# `metre.sty`
# A LaTeX package for classicists

Version 1.0 (2001/12/05)

Gianfranco Boggio –Togna *
*gbt@acm.org*

*Questo piccolo lavoro è dedicato ai*
*miei insegnanti di greco e latino del*
*Liceo Ginnasio 'Girolamo Rossi' di Ventimiglia.*

## License



This program consists of the files `metre.dtx` and `metre.ins`.

## Acknowledgement

The author wishes to thank Dr. Rainer Thiel of Philipps–Universität, Marburg, for providing him with much material and expert advice on the *arcana* of classical philology. Without Dr. Thiel's assistance, the package would never have seen the light of day.

*Author's address: C.P. 14021, I-20140 Milano

## Introduction

The package provides classicists with some of the tools that are needed for typesetting scholarly publications dealing with Greek and Latin texts, with special emphasis on Greek verse.

As the name suggests, the core of the package is a comprehensive set of commands for generating metrical schemes and for placing prosodical marks on text set in the Latin or the Greek alphabet.

The rest of the package provides a miscellany of commands for symbols (most of them not directly related to metre) that are often used in critical editions of classical texts.

The package does not require any special font: all symbols are taken from the Computer Modern OT1 fonts (included in all distributions of TEX) or are generated by means of TEX primitives.

## Notation

The notation used for quantities that can appear as command arguments is that of *The TEXbook*:

| | |
|---|---|
| ⟨*dimen*⟩ | a legitimate TEX dimension. |
| ⟨*number*⟩ | a (signed) integer. |
| ⟨*factor*⟩ | a (signed) decimal constant. |

Many commands accept an ⟨*optional argument*⟩ that can be used to modify the effect of the command. The ⟨*optional argument*⟩ is specified according to the usual LATEX conventions: within square brackets, after the command name.

# Contents

# 1 Package option

The package has one option, which can be specified in the usual way within square brackets (e.g. `\usepackage[en]{metre}`). The option determines the vertical placement of the metrical symbols generated by `\metra`.

The default style is the one that is commonly used in Germany (and universally in Italy). The base of the metrical symbols is aligned with the baseline of the surrounding text and composite symbols are built upwards from the baseline: _ ◡ ⏑ ⏒ ⏓.

The `en`[1] option selects an alternative style which is usual in English typography (for instance, in books published by Oxford University Press). The horizontal axis of the metrical symbols is aligned with the axis of characters such as `=` in the surrounding text and symbols, whether simple or composite, are roughly symmetrical with respect to the axis: – ◡ ⏑ ⏒ ⏓.

You can change the style at any time with the `\MetraStyle` command (see sect. 2.1.4).

[1] `En` selects the same style as `en` with one difference: the *macron* in ⏑ is placed on the horizontal axis –⏑–, while the `[en]` option puts the *breve* on the axis –⏑–.

# 2 Metrical notation

## 2.1 The \metra command

### 2.1.1 Metrical symbols

\metra The \metra command takes one argument and generates the metrical scheme defined by the commands given within the argument.

The list of metrical symbols and corresponding commands is on page 8; it includes all the symbols (marked [*W*] in the table) used by M. L. West in *Greek Metre* (Oxford, 1982). The commands for metrical symbols are defined only within the argument of \metra and are not available outside it. The text typeset by \metra is never split across lines; if it does not fit within the line, TEX gives an Overfull box message.

The \r, \R, \t and \T commands take an ⟨*optional argument*⟩, a ⟨*factor*⟩ which is multiplied by the ex value for the font in use for the metrical symbols; the result is the amount by which the symbol is raised (⟨*factor*⟩ $> 0$) or lowered (⟨*factor*⟩ $< 0$).

The metrical symbols are separated by a short space, whose size may be set with the \InterSigna command (see sect. 2.1.4). To generate a larger
\s space (typically between feet) use the \s command; you can modify the size of the space generated by \s with the \InterPedes command (see sect. 2.1.4). You can also use the standard TEX and LATEX commands for horizontal spacing.

\v The \v command inserts a comma.

### 2.1.2 Placing marks on symbols

\' The \' and \‘ commands place an accent (respectively acute or grave)
\‘ over the metrical symbol which follows: \'\m yields ⏜, \‘\m yields ⏝. Both commands accept an ⟨*optional argument*⟩, a ⟨*factor*⟩ which specifies the displacement of the accent to the right, as a fraction of the width of the symbol. The default value (.5) places the accent over the midpoint of the symbol: \'[.25]\bb yields ⏜, \'\bb yields ⏜, \'[.75]\bb yields ⏜. For convenience, a set of commands (see page 9) is provided for generating the most common combinations of accents and symbols.

\k The \k command places the symbol for *catalexis* under the symbol which

`\K` follows: `\k\m` yields $\underset{\frown}{}$, `\k\bm` yields $\underset{\frown}{\boldsymbol{}}$. The `\K` command generates a slightly larger symbol: `\K\m` yields $\underset{\frown}{}$, `\K\bm` yields $\underset{\frown}{\boldsymbol{}}$.

`\q` The `\q` command encloses the argument in some kind of *"quotation marks"*; the type of mark used is selected by the character that follows the command.

`\Q` The `\Q` command generates slightly larger marks.

| *Type* | *to get* |
|---|---|
| `\q(\bbm` | ‘$\underset{\smile}{}$’ |
| `\Q(\bbm` | ‘$\underset{\smile}{}$’ |
| `\q'\bbm` | ‘$\underset{\smile}{}$’ |
| `\Q'\bbm` | ‘$\underset{\smile}{}$’ |
| `\q"\bbm` | “$\underset{\smile}{}$” |
| `\Q"\bbm` | “$\underset{\smile}{}$” |
| `\q!\bbm` | '$\underset{\smile}{}$' |
| `\Q!\bbm` | '$\underset{\smile}{}$' |
| `\q<\bbm` | «$\underset{\smile}{}$» |
| `\Q<\bbm` | «$\underset{\smile}{}$» |

`\d` The `\d` command places the symbol for *caesura* at the middle point of the symbol which follows: `\d\mbb` yields $\dagger$, `\d\bbm` yields $\dagger$.

`\S` The `\S` command has one argument: a sequence of characters, which is raised from the baseline by 1.25 times the `ex` value for the font in use for the metrical symbols and set in a smaller size: `\C\S{H}` yields $|^{\mathrm{H}}$. You can change the amount by which the argument is raised by specifying an ⟨*optional argument*⟩. The argument is a ⟨*factor*⟩ which is multiplied by the `ex` value for the font in use for the metrical symbols; the result is added to the default value. `\S[-.3]?\mb` yields $^{?}\breve{}$, `\bm\S[-.3]?` yields $\breve{}^{?}$.

### 2.1.3 Placing annotations on symbols

`\n` The `\n` command has two arguments: the first is the text to be placed over
`\ni` a symbol; the second is the command for the symbol. The `\ni` command places the text *under* the symbol. `\n1\m` yields $\perp$, `\ni1\m` yields $\underset{1}{}$.

`\N` The `\N` command has three arguments: the first is the text to be placed over a symbol; the second is the command for the symbol itself; the third is the text to be placed *under* the symbol. The `\mbbx` and `\bbmx` commands are meant to be used in conjunction with `\N`: `\N{25}\mbbx{36}` yields $\overset{25}{\underset{36}{}}$, `\N{25}\bbmx{36}` yields $\overset{25}{\underset{36}{\boldsymbol{}}}$.

In some circumstances (for instance, if the metrical symbol is surrounded by taller symbols) it may be necessary to increase the spacing between the symbol and the text above or under it. This can be achieved by specifying an ⟨*optional argument*⟩, a ⟨*factor*⟩ which is multiplied by the value of the default spacing. For the `\N` command, the new spacing applies to both sides.

In metrical schemes, `\n` and `\N` can be used to give a reference to one or more line numbers within the text. It is advisable to restrict such references to one or two numbers. For more than two numbers (and whenever the numbers placed over different symbols overlap) it is usually better to place a single mark over the symbol. The corresponding list of line numbers can be given in a separate note, which may be generated with the `\numeri`

`\numeri` command: the argument is a sequence of characters that *must* contain a colon: the text preceding the colon is set in italics (it is assumed that the same text, also set in italics, appears over the symbol).

It is also possible to stack the numbers up vertically by using the `\structa` command (see page 20):

`\n{\structa{20}{36}}\mb` yields $\overset{36}{\overset{20}{\cup}}$

`\n{\structa{31}{\structa{54}{108}}}\mb` yields $\overset{\overset{108}{54}}{\overset{31}{\cup}}$

`\n{\structa[.1]{31;54}{108}}\mb` yields $\overset{108}{\underset{\cup}{31;54}}$

# METRICAL SYMBOLS

| | | |
|---|---|---|
| `\m` | $\bar{}$ | long [*W*] |
| `\b` | $\breve{}$ | short [*W*] |
| `\a` | $\times$ | anceps [*W*] |
| `\ma` | $\bar{\times}$ | long syllable in anceps position [*W*] |
| `\bba` | $\breve{\times}$ | 'teilbares Anceps' |
| `\bm` | $\underline{\bar{}}$ | usually long [*W*] |
| `\mb` | $\underline{\breve{}}$ | usually short [*W*] |
| `\bbmb` | $\breve{\underline{\breve{}}}$ | short replaceable by resolvable long |
| `\bbm` | $\underline{\underline{\breve{}}}$ | resolvable long [*W*] |
| `\bbmx` | $\underline{\underline{\breve{}}}\breve{}$ | *see sect. 2.1.3 (page 6)* |
| `\bb` | $\underline{\breve{}}\,\underline{\breve{}}$ | resolved long [*W*] |
| `\bbb` | $\breve{\underline{\breve{}}}$ | 'teilbares Breve' |
| `\mbb` | $\breve{}\,\underline{\breve{}}$ | contractible biceps [*W*] |
| `\mbbx` | $\breve{}\,\underline{\underline{\breve{}}}$ | *see sect. 2.1.3 (page 6)* |
| `\pm` | $\underline{\cdot}$ | |
| `\ppm` | $\underline{\cdot\cdot}$ | contracted biceps [*W*] |
| `\vppm` | $\underline{\vdots}$ | |
| `\vpppm` | $\underline{\vdots}$ | |
| `\tsmb` | $\lceil\!\underline{\quad}$ | triseme (equivalent to $\bar{}\,\breve{}$) [*W*] |
| `\tsbm` | $\underline{\quad}\!\rfloor$ | triseme (equivalent to $\breve{}\,\bar{}$) [*W*] |
| `\tsmm` | $\lceil\!\underline{\quad}\!\rfloor$ | tetraseme (equivalent to $\bar{}\,\bar{}$) [*W*] |
| `\ps` | $\lceil\!\underline{\quad}\!\rfloor$ | pentaseme |
| `\oo` | $\circ\circ$ | two positions of which one at least must be long [*W*] |
| `\C` | $\vert$ | word-end [*W*] |
| `\Ppp` | $\vdots$ | often word end [*W*] |
| `\Pp` | $:$ | less often word end [*W*] |
| `\Pxp` | $:$ | |
| `\Pppp` | $\vdots$ | |
| `\Ppppp` | $\vdots$ | |
| `\Cc` | $\Vert$ | period-end (or beginning) [*W*] |
| `\Ccc` | $\Vert\!\Vert$ | strophe-end (or beginning) [*W*] |
| `\c` | $\vert$ | (*caesura*) same as `\C` but with wider spacing |
| `\cc` | $\Vert$ | same as `\Cc` but with wider spacing |
| `\ccc` | $\Vert\!\Vert$ | same as `\Ccc` but with wider spacing |
| `\ppp` | $\vdots$ | same as `\Ppp` but with wider spacing |
| `\pp` | $:$ | same as `\Pp` but with wider spacing |
| `\pxp` | $:$ | same as `\Pxp` but with wider spacing |

| | | |
|---|---|---|
| \pppp | ⋮ | same as \Pppp but with wider spacing |
| \ppppp | ⋮ | same as \Ppppp but with wider spacing |
| \p | . | syncopation |
| \r \R | ∼ ∼ | in responsion with $[W]$ |
| \t \T | ⊗ ⊗ | beginning or end of composition $[W]$ |
| \x | :: | change of speaker $[W]$ |

---

# SYMBOLS WITH ACCENT

| | | | |
|---|---|---|---|
| \M | ـَ | \gM | ـً |
| \B | ـُ | \gB | ـٌ |
| \Bm | ـِ | \gBm | ـٍ |
| \Mb | ـْ | \gMb | ـّ |
| \Mbb | ـّـْ | \gMbb | ـّـً |
| \mBb | ـّـُ | \gmBb | ـّـٌ |
| \mbB | ـّـِ | \gmbB | ـّـٍ |
| \BBm | ـُـِ | \gBBm | ـُـٍ |
| \Bbm | ـُـْ | \gBbm | ـُـً |
| \bBm | ـِـُ | \gbBm | ـِـٌ |
| \BB | ـُـُ | \gBB | ـُـٌ |
| \Bb | ـُـْ | \gBb | ـُـً |
| \bB | ـِـِ | \gbB | ـِـٍ |

### 2.1.4 Changing the appearance of the symbols

You can change the appearance of the metrical symbols with these commands:

`\MetraStyle` `\MetraStyle` changes the style (see sect. 1) for the metrical symbols. The argument is either `en` (or `En`) for the 'English' style or `st` for the standard, default style.

`\InterSigna` `\InterSigna` sets the size of the space between adjacent metrical symbols. The argument is a ⟨*factor*⟩ which is multiplied by the width of a *macron* (default: .2).

`\InterPedes` `\InterPedes` sets the size of the space generated by the `\s` command. The argument is a ⟨*factor*⟩ which is multiplied by the width of a *macron* (default: .67).

`\SubSigna` `\SubSigna` modifies the distance between the horizontal axis of the metrical symbols and the baseline of the surrounding text. The parameter is a ⟨*factor*⟩ which is multiplied by the `ex` value for the font in use immediately before the `\metra` command; the result is added to the default displacement of the axis from the baseline.

`\Intervallum` `\Intervallum` sets the spacing between the components of composite symbols such as `\bm` or `\mb`. It may be necessary to increase the spacing if the output is to be printed on a low-resolution device. The parameter is a ⟨*factor*⟩ which is multiplied by the `ex` value for the font in use for the metrical symbols (default: .08).

`\Magnitudo` `\Magnitudo` modifies the size of the metrical symbols. The parameter is a ⟨*number*⟩: a positive value increases the size, a negative value decreases it. `\Magnitudo{+1}` selects the next higher font size in the LaTeX font hierarchy extending from `\tiny` to `\Huge`; `\Magnitudo{-1}` the next lower.

The effect of these commands, unless they occur inside a TeX *group*, applies to the rest of the document from the point of use forwards. You can confine the effect to the argument of a single `\metra` command by placing the command(s) inside an ⟨*optional argument*⟩. For convenience, an abbreviated form of the command names is provided for use (only) within the optional argument: `\ms`, `\is`, `\ip`, `\ss`, `\i`, `\m`, with the obvious meaning. `\en` and `\st` may be used as shorthand for `\ms{en}` and `\ms{st}`.

## 2.2 The environments

The two environments, `metrica` and `metrike`, allow you to place prosodical marks over text set in, respectively, the Latin and the Greek alphabet.

`\=` `\-` To place a *macron* over a letter, put `\=` in front of the letter; to place a *breve*, put `\-` in front of the letter (the standard TEX control sequence `\u` may also be used). `\begin{metrica} \=a\-e \end{metrica}` yields āĕ .

To place a *macron* or *breve* over a diphthong, enclose the diphthong in braces: `{\metrica \={ae}}` yields āe͡.

The environments appear to be compatible with most LATEX environments. They are **not** compatible with the LATEX `tabbing` environment, which redefines some critical control sequences.

`\Elevatio` By default, the *macron* and *breve* are placed at the natural height for an accent. In some cases, this may make the metrical structure difficult to grasp at a glance. The `\Elevatio` command causes the prosodical marks to be placed at a constant distance from the baseline; the command argument is the distance, a ⟨*factor*⟩ of the `ex` value for the font in use. Note that the distance may be negative, in which case the metrical symbol is placed *under* the letter. `\Elevatio{0}` restores the default.

The effect of the `\Elevatio` command, unless it occurs inside a TEX *group*, applies to the rest of the document from the point of use forwards. To modify the vertical spacing for a single instance of the environment, give the command inside the environment.

`\Translatio` The `\Translatio` command is analogous to `\Elevatio` and controls the horizontal position of the prosodical marks. The argument is a ⟨*factor*⟩ of the `em` value for the font in use; prosodical marks are shifted to the right by this amount. A value around .1 is generally satisfactory for text set in italics.

Within the environments, the following commands are defined:

`\c`, `\cc`, `\ccc`, `\C`, `\Cc`, `\Ccc`, `\p`, `\pp`, `\ppp`, `\pppp`, `\ppppp`, `\Pp`, `\Ppp`, `\Pppp`, `\Ppppp`, `\pxp`, `\Pxp` with the meaning they have within the argument of `\metra`.

`\pos`, `\Pos`, `\!`, as abbreviations for `\positio`, `\Positio`, `\dubia`. (see sect. 3.3)

`\e` and `\t` as abbreviations for `\Elevatio` and `\Translatio`.

In case of conflicts, definition of these commands may be suppressed by specifying an ⟨*optional parameter*⟩ for the environment, with the value `n` or `N` (e.g. `\begin{metrike}[n]`, `\begin{metrica}[N]`).

### 2.2.1 The `metrica` environment

`metrica` To place an accent over a letter carrying a prosodical mark, put the accent before the letter: `\=‘a` yields ā̀, `\-’a` yields ắ, `\=’{ae}` yields ǽ.

### 2.2.2 The `metrike` environment

`metrike` If a letter carries diacritical marks, the letter and **all** its marks (including the character for *iota* subscript) must be enclosed in braces to form a single argument for `\=` and `\-`.

If either `ibycus4` or `Babel` (with the `polutonikogreek` option) is loaded, it is used automatically; if both are present, `ibycus4` is selected. If you
`\MetrikeFont` use another package for Greek fonts, you must issue the `\MetrikeFont` command before the first use of the environment. The argument of `\MetrikeFont` is the sequence of commands that selects the Greek fonts. The command also accepts two single-letter symbolic parameters: `i` or `I` to select the `ibycus4` fonts, `b` or `B` to select the `Babel` fonts.

The effect of the `\MetrikeFont` command, unless the command occurs inside a TEX *group*, applies to the rest of the document from the point of use forwards.

## 2.3 The `\sigla` command

`\sigla` The command `\sigla` generates abbreviations for the names of metres (but it may also be used to generate other types of abbreviations). Within the argument of `\sigla` some characters and control sequences have a special meaning:

| *Type* | *to get* | *Type* | *to get* |
|---|---|---|---|
| `l_` | $l_{\wedge}$ | `_l` | ${}_{\wedge}l$ |
| `l/` | $l^{\div}$ | `/l` | ${}^{\div}l$ |
| `l:` | $l^{\cdot\cdot}$ | `:l` | ${}^{\cdot\cdot}l$ |
| `l\+` | $l^{+}$ | `\+l` | ${}^{+}l$ |
| `\d` | $\delta$ (isolated) | `\D` | $\delta$ (next to a letter) |
| `\s` | $\int$ | | |

The `\charcolon` and `\charslash` commands yield ':' and '/'. The `\S` command is available with the meaning it has within the argument of `\metra`.

The following commands are also defined, with the meaning they have within the argument of `\metra`: `\r`, `\x`, `\c`, `\cc`, `\ccc`, `\C`, `\Cc`, `\Ccc`, `\p`, `\pp`, `\ppp`, `\pppp`, `\ppppp`, `\Pp`, `\Ppp`, `\Pppp`, `\Ppppp`, `\pxp`, `\Pxp`. Definition of these commands may be suppressed by specifying an ⟨*optional parameter*⟩ for the `\sigla` command, with the value `n` or `N` (e.g. `\sigla[n]`).

For complicated abbreviations, some explicit adjustment of spacing may be necessary; this can be achieved by using the characters '`>`' and '`<`'. '`>`' inserts a space of approximately 1/18th of `1em`; '`<`' inserts a *negative* space of the same length.

For example, the following command:

```
\sigla{gl>\S[.4]{\metra{\bb}}\c \S{\metra{\bb}}<<cr
ia_ /3io k>\D{} \d{} D\S3 E>\S2 ph<\S{3d} tl<:
```

yields: $gl^{\smile}$ | ${}^{\smile}cr$ $ia_{\wedge}$ ${}^{\div}3io$ $k\delta$ $\delta$ $D^{3}$ $E^{2}$ $ph^{3d}$ $tl^{\cdot\cdot}$

By default abbreviations are set in italics. You can change the style with
`\FaciesSiglorum` the `\FaciesSiglorum` command: the argument is the sequence of LaTeX or TeX commands to be used for setting the text. The effect of the command, unless it occurs inside a TeX *group*, applies to the rest of the document from the point of use forwards.

# 3 Other signs

## 3.1 Brackets

Each command yields the symbol on the right:

| | | | | |
|---|---|---|---|---|
| `\angus` | ⟨ | `\Angus` | ⟨ | *(uncus angulatus)* |
| `\angud` | ⟩ | `\Angud` | ⟩ | |
| `\quadras` | ⟦ | `\Quadras` | ⟦ | *(uncus quadratus)* |
| `\quadrad` | ⟧ | `\Quadrad` | ⟧ | |
| `\alas` | { | `\Alas` | { | *(uncus alatus)* |
| `\alad` | } | `\alad` | } | |
| `\semi[` | ⌞ | `\semi]` | ⌟ | |
| `\crux` | † | `\Crux` | † | |

## 3.2 Special symbols

Each command yields the symbol on the right:

| | | | |
|---|---|---|---|
| `\anaclasis` | ÷ | `\Anaclasis` | ÷ |
| `\antisigma` | ⊃ | `\Antisigma` | ⊃ |
| `\asteriscus` | ※ | `\Asteriscus` | ※ |
| `\catalexis` | ∧ | `\Catalexis` | ∧ |
| `\diple` | > | `\Diple` | > |
| `\diple*` | ⪢ | `\Diple*` | ⪢ |
| `\antidiple` | < | `\Antidiple` | < |
| `\antidiple*` | ⪡ | `\Antidiple*` | ⪡ |
| `\obelus` | — | `\Obelus` | — |
| `\obelus*` | ⨪ | `\Obelus*` | ⨪ |
| `\respondens` | ∼ | `\Respondens` | ∼ |
| `\terminus` | ⊗ | `\Terminus` | ⊗ |
| `\terminus*` | ⊕ | `\Terminus*` | ⊕ |

All the commands for special symbols take an ⟨*optional argument*⟩, a ⟨*factor*⟩ which is multiplied by the `ex` value for the font in use; the result is the amount by which the symbol is raised (⟨*factor*⟩ $> 0$) or lowered (⟨*factor*⟩ $< 0$).

Some of the characters (e.g. `\diple`, `\obelus`, `\antisigma`) are usually placed in front of a line of text, in the left margin. The `\margini` command puts the text given in the argument in the left margin of the page;

`\margini`

\margini* `\margini*` puts the text in the right margin. The argument is implicitly considered to form a TEX *group*, as if it were enclosed by an additional set of braces. An ⟨*optional argument*⟩ can specify a ⟨*dimen*⟩ which is added to the default displacement from the margin of the main text (a positive value moves the argument further away from the center of the page).

## 3.3 Diacritics

Each of the following commands places the corresponding mark over the letter given as argument:

| | | |
|---|---|---|
| \macron | `\macron e` | ē |
| \breve | `\breve e` | ĕ |
| \acutus | `\acutus e` | é |
| \gravis | `\gravis e` | è |
| \circumflexus | `\circumflexus e` | ẽ |
| \diaeresis | `\diaeresis e` | ë |

The commands take an ⟨*optional argument*⟩, a ⟨*factor*⟩ which is multiplied by the `ex` value for the font in use; the result is the amount by which the mark is raised (⟨*factor*⟩ > 0) or lowered (⟨*factor*⟩ < 0). *Note*: the TEX mathematical accent normally associated with `\breve` may be generated with the command `\Breve`.

\cons The `\cons` command places a semicircle under the letter that follows, to indicate *consonantization* of a vowel: `\cons{u}` yields u͡.

\dubia The argument of the `\dubia` command is a letter, under which the command places a small dot to show that the letter is only partially preserved in a manuscript or epigraphical text. If the letter carries diacritical marks, these must all be included in the argument.

\dubiae The `\dubiae` command has one parameter, a string of letters, and places a small dot under each of the letters. If a letter within the string carries diacritical marks, it must be enclosed, together with all the marks, within braces.

\dubia* \dubiae* The `*`-forms are meant to be used in conjunction with fonts, like those of `ibycus4`, that have ligatures for all combinations of a letter and the dot. The use of ligatures allows a more precise positioning of the dot, taking the shape of the letter into account.

\erasa The `\erasa` command yields |||.

\positio The `\positio` command yields the symbol ˈ, which is used between a mute and a liquid to show that the combination *makes position*, i.e. closes and
\Positio thereby lengthens the preceding syllable. `\Positio` generates a slightly larger symbol ˈ.

\punctum The `\punctum` command places a small dot under a space, to show that a
\Punctum letter is missing. `\Punctum` yields a dot enclosed within parentheses. Both commands take an ⟨*optional argument*⟩, a ⟨*number*⟩ that is interpreted as a repetition count. *Caution*: sometimes the dot is immediately followed, in the text, by a left square bracket; if you write `\punctum[` TEX will interpret the bracket as the start of the optional argument and give an error message. Use either `\punctum{}[` or `\punctum\lbrack`.

### 3.4 Ties

The commands described in this section allow you to place *tie* symbols above a sequence of up to three adjacent characters or under it. All combinations are allowed, including some that are not particularly sensible or useful.

\tie `\tie` places ◠ above the characters.

\itie `\itie` places ◡ below the characters.

\tie* `\tie*` places ◠ below the characters.

\itie* `\itie*` places ◡ above the characters.

The size of the symbol is chosen so that it embraces all the text. In some cases, you may wish to use a smaller or larger size. An ⟨*optional argument*⟩ can specify a ⟨*factor*⟩ which is multiplied by the width of the text: the size of the symbol is then chosen as if the text had that width. Keep in mind, however, that only a limited range of symbol sizes is available in the Computer Modern fonts.

When any of the *tie* commands is used within the argument of `\metra`, a ⟨*factor*⟩ of .6 is applied automatically and an explicit ⟨*factor*⟩, if specified, applies to the width as modified by the automatic factor. While `\tie{a e}` yields a͡ e, `\metra{\tie{\m\s\m}}` yields ‿

Examples:

| | | | |
|---|---|---|---|
| `\tie{ae}` | a͡e | `\itie{ae}` | a͜e |
| `\tie{uae}` | u͡ae | `\itie{uae}` | u͜ae |

```
\tie{a e}       a͡e          \itie{a e}      a e
a\tie{ }e       a͡e          a\itie{ }e      a e
\tie[.6]{a e}   a͡e          \itie[.6]{a e}  a e
\tie[.8]{a e}   a͡e          \itie[.8]{a e}  a e
```

```
Oceanum\itie{ }interea surgens Aurora reliquit.
It portis iubar\itie[.6]{e e}xorto delecta iuventus.
```

Oceanum interea surgens Aurora reliquit.
It portis iubare exorto delecta iuventus.

### 3.5 Interlinear signs

The signs generated by the commands described in this section are those commonly used for marking strophic divisions in Greek lyrical poetry. The symbols are inserted between lines or placed at the end of the text.

The commands must be used inside a LaTeX environment (such as `verse`) in which each line is terminated by `\\`, or else within the scope of the TeX `\obeylines` command. The normal form of the commands is used within LaTeX environments, while the `*`-form must be used within the scope of `\obeylines`.

The commands must appear at the end of a line and, if used within a LaTeX environment, must *not* be followed by `\\`. The symbol is inserted after the line on which the command occurs and the distance between this line and the line that follows (which may be empty for symbols that are placed after the last line of the text) is increased by 15%.

The automatic increment of the interline spacing may be modified by an ⟨*optional argument*⟩, a ⟨*factor*⟩ which is multiplied by the current value of `\baselineskip`; the result is added to the default increment.

`\linea` The basic command is `\linea` which generates a short line —— extending to the left of the margin of the surrounding text by half its length. There are two variants of `\linea`, both available in two sizes:

`\bifida` ≻—— `\lineabifida` or `\bifida`

`\Bifida` ≻—— `\Lineabifida` or `\Bifida`

`\lunata` )—— `\linealunata` or `\lunata`

\Lunata )—— `\Linealunata` or `\Lunata`

\geminata The `\geminata` command yields a pair of lines, each identical to the line generated by `\linea`: ═.

You can place some text (for instance, parentheses or a question mark) on either side of the symbols generated by the commands listed above
\antelineam by means of the `\antelineam` and `\postlineam` commands. The com-
\postlineam mands must appear at the beginning of the line which follows the symbol. Each command takes one argument: the text to be placed before or after the symbol. The text is implicitly considered to form a TEX *group*, as if it were enclosed by an additional set of braces. Within the argument, any command selecting the font to be used for the text must *not* appear within a *group*; if, for instance, you wish to use parentheses taken from the smallest font available to LATEX you should write: `\antelineam{\tiny(}` and `\postlineam{\tiny)}` and **not** `\antelineam{{\tiny(}}` or `\postlineam{{\tiny)}}`.

\coronis The `\coronis` command generates a stylized representation of the symbol sometimes found in manuscripts. Because of its peculiar shape, the symbol is not placed between lines but to the left of the text.

### Generating the signs within the text

To generate one of the signs described in this section within ordinary text,
\imago use the `\imago` command. The argument is the name of the command yielding the sign: `\imago{\bifida}` yields ≻—. The sign is generated at its natural distance from the baseline. The distance may be modified by an ⟨*optional argument*⟩; the argument, a ⟨*factor*⟩, is multiplied by the current value of `\baselineskip` and the result is added to the default distance. `\imago[-.6]{\bifida}` yields ≻—.

\imago* The *-form of the command automatically aligns the sign with the surrounding text: `\imago*{\bifida}` yields ≻—.

### Defining new signs

\novalinea The `\novalinea` command defines a new command that yields a line placed at the same height as the line generated by `\linea` but having a different length and/or indentation from the text margin. The command has four parameters:

*1* the name of the new command;
*2* the indentation, relative to the left margin of the text, of the line;
*3* the length of the line;
*4* the amount by which the interline spacing should be increased, given as a ⟨*factor*⟩ that is to be multiplied by the value of `\baselineskip`.

For instance, the command `\novalinea{\Linea}{-.5em}{2em}{.2}` defines `\Linea` as a command that yields a line of length `2em`, indented to the left of the text margin by 1/4 of its length; the interline spacing is increased by 20%.

`\novageminata` The `\novageminata` command, with the same parameters as `\novalinea`, defines a new command that yields a pair of lines similar to those generated by `\geminata`.

Definitions made with these two commands are valid within the TEX *group* within which they occur. It is therefore possible to restrict their effect to a section of the document by enclosing the section within braces and placing the definitions after the opening brace.

Note that it is possible to change the definition of `\linea` by saying `\novalinea{\linea}...` Changing the definition of `\linea` automatically changes the appearance of `\geminata`, `\bifida`, `\Bifida`, `\lunata` and `\Lunata`.

If you prefer a typographical style that puts the interlinear signs into the margin, away from the text, you need only change the definition of `\linea` (setting the appropriate value for indentation) in the preamble and all the document will be typeset in this style.

### 3.6 *Miscellanea*

`\lineola` The `\lineola` command generates, at the point where it is used, a horizontal line having the length given in the command argument. The line is placed at the same height, relative to the baseline of the text, as the line generated by `\linea`. You can change the height by specifying an ⟨*optional argument*⟩: a ⟨*factor*⟩ which is multiplied by the value of `\baselineskip`; the result is added to the default displacement from the baseline.

`\Lineola` The `\Lineola` command generates a horizontal line over the text given in the command argument. The command accepts an ⟨*optional argument*⟩, with the same meaning as the corresponding argument of `\lineola`. `\Lineola[-.25]{mrs}` yields $\overline{\text{mrs}}$.

`\structa` The `\structa` command takes two arguments and places the text given in the second argument over the text given in the first argument. Both arguments are implicitly considered to form TEX *groups*, as if each argument were enclosed by an additional set of braces. The command takes an ⟨*optional argument*⟩, a ⟨*factor*⟩ which is multiplied by the `ex` value for the font in use; the result is the amount by which the text in the second argument is raised (⟨*factor*⟩ > 0) or lowered (⟨*factor*⟩ < 0). `w\structa{xy}{\tiny ab}z` yields w$\overset{\text{ab}}{\text{xy}}$z.

Happy TĔXing !

# 4 Index of commands

The index does not include the commands listed on pp. 8 and 9