RyDArab — Typesetting Arabic mathematical expressions

Azzeddine Lazrek

Abstract

TEX was adapted to handle passages of Arabic script some years ago, via packages such as ArabTEX or Ω. The package RyDArab, presented herein, extends these to handle mathematics in an Arabic presentation. That means expressions with specific symbols flowing from right to left, according to the Arabic writing, as they can be found in Arabic mathematical handbooks.

1 Overview

Although TEX [3] was designed according to the conventions of English and Western languages, it is also able to handle passages of Arabic script [4]. Systems such as ArabTEX, by K. Lagally [5], and Ω, by Y. Haralambous and J. Plaice [2], allow generating documents with passages in Arabic or some other language using the Arabic script. Even though many Arabic mathematical texts present mathematical expressions as they are in English or in French texts, a large number of Arabic handbooks display mathematical expressions using specific symbols in a writing flowing from right to left. Arabic mathematical document processing has been discussed in [10] and [11].

The RyDArab system presented here is designed for generating Arabic texts including such mathematical writing. The system consists of a set of TEX macro packages, some additional extensions and a family of symbol fonts. It will run under the Plain TEX or LATEX [6] formats. The present paper was typeset with this package.

This paper is organized as follows: in section 2, we show how to load the package. In section 3, we present some package options. In section 4, we go over commands that can be used to typeset Arabic mathematical expressions and show some examples. Of course, some problems are still to be solved. In particular, some compatibility questions are under investigation. There are still open questions that are outside the scope of this paper.

Throughout the present paper, we speak about "Arabic mathematical" texts, documents, expressions, and so on. Of course, mathematics is unique; it has nothing to do with any national specificity. When we use this appellation, it refers only to the way in which mathematics is presented in most common Arabic writing.

Editor's note: This article originally appeared in *Die TEXnische Komödie* 2/2001, and is reprinted here by permission, with many improvements by the author.

2 Preamble

To use RyDArab to generate Arabic mathematical expressions, first load the system by putting `\input rydarab` (for Plain TEX) or `\usepackage{rydarab}` (for LATEX) in the preamble of your document.

The ArabTEX package or Ω has to be loaded first. That will be the system for typesetting the textual component of the document.

3 Options

The commands defined in RyDArab are prefixed with the initials "`am`" (for Arabic mathematical). This helps to distinguish RyDArab's commands from the basic commands of TEX, LATEX, or other packages. The other part of the command names derive from the corresponding TEX or LATEX commands.

The following options are offered as commands or options of the package:

arabmath to begin an environment where Arabic mathematical expressions are generated (e.g. $\bar{c}\vee$). This is the default.

latinmath to begin an environment where mathematical expressions are in their Latin form (e.g. $\sqrt{j}$).

warabnum for using the standard western Arabic digits ($\{0,1,2,3,4,5,6,7,8,9\}$). These digits — known as Arabic digits — are used in North Africa. This is the default.

earabnum for using the eastern Arabic digits ({٠, ١, ٢, ٣, ٤, ٥, ٦, ٧, ٨, ٩}). These numerals — known as Hindi digits — are used in the Middle East.

alpwithoutdots for using the alphabetic symbols without dots. This is the default.

alpwithdots for using the alphabetic symbols with dots.

funwithdots for using abbreviations representing elementary functions with dots. This is the default.

funwithoutdots for using abbreviations representing elementary functions without dots.

If these options are specified in the preamble of the input file, they work for the whole document. If they are specified at the beginning of an environment, in mathematical mode, the option is valid only through the end of the environment.

4 Commands

In addition to the basic set of TEX commands, there are new commands and commands resulting from some transformations of similar commands in TEX.

All these work in math mode only, either in display or inline environments.

The commands are listed below in an Arabic mathematical environment. The resulting text will appear in the frame following or in front of the command.

4.1 Inversion of direction

`\amrl{`*expr*`}` inverts the direction of writing in order to generate an Arabic mathematical expression *expr* from right to left. Notice that the command `\amrl` does not change the direction of text portions given in additional braces:

```
$ {\amrl{ abjd }}$
$ {\amrl{ {abjd} }}$
$ {\amrl{ a{bj}d }}$
```

If the command `\amrl` is used in the argument of `\amrl`, both it and its argument must be enclosed in braces:

```
$ {\amrl{ a\amrl{bj}d }}$
$ {\amrl{ a{\amrl{bj}}d }}$
$ {\amrl{ \amrl{abjd} }}$
$ {\amrl{ {\amrl{abjd}} }}$
```

In general, all commands with arguments that are used in an `\amrl` command, and their arguments, should be written in braces unless the argument consists of a single character. In the latest versions of RyDArab, the use of the `\amrl` command in an Arabic mathematical environment is transparent for the user. It is added automatically to the expression in an Arabic environment.

```
\arabmath
${\hat b} , b{\sp{{17}}} ,
 {\sqrt{s+3}}$
```

4.2 Alphabetic symbols

The RyDArab system provides various Arabic characters without dots or vowels or diacritics, including three shapes of Arabic characters (initial, isolated and with a tail) in bold or contour forms.

Arabic literal symbols are given via the transliteration TransTec [1] (see Table 1), which is an adaptation of ArabTEX's [7] font `xnsh`. This coding differs slightly from the basic one used in ArabTEX to generate text in Arabic. This transliteration can be used either in text or mathematical expressions. In order to use this transliteration in text also, the user should load the transtec[1] package and add the command `\setcode{transtec}` in the preamble. The transliteration in use is not optimal. Since the package is intended for an Arab user, it would be better to get a direct coding scheme from the keyboard.

Some Arabic literal symbols can be obtained from the font NasX [8] also. In order to use this font, the user should load the NasX[2] package unless

Letter	Text	Mathematical expression					
ا	a	ا	a				
ب	b	ب	b	ر	B	ـبـ	\amb
ت	t						
ث	F						
ج	j	ح	j	ـحـ	J	ـجـ	\amj
ح	H						
خ	X						
د	d	د	d				
ذ	Z						
ر	r						
ز	z	ر	z				
س	s	س	c	ـسـ	C	ـسـ	\amc
ش	C						
ص	S	ص	s	ـصـ	S	ـصـ	\ams
ض	D						
ط	T	ط	T			ـطـ	\amt
ظ	V						
ع	E	ع	e	ـعـ	E	ـعـ	\ame
غ	G						
ف	f	ف	f	ـفـ	F	ـفـ	\amf
ق	q	ق	q				
ك	k	ك	k			ـكـ	\amk
ل	l	ل	l	ـلـ	L	ـلـ	\aml
م	m	م	m	ـمـ	M	ـمـ	\amm
ن	n	ن	n	ـنـ	N		
ه	h			ـهـ	H	ـهـ	\amh
ط	Q						
و	w	و	w				
ي	y	ى	y	ـيـ	Y	ـيـ	\amy
ى	Y						
ـَ	e						
ـُ	u						
ـِ	i						
ـٖ	eN *or* ee						
ـٗ	uN *or* uu						
ـٕ	iN *or* ii						
ـّ	W						
ـْ	o						
ء	A-	ء	A				
أ	Aa						
إ	AY						
ؤ	Aw						
ئ	Ay						
آ	Ma						
آ	La						
ـ	K *or* -						

Table 1: TransTec transliteration

[1] The transtec package was developed by K. Lagally following our propositions. Our thanks to K. Lagally.

[2] The NasX package is the core of an Arabic mathematical font. It was developed in order to have multiple glyphs of Arabic literal symbols directly in METAFONT.

the CurExt[3] [9] package is already in use. Literal symbols are obtained via commands and not directly (see Table 2).

The Latin literal symbols are also offered:

```
${\latinletter A} ,
 {\latinletter B} ,
 {\latinletter C}$
```

$C\,،\,B\,،\,A$

or

```
{\latinletter $A , B , C$}
```

4.3 Accents

Accents, in various shapes, can be combined with one or several symbols. Accents can be placed beside the symbol on its left.

The prime accent can be oriented to the left:

```
$a{'} , b{'} , j{'}$
```

or

```
$a{}^{\prime}, b{}^{\prime},
 j{}^{\prime}$
```

ا′، ب′، ح′

and the prime accent can be oriented to the right:

```
$a{}^{\amprime} ,
 b{}^{\amprime} ,
 j{}^{\amprime}$
```

ا‵، ب‵، ح‵

We can also have multiple prime accents oriented to the left:

```
$b{'} , b{'}{'} ,
 b{'}{'}{'}$
```

or

```
$b{}^{\prime} ,
 b{}^{\prime\prime} ,
 b{}^{\prime\prime\prime}$
```

ب′، ب″، ب‴

and multiple prime accents oriented to the right:

```
$b{}^{\amprime} ,
 b{}^{\amprime\amprime} ,
 b{}^{\amprime\amprime\amprime}$
```

ب‵، ب‶، ب‷

Several accents are offered:

```
${\hat b} , {\check b} , {\tilde b} ,
 {\acute b} , {\grave b} , {\dot b} ,
 {\ddot b} , {\breve b} , {\bar b} ,
 {\vec b} , {\overleftarrow{b}} ,
 {\overrightarrow{b}}$
```

$\overrightarrow{b}\,،\,\overleftarrow{b}\,،\,\vec{b}\,،\,\bar{b}\,،\,\breve{b}\,،\,\ddot{b}\,،\,\dot{b}\,،\,\grave{b}\,،\,\acute{b}\,،\,\tilde{b}\,،\,\check{b}\,،\,\hat{b}$

[3] CurExt is an application for composing variable-sized curvilinear symbols. It allows looking after the typography of symbols such as brackets or kashida of Arabic letters in the composition of Arabic mathematical symbols or the justification of Arabic cursive texts according to strict rules of Arabic calligraphy.

Command	Glyph	Command	Glyph	Command	Glyph
\alef	ا	\aalef	ا		
\beh	ب	\bbeh	ب	\BEH	ب
\jeem	ج	\jjeem	ج	\JEEM	ج
\dal	د	\ddal	د		
\waw	و	\wwaw	و		
\zain	ر	\zzain	ر		
\tah	ط	\ttah	ط	\TAH	ط
\yeh	ى	\yyeh	ى		
\lam	ل	\llam	ل	\LAM	ل
\meem	م	\mmeem	م	\MEEM	م
\noon	ن	\nnoon	ن		
\seen	س	\sseen	س	\SEEN	س
\ain	ع	\aain	ع	\AIN	ع
\feh	ف	\ffeh	ف	\FEH	ف
\sad	ص	\ssad	ص	\SAD	ص
\qaf	ق	\qqaf	ق		
\hamza	ء	\hhamza	ء	\HEH	ه
\lamalef	لا	\llamalef	لا	\KAF	ك
\Yeh	ے	\YYeh	ے	\TTAH	ظ
\Noon	ں	\NNoon	ں		
\Zain	ڑ	\ZZain	ڑ		
\MEem	۾	\MMEem	۾		
\Beh	ہ	\BBeh	ہ	\BBEH	ہ
\Jeem	ځ	\JJeem	ځ	\JJEEM	ځ
\Heh	ھ	\HHeh	ھ	\HHEH	ھ
\Kaf	ڪ	\KKaf	ڪ	\KKAF	ڪ
\Lam	ڵ	\LLam	ڵ	\LLAM	ڵ
\Meem	ھ	\MMeem	ھ	\MMEEM	ھ
\Seen	۽	\SSeen	۽	\SSEEN	۽
\Ain	۽	\AAin	۽	\AAIN	۽
\Feh	۾	\FFeh	۾	\FFEH	۾
\Sad	ص	\SSad	ص	\SSAD	ص

Table 2: Arabic literal symbol font NasX

Some accents are variable-sized:

```
${\underline{abj}} , {\overline{abj}} ,
 {\widehat{abj}} , {\widetilde{abj}} ,
 {\overleftarrow{abj}} ,
 {\overrightarrow{abj}} ,
 {\overbrace{abj}} , {\underbrace{abj}}$
```

ابج ، ابج ، ابج ، ابج ، ابج ، ابج ، ابج

4.4 Digits

Eastern or western Arabic digits can be chosen according to the author's specifications, as follows.

For Western Arabic digits:

```
\warabnum
$0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9$
```

9 ، 8 ، 7 ، 6 ، 5 ، 4 ، 3 ، 2 ، 1 ، 0

Eastern Arabic digits:

```
\earabnum
$0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9$
```

٩ ، ٨ ، ٧ ، ٦ ، ٥ ، ٤ ، ٣ ، ٢ ، ١ ، ٠

Western Arabic digits in old style:

```
$ {\oldstylenum{\amrl{
 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9}}}$
```

or

```
${\oldstylenum{0}} , {\oldstylenum{1}} ,
 {\oldstylenum{2}} , {\oldstylenum{3}} ,
 {\oldstylenum{4}} , {\oldstylenum{5}} ,
 {\oldstylenum{6}} , {\oldstylenum{7}} ,
 {\oldstylenum{8}} , {\oldstylenum{9}}$
```

9 ، 8 ، 7 ، 6 ، 5 ، 4 ، 3 ، 2 ، 1 ، 0

4.5 Numbers

Numbers have to be enclosed within braces. The braces can be omitted only if a number consists of a single digit. The sign of the number should be put outside the braces.

Numbers can be given with or without a fractional part, separated either by a decimal comma or a decimal dot. The numbers can be prefixed by an optional sign. The first example shows the formatting according to North African typographic conventions and the second according to Middle Eastern.

```
$7 , {5} , +{92} , -8 , {107} ,
 {12.345} , {3{\latinmath ,}14}$
```

3,14 ، 12345 ، 107 ، 8− ، 92+ ، 5 ، 7

```
\earabnum$7 , {5} , +{92} , -8 , {107} ,
 {12.345} , {3{\latinmath ,}14}$
```

٧ ، ٥ ، +٩٢ ، −٨ ، ١٠٧ ، ١٢٬٣٤٥ ، ٣٫١٤

4.6 Punctuation

The Arabic punctuation system is provided:

```
$, . ; : ! ? - \cdots$
```

، . ؛ : ! ؟ − ⋯

```
\latinpunct $,$ or \latinmath $,$
or $\lating$
```

،

Of course, Latin dotting is also available:

```
\latinpunct
$, . ; : ! ? - \cdots$
```

, . ; : ! ? − ⋯

4.7 Delimiters

Variable-sized delimiters with automatic adjusting are obtained as follows:

```
$( , | , [ , || , \{ ,
 \} , || , ] , | , )$
```

(، | ، [، || ، {،} ، || ،] ، | ،)

and

```
$\langle , \lbrace ,
 \rbrace , \rangle$
```

⟨، {،} ،⟩

4.8 Symbols

The basic symbols have been adapted as follows:

1. Addition, subtraction, multiplication, equality:

```
$+ , - , * , \times , =$
```

=، × ، ∗ ، − ، +

2. Division, western percentage and per mille and eastern percentage and per mille:

```
$/ , \% , \wpermille ,
 \epercent , \epermille$
```

/، ٪ ، ‰ ، ٪ ، /

3. Inferior, superior, membership and capacity:

```
$< , > , \in , \ni$
```

∈،∋،<،>

4. Assignment, equality and equivalence by definition:

```
$\seqm , \leqm , \leqv$
```

تعر تعر
≔،=:،⟺

5. Radical, angle, existential and universal quantifier:

```
$\surd , \angle ,
 \exists , \forall$
```

∀ ، ∃ ، ∠ ، √

6. Negation:

```
${\not=} , {\not<} , {\not\in}$
```

≠،≮،∉

4.9 Superscript and subscript

Superscripts and subscripts can be placed at the left of any symbol of the equation.

The command `\sp{`*expr*`}` or `^`*expr* produces *expr* as an exponent. The exponent *expr* should be given within braces unless it consists of a single token. The command `^` does not change the direction of *expr*. It can therefore be used only for a single token.

```
$b{{}\sp{{17}}}+5 ; b{{}\sp{2}}+5*s{{}^b}$
```

$ب^{17}+5$ ؛ $ب^{2}+5*ص^{ب}$

The command \sb{*expr*} or _*expr* gives *expr* as an index. The index *expr* should be given within braces unless it consists of a single token. The command _ does not change the direction of *expr*. It can therefore be used only for a single token.

```
$b{{}\sb{{17}}}+5 ; b{{}\sb{2}}+5*s{{}_b}$
```

$ب_{17} + 5 \quad ؛ \quad ب_2 + 5 * ص_ب$

The empty braces {} are necessary to get the exponent or the index closer to the basic symbol.

4.10 Common functions

There are symbols for the usual abbreviations representing elementary functions in use in mathematics. Table 3 shows the predefined names assigned according to typographical conventions

Generally, the abbreviations representing elementary functions are used with dots. Sometimes, they are noted without dots.

```
\funwithdots
$\sin c + \tan s$
\funwithoutdots
$\sin c + \tan s$
```

جا س + ظا ص

حا س + طا ص

4.11 New function

The command \newfunc{*fname*} defines a function named *fname*.

```
${\newfunc{SGr}}(c) = \cos(c{{}\sp 2}) - 6$
```

صغر (س) = جتا $(س^2) - 6$

4.12 Function defined with cases

The command \cases{*array*} generates a function defined with different cases presented in *array*.

```
$d(c) = {\cases{
 {-4c} & {{\arhbox{ AYZa kan }} c<0} \cr
 { 4c} & {{\arhbox{ AYZa kan }} c>0} \cr
 {-2}  & {{\arhbox{ Gyr Zlk }}     } \cr
 }}$
```

$$د(س) = \begin{cases} -4س & \text{إذا كان } س > 0 \\ 4س & \text{إذا كان } س < 0 \\ -2 & \text{غير ذلك} \end{cases}$$

Mathematical Arabic symbols that stretch or shrink according to the context are provided by the system as well.

4.13 Root

The command \sqrt{*expr*} gives the square root of *expr*.

```
${\sqrt{{{ {b*9} \over {lc} }}}} -
 {\sqrt{c{{}\sp 2}}} + {\sqrt{5a}}$
```

$$\sqrt{\sqrt{\frac{ب*9}{لا}} - \sqrt{س^2} + \sqrt{ا5}}$$

Name	Example	Result
Sine	$\sin c$	جا س
Cosine	$\cos c$	جتا س
Tangent	$\tan c$	ظا ص
Cotangent	$\cot c$	ظتا س
Secant	$\sec s$	قا س
Cosecant	$\csc c$	قتا س
Arc cine	$\arcsin c$	زجا س
Arc cosine	$\arccos c$	زجتا س
Arc tangent	$\arctan c$	زظا س
Arc cotangent	$\arccot c$	زظتا س
Arc secant	$\arcsec c$	زقا س
Arc cosecant	$\arccsc c$	زقتا س
Hyperbolic sine	$\sinh c$	جاز س
Hyperbolic cosine	$\cosh c$	جتاز س
Hyperbolic tangent	$\tanh c$	ظاز س
Hyperbolic cotangent	$\coth c$	ظتاز ص
Hyperbolic secant	$\sech s$	قاز س
Hyperbolic cosecant	$\csch c$	قتاز س
Arc hyperbolic sine	$\arcsinh s$	زجاز س
Arc hyperbolic cosine	$\arccosh c$	زجتاز س
Arc hyperbolic tangent	$\arctanh c$	زظاز س
Arc hyperbolic cotangent	$\arccoth c$	زظتاز س
Arc hyperbolic secant	$\arcsech c$	زقاز س
Arc hyperbolic cosecant	$\arccsch c$	زقتاز س
Logarithm	$\lg c$	لو س
Exponent	$\exp c$	قهس

Table 3: Usual functions

The command \root{*expr1*} \of {*expr2*} gives the *expr1* root of *expr2*.

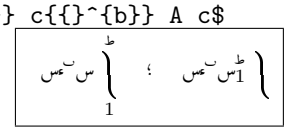
```
$ {\root{3b} \of
 {2+{\frac{b*9}{c}}}}$
```

$$\sqrt[3]{2 + \frac{ب*9}{س}}$$

4.14 Integral

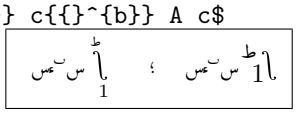
The command \lint_{*expr1*}^{*expr2*} gives the integral from *expr1* to *expr2* using the reversed symbol ⨕.

```
${\lint_{1}^{T}} c{{}^{b}} A c ;
 {\lint\limits_{1}^{T}} c{{}^{b}} A c$
```



The command `\sint_{`*expr1*`}^{`*expr2*`}` gives the integral from *expr1* to *expr2* using the reversed symbol ⨑.

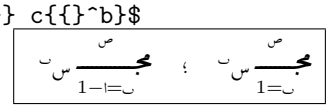
```
${\sint_{1}^{T}} c{{}^{b}} A c ;
 {\sint\limits_{1}^{T}} c{{}^{b}} A c$
```



4.15 Sum

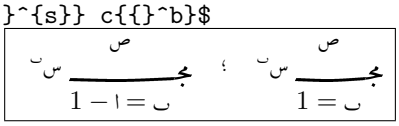
The command `\lsum_{`*expr1*`}^{`*expr2*`}` produces the sum from *expr1* to *expr2* using the conventional Arabic symbol ⅀.

```
${\lsum_{b=1}^{s}} c{{}^b} ;
 {\lsum_{b=a-1}^{s}} c{{}^b}$
```



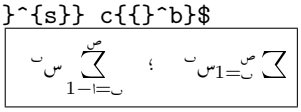
The command `\csum_{`*expr1*`}^{`*expr2*`}` produces the sum from *expr1* to *expr2* using the conventional Arabic curved symbol ⅀. This command needs the CurExt application.

```
${\csum_{b=1}^{s}} c{{}^b} ;
 {\csum_{b=a-1}^{s}} c{{}^b}$
```



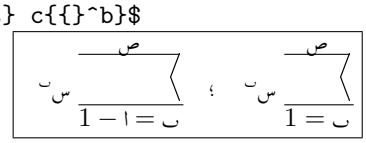
The command `\ssum_{`*expr1*`}^{`*expr2*`}` produces the sum from *expr1* to *expr2* with the inverted symbol ⅀.

```
${\ssum_{b=1}^{s}} c{{}^b} ;
 {\ssum\limits_{b=a-1}^{s}} c{{}^b}$
```



The command `\ssum_{`*expr1*`}^{`*expr2*`}` produces the sum from *expr1* to *expr2* using the inverted big symbol as shown (the broken corner in this symbol is a known flaw to be repaired in a future version).

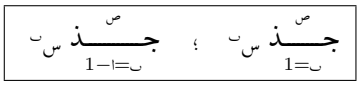
```
${\gsum_{b=1}^{s}} c{{}^b} ;
 {\gsum_{b=a-1}^s} c{{}^b}$
```



4.16 Product

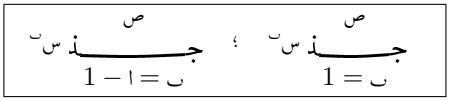
The same can be done with the product symbol of product. The command `\lprod_{`*expr1*`}^{`*expr2*`}` gives the product from *expr1* to *expr2* using the conventional Arabic symbol جـــذ.

```
${\lprod_{b=1}^{s}} c{{}^b} ;
 {\lprod_{b=a-1}^{s}} c{{}^b}$
```



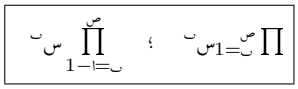
The command `\cprod_{`*expr1*`}^{`*expr2*`}` gives the product from *expr1* to *expr2* using the conventional Arabic curved symbol جـــذ. This command needs the CurExt application.

```
${\cprod_{b=1}^{s}} c{{}^b} ;
 {\cprod_{b=a-1}^{s}} c{{}^b}$
```



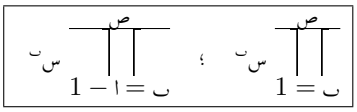
The command `\sprod_{`*expr1*`}^{`*expr2*`}` gives the product from *expr1* to *expr2* using the symbol $\prod$.

```
${\sprod_{b=1}^{s}} c{{}^b} ;
 {\sprod\limits_{b=a-1}^{s}} c{{}^b}$
```



The command `\gprod_{`*expr1*`}^{`*expr2*`}` gives the product from *expr1* to *expr2* using the big symbol ⨘ (see remarks about the big summation symbol).

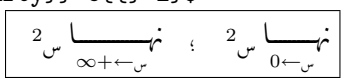
```
${\gprod_{b=1}^{s}} c{{}^b} ;
 {\gprod_{b=a-1}^s} c{{}^b}$
```



4.17 Limit

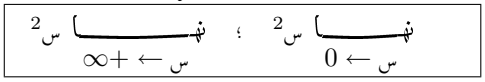
The command `\lim_{`*expr1*`} \to {`*expr2*`}` gives the limit when *expr1* tends to *expr2* using the conventional Arabic symbol نهـــا.

```
${\lim_{c \to 0}} c{{}^2} ;
 {\lim_{c \to +\infty}} c{{}^2}$
```



The command `\clim_{`*expr1*`} \to {`*expr2*`}` gives the limit when *expr1* tends to *expr2* using the conventional Arabic curved symbol نهـــا. This command needs the CurExt application.

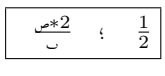
```
${\clim_{c \to 0}} c{{}^2} ;
 {\clim_{c \to +\infty}} c{{}^2}$
```



4.18 Fraction

The command `\frac{`*expr1*`}{`*expr2*`}` gives a fraction with *expr1* as numerator and *expr2* as denominator.

```
${\frac{1}{2}} ;
 {\frac{2*s}{b}}$
```



4.19 Equation numbering

Equations can be numbered or labeled at the right or the left in display math mode.

```
$$2c{{}\sp 2} - 3c + 5 = 0 \leqno (3a)$$
```

(٣) $0 = 5 + س3 - {}^{2}س2$

```
$$2mc{{}^2} - 3c + 5 = 0 \eqno (3b)$$
```

$$0 = 5 + س3 - {}^{2}مس2 \qquad (ب3)$$

4.20 Matrix

The command `\matrix{`*array*`}` generates a matrix. The element's *array* flows from right to left.

The brackets are rendered linear by `\left(` and/or `\right)` or curvilinear by `\parentheses` as well as `\openparentheses`/`\closeparentheses`. In the last case, these commands need the CurExt application.

```
$m = {\left(}
 {\matrix{
 {1}&{2s}&{3}&{+5}\cr
 {1}&{2s}&{3}&{+5}\cr
 {4a}&{5}&{6}&{-2}\cr
 {4a}&{5}&{6}&{-2}\cr
 }} {\right)}$
```

$$\begin{pmatrix} 5+ & 3 & ص2 & 1 \\ 5+ & 3 & ص2 & 1 \\ 2- & 6 & 5 & ١4 \\ 2- & 6 & 5 & ١4 \end{pmatrix} = م$$

```
$m = {\parentheses{
 {\matrix{
 {1}&{2s}&{3}&{+5}\cr
 {1}&{2s}&{3}&{+5}\cr
 {4a}&{5}&{6}&{-2}\cr
 {4a}&{5}&{6}&{-2}\cr}}}$
```

$$\begin{pmatrix} 5+ & 3 & ص2 & 1 \\ 5+ & 3 & ص2 & 1 \\ 2- & 6 & 5 & ١4 \\ 2- & 6 & 5 & ١4 \end{pmatrix} = م$$

4.21 System of equations

The command `\system{`*system_of_equations*`}` generates a system of equations with a big closing brace on the right, as in Latin notation — but that is an opening brace in Arabic notation.

```
$$ {\system{
 {5c - 4s + 6} & { = 8s} \cr
 {5c - 7s + 9} & { = 3} \cr
 {5c - 7s + 9T} & { = {12}c} \cr }}$$
```

$$\left.\begin{array}{rl} 8s = & 6 + 4ص - 5س \\ 3 = & 9 + 7ص - 5س \\ 12س = & 9ط + 7ص - 5س \end{array}\right\}$$

and there is also:

```
$${\left\{}
{\offinterlineskip \openup1\jot
 \tabskip=0pt plus1fil
 \vcenter{\halign{\tabskip=0pt
 & \hfil$#${\ }& \hfil$#${\ }\cr
 \amrl{&{3c}&+&{5s}&&& =&8}\cr
 \amrl{&&-&{6s}&-&{4m}& =&9}\cr
 \amrl{-&{5c}&&&+&{6m}& =&2}\cr}}}
 {\right.}\leqno (1b)$$
```

(ب1) $\left.\begin{array}{rl} 8 = & 3س + 5ص \\ 9 = & 4م - 6ص \\ 2 = & 6م + 5س - \end{array}\right\}$

4.22 Array

The system can also compose tables, including the possibility of combining lines and columns.

```
$${{{{\vbox{\tabskip=0pt
\offinterlineskip\def\tablerule{\noalign
{\hrule}}\halign{\strut#&\vrule#\tabskip
=1.5em plus 2.5em&\hfil#&\vrule#&\hfil#
\hfil&\vrule#&\hfil#&\vrule#\tabskip=0pt
\cr\tablerule\amrl{&{\multispan5\hfil
{\arhbox{almbyan}}\hfil}&&}\cr\tablerule
\amrl{&&&&{\omit\hidewidth{\arhbox
{alAYHdaFyat}}\hidewidth}&&&}\cr\amrl{&
{\raisebox{2ex}[0cm]{\arhbox{alnqT}}}&&
{\omit\hidewidth{\arhbox{alsynyQ}}
\hidewidth}&&{\omit\hidewidth{\arhbox
{alSadyQ}}\hidewidth}&&}\cr\tablerule
\amrl{&$a$&&&$5$&&&$7$&&&}\cr\tablerule
\amrl{&$b$&&&$8$&&&$9$&&&}\cr\tablerule
\noalign{\smallskip}}}}}}}$$
```

المبيان		
الإحداثيات		النقط
الصادية	السينية	
7	5	ا
9	8	ب

4.23 Size variation

Some symbols, as well as superscripts, subscripts and delimiters, vary their size according to nesting.

```
$\sint , \ssum , \sprod$
```

$\int_{ا} \sum_{ب} \prod_{ج}$

```
$$\sint , \ssum , \sprod$$
```

$$\int_{ا} \sum_{ب} \prod_{ج}$$

```
$b{\sp{b{\sp{b{\sp{b}}}}}}$
```

$ب^{ب^{ب^{ب}}}$

```
$b{\sb{b{\sb{b{\sb{b}}}}}}$
```

$ب_{ب_{ب_{ب}}}$

```
${\Bigg(}{\bigg(}{\Big(}
 {\big(}( ){\big)}
 {\Big)}{\bigg)}{\Bigg)}$
```

$$\Big(\Big(\big((()\big)\Big)\Big)$$

```
${\left(}{\left[}{\Bigg(}{\left[}{\bigg(}
 {\left[}{\Big(}{\left[}{\big(}{\left[}()
 {\right]}{\big)}{\right]}{\Big)}{\right]}
 {\bigg)}{\right]}{\Bigg)}{\right]}{\right)
 }$
```

$$\left(\left[\left(\left[\left(\left[\left(\left[\left[()\right]\right]\right)\right]\right)\right]\right)\right)$$

4.24 Translation

The system can translate mathematical expressions and link-words from Arabic to French or English and vice versa. The user can get different results from the following mathematical expression `\expression` depending on the environment, which is easily specified:

```
$$d(c) = {\cases{
 {{\ssum\limits_{b=1}^{s}} c{{}^b}} &
  { {\amhbox{AYZa kan} } c < 0} \cr
 {{\sint\limits_{1}^{s}} c{{}^b}}   &
  { {\amhbox{AYZa kan} } c > 0} \cr
 {{\sin} \pi} & {{\amhbox{Gyr Zlk}}} \cr
 }}$$
```

`\arabmath \artoar \expression`

$$d(س) = \begin{cases} \sum\limits_{ب=1}^{ص} س_{ب} & \text{إذا كان } س < 0 \\ \int\limits_{1}^{ص} س_{ب} & \text{إذا كان } س > 0 \\ \text{جا } \pi & \text{غير ذلك} \end{cases}$$

Figure 1: Egyptian Arabic

`\latinmath \artoar \expression`

$$d(c) = \begin{cases} \sum\limits_{b=1}^{s} c^b & \text{إذا كان } c < 0 \\ \int\limits_{1}^{s} c^b & \text{إذا كان } c > 0 \\ \sin\pi & \text{غير ذلك} \end{cases}$$

Figure 2: Moroccan Arabic

`\latinmath \artofr \expression`

$$d(c) = \begin{cases} \sum\limits_{b=1}^{s} c^b & \text{si } c < 0 \\ \int\limits_{1}^{s} c^b & \text{si } c > 0 \\ \sin\pi & \text{sinon} \end{cases}$$

Figure 3: French

`\latinmath \artoen \expression`

$$d(c) = \begin{cases} \sum\limits_{b=1}^{s} c^b & \text{if } c < 0 \\ \int\limits_{1}^{s} c^b & \text{if } c > 0 \\ \sin\pi & \text{otherwise} \end{cases}$$

Figure 4: English

4.25 Miscellaneous

The command `\arhbox{`*string*`}` introduces the Arabic string *string* in an expression, `\time` the current time, `\day` the current day, `\month` the current month, `\year` the current year.

`$ {\arhbox{mFal}}$` → مثال

`$\time , \day , \month , \year$` → 2005 ، 6 ، 6 ، 872

The command `\amlwm` lists western Arabic month names.

`$\amlwm$` →

يناير، فبراير، مارس، أبريل، ماي، يونيه،
يوليوز، غشت، شتنبر، أكتوبر، نونبر، دجنبر

The command `\amlem` lists eastern Arabic month names.

`$\amlem$`

كانون الثاني، شباط، آذار، نيسان، أيار، حزيران، تموز،
آب، أيلول، تشرين الأول، تشرين الثاني، كانون الأول

The command `\wtoday` gives the current date consisting of the day, the western Arabic month name and the year. To give the eastern Arabic month names, `\etoday` can be used; `\numtoday` shows a numerical month.

`\warabnum$\wtoday$` → 6 يونيه 2005

`\earabnum$\etoday$` → ٦ حزيران ٢٠٠٥

`\warabnum$\numtoday$` → 2005/6/6

`\earabnum$\numtoday$` → ٢٠٠٥/٦/٦

Notes:

- One should not put mathematical expressions in display mode in the interior of an Arabic environment.
- Any command that requires arguments must be between braces (for example, `{\command{arg_1} ... {arg_n}}`).
- If an expression is made up of only one element with at least one argument, it should be preceded by a space (e.g. `$ {\command{arg}}$`).

5 Conclusion

The RyDArab package can be adapted to different needs and situations. Styles can be designed according to the typographic context. The present transliteration is still hard to use, and many open questions remain about this issue. The user should be aware of the use of braces but should not need to know all the details of the `\amrl` command. In recent

versions, automatic management of spaces among terms works well.

All commands can be renamed. For instance, the command `\amsqrt` can be changed into `\jdr` in order to get a name closer to the Arabic pronunciation of the symbol.

RyDArab has been improved, mainly with respect to the transparency of the command `\amrl` for inversion of writing expressions, and the generalization of the possibility to use the same names of commands as for Latin mathematics. Therefore, the system can provide an automatic translation of mathematical expressions from Arabic to Latin and vice versa.

The system can be used with Ω as well as with ArabTEX. In the near future, the system will be able to provide a multilingual scientific e-document by encoding with Unicode, structuring in XML and MathML and the use of a new Arabic mathematical font.

Acknowledgements: Thanks to all the people who have helped to improve the earlier versions. A special thanks to Barbara Beeton, Karl Berry and Steve Peter for proposing this work for reprinting, and their editorial corrections and contributions.

References

[1] Mostafa Banouni, Azzeddine Lazrek et Khalid Sami. Une translittération arabe/roman pour un e-document. In *5e Colloque International sur le Document Électronique*, pages 123–137, Conférence Fédérative sur le Document, Hammamet, Tunisi, 2002. `http://www.ucam.ac.ma/fssm/rydarab/doc/communic/cide5.pdf`.

[2] Yannis Haralambous and John Plaice. Multilingual typesetting with Ω, a case study: Arabic. In *Proceedings of the International Symposium on Multilingual Information Processing*, pages 137–154, Tsukuba, 1997.

[3] Donald Ervin Knuth. *The TEXbook*. Addison-Wesley, 1984.

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

[5] Klaus Lagally. ArabTEX — Typesetting Arabic with vowels and ligatures. In *EuroTEX'92*, pages 153–172, Prague, 1992.

[6] Leslie Lamport. *LATEX: A Document Preparation System*. Addison-Wesley, 1986.

[7] Azzeddine Lazrek. Aspects de la problématique de la confection d'une fonte pour les mathématiques arabes. *Cahiers GUTenberg* 39–40, Le document au XXIe siècle: 51–62, 2001. `http://wwww.ucam.ac.ma/fssm/rydarab/doc/communic/gut39_40.pdf`.

[8] Azzeddine Lazrek. *Vers un système de traitement du document scientifique arabe*. Thèse d'État ès-Science, Université Cadi Ayyad, Marrakech, 2002. `http://www.ucam.ac.ma/fssm/rydarab/doc/communic/these.pdf`.

[9] Azzeddine Lazrek. CurExt, Typesetting variable-sized curved symbols. *TUGboat* 24(3):323–327, 2003. Proceedings of EuroTEX 2003: 14th European TEX Conference, Brest, France. `http://www.ucam.ac.ma/fssm/rydarab/doc/communic/curext.pdf`.

[10] Azzeddine Lazrek and Khalid Sami. Arabic mathematical document processing perspectives. In *Symposium on computer science and software development and its future impact*. Federation of Arab scientific research councils and University AlHadbaA Faculty, Iraq, 2000.

عزالدين لزرق وخالد سامي، آفاق معالجة النصوص الرياضياتية باللغة العربية، مؤتمر المعلوماتية والصناعة البرمجية ودورها المستقبلي، اتحاد مجالس البحث العلمي العربية وكلية الحدباء الجامعة، الموصل، العراق، 2000

[11] Azzeddine Lazrek and Khalid Sami. Towards a system for Arabic mathematical document processing. In *7th symposium of Sciences Arabization, Sciences Arabization in development system*. Egyptian Society for Arabizing Science and Ayn shams University, Egypt, 2000.

عزالدين لزرق وخالد سامي، نظام مبتكر لمعالجة النصوص الرياضياتية العربية، المؤتمر السابع لتعريب العلوم، تعريب العلوم في منظومة التنمية القومية، الجمعية المصرية لتعريب العلوم وجامعة عين شمس، القاهرة، مصر، 2001

◇ Azzeddine Lazrek
Department of Computer Science,
Faculty of Science,
University Cadi Ayyad, P. O. Box 2390,
Marrakech, Morocco
Phone: +212 44 43 46 49
Fax: +212 44 43 74 09
`lazrek@ucam.ac.ma`
`http://www.ucam.ac.ma/fssm/rydarab`