8) and can typeset *classical Arabic poetry*.

The package is distributed under the L^AT_EX Project Public License (LPPL), and has the LPPL maintenance status “author-maintained”. It can be used *freely* (including commercially) to produce beautiful texts that mix Arabic, Farsi and Latin (or other) characters.

ملخص

رزمة العربي نظام يتيح إمكانية استعمال الحروف العربية واللاتينية جنباً إلى جنب في مستند واحد باستعمال نظام « تيخ » T_EX لتصنيف الحروف. رزمة العربي تضيف إمكانية استعمال اللغتين (عربي و فارسي) مع نظام تيخ ومنذ البداية ، فهذا النظام يتميز بكونه محمولاً ويتمتع بقدر كبير من المرونة ، لأنه قابل للاستعمال مع معظم ما تم إنجازه من إضافات . إضافة إلى أنه لا يحتاج إلى أي معالج خارجي لتحديد أشكال الحروف في الكلمة . يقدم العربي حالياً مصحوباً بمجموعة خطوط حرة الاستعمال كما يمكنه استعمال عدد من الخطوط التي تأتي مع نظام ويندوز مثلاً . كما هو الحال بالنسبة لنظام تيخ ، فإن العربي مجاني ولا يكلف مستعمله إلا عناء الاستعمال .

1 Introduction

The development of Arabi¹ was a response to the absence of a package that manipulates the Arabic script and fulfills the following requirements:

1. L^AT_EX 2_ε and Babel compliant, this combination format/package being the most widely used in our opinion when mixing different languages.
2. The possibility of using 8-bit input text including already existing Arabic texts, on different systems.
3. Able to use existing, commercial and free, beautiful Arabic fonts.

¹ The name of the package should not be misunderstood. It is not designed to support only the Arabic language, but all languages that use the *Arabic script*. Technically speaking, for Babel, they will all be considered as *dialects* of Arabic.

4. Free (as in freedom), meaning a license like the GNU GPL or LPPL.

Arabi comes with an extensive user manual; this article gives a general overview of the system.

2 Typesetting Arabic with T_EX: the existing possibilities

T_EX and the Arabic script have a long history.

One might imagine that enabling T_EX to write in both directions Right-to-Left (R2L) and Left-to-Right (L2R) with an Arabic font suffices to typeset Arabic with T_EX.

Unfortunately, although such an extended T_EX may perhaps be used to typeset a R2L language like Hebrew, this is far from sufficient for a complex script like Arabic, where the shapes of the glyphs

depend on the context, and may take many forms (at least four forms for the majority of Arabic characters even in the simplest² cases).

Many early attempts have been made; they all relied on a preprocessor that does the contextual analysis (also known as the *shaping algorithm*).

One attempt, not widely known, due to Terry Regier from the University of California, Berkeley, dating from December 1990, relied on the famous macros of D. Knuth and P. MacKay:

```
%The lines below are from Knuth and MacKay
%   TUGboat vol.8, #1, page 14.
\font\revrm=xbmc10   \hyphenchar\revrm=-1
\catcode'\|= \active
\def|#1|{\{\revrm \reflect#1\empty\tceller}}
\def\reflect#1#2\tceller{\ifx#1\empty\else%
\reflect#2\tceller#1\fi}
```

to do the reflection, after a preprocessor has done a rough contextual analysis.

The pioneering work by Knuth and MacKay [11], who implemented the T_EX bidirectional algorithm (which is unrelated to the Unicode Bidirectional Algorithm; the latter implicitly chooses the directions of the text) and added to T_EX the four primitives (`\beginL`, `\endL`, `\beginR` and `\endR`) made things much better!

Some early attempts were also carried out by Y. Haralambous, who used the new extended engine T_EX--X_ET. This includes the non-free A1-Amal³ (1992, [6]), and the free ArabiT_EX⁴ (April 1995).

The most widely used system at present is probably K. Lagally's ArabT_EX [13]. It is a package for writing Arabic in several languages using the Arabic script. It consists of a T_EX macro package and *one* Arabic Naskhi-like font. ArabT_EX will run with Plain T_EX and L^AT_EX; and work with any T_EX engine, because it uses *its own* bidirectional algorithm. So, no preprocessor is needed! This makes it a little slow but with today's computer power, this is not really a problem. Its real drawback lies in the fact that the macros apparently *depend heavily* on the glyphs of the font it uses, making it quite impossible to use any other fonts that may be available to the user.

For courageous users, there also exist two more powerful systems

- Ω by Y. Haralambous and J. Plaice, and
- X_qT_EX by J. Kew, if you have the right system and the right fonts.

² Through typographical simplifications. Some aspects of traditional Arabic typography are described in [5].

³ We did not review it, as it was not available to the public as far as we know.

⁴ The source and a DOS executable of the preprocessor were available through the French TUG.

3 Arabic script specifics

The Arabic script is one of the most widely used scripts on earth. It dominates in Arabic countries, of course, but has a special place for all Muslims because it's the script used to write the Koran, the holy book of Muslims.

The Arabic script, like all other Semitic languages, is written from *Right-to-Left*.

Another important aspect of the Arabic script is that *no hyphenation* is needed, or allowed at all. So, *no hyphenation patterns* are needed for any languages that uses the Arabic script. In very old Arabic documents, words could be split after a non-connecting character, while characters that connect were never split. In modern Arabic, hyphenation is forbidden completely. This makes it more difficult to get justification when long words occur at the end of a line, but Arabic is also cursive and has (in modern fonts mimicking the handwritten forms) a special character called *kashida* or *tatweel* (*keshideh* is a Farsi word that means *stretch*) that may be used between adjoining characters to make the word become longer. An example is the following word:

مثال that may be written to occupy longer مثال and longer مثال and much more longer space مثال.

3.1 The Arabic alphabet

The Arabic alphabet is caseless, but most letters have either two or four forms. The different forms are used according to the letter's position in the word (*initial*, *medial*, *final* and *isolated*). The alphabet is constituted in its basic form by

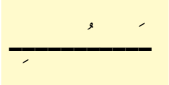

- 28 *consonants* (29 if we count the *hamza*). But the number of 28 characters can exceed easily 1000 glyphs per font if all ligatures are present!

Isolated	Initial	Medial	Final
ب	بـ	بـ	بـ
ج	جـ	جـ	جـ
ع	عـ	عـ	عـ
هـ	هـ	هـ	هـ

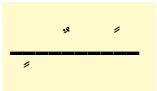
Table 1: Some characters' contextual forms

- Seven diacritical marks specifying the *vowels*. They are not used in typical Arabic texts but appear in poetry, textbooks for people learning the Arabic language, and some religious texts. They can be typed and then at the moment of compiling the document, can be either included or omitted according to the author's wish! The

three basic ones are called *fatha*, *damma* and

kasra: ; the *sukun*  is used

for the absence of vowels; and there are three *tanwin* forms written by doubling the three basic ones:

.

The vowel marks are written somewhat like accents in the Latin script. Above, the drawn line represents the baseline, with the vowels that appear above the line being typeset above letters, while those below the line are typeset below letters.

3.2 Arabic typography

This aspect of Arabic merits much investigation and so much can be said about it. But in order not to be too lengthy, we will just cite three points.

In the classical Arabic literature, there are no typographical styles like bold, italic, etc. Different classical typefaces are used instead (req'a, naskhi, thuluth, etc.) to distinguish between different logical parts of the text. In modern literature, that depends heavily on computers made by people who are either unaware of the rules of Arabic typography or do not have enough time or money to develop such possibilities, we use more and more boldface and italics (slanted to the wrong side many times, unfortunately).

Concerning spacing and punctuation, there is a lot of change between books published early in this century by mechanical means and some more recent ones typeset using computer programs. It seems that different editors adopted different rules. Some use English or French rules, while others insert space before and after each sign — which was the rule in the older texts!

In general, in Arabic texts, enumerated lists use the *abjad* system using letters, in a particular order, instead of numbers, but numbered lists are used also.

4 The Arabi system

The two main problems faced when typesetting Arabic with T_EX are managed by Arabi as follows.

1. The *bi-directional capability* supposes that the user has a T_EX engine providing the four primitives `\beginR`, `\endR`, `\beginL` and `\endL`. This is the case with the T_EX--X_EL and ϵ -T_EX engines.
2. The contextual analysis does not need/use any pre-processor; this is done completely in the

fonts, using the (quite limited) ligature possibilities of METAFONT.

This second point is the whole secret of Arabi's compatibility with most available packages. We tried to shorten T_EX coding to deal with the specifics of the Arabic script as much as we could, to avoid eventual conflicts and clashes with other code.

The system is also *compatible* with *all other formats*, such as plain or ConT_EXt. This too is because the whole contextual analysis is done in the fonts!

4.1 Input and font encodings

Typesetting Arabic and Farsi texts with T_EX implies the use of special *input* and *output encodings*, so we need to use the standard packages `inputenc` and `fontenc`.

We use two special font encodings. For Arabic, we use LAE for *Local Arabic Encoding*, while for Farsi we use LFE that stands for *Local Farsi Encoding*. These two encoding *are not final*. Some character positions may change, and some empty slots will be filled with new characters.

Concerning the input encoding, the user simply creates an ordinary L^AT_EX file, in which he can use 8-bit Arabic characters, typed visually on some system that supports the Arabic script.

For now, the Arabi system supports the following input code pages:

1. Arabic Windows CP-1256 for Arabic and Farsi.
2. ISO-8859-6 for Arabic, not suitable for Farsi because many Farsi characters are missing.
3. The multibyte Unicode UTF-8 (ISO-10646) for Arabic and Farsi.

4.2 What has been done so far?

Currently, with Arabi you can typeset correctly, while mixing the Arabic and Latin scripts, according to the context:

- Footnotes, appearing on the right side of the page.
- Lists, both itemized and enumerated. The standard `enumerate` environment uses the *abjad* system mentioned earlier.
- Floats are typeset with the right caption form and the appropriate entry is added to the table of contents.

Moreover, Arabi takes care of the bidirectional formatting of sectioning, chapters, (sub-)sections, etc., according to the context. And the tables (of contents, figures and tables) are typeset all according to one global text direction, which is the main text direction as specified by the user. This is meant to

```

\documentclass{article}
\usepackage[cp1256]{inputenc}
\usepackage[arabic,english]{babel}
\begin{document}
\selectlanguage{arabic}
بسم الله الرحمن الرحيم .
\\
الفصل السادس عشر في الاستخارة

في صحيح البخاري عن جابر قال كان رسول الله يعلمنا الاستخارة في الامر كما يعلمنا السورة من القران اذا هم
احدكم بالامر فليركع ركعتين من غير الفريضة ثم ليقل اللهم اني استخيرك بعلمك واستقدرك بقدرتك واسالك من
فضلك العظيم فانك تقدر ولا اقدر ونعلم ولا اعلم وانت علام الغيوب اللهم ان كنت تعلم ان هذا الامر وبسمى
حاجته خير لي في ديني ومعاشي وعاقبة امري فاقدري له ويسره لي ثم بارك لي فيه وان كنت تعلم ان هذا الامر شر
لي في ديني ومعاشي وعاقبة امري فاصرفه عني واصرفني عنه واقدر لي الخير حيث كان ثم ارضني به وفي مسند
الامام احمد من حديث سعد بن ابي وقاص عن النبي ص انه قال من سعادة ابن ادم استخارة الله ومن سعادة ابن
ادم رضاه بما قضى الله ومن شقوة ابن ادم نركه استخارة الله ومن شقوة ابن ادم سخطه بما قضى الله وقد قال
سبحانه ونعالى
[\textmash{
وشاورهم في الامر فاذا عزمت فتوكل على الله
}]
وقال قتاده ما نشاور قوم ينتغون وجه الله الا هدوا الى ارشد امرهم
\L{This is a simple example of Arabic text you may want to type}
ثم والحمد لله رب العالمين.
\end{document}

```

Figure 1: Sample Arabic input

be the one that dominates your text, either an Arabic (script) document with small amounts of Latin text included, or a Latin one that contains Arabic.

Arabi has also a *limited*, but almost good, capability of vocalizing. Some more work needs to be done in that direction. Things would have been certainly better if METAFONT had more powerful ligature possibilities! But if you use X_YTEX and have the *right* fonts, then things are certainly better.

The package also comes with extensive, and, we hope, clear documentation.

4.3 Current status

At the time of this writing, Arabi is at version 1.0, and already included in some distributions like MikTEX and BakomaTEX.

The latest version is always available from the CTAN archives. You should find it at `CTAN:tex-archive/language/arabic/arabi`

As mentioned earlier, the package is distributed under the LPPL, and has the status “author-maintained”. It can be used *freely* (including commercially) to produce beautiful texts that mix Arabic with characters from other scripts.

Figure 1 shows a sample Arabi input document, and figure 2 the corresponding output.

4.4 Babel compliance

Arabi is fully L^AT_EX 2_ε and Babel compliant. It provides almost all the language-dependent strings for the *Arabic* and *Farsi* languages and can generate automatically the official *Jalali calendar*. The Farsi captions and the code for the Farsi date are from the FarsiTEX system.⁵ Moreover, all Babel language switching commands apply.

⁵ The FarsiTEX system seems unfortunately still not available with L^AT_EX 2_ε. We hope that the Farsi support offered by Arabi and the Farsi fonts from the Farsi web project that come with Arabi will be useful to all Farsi users.

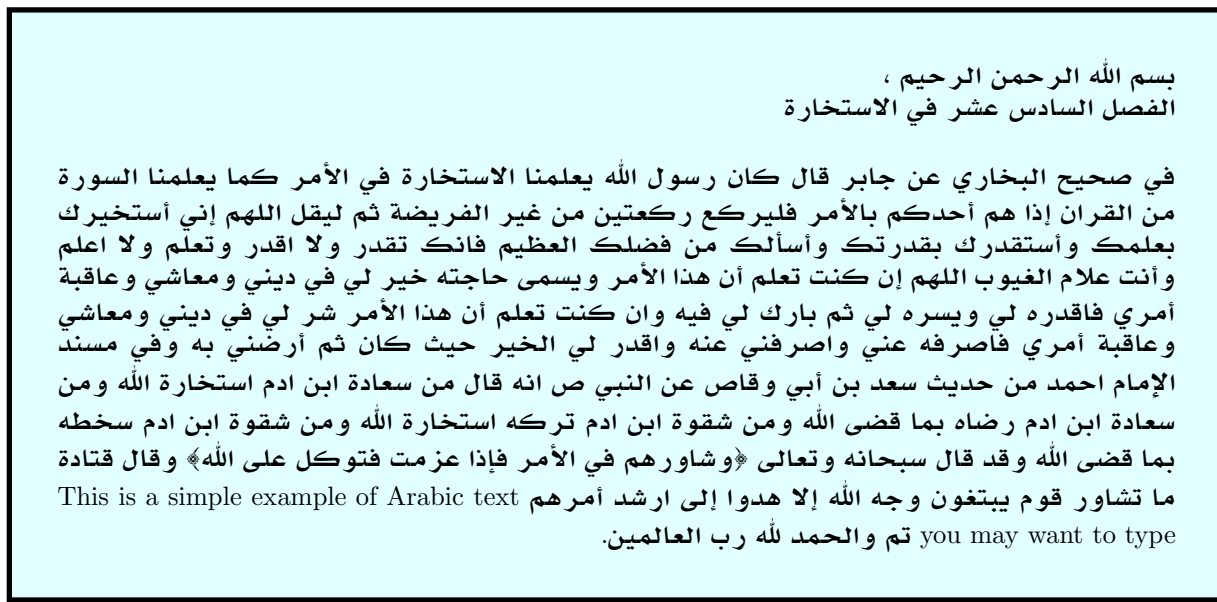


Figure 2: Sample Arabi output

4.5 Compatibility

The Arabi package has been tested successfully with packages such as `parshape`, `poster`, `pstricks` (and many of its derivatives), and, to our great surprise and pleasure, ArabT_EX. It has been tested also on a Mac OS X system with the t_EX distribution and TeXShop (see figure 3).

4.6 Arabic fonts

One of the good features in Arabi is its ability to use any existing fonts that the underlying T_EX engine can access. Arabi comes with a collection of Arabic and Farsi GNU fonts from, respectively, the Arabeyes and Farsi Web projects. The TFM files of some widely available commercial fonts are also included in the distribution, but the user still has to manage telling his engine where to find the corresponding font.

One remark to make here is that when preparing the vector encoding files for the different fonts, we learned that there is *no* standard. Even some corporations who produce and distribute applications and fonts that support the Arabic script for many years use so many names for the same glyphs that we arrived at the conclusion that one can never know what will be found when the font is opened!

5 Some bells and whistles

Arabi comes also with an *experimental* module that produces a *transliteration* of Arabic texts. No counterpart has been done for Farsi yet.

When texts are in general not fully vowelized, the transliteration cannot be expected to be correct. Moreover, when writing using some 8-bit input encoding there is absolutely no way to distinguish between long vowels **و ي ا** and the letters *alif*, *yaa* and *waw*. Neither, it is possible to write correctly the *hamza* when on *ʾalif*, *wāw*, or *yā*.

To use it, just load the package `translit` as with any other package, and type Arabic text in 8 bits in a Latin context, that is, without issuing a command that switches to the Arabic language.

1	<code>ʾabw āl-lāʾ ālm-r̄y</code>		أبو العلاء المعري	1
2	<code>matnuN mubārakuN</code>		مَنْ مَبَارَكٌ	2
3	<code>ḥġ mbrwr</code>		حج مبرور	3

Table 2: A little example of transliteration

Classical poetry, in both Arabic and Farsi, is formatted in two “parallel” verses that begin and end at the same positions. When verses are too short, they are written closer to the (vertical) center of the page, as in the next example. Arabi relies on the same idea⁶ of spreading the *keshida* glyph used by ArabT_EX.

⁶ A contribution by the author to ArabT_EX a long time ago. ArabT_EX uses a variable width horizontal “line” while we stack the *keshida* glyph the necessary number of times to get the right width!

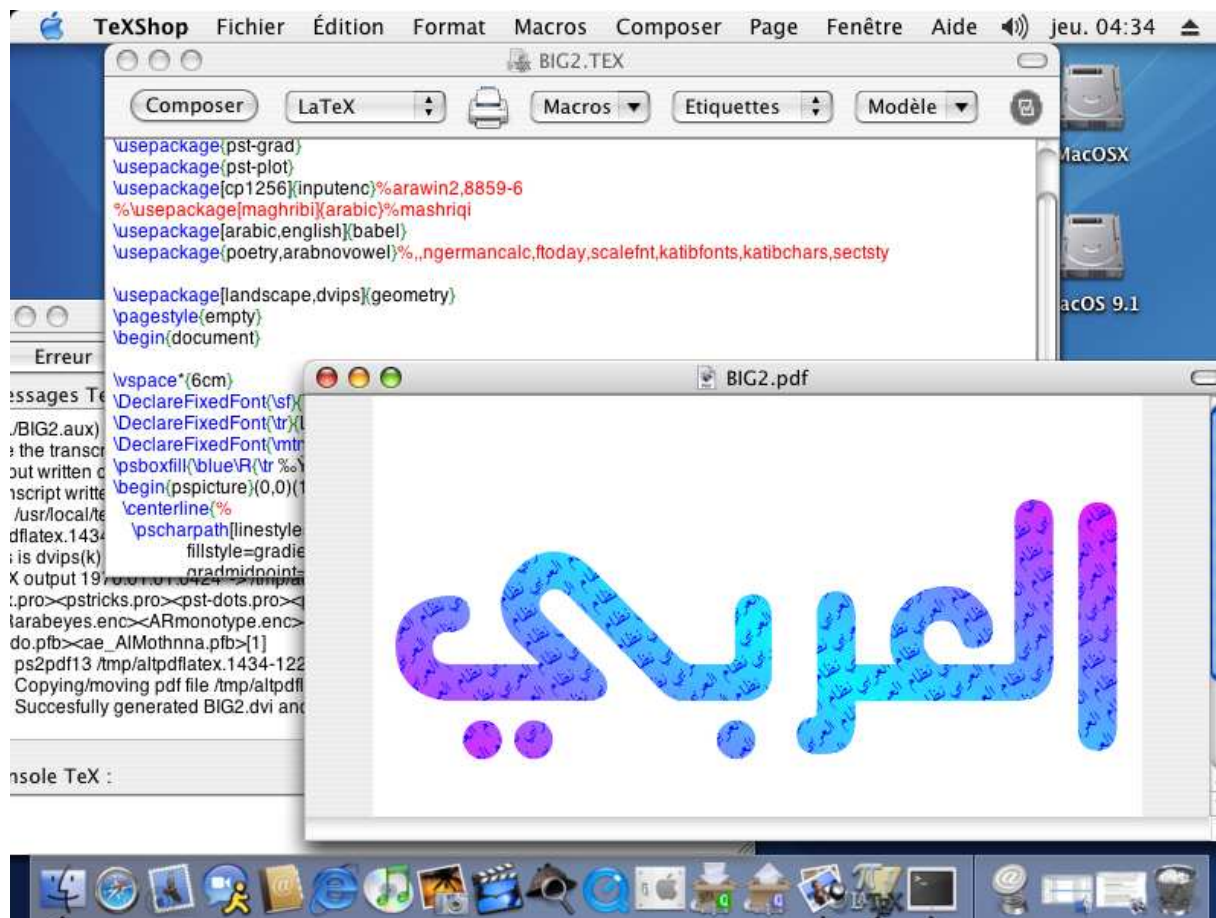


Figure 3: Arabi running on Mac OS

﴿ قَالَ الطُّغْرَائِي فِي لَامِبَةِ الْعَلَمِ ﴾

حب السلامة يثني هم صاحبه
 عن المعالي ويغري المرء بالكسل
 فإن جنحت إليه فاتخذ نفقا
 في الأرض أو سلما في الجو واعتزل
 لو كان في شرف المأوى بلوغ منى
 لم تبرح الشمس يوما دارة الحمل
 وشأن صدقك عند الناس كذبهم
 وهل يطابق معوج بمعتدل
 ملك القناعة لا يخشى عليه ولا
 يحتاج فيه إلى الأتصار والخول
 ترجو البقاء بدار لا ثبات لها
 فهل سمعت بظل غير منتقل

Figure 4: Some Arabic poetry

6 The limits and the problems

The main limit seems to be the capacity of the TFM format:

- First in its 256-glyph limit, which certainly is not a sufficient number for a modern font, not to talk about an Arabic one!
- And second in the very limiting way it handles ligatures. In a script like Arabic, three-character ligatures are the rule, while there are even four letter ligatures, e.g., محمد. But if we also want to manage diacritics, which we can recall play in Arabic the role of vowels in Latin languages, things become even worse.

There is also an important ϵ -TeX issue, that the R2L direction is not supported in Mathematics. So we have to rely on some script *à la* Knuth and MacKay to reverse the characters and the words.

7 The future for Arabi

Concerning the future developments of Arabi. In these early times, we focus on keeping it alive and

bringing it to maturity by correcting any bug that appears and completing the already existing functions, as no one is perfect. Let us cite the poet *al-mutanabbi*: **قال أبو الطيب المتنبّي**

ولم أر في عيوب الناس عيباً
كنقص القادرين على التمام

Please do not hesitate to forward suggestions, questions, or comments on Arabi. Thanks for your interest.

References

- [1] B. Esfahbod and R. Pournader. FarsiT_EX and the Iranian T_EX community. *TUGboat* 23(1), 41–45, 2002.
- [2] J. Braams. Babel, a multilingual style-option system for use with L^AT_EX’s standard document styles. *TUGboat* 12(2), 291–301, 1992.
- [3] J. Braams. An update on the Babel system. *TUGboat* 14(1), 60–61, 1993.
- [4] Michel Goossens and Frank Mittelbach, with Johannes Braams, David Carlisle, and Chris Rowley. *The L^AT_EX Companion*. Addison-Wesley, 2nd edition, 2004.
- [5] Y. Haralambous. Towards the revival of traditional Arabic typography ... through T_EX. Proceedings of the EuroT_EX’92 conference, Prague, 1992.
- [6] Y. Haralambous. Typesetting the Holy Qur’ān with T_EX. Proceedings of the 2nd International Conference on Multilingual Computing (Latin and Arabic script), Durham, 1992.
- [7] A. Hoenig. *T_EX Unbound: Strategies for Fonts, Graphics, and More*. Oxford University Press, 1998.
- [8] D.E. Knuth. *The T_EXbook*. Addison-Wesley, Reading, MA, USA, 1986.
- [9] D.E. Knuth. *The METAFONTbook*. Addison-Wesley, Reading, MA, USA, 1986.
- [10] D.E. Knuth. Virtual Fonts: More fun for grand wizards. *TUGboat* 11(1), 13–23, April 1990.
- [11] D.E. Knuth and P. MacKay. Mixing right-to-left texts with left-to-right texts. *TUGboat* 8(1), 14–25, April 1987.
- [12] A. Lakhdar-Ghazal. *Caractères arabes diacritiques selon l’ASV-CODAR (pour imprimer les langues arabes)*. Institut d’Études et de Recherches pour l’Arabisation, Rabat, 1993.
- [13] K. Lagally. ArabT_EX — Typesetting Arabic with vowels and ligatures. Proceedings of the EuroT_EX92 conference, Prague, 1992
- [14] K. Lagally. *ArabT_EX Arabic and Hebrew, (Draft) User Manual Version 4.00*. March 11, 2004.
- [15] L. Lamport. *L^AT_EX: A Document Preparation System: User’s Guide and Reference Manual*. Addison-Wesley, Reading, MA, USA, second edition, 1994.
- [16] P. MacKay. Typesetting problem scripts. *BYTE* 11(2), 201–218, 1986.
- [17] The FarsiT_EX Project. <http://www.farsitex.org/>
- [18] The FarsiWeb Project. <http://www.farsiweb.info/>
- [19] Institute of Standards and Industrial Research of Iran. <http://www.isiri.com>
- [20] Microsoft. Free download of the Arabic font pack, `arafonts.exe`. <http://office.microsoft.com/arabicregion/Downloads/2000/arafonts.aspx>
- [21] X_YT_EX web site and mailing list. <http://scripts.sil.org/xetex>
- [22] The Unicode standard. <http://www.unicode.org/>