ArabTEX, a System for Typesetting Arabic

Klaus Lagally*

Abstract

TEX, the powerful computer typesetting system by D.E. Knuth, and LATEX, its extension by L. Lamport to handle structured documents, have both been adapted to handle passages of arabic script. Our new system, ArabTEX, will accept as input an ASCII encoding of the standard transliteration of Arabic, and will generate the arabic writing with a full complement of vowel marks, automatically producing most of the usual ligatures, and obeying the common writing rules. Likewise, or additionally, the standard transliteration can also be generated from the same input. The notation is easily readable and suitable for electronic transmission. For non-vocalized writing, a reduced input notation is available, as are some extensions for other languages using the arabic script, and for reproducing erroneous or archaic documents.

ArabTEX will run on a wide range of hardware platforms in conjunction with any TEX implementation that can be extended by loading additional macros and additional fonts. No special equipment is required.

1 Overview

We present ArabTEX, a system for preparing documents in some European language which contain passages in Arabic or some other language using the arabic script.

ArabTEX is not a general purpose word processing system for producing, e.g., arabic newspapers, or for everyday office use in an arabic environment. For these application fields good systems already exist; however, these systems are usually based on dedicated hardware equipment, and they also require special training for efficient use. ArabTEX, on the other hand, is mainly targeted towards use by scholars with little or no computer or typesetting experience, and who have no specialized equipment available besides a standard PC or work station with a high resolution printer.

1.1 On Mathematical Typesetting

In order to explain the operation of ArabTEX, let us consider the related problem of mathematical typesetting first.

Typesetting mathematical formulas is inherently difficult for several reasons:

- there is a very large number of different symbols, some of them occurring in various shapes and sizes;

*CV: born in 1937, Public School, Graduation 1956, University Studies in Mathematics and Physics, Ph.D. in Theoretical Physics 1967, Work on Operating Systems and Programming Languages, Professor of Computer Science 1976, Universität Stuttgart, Germany

$$Y_l^m = e^{im\phi} \cdot \sqrt{\frac{2l+1}{4\pi}} \cdot \sqrt{\frac{(l-m)!}{(l+m)!}} \cdot \frac{(-1)^{l+m}}{2^l \cdot l!} \cdot \sin^m \theta \cdot \left(\frac{\mathrm{d}}{\mathrm{d}\cos\theta}\right)^{l+m} \sin^{2l} \theta$$

Figure 1: Example of mathematical text

- these symbols have to be arranged in a two-dimensional pattern according to the structure of the formula;
- frequently there are several possibilities of rendering a given formula, and choosing the optimal way requires knowledge both in typesetting and mathematics.

This is a rare combination of skills, and thus there are not too many publishing houses capable of producing printed mathematical text of high quality. Also, specialized equipment is needed, and the communication between the author and the publisher, due to their different views, might not be too easy. Thus, producing a mathematical textbook is a time-consuming and expensive task, and publications on a smaller scale are usually reproduced from a camera-ready original prepared by the author himself, either using a specialized typewriter or inserting the formulas by hand, (as this author had to do when preparing his Ph.D. thesis.)

This state of affairs led D.E. Knuth to the development of TEX [Knuth84], a computer program that enables the author to do the typesetting himself. The basic idea behind it is that the author will input his text concentrating on its content and logical structure, and the program will do the typesetting automatically, drawing on a large body of specialized knowledge. The results look impressive, and indeed several textbooks have been produced using TEX.

Even when using TEX, the production of a fair-looking document is not an easy task, as there still is a very high variety of possible layouts for the same text, and a casual user might easily go astray. Fortunately TEX is not only a typesetter, but also an interpreter for its special (macro) programming language, and most documents are structured according to one of a few standard patterns. Thus by providing a set of standard definitions, e.g. the macro package LATEX by L. Lamport [Lamport86], the user's task can be reduced to "filling in the blanks", and the macros will automatically take care of clerical tasks like section and subsection numbering, placement of figures and tables, building of itemized or enumerated lists, cross-referencing, indexing, etc.

1.2 Some Problems with Arabic

Typesetting arabic text inside a document in some European language has to cope with the same problems the user faces when including mathematical formulas, and a few more:

- Arabic runs from right to left.

  This means that the sophisticated line-breaking algorithm of TEX will not work for arabic text.

- Arabic is written in a cursive style.

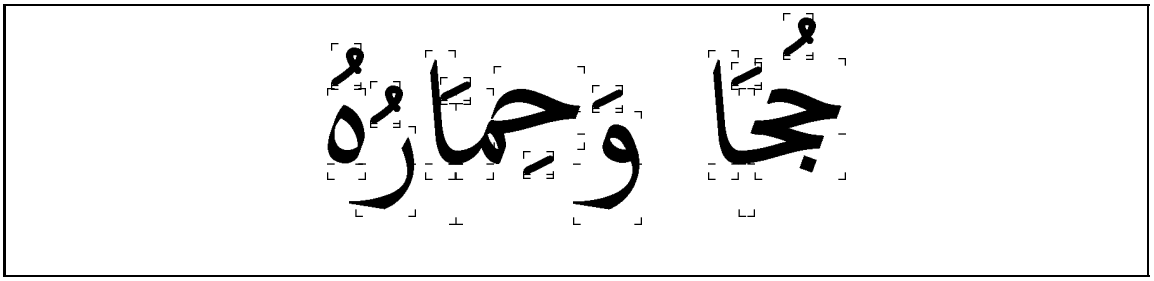

Figure 2: Character assembly with components shown.

We will have to join the individual graphical symbols arising from the arabic characters into a softly flowing curve, and the form of these symbols is strongly context-dependent.

- There is a large number of ligatures, and some of these are mandatory.

  A mathematical formula is a highly structured entity, and its two-dimensional layout is basically determined by this structure. Whereas this is also true for the arabic writing, the user will usually think of an arabic word as being a linear sequence of individual letters. We ought not to emburden him with the task of specifying exactly the graphical representation which is influenced by a number of well-known and time-honoured rules.

- There is a complex set of diacritical marks whose use should be controllable by the user.

  Diacritical marks are usually redundant and therefore omitted in most cases, but under special circumstances they are essential, and the user should have full control over their use.

- Sometimes also the standard transliteration is required.

  Whereas this is not a problem of arabic typesetting at all, the transliteration uses many symbols outside the standard character set which cannot be coded directly, and to avoid confusion, we should be able to obtain it from an input notation not too different from that for denoting the arabic writing.

In addition we require that the user need not be a computer expert, and that the input notation for arabic text be easily readable and suitable for electronic transmission.

It turns out that the technique of extending TEX, or LATEX, by still another macro package is sufficiently powerful to construct a system that fulfills the given requirements. Technical details are outside the scope of this presentation; they can be found, e.g., in [Lagally92b].

2 Input Notation for Arabic

TEX is not an interactive system. It will transform a given input file into a device-independent output representation which afterwards, by some device-specific driver program, can be sent, e.g. to a laser printer, to a photo-typesetter, or can be viewed on a high

resolution computer screen. If, as is normally the case, the user has to correct some typos or wants to change the appearance of the document, he has to edit the input file by means of some text editor, and therefore the input representation must be easily readable and in close correspondence to the desired output.

For Arabic, a plausible candidate seems to be the standard transliteration [DIN31635], [ISO/R233]; however, it cannot be used directly as it makes heavy use of diacritical marks, most of which are not readily available on the common computer keyboards, and whose internal coding unfortunately is in no way standardized. Also there are many national keyboard variants handling these special symbols differently. For a truly portable system, we should use only the symbols common to all keyboards, i.e. letters, figures, and punctuation marks.

2.1 The Basic Coding Scheme

a	ا	*a*	b	ب	*b*	p	پ	*p*	t	ت	*t*
_t	ث	*t̲*	^g	ج	*ǧ*	.h	ح	*ḥ*	_h	خ	*ḫ*
c	ځ	*c*	^c	چ	*č*	,c	څ	*ć*	d	د	*d*
_d	ذ	*d̲*	r	ر	*r*	z	ز	*z*	^z	ژ	*ž*
s	س	*s*	^s	ش	*š*	.s	ص	*ṣ*	.d	ض	*ḍ*
.t	ط	*ṭ*	.z	ظ	*ẓ*	‘	ع	‘	.g	غ	*ġ*
f	ف	*f*	q	ق	*q*	v	ڤ	*v*	k	ك	*k*
g	گ	*g*	l	ل	*l*	m	م	*m*	n	ن	*n*
h	ه	*h*	w	و	*w*	y	ي	*y*	T	ة	*t*

Figure 3: Coding of arabic characters

The ArabTEX encoding is based on the standard transliteration, but uses one- and two-character encodings according to the following rules (see Figure 3):

- whenever the transliteration uses just a single letter, we also use that letter;
- whenever the transliteration uses a letter with a diacritical mark, we use the same letter and *precede* it with the punctuation mark most closely resembling the diacritic.
- `<A>, <I>, <U>` denote the long vowels, `<a>, <i>, <u>` the short vowels if required.
- `<'>` (right quote) is *hamza* (glottal stop). If the arabic writing mode has been selected, its carrier will be determined by the context according to the full *hamza* rules, otherwise by a following short vowel.
- `<'A>` generates *madda*.
- `<T>` is *ta' marbuta*, `<N>` is *tanwin*, `<Y>` is *alif maqsura*.

- Doubled consonants are written twice (*shadda*).

This is easily remembered, fairly readable, and works well because punctuation marks (except hyphen) never occur within a word.

2.2 Variants and Extensions

The coding scheme as given contains the full information necessary for obtaining both the fully vocalized arabic writing and the standard transliteration. For an experienced user, some simplifications are possible: if neither the vocalization nor the transliteration are required the user does not need to denote the short vowels except in cases where they influence the *hamza* writing. Also there are many additional options:

- the vocalization can be controlled in three levels, and locally;
- there are language-specific modes for Farsi, Urdu, and Pashto with the necessary additional input codings;
- the generation of ligatures can be locally modified;
- all default settings can be locally overridden.

Full details are given in [Lagally92a].

3 Examples of Document Structure

ArabTEX follows the TEX paradigm of the user specifying the logical structure of the document, and letting the computer worry about the typesetting details. TEX itself offers a large set of mechanisms for describing the desired appearance of the document, and ArabTEX just adds a few commands for indicating arabic text, plus the internal routines responsible for the language-specific processing. As the basic TEX mechanisms are comparatively low-level, an inexperienced user will rather start with LATEX to handle the standard cases in a convenient way. Users preferring to work with Plain TEX can of course do so.

A LATEX document consists of a header specifying the document style and possibly modifying some parameters, and a document body containing the text of the document in free format and grouped according to the logical structure of the document.

For an example of the input format, see Figure 4 which should be fairly self-explanatory. Note that the percent mark indicates a comment for a human reader of the source text, that is not otherwise processed. The output for the same text is shown in Figure 5.

Figure 6 shows a more realistic example. Here both the arabic writing and the transliteration have been switched on, and the transliteration output is interleaved automatically with the arabic writing.

Still another example is this paper itself [Lagally92c]: it has been produced using LATEX and ArabTEX without any manual cutting and pasting. (We had to use some technical tricks to produce Figure 2 which is non-standard.)

```
\documentstyle[11pt,dina4,arabtex,atrans]{article}
% choose a document style, the type size and paper format
% load the ArabTeX macros and the transliteration module
\setarab  % select language-specific processing, e.g. for <hamzaT>
\vocalize % indicate short vowels by diacritics
\begin{document}

This is a short demonstration. We start with an arabic insertion
<^gu.hA wa-.himAruhu>
inside a line of English text.
Please note the automatic formatting of this paragraph
which ends with a blank line.

By changing some switch settings,
\arabfalse  % no arabic writing
\transtrue  % transliterate
we can also produce the transliteration from the same input:
<^gu.hA wa-.himAruhu>,
and we should not forget to switch back!
\arabtrue   % arabic writing on again
\transfalse % no transliteration

For longer arabic texts \ArabTeX\ has to do the line-breaking:
\begin{arabtext}
'at_A .sadIquN 'il_A ^gu.hA ya.tlubu minhu .himarahu li-yarkabahu
fI safraTiN qa.sIraTiN wa-qAla lahu :
sawfa 'u`Iduhu 'ilayka fI al-masA'i , wa-'adfa`u laka 'u^graTaN .
\end{arabtext}
As we see, including arabic text is not difficult.
\end{document}
```

Figure 4: Input for a sample LaTeX document

This is a short demonstration. We start with an arabic insertion جُحَا وَحِمَارُهُ inside a line of English text. Please note the automatic formatting of this paragraph which ends with a blank line.
By changing some switch settings, we can also produce the transliteration from the same input: *ǧuḥā wa-ḥimāruhu*, and we should not forget to switch back!
For longer arabic texts ArabTeX has to do the line-breaking:

أَتَى صَدِيقٌ إِلَى جُحَا يَطلُبُ مِنهُ حِمَرَهُ لِيَركَبَهُ فِي سَفرَةٍ قَصِيرَةٍ وَقَالَ لَهُ : سَوفَ أُعِيدُهُ إِلَيكَ فِي المَسَاءِ ، وَأَدفَعُ لَكَ أُجرَةً .

As we see, including arabic text is not difficult.

Figure 5: Output for the sample LaTeX document

جُحَا وَحِمَارُهُ *ǧuḥā wa-ḥimāruhu*

›atā ṣadīqun ›ilā ǧuḥā yaṭlubu minhu ḥimārahu li-yarkabahu fī safratin qaṣīratin wa-qāla lahu: sawfa ›u‹īduhu ›ilayka fī 'l-masā›i

أَتَى صَدِيقٌ إِلَى جُحَا يَطلُبُ مِنهُ حِمَارَهُ لِيَركَبَهُ فِي سَفرَةٍ قَصِيرَةٍ وَقَالَ لَهُ : سَوفَ أُعِيدُهُ إِلَيكَ فِي المَسَاءِ

, wa-›adfa‹u laka ›uǧratan.

، وَأَدفَعُ لَكَ أُجرَةً .

fa-qāla ǧuḥā: ›anā ›āsifun ǧiddan ›annī lā ›astaṭī‹u ›an ›uḥaqqiqa laka raġbataka, fa-'lḥimāru laysa hunā 'l-yawma.

فَقَالَ جُحَا : أَنَا آسِفٌ جِدًّا أَنِّي لَا أَستَطِيعُ أَن أُحَقِّقَ لَكَ رَغبَتَكَ ، فَالحِمَارُ لَيسَ هُنَا اليَومَ .

wa-qabla ›an yutimmu ǧuḥā kalāmahu bada›a 'l-ḥimāru yanhaqu fī 'ṣṭablihi. fa-qāla lahu ṣadīquhu: ›innī ›asma‹u ḥimāraka yā ǧuḥā

وَقَبلَ أَن يُتِمَّ جُحَا كَلَامَهُ بَدَأَ الحِمَارُ يَنهَقُ فِي اصطَبلِهِ . فَقَالَ لَهُ صَدِيقُهُ : إِنِّي أَسمَعُ حِمَارَكَ يَا جُحَا

yanhaqu.

يَنهَقُ .

fa-qāla lahu ǧuḥā: ġarībun ›amruka yā ṣadīqī! ›atuṣaddiqu 'l-ḥimāra wa-tukaḏḏibunī?

فَقَالَ لَهُ جُحَا : غَرِيبٌ أَمرُكَ يَا صَدِيقِي ! أَتُصَدِّقُ الحِمَارَ وَتُكَذِّبُنِي ؟

Figure 6: Arabic text with simultaneous transliteration.

4 Availability and first Experiences

ArabTEX is freely available for scientific and private use, without any guarantee for correctness, and without any explicit or implied warranty. It can be picked up via FTP from `ifi.informatik.uni-stuttgart.de` (129.69.211.1), directory `pub/arabtex`. Prospective users without Internet access should contact the author.

A first version of ArabTEX has been distributed on the Internet in August 1991. In the sequel, many error reports and suggestions for improvement reached us, which led to a second, much improved and expanded, version in May 1992. The system is still in an experimental stage; yet about 500 persons and institutions downloaded it up to now, and presently about a dozen errors are known and being corrected. We know of a few institutions using the system for production work; e.g., the American Arab Scientific Society is typesetting its quarterly newsletter using ArabTEX.

5 Acknowledgments

The development of ArabTEX would not have been possible without the assistance of many people. Apart from our local team, Udo Merkel and Heribert Schlebbe, helpful advice came among others from Ivan Derzhansky, Wolfdietrich Fischer, Ahmed El-Hadi, Abdelsalam Heddaya, Iqbal Khan, Tom Koornwinder, Eberhard Krüger, Asif Lakehsar, Jan Lodder, Richard Lorch, Eberhard Mattes, and Bernd Raichle. We also have to thank the many users who sent error reports, comments, and suggestions.

References

[DIN31635] DIN 31 635: *Umschrift des Arabischen Alphabets*, Deutsches Institut für Normung e.V., 1982.

[ISO/R233] ISO/R 233 - 1961: *International System for the Transliteration of Arabic Characters*, International Standards Institution, 1961.

[Knuth84] Donald E. Knuth, *The TEXbook*, Volume A of *Computers & Typesetting*, Addison-Wesley, Reading, Mass., 1984.

[Lagally92a] Klaus Lagally, *ArabTEX, a System for Typesetting Arabic*, User manual. Report 1992/06, Fakultät Informatik, Universität Stuttgart, 1992.

[Lagally92b] Klaus Lagally, *ArabTEX - Typesetting Arabic with Vowels and Ligatures*, in: *EuroTEX '92, Proceedings of the 7th European TEX Conference, Prague, Czechoslovakia, September 14-18, 1992.* Also available as: Report 1992/07, Fakultät Informatik, Universität Stuttgart, 1992.

[Lagally92c] Klaus Lagally, *ArabTEX, a System for Typesetting Arabic*, Paper to be presented at the *3rd International Conference and Exhibition on Multi-lingual Computing (Arabic and Roman Script), University of Durham, UK, December 10-12, 1992.* Report 1992/11, Fakultät Informatik, Universität Stuttgart, 1992.

[Lamport86] Leslie Lamport, LaTeX, *a Document Preparation System*, Addison-Wesley, Reading, Mass., 1986.

Author's address:

Prof. Klaus Lagally
Institut für Informatik
Universität Stuttgart
Breitwiesenstraße 20-22
W-7000 Stuttgart 80
GERMANY

`lagally@informatik.uni-stuttgart.de`