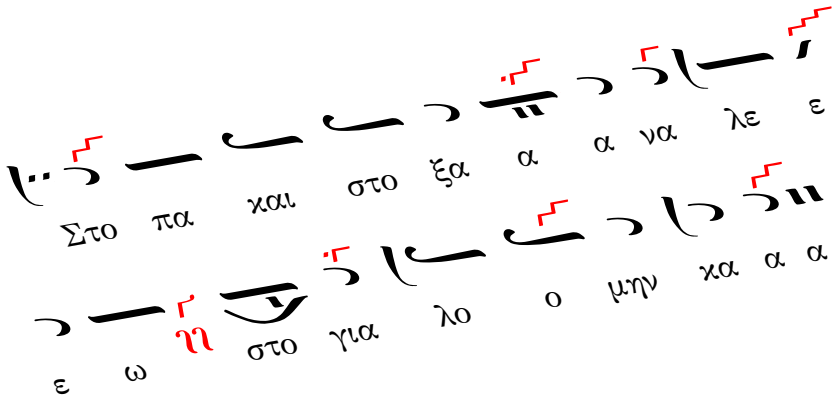
Documentation for the byzantinemusic.sty package.

Charalampos Milt. Cornaros*

2016/04/15



Abstract

The byzantinemusic package gives us the ability to directly type Byzantine Ecclesiastical musical and traditional Panhellenic and Eastern folk music using the XHLATEX machine and BZ-fonts font family.

Contents

 Questions, commentary and bug reports to kornaros@aegean.gr please.

1 Introduction

Byzantine music(BM) is the official music of the Greek Orthodox Church for its liturgical needs and as such is used for recording traditional music in Greece and Cyprus. The problem arises when typing a musical text of BM. Although wonderful fonts are already available, there is no free nor simple tool which can be used to type its symbols appropriately. The reason is simple. There are thousands of musical symbol combinations and therefore it is almost impossible to place them properly, in relation to each other, all the symbols of any combination, in the right order, distance or size. The LaTeX engine with the command \put enables us to place our symbols with absolute accuracy at any point on the page without the need for further automations. This command, with the cooperation of the impressive package \stackengine recently written by Mr. Steven B. Segletes (see `ftp.dante.de/tex-archive/macros/latex/contrib/stackengine/stackengine.pdf`), for easy vertical placement of objects, has enabled us to realize our dream i.e. to make a handy package for typing BM using the XeLaTeX engine.

With this package, every musical symbol or combination of musical symbols is a simple and memorable command (automation). After one gives a straightforward name to each basic musical symbol then one could start the construction of complex combinations using some elementary commands and allowing more complex names for the new structures as in chemistry with regard to atoms and molecules. For example, the symbol of ίσον(isson) ( ) has the name \iso. The γοργόν(gorgon) is an auxiliary musical symbol ( ) with the name \g. Using the letters g and iso, we constructed the command \giso ( ) where the ίσον is below the γοργόν. Of course, it is possible to forget some useful combinations of musical symbols, yet we believe that with each new version, the package will be increasingly perfected.

It is my opinion, that we could not have completed the construction of our package without the existence of the beautiful font family

BYZANTINA 1.1 (http://www.melodima.gr/index.php?pg=downloads&lg=gr). Our package does not correct (for now) spelling mistakes that someone will likely make violating the spelling rules of BM. However it provides the opportunity for easy manual correction of those errors without the need of any extra automation. It also offers many other possibilities, such as easier typing, as well as an an interchange of musical documents among the users of the package and the construction of a large electronic base with BM texts. In addition, it offers the possibility of notating all the instruments of a traditional folk orchestra and the works of musicians from countries that have true roots and tradition in BM. In Eastern Europe, the staff is being used to note down (with the exception of Greece and Cyprus). Generally, with today's options of TEX offsprings any automation or other feature that may already exist in a notation program for BM could be achieved with our package. This can be done in a professional way and totally free!

2 Basic commands

Each simple symbol of Byzantine music corresponds to a command with a short name, usually up to three letters. Simple symbols are divided into two categories: main and auxiliary. Auxiliary symbols usually have at the right of their names some additional indications marked with capital letters. The most common signs are the M, D, A, P, K and DD. These are always assigned in relation to the normal position of the auxiliary symbol, when it is placed over or under the main symbol \iso of the BZ-fonts family, exactly as designed by the manufacturer of BZ-fonts. The M, for example, means that its position is just below the *ίσον*. The K sign is in line with the M but in a lower position. The D stands for the right side of the *ίσον* and the A for the left. The P indicates that it is located far above the center of the *ίσον*, while the DD means far to the right. The construction of a composite combination usually starts from a main musical symbol(i.e. command) and places one or more auxiliary symbols to the right.

Example

Within the font BZByzantina.otf there are many forms of the auxiliary symbol \g. The positions relative to the \iso are: \gPA ⇒  , \gPD ⇒  , \gMD ⇒  , \gDD ⇒  , \gM ⇒  , \gKD ⇒  , \gK ⇒  , \gD ⇒  , \gMD ⇒  and \g ⇒  .

The auxiliary symbols ending in D e.g. KD, PD are commonly used in the combination of medium width BM symbols, such as \apo (apostrophe), whereas auxiliary symbols ending in DD are used with short main symbols, such as \ypo (ypporoe). Perhaps, it is not necessary to be familiar with all these signs. When we make a complex combination from simpler symbols, the new name will rarely have all these additional indicators(with a few exceptions). Each musical combination is read out and typed down from *left to right and top to bottom*. With this simple mnemonic rule all (composite) names of the new combinations actually have a very simple construction.

Example

Instead of writing \isogM we have to write \isog to get  . If we change the order of g and iso we will take the \giso .

We have built many such new commands to combine simple symbols. If we have forgotten a combination, then we can define it with a single name in the preface (preamble) of our TEX file using the command **\newcommand** of LATEX and some commands like those described below. By the way, we should mention for those who do not know the LATEX programming language, that LATEX is completely different from ordinary editors. Rather than pressing buttons e.g. to change the font

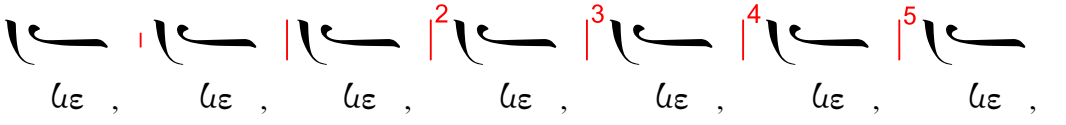
of the text, we use a specific command to achieve this. Commands are much better than buttons! LaTeX does not attempt to alter the colors, position or function of any of the commands given. They give us complete control over our work and can be combined for incredible automation and typographic perfection! To make possible the easier handling of our \byzantinemusic package for anyone who might decide to use it, without having sufficient knowledge of XeLaTeX, we constructed additional useful commands described below. All these commands are taking some arguments in square brackets [and] or braces {}. Each command has some simple syntax rules that we must respect. If the argument is mandatory then it must be put in braces, otherwise in brackets [ ]. E.g. we will describe the basic command of BM that takes a syllable along with the corresponding musical notes, that are exactly above it. This is the **\th**. It contains only two arguments and has the syntax \th{musical symbols}[syllable]. We used { and } to enclose the musical symbols and the [ ] brackets for the syllable. We constructed it in this way, because a musical note may exist in a staff without a syllable, but the opposite does not occur. There is no syllable without a note (or multiple notes) above it! Therefore, we chose the { } pair for the mandatory arguments and the brackets for the optional! This structural rule applies to all commands we built! Generally some knowledge of LaTeX language is needed before trying to write BM in order to avoid any disappointment!

As already mentioned, the basic musical symbols placement command of BM is \th{ }[]. th is derived from the letter Θ of the greek word Θέσε i.e. put, insert characters in the current line. Within the brackets [] we usually insert the syllables of the text that follow the musical line above it. We can leave it empty though, because it is not necessary. Within the pair of {} we place what we usually put over the syllables like notes and isokratema above notes.

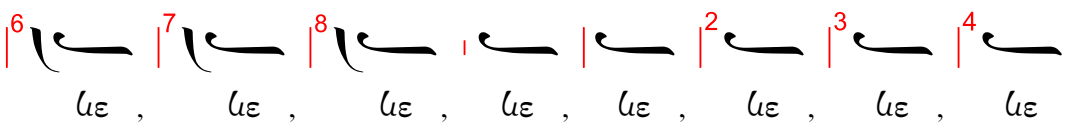
With the help of \th we also produced auxiliary commands \barth, \lbarth, \Lbarth, \LIIbarth, \LIIIbarth, \LIVbarth, \LVbarth, \LVIbarth, \LVIIbarth, \LVIIIbarth, \lth, \Lth , \LIIth, \LIIIth , \LIVth, \LVth, \LVIth, \LVIIth. \LVIIIth, \LLth, \llth, \Lyfeth, \lyfeth, \synl, \synlyfe and \syn (the last three for the construction of the συνεχές ελαφρόν musical symbols).

Example:

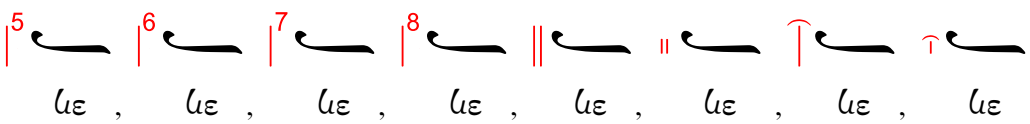
With \th{\iso}[\ne ε] we achieve ιε . (Using the command \ne we achieve ι.) Similarly, using the other commands we get respectively:



ιε , ιε , ιε , ιε , ιε , ιε , ιε ,



ιε , ιε , ιε , ιε , ιε , ιε , ιε , ιε ,



ιε , ιε , ιε , ιε , ιε , ιε , ιε , ιε ,

ιε , ιε , and ιε .

(The last result is typed as \syn{\ela}[\ne ε])

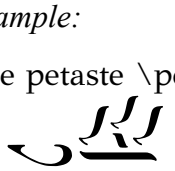
The **\bm** is the most useful command for placing a character, multiple characters or generally any composite object at a specific point on the page. It gives us the opportunity to feel like typographers that could place our symbols (metallic movable types) on any point of the "matrix", as it was used in the good old days ... Its syntax is

\bm[object1, horizontal - displacement1 , vertical-displacement1][object2, horizontal-displacement2 , vertical displacement2] etc...

The horizontal and vertical displacements are determined by small decimals related to the length and height of the ίσον(\iso) symbol of BM. The zero value (0) in any given direction does not move the object in that direction.

Example:

The petaste \pet( ) from the left of \ypsypskypsoli( )



⇒ and below

\bm[\pet,0.12,-0.28] \ypsypskypsoli ⇒

The number 0.12 is positive, which means a horizontal movement to the right while -0.28 is negative which means placement vertically downwards. All in all, petaste was moved downwards and to the right of \ypsypskypsoli as we would do by hand if we wanted to place the petaste under it.

Another example: the  is not correctly placed because the δίεση (=sharp) \dMD ( ) is overlaying the έτερον \eteKDD ( ). The combination is written more correctly with proper movement of the δίεση upwards and to the left of its normal position.

\gapot\bm[\dMD,-0.05,0.1]\eteKDD ⇒ 

3 Stacking objects

The \bm command requires a measure of test and trial to find the proper values in order to place our objects in the desired position. If we are interested only in stacking an object above or below another then it is preferable to use some other LaTeX commands or commands of the stackengine package. For this reason we created some additional auxiliary commands for the vertical positioning of isokratima, martyries and other symbols.

In order to place isokratema(ισοκράτημα) over notes we use the command **\s**. The name s is derived from the first letter of the \stackengine command to remind us of the use of this command in the construction

of \s. The **\marts** command is similar to \s but is used in the construction of martyries(μαρτυρίες) and finally, the **\xrs** command is used when we want to introduce time signatures in our musical texts at various appropriate heights. To construct rhythmic patterns(meters) with \arsi (ι), \thesi (ο) and \makra (.), we use the command **\stackon** of the stackengine package. The \stackon command is a simplification of the command \stackengine. In \stackon the first argument is optional and as such is included in []. Two mandatory arguments follow and they are included in {}. In the optional argument we insert the intermediate distance of the stacked objects using some proper measurement unit lengths like 0.2em or 10pt, while for the next two arguments we put the bottom object (located at 0 level) and then the object above it (located on the first level). The syntax of \s, \marts, and \xrs is the same. There are only two optional arguments.

Example:

\s[(K)][\iso] ⇒

\marts[\padiaa][\dik] ⇒

\xrs[\bbbmchi][\dibm] ⇒

\stackon[0.2em]{\arsh}{\red\makra\makra\makra} ⇒ . The \red command is used when we want to color an object red. Regarding the colors, we will give more details below.

3.1 Building brackets and positioning them

We can properly use the **\bsline** to put braces, brackets, parentheses etc. of variable heights and width directly on the baseline. *Brackets* of variable height are easily constructed with the help of the command **\agkylh**. Let us note the difference:

\agkylh[1]{\{} $\Rightarrow \left\{ \right.$ and \bsline{\agkylh[1]{\{}} $\Rightarrow \left\{ \right.$.

The symbols {and } are used exclusively for the construction of LaTeX commands and should never be inserted in the text. If we want to insert the braces somewhere in the text then we have to use the backslash \ and type \{ or \}. For this reason we used the symbol \{ inside the command \agkylh[...]{ ... }. Moreover, instead of \{ we can alternatively insert some other commands (constructing mathematical figures) or other simple bracket-like shapes like parentheses (,), ‖(=\Vert), ⟩(=\rangle), ⟨(=\langle), \}, \| (but excluding the brackets] and [ ). The \bsline{object} is the abbreviation of the \stackon[0ex]{ }{object} command. The main disadvantage of \agkylh is that it accepts only mathematical shapes of TeX and not something else, such as a letter. All symbols mentioned above are considered mathematical symbols and for this reason can be used as an argument of \agkylh.

The 1 inside the \agkylh command sets the amount of the height of the left parenthesis { (as a percentage of the total height of the musical character \iso, ie. 1 = 100 % of the height of \iso) and can be changed to any integer or decimal value we like.
Instead of \agkylh command there are also **\leftbracket[number]** and **\rightbracket[number]** commands to take the [and] with a variable in height.

Example: \rightbracket[1] $\Rightarrow \left]\right.$

There are also starred versions eg. \agkylh*[number] {figure} and \leftbracket*[number] to position our brackets *horizontally*.

If we want to simultaneously change the thickness and height of our braces we make the **\thickshape** command available with the syntax

\thickshape[! or a thickness unit eg. 10 pt][a number for the enlargement] {any shape}.

The ! is the normal thickness of the shape for this specific enlargement. Otherwise, we must insert a number in some typographic unit of the measurement of the thickness eg. the pt (point). The \thickshape is much more general than \agkylh, because the latter can not accept any

brace-like shape or other shape as its argument, other than those already mentioned, while the \thickshape command accepts everything! Of course, we could integrate into the command \thickshape any other command like the \bsline, so the final result will be set on the (base)line.

Example:

\thickshape[!][2]{[} $\Rightarrow$  and \thickshape*[10pt][1]{)} $\Rightarrow$  and

\thickshape[10pt][1.5]{\bsline{(}} $\Rightarrow$  and \thickshape[!][1]{A} $\Rightarrow$

3.2 Construction of initial martyries

Martyries(μαρτυρίες) show the starting note of the melody from which we begin to chant. They are not interval signs, so they are not chanted. Martyries are divided into three groups: diatonics, chromatics and enharmónics. We can use the \stackengine command to achieve our own beautiful combinations of initial martyries using any text font.

Eg. we could place the (=\nhdiafM) over Nη with the command

```
\bm[\nhdiafMD,0.4,0.1]\mbox{Nη}
```

or with the command

```
\stackengine{0.5ex}{Nη}{\nhdiafMD}{O}{r}
{F}{\useanchorwidth}{L}
```

to obtain . Also with

```
\stackengine{1.2ex}{π}{λ}{O}{c}{F}{\useanchorwidth}{L}
```

we get $\overset{\lambda}{\pi}$ (the 1.2ex above determines the vertical distance between the letters π and λ. We may reduce or increase the 1.2 value depending on the desired distance). The last example can also be written in a simpler fashion by means of using as an alternative the \bm command or the

\stackon command as follows:
\stackon[0.2ex]{π}{λ} $\Rightarrow \overset{\lambda}{\pi}$

We could also construct various other commands such as the above to build new initial martyries using any font. Our opinion is to use exclusively the BZByzantine.otf font, as well as, the rest of the fonts of the BZ-fonts package, so our musical text has a robust and consistent appearence throughout its length. Please note that most of the complex initial martyries have already been built by us with the help of commands described (see table at the end of the documentation), so there is no need to build a new initial martyria.

3.3 An example of the construction of a tower.

The \stackenginepackage gives us, in addition to the basic \stackengine command and auxiliary command \stackon, many other auxiliary commands, even for the building of an entire "tower" of stacked objects. Without the willingness to expand on the details of this package, we will simply give a vague idea as to how we could write a musical text with multiple musical parallel lines (eg. when we want to build Russian or other Eastern Church polyphonic music or we could use two lines, the above line is with the pre-reformation of 1814 AD writing style while on the other line its the reformed style after 1814 AD). We will use the **\Longstack** command of the above package to construct the tower. Its general form is

\Longstack[r or c or l] {beginning on the top floor, followed by each consecutive floor thereafter, which are shown as enclosed in separate pairs of {}, until finally reaching the "ground " floor, which is not in braces }.

The **r**, **c**, **l** denote right, central and left aligned stack.

With the help of \rbox* (see details below) we can have absolute control of the intermediate distance between floors of the tower.
Eg. using

\Longstack[r]{ \rbox*[4.3]{Ὑψίφωνος} {\rbox*[4]{μονῳδὸς}}
{\rbox*[3] {Ὑψίφωνος }} {\rbox*[2]{Μεσόφωνος}}
{\rbox*[1]{Ὀξύφωνος}} Βαρύφωνος}

we take:

Ὑψίφωνος
μονῳδὸς

Ὑψίφωνος

Μεσόφωνος

Ὀξύφωνος

Βαρύφωνος

We met the above example in the Εκλογή αρμονικών μαθημάτων of the book "New Anthology" of Emm. Vamvoudaki.

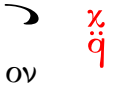
4 Variable distances between characters

There are several ways to distance or bring closer **horizontally** two characters. We constructed three commands for this purpose. The first **\p** separates two characters into a constant distance as the width of the Latin letter i. The second command, **\keno**(from κενό =space) could be used to join or to separate two characters at any distance. Finally, the **\met** (from the initials of the greek word μετακινώ = move) has a similar syntax with \keno. The \met moves the character or object that is on its right in a positive or negative percentage. Each of these commands has its own advantage and disadvantage. The \p is essentially the i, but invisible, so we have the impression of there being an empty space between the two characters. The \p does not get any argument so if a wider gap is desired then we must repeat the command \p \p ... several times. In practice, we usually use the \keno to control the exact space between two musical syllables, except in some rare cases (in which a visual problem arises in the final result). In this situation we have the choice of \p or \met commands or perhaps one of the dozens of the TEX). The \keno uses the command \hspace internally, which is a command for making horizontal gaps. The \met uses the command \kern for the left or right

shift of the character after the \met. The \met and \keno may cause problems when using large negative values because they could move all the characters on their right hand side to the left)

Example:

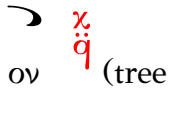
\th{\apo}[ον]\keno\th{\kediam} ⇒

\th{\apo}[ον]\keno[0.5]\th{\kediam} ⇒ 

\th{\apo}[ον]\th{\kediam} ⇒  (the \th commands without \keno and without intermediate space)

\th{\apo}[ον] \th{\kediam} ⇒  (the \th commands without \keno and with an intermediate space)

\th{\apo}[ον]\keno[-0.1]\th{\kediam} ⇒  (negative \keno)

\th{\apo}[ον]\p\p\th{\kediam} ⇒  (tree \p in line)

\iso\met\iso ⇒ (a small displacement)

\iso\met[0.5]\iso ⇒ (a bigger one)

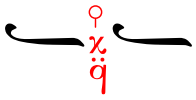
\iso\met[-1.5]\iso\apo ⇒

A problem that may arise with the negative met: the second red iso is shifted too far to the left, affecting the apostrophe, which falls on the first iso!) Using the command \bm the problem can be solved.

\iso\bm[\iso, -1.5]\apo ⇒ The \bm command always moves its objects without affecting the other objects around it!

Example:

♀ (=\padiafD) is positioned on  (=\kekediam) without affecting \iso:

\iso\bm[\padiafD,0.3]\kekediam\iso ⇒ 

For an exact **vertical** movement of our musical characters or combination of characters, as well as any kind of objects, we also created the command **\rbox** with the following syntax

\rbox[+ or - number] {character (to be moved) }.

This is derived from the initial letters r and box of the LaTeX command **\raisebox** (with the help of which the \rbox command is defined). It can be combined with any other commands such as the \s command.

Example:

\s[\rbox[-0.8]{(NH)}][εαν] ⇒ $\overset{\text{(NH)}}{\varepsilon\alpha\nu}$. And

\s[(NH)][oh!] ⇒ oh! (without the \rbox the isokratema (NH) is much higher than the oh! and almost touches the top line!)

\th{\rbox[0.3]{\iso}}[up]⇒ up , and without \rbox: \th{\iso}[up]⇒

up .
Lowering with a negative value \rbox[-1.3]{\iso}⇒

The use of \rbox has a drawback. When we move a character too high or too low it crosses over onto the previous line or the next line. This effect is not always desirable. For this reason we created a command similar to the \rbox. The command **\rbox*** helps us to enlarge the distance between consecutive lines so that the objects do not cross. This is the main reason we have used the \rbox* rather than \rbox for the construction of levels in the example above. Of course, for small displacements it is not preferable to change the distance between consecutive lines. In these cases it is better to use the \rbox.

5 Invisable vertical lines

To construct the commands \th, \agkylh, towers etc. there needs to be some invisible vertical lines (of zero thickness) of a fixed or variable height so that we can control the overall height of the shape or of the stack, regardless of the height of objects stacked. For example, the command **\rul** has been constructed to be used exclusively with the \th command. If we are not satisfied with the total amount of height of \th, then it is possible to change it using the command **\changethheight**, eg. \changethheight[1.7]. We have selected 1.9 as the pre-determined \rul height but with \changethheight we can give any value to it.

With the command **\katheth[number]** we make an invisible vertical line of variable height, which is determined by the number, eg

\katheth[2] ⇒ (in this example we revealed the \katheth[2] just to see the actual height and thus the greater distance we have achieved between the line in which the \katheth is placed and the line above it.)

6 Basic parameters and the change commands

There are some additional parameters and commands that determine the final appearance of our musical text. The **\ypsosisokrathmatos** has a default value of 1.3 (% percentage of the height of \iso symbol) and affects the command \s i.e. it sets the height of isokratema over the musical line. The **\ypsosmartyrias** (with a pre-determined value 0.02 % of the value of \iso) is similar to \ypsosisokrathmatos but affects the command \marts and finally the **\ypsosxronou** (with the pre-determined value 1.02 % of the \iso's height) is used in \xrs. The command **\changemusictextgap** is used to change the gap between a musical note and the syllable located exactly below it. The pre-determined value has been chosen 0.27 (i.e. the gap is 0.27 % of the total height of the \iso symbol in our musical font). If this distance does not suit us then we can decrease or increase it depending on the visual effect we want to achieve. We can compare the distances as given in the following example.

Example:

\changemusictextgap[0.87] \th{\iso}[one] ⇒ one and

\changemusictextgap[0.27] \th{\iso}[two] ⇒ two

Naturally, it is not a good idea to change this value unnecessarily, in order to always have a uniform gap between notes and syllables for the whole song. We could also change the pre-determined values of the parameters mentioned above with similar change commands such

as \changeypsosisokrathmatos, \changeypsosmartyrias and \changeypsosxronou respectively. I should have mentioned that there are many other commands with prefix **\change**.
The **\changetextfont** with this syntax

\changetextfont[font name]

has been naturally designed to change the text-font and uses internally the the \fontspec package. Currently there is no option to change the musical font, because most well-designed musical fonts of BM are not free.

Example:

\changetextfont[GFS Artemisia] ⇒I was written using GFS Artemisia font while \changetextfont[GFS Neohellenic] ⇒ I was written using GFS Neohellenic font!

Of course we can return to the original chosen font i.e. our main text font using the command \normalfont or the \changetextfont[] command with no arguments.
Obviously, fonts must have already been installed either in our computer or in our TEX distribution or otherwise pre-exist in the same folder with the current musical folder, so that the \fontspec package could identify them when the \changetextfont command is used! It is surprising that we can access hundreds, perhaps even thousands of fonts ready for use with XHTEX, without them needing to be installed! Note that the font name must be correctly typed. Never put a space before or after the font name, for example, \changetextfont[Arial] could cause a serious problem! Our fonts must be of .ttf or .oft type.

Another useful and relevant command is **\changetextscale** that gives us the ability to change the font size of the text by a percentage as well as **\changetextsize** to change its size. The **\changetextcolor** changes the text color from black to a wide array of choices.

Example:

\changetextscale[1.5] ⇒ I have grown taller 1.5 times and my color is now \changetextcolor[red] ⇒ red! \changetextscale[]. \changetextsize[10] ⇒ My font size

is only 10pt. To return to the pre-determined size and color i.e. 12pt and black respectively, we could write simply \changetextcolor[] and \changetextsize[]. We can define our own "mycolor" color with the command **\definecolor** and colorize any text!

Example:

\definecolor{mycolor}{RGB}{50,150,220}Then we can use our color to change the text color:
\changetextcolor[mycolor] Do you like this color?

Here we should note that the $\Rightarrow$ is not a character among any font but is a mathematical shape (an arrow defined by the command \Rightarrow) and has been designed exclusively in TEX. There are hundreds of such symbols as the square root ($\sqrt{\ }$) or \Vert command as encountered above. This is the reason that $\Rightarrow$ has not changed its size or color in the above example. The \agkylh, as designed, can accept only mathematical symbols of TEX in its arguments, as already mentioned, and therefore is also not affected with text commands like the \changetextcolor. In order to change the color of a mathematical TEX shape, we should use the command **\color** with syntax \color{a color }, while to change the size of it we could use other appropriate commands such as \thickshape. It should also be noted that when we intend to insert a field (i.e. a string) of mathematical symbols in our text, then we must use some special characters at the beginning and at the end of the field to inform the TEX engine about our intention. The most common character for mathematical fields is the dollar sign $ which is inserted at the beginning and at the end of the field.

Example:

$\color{red} \Rightarrow$ to achieve $\Rightarrow$ and
\thickshape[!][0.5]{$\color{red}\Rightarrow $} to achieve  and
\color{red} \thickshape[!][0.5]{$\Rightarrow $} to achieve
and {\color{red} I am colored red!} to I am colored red! I am not!

7 Text coloring

As can be seen from the above examples, the command \color can be used both to color a mathematical field and any other text! If we enclose the \color inside the braces {} then we limit this specific color to those objects enclosed by these braces. So in the last example, I'm not! remained black because it is outside the braces. If there were no braces around the \color{red} command then everything after this command would be red!

The command \color could take any ordinary color names such as black, red, green, blue, cyan, magenta, yellow and so on. We could use thousands of different colors (for details see `http://texdoc.net/texmf-dist/doc/latex/xcolor/xcolor.pdf`) Due to the great practical usefulness of the colors we have made two more auxiliary commands using the \color. One is the \red and the other is \R with different syntaxes each. The \R is used mostly to color a character, or at most, a short text of characters e.g. some characters inside the \th command.

Example:

{\red a very long text } ⇒ for a very large text whereas \R {\oli } ⇒

. We could change the default color (that is red) of the \R command with the help of the **\changeRcolor** with the syntax \changeRcolor[name of the new color]

Example:

\changeRcolor[mycolor] and \R {\oli } .

Each of the qualitative and time musical characters have a pre-defined color. We could change these default colors according to our own preferences. For example, the color of the characters of the gorgon(γοργόν) family is determined by the command **\gcolor**, which has red as its default value. Red is also given to the following parameters:

\trocolor, **\parcolor**, **\etecolor**, **\argcolor**, **\edocolor**, **\ekscolor**, **\piacolor**, **\ydcolor**, **\lygcolor**, **\metracolor**, **\martcolor**, **\fthcolor**, **\isokrthmacolor**

corresponding to tromikon, paraklitiki, eteron, argon, endofono, ekstrepto, piasma, flat and sharps, lygisma, meters, martyries, fthores and isokratema respectively. The antikenoma, psifiston, aple, klasma and tempo signatures have their color parameters **\antcolor**, **pshcolor**, **aplhcolor**, **klacolor** and **tempocolor** respectively in black, the παῦσεις(rests) have also color parameter **\paycolor** in black and the color parameters for anapnoi **\anacolor**, stayros **\stayroscolor** and corona **\korwnacolor** have blue as their default value. To change these default values we can use their corresponding change commands.

Example:

+

\changestayroscolor[mycolor] . We could come back to the pre-defined color, if it is desired, simply by typing the command with an empty argument eg. \changestayroscolor[].

8 Some other commands

Besides the above \change commands we could also use **\changemusicscale** to re-scale our musical symbols by a percentage and the **\- changemusicsize** command to change the size of the musical symbols. Their default values are 1 and 22 respectively, but we could give them any other value. 1 means 100% i.e. no zooming in or out while 22 means 22 pt (points). We are also concerned with the isokratema font. The **\changeisokrathmasize** command changes the size of the isokratema font.

Example:

\changemusicscale[0.8] ⇒  and

\changemusicsize[40] \changeisokrathmasize[30]⇒ .

To return to the default size of the musical fonts (ie. to 22pt or to 100 %) we simply write \changemusicalsize[] and \changemusicscale[] respectively. 40 in \changemusicsize[40] refers to the musical font size of 40 pt. We never write the pt next to the number in the \changemusicsize eg. \changemusicsize[40pt] because it will create a problem! It should also be noted that when we make these changes in size then the distance \rul (see above) is automatically altered; thus all the visual effects we get with the command \th. There is no need to simultaneously change the \changemusicscale and \changemusicsize. The basic difference is that \changemusicscale will simultaneously change the size of the isokratema font while \changemusicsize will not. If we prefer to use the \changemusicalsize command and at the same time want to make a change in the size of the isokratema font, then we should use the command \changeisokrathmasize in addition to. \changeisokrathmasize.

Changetolerance and changespaceskip commands

Sometimes our musical syllables are not well aligned with the right margin and extend themselves further. In such cases it is advisable to use the auxiliary command **changetolerance** with the syntax

\begin {changetolerance}[number][a number in a typographical unit length]
text ...
\end{changetolerance}

With appropriate changes to the values of the two arguments we can have a very good alignment of our musical text, no matter how big or small \changemusicscale we use.

After we complete the paragraph we should finalize with \end{changetolerance} The commands starting with \begin and ending with the \end{name of the command} are called *environments* in the terminology of LaTeX. We found the above command in the `http://tex.stackexchange.com/questions/84510`. Its syntax needs some attention. We should not omit the pair of brackets [] even if we have empty arguments. For example, the \begin {changetolerance }[2em] will generate an error while \begin {changetolerance }[][2em] will not. In other cases we would like the notes to be as tight as possible so that it takes up less space while at other times we can spread things out. For these situations we created the **changespaceskip** environment. For example, \begin {changespaceskip}[0][][] musical notes... \end {changespaceskip} will put the notes ... to zero (0) apart. We could even put a negative value eg. -0.2 for even greater economy on ... paper! The **changespaceskip** environment, not only arranges the gap (space) between words, but the additional percentage can stretch or shrink. Consequently it has three arguments for space, stretch and shrink. Its syntax is

\begin {changespaceskip}[a number for the space][a number for the stretch][and one for shrink] our text ... \end {changespaceskip}

The *command* (not environment) **\changeletterspace** is similar to changespaceskip and is built with the help of another command of \fontespec package. This command gives very good results when we have some text in capitals (like titles, headers etc.) and we want to spread it out. Below we show the same text in many different ways for the difference to be seen.

Example:

\\\changeletterspace[20]

After we complete the paragraph, we should finalize with \end{changetolerance} The commands starting

with \begin and ending with the \{name of the command} are called *environments* in the terminology of LaTeX. \\
\changeletterspace[]
\begin{changespaceskip}[0][-1][1]

After we complete the paragraph, we should finalize with \end{changetolerance} The commands starting with \begin and ending with the \end{name of the command} are called *environments* in the terminology of LaTeX. \\
\end{changespaceskip}
\begin{changespaceskip}[5][][]\\

After we complete the paragraph, we should finalize with \end {changetolerance } The commands starting with \begin and ending with the \end{name of the command} are called *environments* in the terminology of LaTeX. \\
\end{changespaceskip}
\begin{changespaceskip}[][5][] \\

After we complete the paragraph, we should finalize with \end {changetolerance } The commands starting with \begin and ending with the \end{name of the command} are called *environments* in the terminology of LaTeX. \\
\end{changespaceskip} \begin{changespaceskip}[][][3]\\

After we complete the paragraph, we should finalize with \end{changetolerance} The commands starting with \begin and ending with the \end{name of the command} are called *environments* in the terminology of LaTeX. \\
\end{changespaceskip}
And finally the original text:

After we complete the paragraph, we should finalize with \end {changetolerance } The commands starting with \begin and ending with the \end{name of the command} are called *environments* in the terminology of LaTeX. \\
\end{changespaceskip}
Please note that before we start writing BM we should set the special

environment **\begin{byzantinemusic}** and when we finish our musical text, the **\end{byzantinemusic}**.

8.1 The formatting commands

Sometimes, we do not have a whole range of fonts of a family at our disposal, ready for use. For instance, we may need a specific font that it is not available to us, or its author did not care to design the thick (bold) or slanted version of it. In such cases, there are appropriate commands in the fontspec package to help us. We have built three more auxiliary text formatting commands for the needs of our package. Of course, we can use many other commands of fontspec or another package to get exactly the desired appearance of our letters. The new formatting commands are **\changetextbold**, **\changetextslant** and **\changetextstretch**. The last command has nothing to do with the stretch of empty space mentioned previously nor does the changespaceskip environment. It just deforms our font letters by a percentage.

Example:

We could use the \changetextbold[0.6] **command to make our letters thicker. We can put an even greater number for even better results: \changetextbold[1.5] Using the command \changetextbold[]** we return to the normal thickness. We can also slant our characters with the command \changetextslant[0.3] *and come back to its normal sloping position with the command \changetextslant[]* . Of course, if we like experimentation we could use the \changetextstretch[1.3] command but the use of large values could give \changetextstretch[2] undesirable results! We could simply return to normal by typing \changetextstretch[].

9 Writing polytonic texts

To write polytonic texts more easily, we should insert in our preamble an appropriate package using the command \usepackage {greektonoi } (for information `http://ctan.sharelatex.com/tex-archive/language/greek/greektonoi/greektonoi.pdf`) to introduce accents and breaths in large

polytonic texts in which there are no LaTeX commands in-between, we should use some proper combinations of keyboard letters without the use of any additional commands. However, if we are going to write just a few polytonic words in a (musical) paragraph that contains several LaTeX commands (such as those we have already defined), then we should *definitely* use specific commands of greektonoi package. These commands are designed with letters of the Greek alphabet and are quite memorable. We use letters β (βαρεία, grave accent), ψ (smooth breathing), τ (acute accent), π (circumflex), δ (rough breathing), λ (diaeresis), μ (iota subdcript), μκρ (long vowel) βρχ (short vowel) and many other commands to achieve any (ancient or not) letter or number of the polytonic and monotonic Greek orthographic system. For example, to get ἅ we can simply type \πδA or \δπA. Warning: if our word is comprised of more than one syllable then we should give the XeLaTeX the ability to recognize where our tonal command ends before the next syllable. This is done by double {} braces eg. we write

\ψα{}νατε\τι{}λ\μη or \ψα{}νατείλ\ημ

to get ἀνατείλῃ. We should note here that acute (\τ) does not always coincide with the τόνος(acute) of the monotonic system in some polytonic fonts. This difference will only be visible if we have a well-designed polytonic font in our computer. Finally, if the use of the pair of braces {} between the syllables looks embarrassing then we can put somewhere in the text before we start writing polytonic, the \tildeON command which causes the tilde (tilde) ~ to play the role of double {}. For example, the word ανατείλη is easily converted to polytonic by typing the appropriate command \ψα~νατεί~λ\μη. Due to this particular use of ~ after the \tildeON, if we need to explicitly insert the tilde symbol somewhere, then we could use the command \char plus its hex value: \char"007E. We should insert \tildeOFF immediately after the end of the polytonic text to return to the normal use of the tilde in TeX.

10 Future plans

Our package should be designed in such a way as to encompass all of the ancient symbols of BM, including ancient Greek Music. It would also be very useful to build a database of musical BM texts written in XeLaTeX.

For your consideration, it is necessary to built an OCR tool that converts printed musical text in XeLaTeX in order to create this database quickly. It would also be useful to make a program that could convert the musical text to Midi files. For the latter we have already progressed somewhat with the help of the abcnotation program. Details are on our website http://samosweb.aegean.gr/mathmusical/. Finally, it would be worthwhile if a simple tool of inserting the staff in the byzantinemusic package can be implemented in order to have a fully powerful tool at our disposal for studying Greek music from ancient times. We would be grateful to anyone who could volunteer to help our efforts.

11 Acknowledgments

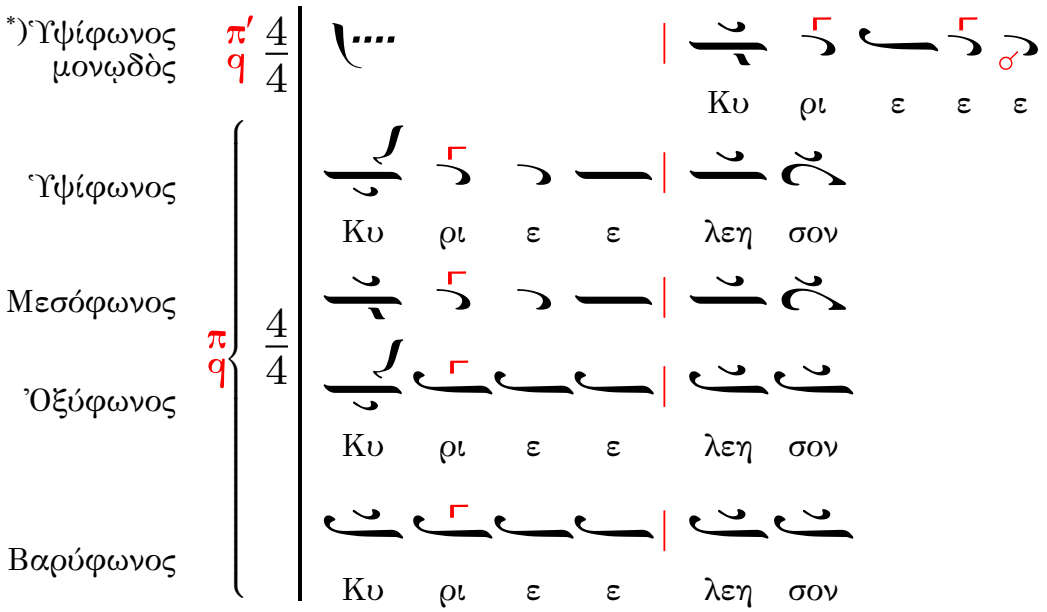
If we did not have the BZ-fonts of Fr. Ephraim of Arizona (https://www.stanthonysmonastery.org/musical/ByzmusicalFonts.html) and Mr. Souldatos (BYZANTINA 1.1) then we could not have designed our package because we could not have created from scratch all the musical symbols. I thank them warmly. I would also like to thank the Assistant Professor Tsolomitis Antonios of our Mathematics Department for his valuable help on the uploading of the package contents at TUG and other issues related to fonts, and Mr. Vamvaka, John and Mr. Kotopouli, Panagiotis for creating BYZLATEX the first BM package in LaTeX (https://www.tug.org/TUGboat/tb26-3/tb84vamvakas.pdf). I couldn't go without mentioning the Samian Professor of BM Mr. Angelinaras, George who made us the manuscript available (it is the second example below). There is a Ph.D candidate at the Department of Inf. and Comm. Systems Engineering of the University of the Aegean, Greece, I must thank deeply Zessis Tsiatsikas for his invaluable assistance, with his Java program "BMeditor"(for details, please refer to the README file of this program). Needless to say, I must also thank my family for the patience that they showed me at the endless hours of package design. In closing with my expressions of gratitude, I would like to dedicate this body of work to the memory of my mother Sophia.

12 Two musical examples

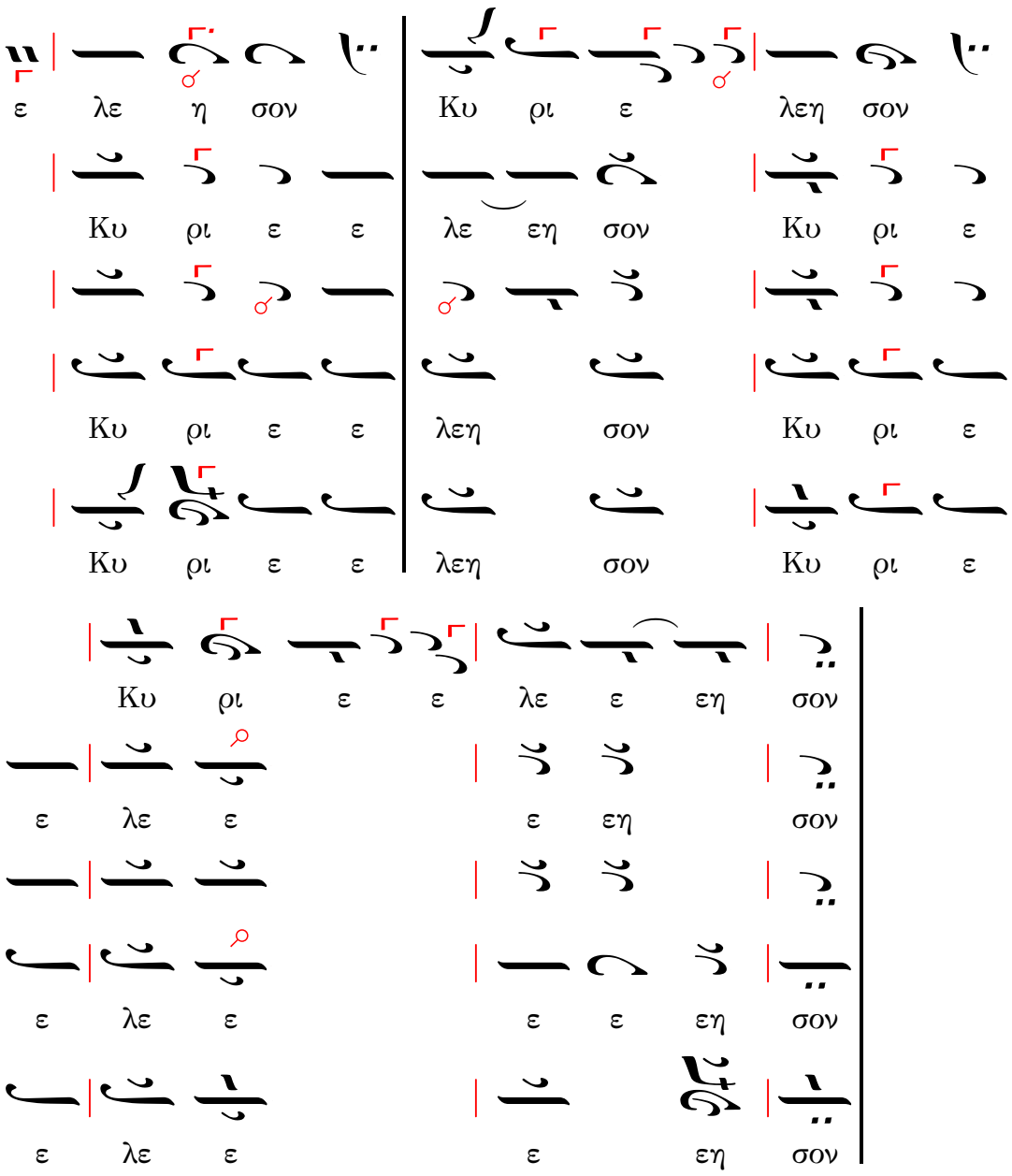
We will give two simple examples using several commands from our package. The first example is the “Lord Have mercy ” for 5 voices by Emmanuel Vamvoudaki. This is a translation and adaptation of Vamvoudakis from the original Russian song and it is in the book "New Anthology" by Emmanuel Vamvoudaki - Samos 1921. We used the command **\Longstack[r]** to create the command **\pyrgos** for the construction of floors. This command is not in the byzantinemusic package, but in the preamble of byzantinemusic-en.tex. Each floor usually gets a single note and its syllables. We did this so that each tower is not very wide and consequently our musical text can be sufficiently well aligned in TEX without going beyond the right margin. We used the command \agkylh[6.5] {\{} to insert the large brace at the beginning of the musical text. Also for the construction of the small hyphens, we used the command \agkylh* (eg \agkylh*[0.5] {) }), although we could also use the commands **\yfa**, **\yfk** (see list of symbols) that are made from BZ-fonts without using any mathematical commands. Certainly \agkylh* is the recommended command in the case of a very large hyphen because it gives a uniform thickness throughout the length of the hyphen (regardless of its size and the font used). The musical measure $\frac{4}{4}$ is written using the “mathematical ” command \frac that has the capacity to make fractions: $ \frac{4}{4 }$. We zoomed this fraction a bit with the command \thickshape[!][1]. If we wanted to, we could make a different construction of the tower that would gives us the ability to put many musical notes at the same level of the stack. This command is called **\pyrgosh**. In this way, the notes of each voice is more condensed but we must be careful of how many notes insert at each level of the stack, because otherwise there is a chance of stretching beyond the margin! Below, we wrote only the first measure of the hymn using this tower to show the difference.

The second example is an old traditional and well-known song from the Lydia Province of the unforgettable Greek Asia Minor. It uses the changespaceskip environment with zero width between the musical words throughout its extent ie. it starts with \begin {changespaceskip }[0][][]. In the title of the piece we used each of the \changetextbold, \changeletterspace, \changetextslant commands for better appearance.

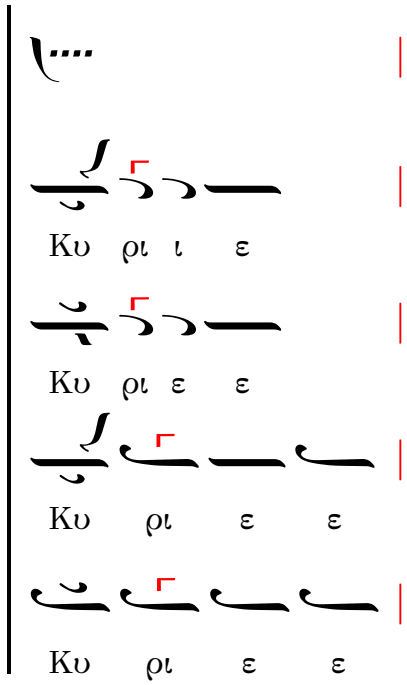
Moreover, we used \changetextscale[1.5] command to increase letters' size by a percentage and soon came back to our original settings, once we had finished the construction of the song's title, using the command \changetextscale[1] (or simply \changetextscale[]). We easily centered the title with the command **\kentro**. The remaining verses after the track are placed using the environment **\begin {verse } ... \end {verse }** of LaTeX.



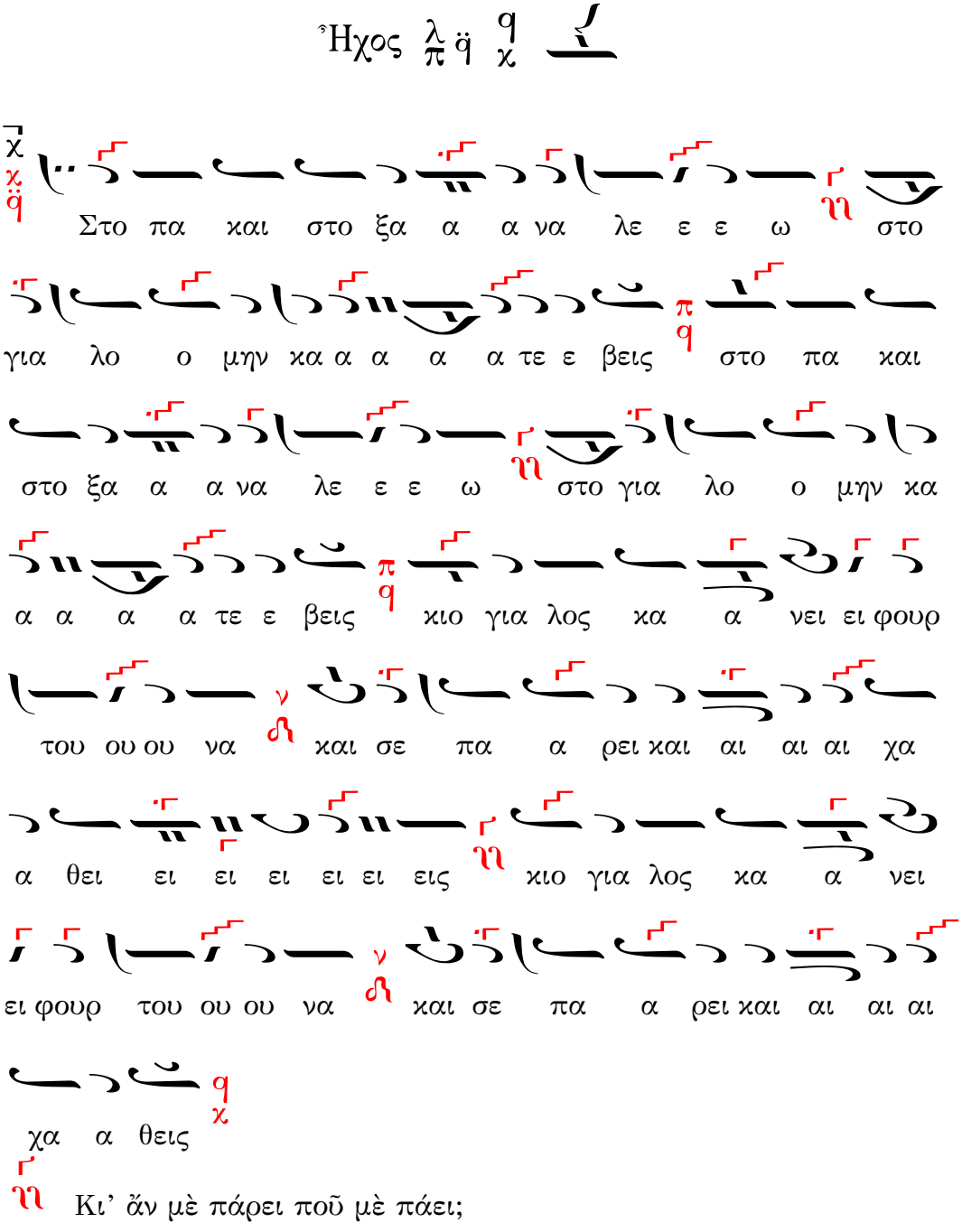
*)Τὰ τρία πρῶτα μέρη θέλουσιν ἐκτελεῖ παῖδες



The following is a different construction of the tower using the command **\pyrgosh** (each level contains more than one notes).


Κυ ρι ι ε
Κυ ρι ε ε
Κυ ρι ε ε
Κυ ρι ε ε

Στό ’πα καὶ στὸ ξαναλέω
(Ἀντικρυστὸ Μικρᾶς Ἀσίας)



– Κάτω στὰ βαθιὰ νερά!
Κάνω τὸ κορμὶ μου βάρκα,
τὰ χεράκια μου κουπιά,
τὸ μαντήλι μου πανάκι,
μπαίνω βγαίνω στὴ στεριά!
Στό 'πα καὶ στὸ ξαναλέω
μὴ μοῦ γράφεις γράμματα
γιατὶ γράμματα δέν ξέρω
καὶ μὲ πιάνουν κλάματα.

13 List of musical commands

The following list contains most of the symbols built. Matches of the commands with their corresponding musical symbols are obvious. So, for example, the command iso stands for ίσον, the oxe for οξεία, the oli for ολίγον, the pet for πεταστή, k for κέντημα, the xk is k right bellow ίσον, the yps (or ypsD) for υψηλή which is positioned to the right(D) of ίσον, the yps (or better ypsA) for υψηλή to the left(A) of ίσον, the kk for κεντήματα, the g for the γοργόν, the gg for δίγοργον, ggg for τρίγοργον etc., The t for απλή(aple), tt for double απλή(diple), ttt for τριπλή(triple), p for dotted g, gg, ggg etc. The synela for συνεχές ελαφρόν, the apo for the απόστροφος(apostrophe), the xapo for apostrophe below oli, the ela for ελαφρόν, the ypo for υπορροή and xam for χαμηλή. In some complex musicalical ascending combinations we also used the word anw (=upwars) and for some descending combinations the word kat (= downwards) plus a Latin number (eg. Xanw = ascent 10 voices) to facilitate typing them. The p sometimes comes from the left of g as pg and sometimes from the right as gp. To align the syllables to the right of συνεχές ελαφρόν we had to create the command \syn with exactly the same syntax as the command th.

Example:

\th{\synela}[\ne e] ⇒  and

\syn{\ela }[\ne e] ⇒ 

13.1 Symbols for ascending and descending

Command	Result	Command	Result
Symbols for ascending			
\isopet		\pet	
\olipet		\kpet	
\ypspet and \ypsDpet		\iso	
\oli		\olixk	
\koli		\ypsoli and \ypsDoli	

\olixkantt and \olixkantapl 	\oxe
\k 	\gkkoli 
\olikkantt 	\kk 
\olikkD 	\kkoli 
\ypskkoli 	\ypsAoli and \Vanwoli 
\VIanwpet and \kypspet 	\VIIanwpet and \ypskpet 
\VIIIanwpet and \ypsypspet 	\kkypsoli 
\Vanwoli and \ypsAoli 	\VIanwoli and \kypsoli 
\VIIIanwoli and \ypsypsoli 	\IXanwpet and \ypskkypspet 

\Xanwpet and \ypskypspet 	\XIanwpet and \ypsypskpet 
\XIIanwpet and \ypsypsypspet 	\XIIIanwpet and \ypsypskkypspet 
\XIVanwpet and \ypsypskypspet 	\IXanwoli and \ypskkypsoli 
\Xanwoli and \ypskypsoli 	\XIanwoli and \ypsypskoli 
\XIIanwoli and \ypsypsypsoli 	\XIIIanwoli and \ypsypskkypsoli 
\XIVanwoli and \ypsypskypsoli 	\XVanwk 
\kkypsolik 	\XVanwpet 
\isokkoli 	\isokkoxe 
\olixkant 	\ggkkoli 

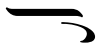
\olik 	\kkolik 
\ypskkolik 	\ypsAoxe 
\VIanwoxe and \kypsoxe 	\VIIanwoxe and \ypskoxe 
\VIIIanwoxe and \ypsypsoxe 	\isooxe 
\oxexk 	\oxek 
\koxe	\ypsoxe and \ypsDoxe 
\ypskkoxe 	\oxexkant 
\ggkkoxe 	\gkkoxe 
\XIanwoxe and \ypsypskoxe 	\XIIanwoxe and \ypsypsypsoxe 

\kkypsoxe 	\oxekkD 
\kkoxe 	\IXanwoxe and \ypskkypsoxe 
\Xanwoxe and \ypskypsoxe 	\isooli 
Symbols for descending	
\isoapo and \isoxapo 	\apokkoli 
\apoapo and \apoxapo 	\synelapet 
\apopet 	\elapet 
\elaapopet 	\gypo
\ypopet 	\synela 
\apo 	\ela 

\elaapo 	\xam 
\ypo 	\apooli 
\ypooli 	\xamoli 
\elaoli 	\elaapooli 
\synelakkoli 	\xamkkoli 
\elakkoli 	\elaapokkoli 
\gpypo 	\pgypo 
\pggypo 	\ggypo 
\pgggypo 	\gggypo 

 \gypokkoli	 \ypokkoli
 \gggypokkoli	 \gggypokkoli
 \gpypokkoli	 \pgypokkoli
 \ggpypokkoli	 \pggypokkoli
 \gggpypokkoli	 \pgggypokkoli
 \xampet	 \Vkatpet and \xamapopet
 \xamapo	 \VIkatpet and \xamelapet
 \xamela	 \xamelaapopet
 \xamelaapo	 \xamxampet

\xamxam 	\xamxamapopet 
\xamxamapo 	\xamxamela 
\XIkat and \xamxamelaapo 	\XIIkat and \xamxamxam 
\XIIIkat and \xamxamxamapo 	\XIVkat and \xamxamxamela 
\XVkat and \xamxamxamelaapo 	
\gypokkoxe 	\pggypokkoxe 
\pgggypokkoxe 	\ypokkoxe 
\ggypokkoxe 	\gggypokkoxe 
\synelaoxe 	\apokkoxe 

\elakkoxe 	\elaapokkoxe 
\xamkkoxe 	\gypooxe 
\synelaoxe 	\apooxe 
\elaapooxe 	\xamoxe 
\ypooxe 	\oliapo 

13.2 Rest symbols

For the rests we use βαρεία(bar) with an απλή(t) to its right. The \bart combination is also defined as pay(from the Greek word παῦσις).

Command	Result	Command	Result
\bart and \pay		\bartt	
\barttt		\bartttt	

\bartg and \bargt	\bartpg and \barpgt
\bartgp and \bargpt	

13.3 Qualitative signs

The quality characters are auxiliary musical symbols that are placed underneath the main musical symbols. These are the έτερον (ete), ομαλόν (oma), λύγισμα (lyg), τρομικόν(tro), εκστρεπτρόν(eks), πίασμα(pia), παρακλητική(par), αντικένωμα (ant), small αντικένωμα (anm), ψηφιστόν(psh) and the opened ψηφιστόν (psa). In some complex cases, we had to use the command \bm to place the quality characters appropriately. Indicators like M, MD, K naturally refer to the position of the character relative to their positions below \iso. Thus, the \etepsaKD is the combination of ete and psa positioned under (K) and to the right (D) of \iso (or any other musical character with the same width as \iso). Here are some examples for your consideration.

Example:

\oli\eteMD \iso ⇒ and \oli\eteMDD \iso ⇒

\iso\eteM⇒

\iso\ttM\eteKDD\iso or simpler \isotteteKDD\iso ⇒

\apot\eteKD\iso ⇒

\oli\ttM\eteK\iso and \olitt\eteK\iso ⇒

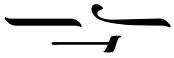
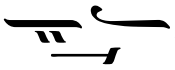
\oli\omaMDD\iso and \oliomaMDD\iso ⇒

\olikk\omaKDD and \olikkomaKDD ⇒ 

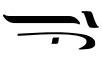
\isopet\eteMDD\apogkkoli and \isopeteteMDD \apogkkoli ⇒ 

\oli\eteMDD\iso\psaM and \olieteMDD \iso \psaM ⇒ 

\olikk\eteKDD\isokkoli\psaK and \olikketeKDD\isokkoli\psaK ⇒ 

Command	Example	Command	Example
symbol+\eteMDD		\eteKDD	
\eteMD		\eteM	
\eteKD		\eteK	
\omaK		\omaM	
\omaMDD		\omaMD	
\omaKD		\omaKDD	

\lygMDD		\lygMD	
\lygM		\troM	
\troMD		\eksMD	
\eksM		\piaMD	
\parP		\parPD	
\edoKD		\eteMDD	
\pshM		\pshMD	
opened ψηφιστό \psaM		\psaK	
\antM		anmMD	

\antK		\antK	
\anmM		\antK	
\anttM		\anmtMD	
\anmtM		\anttK	
\olieteMDD		\isopeteteMDD	
Difficult cases. They are made using \bm.			
\apoeteMDD		\olikketeKDD	
\petant		\olikant and \oliantk	
\petantt		\olikantt and \olianttk	
\etepsaKD		\etepsaKDD	

13.4 Aple(small dot) - diple - triple

The aple, diple and triple are denoted by the letters t, tt and ttt respectively. There are some difficult cases of musical combinations in which the \t, \tt, or \ttt require small displacements using \bm or other commands for better visual effect. In such cases we designed additional commands.

Command	Example	Command	Example
\oli\tM		\iso\ttM	
\iso\tttM		\apo\tMD	
\apo\ttMD		\apo\tttMD	
\ypo\tMD		\ypo\ttMD	
\ypo\tttMD		\apoapo\tKD	
\apoapo\ttKD		\apoapo\tttKD	
\oxe\tM		\xamoli \tM	
Difficult cases			

\XVkatt	\xamxamapot and \IXkatt 
\xamxamapopett and \IXkatpett 	\Xkatt and \xamxamelat 
\isoapot 	\oliapot 
\elat 	\elaapot 
\pett 	\olitk and \olikt 
\xamt 	\olixkt 

13.5 Small ison and musical hyphen

To ισάκι(smallon iso) is an auxilary symbol and has two forms: the normal \isa and lowered \isk. If the isaki has an additional positioning indicator (D or A) at its end then it should be located to the right of the musical character to which it is going to be placed (if there is no such sign then we can put isaki on any side). The hyphen and other patterns of variable length may be made by the \agkylh command without the use of any font. They could also be made directly from some symbols of our BZ-fonts, if desired, changing their length and position with appropriate commands. The commands \yfma and \yfmk (small hyphens) always have a fixed width while \yfa and \yfk have a variable width and position. All these hyphens are placed between the hyphened

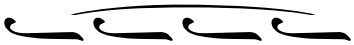
characters. Otherwise, we need to use the command \bm or possibly change the pre-determined values of the second and third argument of \yfa and \yfk (corresponding to horizontal and vertical displacement of the hyphen, respectively) for better positioning.

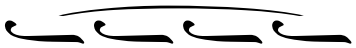
Example:

\iso\isaA ⇒ 

\apo\isaD ⇒ 

\isa\iso ⇒ 

\iso\yfmk\iso ⇒ 

\iso\iso\yfa[3.5]\iso\iso ⇒ 

\iso\iso\yfa[3.5,-0.04,0.05]\iso\iso ⇒

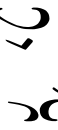
Command	Example	Command	Example
\isk		\isaA is positioned from the left	
use \isaD for small symbols		\isa	
\yfma (small hyphen) from above		\yfmk (small hyphen) below	

variable witdth \yfa		variable witdth \yfk	

13.6 Signs that add duration

The kla command is used to double the duration. There are various forms of the \kla according to their normal position over the iso sign. The following are example of kla commands: kla, klaA, klaD which put the kla at the middle, to the left or the right, while klaM, klaMD, klaKD commands place kla below the center, center right and below to the right, but lower than klaMD. Usually, we would not need to use all these kla symbols in a composition because we have already built common combinations containing kla covering most of our needs. For example, instead of typing \iso \kla can simply insert \klaiso. Instead of \iso \klaM we can simply write \isokla (as we have already mentioned we usually do not use the additional signs M, D, DD .. in compound command names, because lengthly names are difficult typed). Exactly the same rules are applied to the \tsa(tsakisma) symbol.

Example:

\pet\tsaK ⇒ 

\synela\kla ⇒

\klaapoolipsh ⇒ 

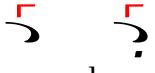
Command	Example	Command	Example
\klaM		\iso\tsaM	

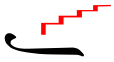
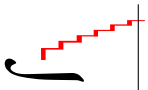
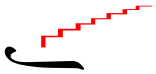
\iso\kla 	\oli\tsa 
\apo\tsaD	\ypsoli\klaA 
\pet\tsaK	\pet\klaKD 
\ela\klaK	\elaapo\klaK 
\synela\kla	\ela\kla
\isopet \klaK 	\xamapo\klaD
\xamela\klaPD 	σύνθεση με το \klaP

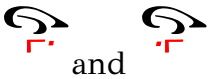
13.7 Signs that divide the duration

p denotes a dot and is used with g (gorgon), gg and other symbols. The gorgon requires special displacement using \bm or
other command, in some difficult cases.

Command	Example	Command	Example

\iso\g and \giso		apo\gD \and \gapo	
\ypo\gDD \gypo		\ypokkoli\gPA	
\kkoli\gP and \gkkoli		\oli\gM and \olig	
\isokkoli\gPD		\xamkkoli\gPD	
\apokkoli\gPD		\elakkoli\gPD	
\isoapo\gDD		\oli\g\k and \golik	
\ela\g and \gela		\apo\gMD and \apog	
\kk\gMD and \kkg		\xam\gD and \gxam	
\xam\gMD and \xamg		\gapo and με t \gapo\tMD and \gapot	 and

 \kkoli\ggP	 \kkoli\gggP
 \ypo\pggDD	 \ypo\ggDD
 \ypo\pgggDD	 \ypo\gggDD
 \ypokkoli\gggPA	 \ypo\ggpDD
 \ypo\gggpDD	 \ypokkoxe\gPA
 \ypokkoxe\ggPA	 \ypokkoxe\gggPA
 \gIV	 \gV
 \gVI	 \gVII
Difficult cases	
\apoapog and \apogapo 	 \elag

\elaapog 	\elaapogp and \elaapopg 
\xamelaapog 	\gkkypsoli 
\gxamelaapo 	\ypog 
\synelagkkoli 	

13.8 The argon, diargon and triargon

Command	Example	Command	Example
\arg		\diaD	
\tri		\dia	
\parg		\argp	
\diap		\pdia	

	
\trip	\ptri

13.9 The cross, apostrophe (breathings) and corona

These commands are always placed on the right side of a composition.

Command	Example	Command	Example
\sta		\staP and \staD	
breath sign \ana			
corona \kor		\korD	
\korP		\korPD	

13.10 Measures and bars

The bars are denoted with the letter l or L depending on their heights. The yfe as an additional indicator is used when we want to put a small hyphen on a bar (as in the case of συνεπτυγμένου rhythm). We use latin numberin to indicate the measures like \VII. We should be careful because the capital I is used for counting while the l ie. the small L is reserved for the bars. The bars are always inserted in the right side of our composition even if they end with an A sign. Alternatively, we can use the \th along with some indicators for the bars and measures (as mentioned above) in order to place them uniformly before the musical

characters eg. \LIIIth

Example:

\iso\l\VIII ⇒

Command	Example	Command	Example
\lA		\lMD	 and
		\llA	
\l		\ll	
\lyfe		\lyfeA	
\V		\VI	
\VII		\VIII	
\L		\LL	

\Lyfe 	\II 
\IIA 	\III 
\IIIA 	\IV 
\IVA 	\VA 
\VIA 	\VIIA 
\VIIIA 	

13.11 Sharps and flats

Sharps(\d) and flats(\y) are stressed by using the l (one stroke). Additional strokes can be added to give the note a deeper quality of either being sharper or flatter. They are colored red to better distinguish them.

Command	Example	Command	Example
\y		\yD	

\yl 	\ylD 
\yll 	\yllD 
\ylll 	\yllD 
\dM 	\dMD 
\dlM 	\dlMD 
\dllM 	\dllMD 
\dlllM 	\dlllMD 
Difficult cases	
\elad 	

13.12 Martyries and note signs

All martyries and note signs are colored red. All martyries ending in P (= above) are placed naturally first in the left argument of \marts command,

while those ending in K (= bottom) always come to the right argument. The letter b denotes a martyria of soft chromatic genus and the letter plab of hard crhomatic genus. The mark dia is used for diatonic matyries and ton when we must incert an accent on a letter. All complex martyries are made by the use of the command \marts.

Example:

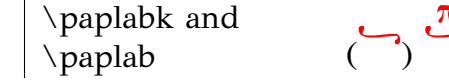
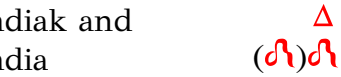
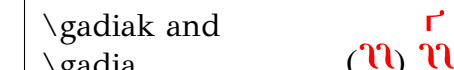
\marts[\pa][\paplab] ⇒ 

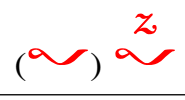
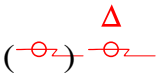
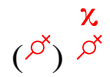
\marts[\nhdia][\gak] ⇒ 

\marts[\di][\rbox[-0.8]{\yD}] ⇒ 

In the following list the comand of interest has been enclosed in parenthesis to distinguish it from the example to its right.

Command	Example	Command	Example
\pa		\boy	
\ga		\di	
\ke		\zw	
\nh		πα with accent \paton	
\boyton		\gaton	

\diton		\keton	
\dik		\kek	
\gak		\boydiaa	
\paplaba		\paplabk and \paplab	
\nhdiak and \nhdia		\nhdiaa	
\padiaa		\padiak and \padia	
gadiaa		\gadiak and \gadia	
\didia		\boydia and \boydiak	
\diplabk and \diplab		\diplaba	

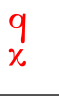
\zwdiak and \zwdia		\dibk, \nhbk and \dib	
\kedia and \kediak		thema aploun \tha	
\kebk, \nhtonbk, and \pab, \keb, \nhtonb			
\gaD		\diD	
\keD		\zwD	
\nhD		\paD	
\boyD		\paMA	
\diMA		\gaA	
\diA		\keA	

	
\zwA	\nhA
	
\paA	\boyA

13.13 Some useful combinations of martyries

In the following combinations there are some cases where a sign of a note is missing for readability reasons only. For example, \nhdiam is the shortcut for \nhnhdiam. It is also worth noting that each of these combination ends with the letter m for better readability in a complex and long musical text.

Command	Example	Command	Example
\nhdiam		\padiam	
\zwdiam (and \zwzwdiam)		\boydiam	
\gadiam		\didiam	
\kediam		\zwtondiam (and \zwtonboydiam)	
\nhtondiam(and \nhtongadiam)		\patondiam	

\boytondiam		\gatondiam	
\ditondiam		\ketondiam	
\zwtontondiam		\nhtontondiam	
diat. martyria lowest ke \kekdiam (and \padiakem)		diat. martyria lowest di \dikdiam (and \nhdiadim)	
diat. martyria lowest ga \gakdiam (and \gadiagam)		diat. martyria lowest bou \boykdiam (and \boydiaboym)	
diat. martyria lowest pa \pakdiam (and \padiapam)		diat. martyria lowest ni\nhkdiam (and \nhdianhm)	
plagal 2nd mode pa \paplabm		plagal 2nd mode bou \boyplabm	
plagal 2nd mode ga \gaplabm		plagal 2nd mode di \diplabm	
plagal 2nd mode ke \keplabm		plagal 2nd mode upper zo \zwtonplabm	

plagal 2nd mode ni \nhtonplabm		plagal 2nd mode upper pa \patonplabm	
\digadiam		2nd mode ni \nhbm	
2nd mode pa \pabm		2nd mode bou \boybm	
2nd mode ga \gabm		2nd mode. di \dibm	
2nd mode ke \kebm		2nd mode upper zo \zwtonbm	
2nd mode upper ni \nhtonbm		\nhtondiplabm	
Ench. mode ni \nhgadiam		Ench. mode πα \panhdiam	
Ench. mode bou \boypadiam		Ench. mode. upper zo \zwtongadiam	
Ench. mode upper ni \nhtondidiam		Ench. mode. bou \boytongadiam	

13.14 Initial martyries of modes

The initial martyries of modes always end in the letter h to be distinguished easily. The modes first, second, third, fourth and baris are marked respectively with symbols \ahxo, \bhxo, \ghxo, \dhxo and \barhxo. We have already built most of them so there will probably be no need for the construction of a new martyria. The command \tesh makes a symbol resembling the number four and is always positioned to the right in the construction of an initial martyria.

Example:

\kediah\bm[\tesh, 0.1]⇒

\synelah\gaboydiah\tosh ⇒

\lph\kediah\rbox[-0.2]{\kediaypsolikeh} ⇒

\zwh\ypskolizwdiah ⇒

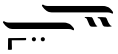
\teskediah\kediakeh ⇒

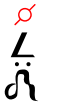
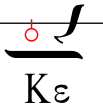
\padiaypsolih \keh ⇒ , \boydiah\tosh ⇒

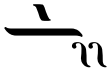
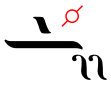
\synelah\boyh ⇒ , \pah\paplabfPD ⇒

Command	Example	Command	Example

\plah	\hxoh and \hxosh
\atonh	\btonh
\gtonh	\dtonh
\amtonh	\bmtonh
\gmtonh	\dmtonh
\tosh	\barysh
\dhxoh	\didiah
\ahxoh	\kediah
\bhxoh	\dibh

 \barhxoh	 \ttghxoh
 \ghxoh	 \olikkh
 \paplabpahxoh	 \tesh
 \dibdihxoh	 \boybboyhxoh and \dibboyhxoh
 \gah	 \dih
 \keh	 \zwh
 \nhh	 \pah
 \boyh	 \padiah
Some useful combinations	
 \nhdianhh	 \nhdiagah

 \didiadih	 \tesdidiah
 \diplabtesdidiah	 \dibdih
 \dibboyh and \boybboyh	 \gaboydiah
 \dibdididiah	 \kediaypsolikeh
 \barhxozwyh	 \ypskolizwyh and \ypskzwyolih
 \padiapah	 \pah\paplabfPD and \paplabpah
 \kolinhdiah and \knhdiaolih	 \ypskolizwdiah and \ypskzwdiaolih
 \kediakeh	 \kolidiplabh and \kdiplabolih
 \dibdididibh	 \padiaypsolih

\dibkolih and \kdibolih 	\ypskoligadiah and \ypskgadiaolih 
 \boydiah	 \koligadiah
\kolidiplabgadiah and \kdiplaboligadiah 	 \synelah
 \gagadiah	 \ypskh
 \ypskdidiah	 \ypskkediah

13.15 ι, ϙ symbols and symbols for isokratimata

The ne(\ne), na(\n), arrow pointing up(\bela) or downwards(\belk)) and the diphthong th(\oy) are positioned with the help of \s over the musical characters.

Example:

\th{\s[(M)][\iso]}[] ⇒
ΔΙ \belk ⇒ ΔΙ↓, ΔΙ \bela ⇒ ΔΙ↑
(NH-B\oy) ⇒ (NH-BϏ)
\n η ⇒ ϙη and \ne ε ⇒ιε

13.16 Tempo signatures

Tempos are divided into slow (marked as b), and fast (marked as x). The P denotes the upper position while the mchi denotes the Greek letter chi which is small. The tempos ending in P are always placed to the right in the construction of combinations. To properly place the tempos we have to use the \xrs command in combination with the mchi commands.

Example:

\th{\xrs[\bbbmchi][\dibm]}[]

The use of tempo commands ending in P should generally be avoided because they do not always give satisfactory results:

\dibm\bbbmchiP ⇒ 

Command	Example	Command	Example
\bbbmchiP		\bbbmegachi	
\bbmegachi		\bmchiP	
\bmegachi		\bxmchiP	
\bxmegachi		\xmchiP	

\xmegachi		\xxmchiP	
\xxmegachi		\xxxmchiP	
\xxxmegachi		\bbmchiP	
\bbbmchi		\bbmchi	
\bmchi		\bxmchi	
\xmchi		\xxmchi	
\xxxmchi		\mchi	χ
\megachi	χ		

13.17 Fthores(destroyers) and chroes(chromatic signs)

All martyries and chroes end with the letter f to distinguish them from the other symbols. Their default color is red. Most of the fthores names

have additional signs like M, MD on their right. If there is no such indicator then the symbol is positioned on the upper middle of a big symbol(ie. one having the same length as the \iso symbol).

Command	Example	Command	Example
spathi to the right \spafD, \spaD		Πα plagal 2nd mode \paplabfD and \keplabfD	
Di 2nd mode \dibfD and \nhbfD and \boybfD		ke (and upper ni) 2nd mode \kebfD, \nhanwbfD and \pabfD	
Di plagal 2nd mode\diplabfD		geneki yfesi \gyfD, \gyD	
zygos to the right \zygfD,\zygD		kliton to the right \klifD,\kliD	
geniki diesi \gdfD,\gdD		upper zo or ga geniki yfesi \zwyfD,\zwyD, \gayfD, gayD	
thema aploun \thaf		thema aploun under iso\thafM,\thaM	
diatonic ni tb nhdiafD		diatonic pa \padiafD	

diatonic bou \boydiafD 	diatonic ga \gadiafD 
diatonic di \didiafD 	diatonic ke \kediafD 
diatonic upper zo \zwdiafD 	diatonic upper ni \nhanwdiafD 
diat. ni under iso\nhanwdiaf 	diat. upper ni under iso \nhanwdiafMD 

13.18 Polytonic texts

In the list below we used the impressive "medieval" font Junicode (`https://www.ctan.org/tex-archive/fonts/junicode`), which contains many letters of the Ancient Greek language. Alternatively, we can use Arial or any other font that suits us. If a ? appears instead of a letter this means that this corresponding character is not available. We have used the command \changetextfont[Junicode] to change the font of the text. If the command \tildeON is inserted somewhere at the beginning of our text (before we start writing polytonic) then it is necessary ~ to be used solely as a binding of the syllable containing the polytonic command with the next syllable (in the same word). In other words, the ~ becomes the braces {} shortcut. We emphasize this mandatory use of ~ in some examples in the list below. For additional examples and instructions regarding the package greektonoi one should read the package documentation. The letter ε in front of some letters provides an alternative version of them. E.g. \εθ ⇒ ϑ. We need to know that in polytonic text writing some letters have one form at the beginning of a word (e.g. the letter β) and another form when they are used somewhere in the middle (e.g. ϐ). The leaf after the end of the table is not an image but a letter of the Junicode font!

Command	Example	Command	Example

\; semicolon	ἐφυγε·	right quotes \>>	»
left quotes \<<	«	\ααπ left. single quot. mark	᾽
\πυ(υ circumflexed)	ῦ	\βα(a with grave)	ὰ
\λυ(υ with diairesis)	ϋ	\μχρυ(long υ)	ῡ
\βρχυ (υ βραχύ short υ)	ŭ	\μΩ	Ω῾
\ψα (a smooth breath)	ἀ	\δα (a rough breath)	ἁ
\εβ~	ϐ	\εθ~	ϑ
\εφ~	ϕ	\επ~	ϖ
\ερ~	ϱ	\εκ~(version of ε)	ϰ

\εε~	?	digamma \ϝ	F
capital koppa \Κοπ	?	small coppa \κοπ	?
capital sampi \Σαμ	ϡ	small sampi \σαμ	ϡ
\ρρ~	ῤῤ	\τ (accute accent)	ά