

TYPESETTING

ASTROLOGY

with

horoscop



MATTHEW SKALA
mskala@ansuz.sooke.bc.ca

Version 1.01, July 31, 2020

CONTENTS



1	Introduction	5
2	Prerequisites and Warnings	8
2.1	Font Support	8
2.2	Chart Graphics	8
2.3	External Programs	8
3	Package Options	10
3.1	Astrological Symbol Options	10
3.2	Calculation Backend Options	10
3.3	Optional Package Components	11
4	General Concepts	12
4.1	Objects and Variables	12
4.1.1	Objects	12
4.1.2	Cusps	13
4.1.3	Variables	13
4.2	Symbols	13
4.2.1	Naming scheme	13
4.2.2	Zodiac signs	14
4.2.3	Luminaries and traditional planets	14
4.2.4	Other objects, cusps, and angles	15
4.2.5	Aspects	16
4.2.6	Text angle and direction symbols	16
5	Calculating Horoscopes	17
5.1	Entering Chart Data	17
5.2	Calculating Object Positions	17
5.3	Entering Positions Manually	19
5.4	Saving and Manipulating Positions	20
5.5	Printing the Results as Text	22
6	Ready-Made Chart Templates	23
6.1	Basic Wheel Charts	23
6.1.1	Notes	23
6.1.2	Aspect webs	23
6.1.3	House and angle markings	26
6.1.4	Smart labels	26
6.2	Dial Charts	29
6.3	Decorative Wheel Charts	31

7	Advanced Topics	40
7.1	Rounding and Mixed-Base Conversion	40
7.1.1	The rounding problem	40
7.1.2	Automatic rounding	43
7.1.3	Manual rounding modes	44
7.2	Adding Custom Objects	45
7.3	Special Charts	45
7.4	Aspects Between Two Charts	48
7.5	Label Adjustments	48
7.6	Designing New Templates	54
7.6.1	Ticks and keys	54
7.6.2	Conditionals for use in templates	54
7.6.3	Drawing sets of graphical elements	55
7.6.4	Single linework objects	55
7.6.5	For-each and things to put in it	56
7.6.6	Drawing the aspect web	57
8	Implementation	58
8.1	Initial Option Handling	58
8.1.1	Symbol choices	58
8.1.2	Calculation backends	58
8.1.3	Higher-level features	58
8.1.4	Processing the options	59
8.2	Symbols	60
8.2.1	General symbols	60
8.2.2	Text abbreviations	61
8.2.3	Unicode astrological symbols	62
8.2.4	Symbols from wasysym	65
8.2.5	Symbols from marvosym	66
8.2.6	Symbols from starfont	67
8.3	Utilities	68
8.3.1	Trimming spaces	68
8.3.2	Redefine after current group	69
8.3.3	Polar to Cartesian	69
8.3.4	For-each	70
8.3.5	Double for-each	70
8.4	Basic Astrological Calculation Routines	71
8.4.1	Cusp information	71
8.4.2	Calculating harmonics	72
8.4.3	Shifting positions	72
8.4.4	Calculating equal cusps	73
8.4.5	Calculating midpoints	73
8.4.6	Angle adjustments	74
8.4.7	Chart rotation	74
8.4.8	Variable copying and saving	75
8.4.9	Sexagesimal arithmetic	76

8.4.10	Setting all data at once	82
8.5	Astrolog Calculation Back End	82
8.5.1	Name translation	82
8.5.2	House systems	83
8.5.3	Interface stuff	83
8.5.4	The actual calculation	84
8.6	Swiss Ephemeris Calculation Back End	87
8.6.1	Command-line options	87
8.6.2	House systems	88
8.6.3	Interface stuff	88
8.6.4	Calculation routines	89
8.7	Low-Level Chart Graphics	90
8.7.1	Configuration stuff	91
8.7.2	Horoscope environment and chart labels	91
8.7.3	Scratch dimension registers	92
8.7.4	Polar-coordinate puts	92
8.7.5	Arrowheads	93
8.7.6	Polar-coordinate curves	94
8.7.7	Locating objects	96
8.7.8	Cusps, ticks, and sign keys	97
8.7.9	Object labels	100
8.8	Aspect Web	105
8.8.1	Configuration settings	105
8.8.2	Per-aspect-type configuration	106
8.8.3	Drawing the aspect web	106
8.9	Internal House Labels	108
8.10	Object and Cusp Adjustment	109
8.10.1	Configuration settings	110
8.10.2	Internal variables	111
8.10.3	Support macros for adjustment iterations	111
8.10.4	Single adjustment cycles	113
8.10.5	Support macros for main loop	115
8.10.6	Main loop	116
8.11	Support Macros for Ready-Made Wheels	117
8.11.1	Recognizing houses	117
8.11.2	Support for Montreal template	119
8.11.3	Support for Quebec City template	119
8.11.4	Support for dial templates	120
8.11.5	Choosing text size	121
8.12	Ready-Made Wheel Templates	122
8.12.1	Basic wheel	122
8.12.2	Harmonic multi-dial charts	125
8.12.3	Decorative wheel charts	126

Chapter One

INTRODUCTION



This document describes the design and use of a \LaTeX package named `horoscop`, which supports typesetting of astrological charts. Features include:

- A unified interface for astrological symbols/glyphs, supporting three different astrological fonts as well as text abbreviations and Unicode astrological symbol code points.
- Support for invoking `Astrolog` or `Swiss Ephemeris` to calculate charts. Positions can also be specified manually.
- Loading and saving object and cusp positions into \TeX macros.
- Typesetting of angles and positions as text.
- Ready-made templates for basic wheel charts, dial charts including multidials with up to four sets of objects, and decorative wheel charts.
- Optional variations of the standard templates: aspect webs, highlighting for angular cusps, choice of what to include in object labels, house labels inside the houses.
- Low-level graphics functions for plotting in polar coordinates and building new templates.
- Labels move, and where necessary houses expand, to prevent crowding.

This package is primarily typesetting software, not astrological software. It is capable of interfacing to external packages to calculate things like object positions for charts, and it performs astrological computations like detection of aspects where that is directly needed for typesetting, but it does not do more generally astrological tasks like counting the objects in different elements. Things like time zones are not directly relevant to typesetting and left to the user to deal with.

Similarly, this documentation is about using `horoscop` to typeset charts into documents. It is assumed that readers have other sources for the astrological knowledge of what charts to typeset, what they mean, and what to say about them in the documents. Other software is probably more convenient for the earlier stages of exploring a chart and constructing an interpretation. This package becomes relevant after the interpretation is decided, when the time comes to publish a chart in an attractive printed form.

Many issues in astrology are subject to opinion, debate, and variations in personal preference. Where possible, `horoscop` avoids enforcing any specific line on such things. When it is necessary to choose a default (for instance, whether to use

♄ or ♅ for Capricorn) the system generally follows the author's preference with as much support for user customization as possible. The intention is to provide reasonably usable typeset charts—at least as good as the output of high-quality commercial charting software—right out of the box for users with minimal L^AT_EX skill, and also provide the capability for advanced users to customize the system to meet their own exact needs.

Some L^AT_EX users have philosophical objections to the very idea of astrology. Such issues are not addressed here.

Basic correctness is a design priority. Some other software will do things like typeset an object in the wrong house if there are too many objects in the right house for them to fit nicely; horoscop should never do that. The garbage in, garbage out principle applies, however, and horoscop is not responsible for problems occurring in external software; so if you ask for Placidus cusps from a birth in the Arctic Circle, horoscop will make its best effort to typeset whatever comes out of the calculation software but it will be your own fault if that is nonsense.

I presented this package at TUG 2016, with a one-page summary in the conference proceedings: Skala, M. 2016. Astrological charts with horoscop and starfont. *TUGboat*, 37(2):p. 182. Proceedings of the 37th Annual Meeting of the TeX Users Group (TUG 2016), Toronto, Ontario, July 25-27, 2016. A PDF of the summary is at <https://tug.org/TUGboat/tb37-2/tb116skala.pdf>; a PDF of the slides from my talk is at <https://tug.org/tug2016/slides/skala.pdf>; and there is video of most of the presentation at <http://zeeba.tv/astrological-charts-with-horoscop-and-starfont/>.

Version 1.0 represents a mature version of the package. The last release, version 0.92, was called a beta test, but it has been in use seven years now without needing any significant bug fixing or updates. The most serious issue in 0.92 was in the documentation of how to enter manually-calculated object positions; the underlying macros for it worked well.

Version 1.0 was primarily an update to the documentation to fix that issue; unfortunately, it still wasn't correct, so 1.01 is meant to fix it for real this time. Version 1.0 made other documentation updates to recognize the fact that the package is mature and fully usable. I also took the opportunity to add the Unicode-symbols feature, to reduce horoscop's ties to classical T_EX and its font systems as users increasingly move toward Unicode engines and Unicode-based font selection; and I have updated such things as Web links and development contacts to reflect the inevitable changes in human institutions over the years.

I encourage readers to think of Axel Harvey, a close friend of mine and a professional astrologer who contributed greatly to the testing of this software and its associated Web service. He died in 2016, and is missed.

There is an online chart service demonstrating horoscop at <https://edifyingfellowship.org/astro/>. I also maintain a page pointing to L^AT_EX astrology resources, at <https://ansuz.sooke.bc.ca/entry/107>.

I no longer encourage the use of GitHub and it is unlikely that the existing GitHub repository for horoscop will be updated further. Bug reports and other correspondence related to horoscop should be directed to mskala@ansuz.sooke.bc.ca.

As of 2016, I am no longer pursuing a career in academic computer science. I have started a business selling modules of my own design for Eurorack modular synthesizers, and kits for building them, through a Web storefront at <https://northcoastsynthesis.com/>. If you are interested in electronic music, please consider buying my products; if not, please pass on the link to others. My ability to spend time making free resources to share, such as the horoscop package, depends on the success of my business.

Chapter Two

PREREQUISITES AND WARNINGS



1. FONT SUPPORT

Typesetting astrological symbols (often called “glyphs”) requires an appropriate font, packaged for \LaTeX . This package supports three, selectable by package options: `marvosym`, `starfont`, and `wasysym`, none of which are included in a typical default \LaTeX installation. The `starfont` package is default and recommended, because the others have many symbols either missing or unsuitable. Instead of using an astrological font at all, the `textsym` option may be used to substitute abbreviations written in \LaTeX ’s ordinary text font, which should work in any \LaTeX environment; or the `unicode` option may be used with $X_{\text{q}}\LaTeX$ or $\text{Lua}\LaTeX$. With `unicode`, `horoscop` will use Unicode code points for the astrological symbols and leave it up to the Unicode-supporting $\text{T}_{\text{E}}\text{X}$ engine to typeset them in an appropriate font.

Sources for astrological font packages:

- `marvosym`: <https://ctan.org/pkg/marvosym>
- `starfont`: <https://ctan.org/pkg/starfont>
- `wasysym`: <https://ctan.org/pkg/wasysym>

2. CHART GRAPHICS

Typesetting astrological charts (which are distinct from symbols—you could use symbols in text without any charts, or conceivably charts without symbols) requires `pict2e` and `trig`; these *are* typically included in \LaTeX installations by default. An earlier version (predating the existence of `pict2e`) required `eepic`, and consequently `latex/dvips`, not `pdflatex`. But the long-promised `pict2e` package finally exists, allowing large circles and arbitrary-slope lines in the `picture` environment in many different engines, and that’s a nicer solution.

3. EXTERNAL PROGRAMS

Calling an external program to compute chart information requires a suitable external program. At present, `horoscop` supports `Astrolog` and `Swiss Ephemeris`. It is possible to typeset charts without using this feature, by coding all the object locations into the \LaTeX source file instead of having them calculated on the fly.

Sources for external calculation programs:

- `Astrolog`: <https://www.astrolog.org/astrolog.htm>
- `Swiss Ephemeris`: <https://www.astro.com/swisseph/>

External calculation programs require `\write18` support in T_EX. That is disabled by default, and should not be enabled lightly. With `\write18` support, T_EX documents can execute arbitrary commands on the host computer system, potentially giving them the ability to cause unlimited damage. No file containing data from an untrusted source should be run on an interpreter with `\write18` enabled. In particular, readers should be especially wary of using horoscop in automated chart-calculation services. If you create a Web form for people to enter birth data and automatically generate a L^AT_EX source file to typeset a chart, then you must carefully sanitize all the user-entered data on the server side.

The procedure for enabling `\write18` will vary depending on your T_EX installation; for obvious security reasons, it must be done out of band, and cannot be turned on by any commands given in the document. On a typical installation `\write18` might be enabled by a special command-line option to the T_EX interpreter. If you don't know what this stuff means and how to deal with it, then you shouldn't attempt to turn on `\write18`.

The external calculation programs' shell commands have only been tested under Linux. They will probably work on any system where the external software can be installed, but no guarantees are offered.

Users of horoscop should be aware of some issues related to the external calculation programs, although these issues are not directly issues with horoscop. First, the author of Astrolog *claims* a right under copyright to forbid commercial use not only of Astrolog, but of the factual information contained in Astrolog's output. Noting that in many jurisdictions copyright does not apply to factual information, the author of horoscop does not endorse the validity of any such claims by third parties; and horoscop itself is public domain and may be used without restriction, even commercially. However, commercial users of horoscop may prefer to select the Swiss Ephemeris support, which is default and not subject to such a claim.

The authors of Swiss Ephemeris restrict commercial distribution of their package, but do not claim to restrict commercial use of its output. Swiss Ephemeris is not intended by its developers to be an end-user product. They market it to other software developers for use as a module in the other developers' products. To use it with horoscop, you must install the library on your system as if you were preparing to write software using it yourself, and make sure all the paths are right so that T_EX can invoke the `swetest` executable and have it really work. This process will normally require at least a minimal level of C programming skill.

Users are obviously at their own risk with regard to license conditions set by the copyright holders of any and all third-party products. The horoscop package itself is released to the public domain in an effort to help stop the insanity of escalating license conditions imposed by other authors of astrological software.

To use the `egrep` option with the Swiss Ephemeris backend, the standard Unix `egrep` program must be available.

Chapter Three

PACKAGE OPTIONS



1. ASTROLOGICAL SYMBOL OPTIONS

The options `textsym`, `unicode`, `wasysym`, `marvosym`, and `starfont` control the font used for astrological symbols. You may choose more than one of these options, in which case more than one set will be available in your document. The set in force at the start of the document will be the last one on that list, from among the ones chosen; so `starfont` overrides `marvosym` which overrides `wasysym` which overrides `textsym`. If none of these options are specified, then `starfont` will be enabled by default.

If you have no astrological font support, or will not be using symbols and want to avoid the resource consumption, choose `textsym` to prevent `horoscop` from trying to load the missing packages. The abbreviations defined by `textsym` (and the `\horotextsym` macro below) are available regardless of package options; `textsym` is provided as an explicit option so users can override the default loading of `starfont`. The abbreviations are also used to replace missing symbols in the other sets.

The `unicode` option is normally only useful in \XeLaTeX and \LuaLaTeX ; it causes `horoscop` to use non-ASCII Unicode characters for astrological symbols. It is up to the \TeX engine, or the user, to select a font or combination of fonts in which these characters can be typeset. Both \XeLaTeX and \LuaLaTeX offer relevant font-selection features, and if `horoscop` detects (using the `iftex` package) some other engine, it will fall back to a text indication of the desired code points, which looks like $\langle\text{U+263D}\rangle$.

`\horotextsym`
`\horounicode`
`\horowasysym`
`\horomarvosym`
`\horostarfont`

The macros `\horotextsym`, `\horounicode`, `\horowasysym`, `\horomarvosym`, and `\horostarfont` switch to the corresponding set of astrological symbols, so that documents can mix the different sets. Any symbols not defined by the new set remain in the state left by the old set. To use these macros the corresponding symbol sets must have been loaded with the appropriate package options, except that `\horotextsym` is always available.

2. CALCULATION BACKEND OPTIONS

The `horoscop` package can get its chart information from either of two external calculation backends, or just use positions supplied by the user in macro definitions. The options `nocalc`, `astrolog`, and `swetest` choose among no backend (user must supply all positions), `Astrolog`, and the Swiss Ephemeris test program respectively. If more than one is supplied, `swetest` overrides `astrolog` which overrides `nocalc`. The default if none is specified will be `swetest`. Note that either external calculation program requires that the corresponding software be

installed properly on the system and that `\write18` be turned on; see the previous chapter for instructions and warnings regarding external software.

When using the `swetest` backend, it is possible to request filtering of `swetest`'s output via the `egrep` option. With this option, `swetest`'s output will be filtered through the `egrep` program (which must also be installed) to eliminate error and warning messages. Normally an error or warning from `swetest` will cause typesetting to fail with an uninformative message. This might occur for instance if the chart's date is not covered by the installed high-accuracy ephemeris files and Swiss Ephemeris reverts to its analytic model. Selecting `egrep` makes horoscopes try harder to typeset the document, working through the error or warning with whatever numbers come out of `swetest`.

3. OPTIONAL PACKAGE COMPONENTS

If this package will be used in a non-graphical document (for instance, to typeset text-based interpretations), it may be desirable to turn off the graphics support and avoid loading the graphics packages. That can be accomplished with the `nowheels` option. Similarly, the ready-made templates can be disabled with `notemplates`, which might be useful in documents that use user-defined templates exclusively. Selecting `nowheels` automatically causes `notemplates` to take effect also.

Chapter Four

GENERAL CONCEPTS

4

1. OBJECTS AND VARIABLES

Calculating and typesetting a chart requires keeping track of a number of pieces of information relating to luminaries, planets, asteroids, derived points, hypothetical bodies, and similar things. The pieces of information are referred to as “variables” and the things that have associated variables are collectively called “objects.” The values for variables are stored in macros named `\horo⟨object⟩⟨variable⟩`; for instance, `\horoSunPos` represents the Sun’s longitude. Numerical values should be stored in the macros as plain decimal numbers (with or without a fractional part, possibly negative where appropriate).

§1.1. Objects.—Standard object names include the luminaries and traditional planets Sun, Moon, Mercury, Venus, Mars, Jupiter, Saturn, Uranus, Neptune, and Pluto, and the asteroids Chiron, Ceres, Pallas, Juno, and Vesta. The asteroid Pallas is named that rather than common variations like “Pallas Athena” because its IAU name is just Pallas. The Lunar North node, as calculated by Astrolog, is called `NorthNode`; with Swiss Ephemeris the names `MeanNode` and `TrueNode` are used instead.¹ The Astrolog version will be whichever one Astrolog is configured to compute; on a default installation of Astrolog that is probably the mean version. The name `Lilith` refers to the Lunar apogee.

`\horoobjects`

Calculation and the chart templates use a macro called `\horoobjects`, which lists the names (comma separated) of all the objects to compute or typeset. The default value is all the object names defined in the previous paragraph except that only one North node will be used: `NorthNode` for Astrolog and `MeanNode` for Swiss Ephemeris. To exclude some objects, or choose a different node calculation, the user can redefine `\horoobjects` to the desired list of objects. Note that this macro, like the variable values, is used and parsed internally and so it should not be given a smart value; it should just be a simple list of names separated by commas. No whitespace either, please.

It is possible to create additional objects by defining the appropriate macros and adding the new names to the `\horoobjects` macro. An example of this process is given in Section 7.2. The Swiss Ephemeris backend will also automatically calculate objects called `Ascendant`, `MC`, `ARMC`, and `Vertex`; these four get calculated every time you calculate a chart whether you request them in `\horoobjects` or not.

¹Calling the osculating node “true” is deprecated, but too entrenched to override. The node is undefined except when the Moon is right on top of it, so the question is which of several reasonable assumptions to make when inventing a value in between those times, and there’s no reason to privilege one choice over the others.

`\horocusps` §1.2. **Cusps.**—House cusps are treated very much like additional objects with the names CuspI, CuspII, up to CuspXII. Like objects from `\horoobjects`, each cusp has an associated set of variables in macros with names like `\horoCuspIPos`. The macro `\horocusps` lists the cusps, like `\horoobjects`, but in general it should not be modified. Cusps will be calculated and used as appropriate; in general they should not be added to `\horoobjects`. In most present-day house systems, CuspI, CuspIV, CuspVII, and CuspX coincide with the ascendant, nadir, descendant, and zenith respectively. For systems where that is not the case, Swiss Ephemeris users may want to add Ascendant and MC to `\horoobjects` to plot those on the chart as additional objects.

§1.3. **Variables.**—The variable Pos, associated with every object and cusp, has already been mentioned. It represents the longitude of the object or cusp, measured in decimal degrees starting from $0^\circ \Upsilon = 0.0$. On that scale $0^\circ \text{♁} = 30.0$, $0^\circ \text{♂} = 60.0$, and so on.

The variable DPos represents “display Pos”; that is the actual location where the label or line representing an object or cusp will be plotted, which might not be the same as its Pos if it had to be moved to prevent interference with another object or cusp.

Vel represents an object’s velocity, in longitude. This should be in degrees per day, but the system actually only looks at its sign: positive for direct and negative for retrograde. The calculation backends calculate it along with Pos and the chart templates check for it to determine whether to display a R_x symbol in the label.

Variables called MPos, MDPos, and SPos are used internally by horoscop code to represent house midpoints, display positions of internal house labels, and saved position values for checking termination of the adjustment cycle. It should not be necessary for users to touch these. Use of a user-defined variable called XPos is demonstrated in Section 7.4.

The Symbol macros (described in the next section) are much like variables, although they have slightly different naming and do not take numeric values. The `\horo⟨object⟩SEOpt` macros used by the Swiss Ephemeris calculation backend are essentially string-valued variables too. Their use is explained in Section 7.2.

`\horocopyvar` The `\horocopyvar{⟨objects⟩}{⟨from⟩}{⟨to⟩}` macro copies the `⟨from⟩` variable to the `⟨to⟩` variable on all the objects in the comma-separated list given by `⟨objects⟩`. That operation is often used internally, and exposed to the user because it is occasionally useful to users also. For instance, if you set up the Pos values for your charts manually with `\def`,² then you must put the same values into DPos before invoking a chart template that will attempt label adjustment. The command `\horocopyvar{\horoobjects}{Pos}{DPos}` will copy all the Pos variables for objects in `\horoobjects` to the corresponding DPos variables.

2. SYMBOLS

§2.1. **Naming scheme.**—For every object there should be a symbol or glyph in a macro named `\⟨object⟩Symbol`. Note the absence of horo at the start of the

²It is traditional to warn readers to use `\newcommand` or `\renewcommand` instead of `\def`.

macro name; that's to save typing because these macros are probably of interest in general text outside the context of horoscop's chart features, and to separate the Symbol variable-like thing from real variables that have numeric values.

There are four basic sets of symbols, based on three font packages and the `textsym` option of using text abbreviations. See the package options discussion above for a description of how the choice between symbol sets is made. Some of the font packages are incomplete, and horoscop will attempt to make substitutions to fill in the blanks. Users can redefine the macros as needed, to use symbols from other fonts or for instance if it's desired to use $\text{\textcircled{3}}$ instead of $\text{\textcircled{B}}$ for Capricorn (starfont provides both).

The main use of these is for wheel charts, where you specify a list of objects and they all get typeset into the chart. It's necessary to provide a user-defined $\langle object \rangle Symbol$ for any new objects you add, if those objects will be typeset into wheel charts. The symbols can also be used by themselves in text. Note that horoscop will always invoke these symbols in text mode; if the desired symbol is a math-mode symbol, then the macro has to include the appropriate shift to math mode.

`\Zodiac` §2.2. **Zodiac signs.**—The system provides a $\backslash\text{Zodiac}\langle sign \rangle$ macro, whose parameter $\langle sign \rangle$ should be an integer from 1 to 12 choosing the sign from Aries=1 to Pisces=12; this typesets the symbol for the appropriate sign. It mimics, and replaces, the similar macro provided by `marvosym` and `starfont`; instead of going directly to the font characters, horoscop's $\backslash\text{Zodiac}$ calls the appropriate macro from $\backslash\text{AriesSymbol}\dots\backslash\text{PiscesSymbol}$, allowing the user to redefine individual sign symbols without needing to redefine all of $\backslash\text{Zodiac}$. Note that all use of sign symbols by horoscop (for instance, in chart wheel labels) goes through $\backslash\text{Zodiac}$, so it is possible to redefine the entire zodiac by redefining $\backslash\text{Zodiac}$ instead of redefining individual symbols.

macro	textsym	unicode	wasysym	marvosym	starfont
$\backslash\text{AriesSymbol}$	Ar	$\langle\text{U+2648}\rangle$	Υ	$\text{\textcircled{Y}}$	$\text{\textcircled{Y}}$
$\backslash\text{TaurusSymbol}$	Ta	$\langle\text{U+2649}\rangle$	$\text{\textcircled{B}}$	$\text{\textcircled{B}}$	$\text{\textcircled{B}}$
$\backslash\text{GeminiSymbol}$	Ge	$\langle\text{U+264A}\rangle$	$\text{\textcircled{C}}$	$\text{\textcircled{C}}$	$\text{\textcircled{C}}$
$\backslash\text{CancerSymbol}$	Cn	$\langle\text{U+264B}\rangle$	$\text{\textcircled{D}}$	$\text{\textcircled{D}}$	$\text{\textcircled{D}}$
$\backslash\text{LeoSymbol}$	Le	$\langle\text{U+264C}\rangle$	$\text{\textcircled{E}}$	$\text{\textcircled{E}}$	$\text{\textcircled{E}}$
$\backslash\text{VirgoSymbol}$	Vi	$\langle\text{U+264D}\rangle$	$\text{\textcircled{F}}$	$\text{\textcircled{F}}$	$\text{\textcircled{F}}$
$\backslash\text{LibraSymbol}$	Li	$\langle\text{U+264E}\rangle$	$\text{\textcircled{G}}$	$\text{\textcircled{G}}$	$\text{\textcircled{G}}$
$\backslash\text{ScorpioSymbol}$	Sc	$\langle\text{U+264F}\rangle$	$\text{\textcircled{H}}$	$\text{\textcircled{H}}$	$\text{\textcircled{H}}$
$\backslash\text{SagittariusSymbol}$	Sg	$\langle\text{U+2650}\rangle$	$\text{\textcircled{I}}$	$\text{\textcircled{I}}$	$\text{\textcircled{I}}$
$\backslash\text{CapricornSymbol}$	Cp	$\langle\text{U+2651}\rangle$	$\text{\textcircled{J}}$	$\text{\textcircled{J}}$	$\text{\textcircled{J}}$
$\backslash\text{AquariusSymbol}$	Aq	$\langle\text{U+2652}\rangle$	$\text{\textcircled{K}}$	$\text{\textcircled{K}}$	$\text{\textcircled{K}}$
$\backslash\text{PiscesSymbol}$	Pi	$\langle\text{U+2653}\rangle$	$\text{\textcircled{L}}$	$\text{\textcircled{L}}$	$\text{\textcircled{L}}$

§2.3. **Luminaries and traditional planets.**—These are well-supported by the various fonts.

macro	textsym	unicode	wasysym	marvosym	starfont
\SunSymbol	Su	⟨U+2609⟩	☉	☉	☉
\MoonSymbol	Mo	⟨U+263D⟩	☾	☾	☾
\MercurySymbol	Me	⟨U+263F⟩	♿	♿	♿
\VenusSymbol	Ve	⟨U+2640⟩	♀	♀	♀
\MarsSymbol	Ma	⟨U+2642⟩	♂	♂	♂
\JupiterSymbol	Ju	⟨U+2643⟩	♃	♃	♃
\SaturnSymbol	Sa	⟨U+2644⟩	♄	♄	♄
\UranusSymbol	Ur	⟨U+2645⟩	♅	♅	♅
\NeptuneSymbol	Ne	⟨U+2646⟩	♆	♆	♆
\PlutoSymbol	Pl	⟨U+2647⟩	♇	♇	♇

§2.4. Other objects, cusps, and angles.—The starfont package provides symbols for asteroids, derived points, angles, and so on. Without it, these things default to the textsym abbreviations. The unicode option also has code points for these. Cusp symbols are listed in this table, but they generally do not appear in the default templates and will seldom be used in actual practice.

macro	textsym	unicode	starfont
\ChironSymbol	Chi	⟨U+26B7⟩	♄
\CeresSymbol	Cer	⟨U+26B3⟩	♁
\PallasSymbol	Pal	⟨U+26B4⟩	♁
\JunoSymbol	Jun	⟨U+26B5⟩	♁
\VestaSymbol	Ves	⟨U+26B6⟩	♁
\NorthNodeSymbol	No	⟨U+260A⟩	♁
\SouthNodeSymbol	SNo	⟨U+260B⟩	♁
\LilithSymbol	Lil	⟨U+26B8⟩	♁
\CuspISymbol	ASC		A ^{sc}
\CuspIISymbol	2 nd		
\CuspIIISymbol	3 rd		
\CuspIVSymbol	IC		I ^c
\CuspVSymbol	5 th		
\CuspVISymbol	6 th		
\CuspVIISymbol	DSC		D ^{sc}
\CuspVIIISymbol	8 th		
\CuspIXSymbol	9 th		
\CuspXSymbol	MC		M ^c
\CuspXISymbol	11 th		
\CuspXIISymbol	12 th		
\AscendantSymbol	ASC		A ^{sc}
\MCSymbol	MC		M ^c
\VertexSymbol	Vtx		V ^x

Note that wasysym provides a symbol it calls \ascnode, but it’s actually identical to the symbol it calls \leo, and the symbol (♁) looks more like a Leo symbol; to prevent insanity, horoscop won’t use that symbol for the node unless you force the issue by redefining \NorthNodeSymbol. The corresponding \descnode symbol is just the Leo symbol again, turned upside-down; it is aesthetically offensive

enough that it also won't be used for South node by default.

`\MeanNodeSymbol` There are no well-agreed standard symbols for “mean” as opposed to “true”
`\TrueNodeSymbol` (osculating) definitions of the Lunar nodes. The macros `\MeanNodeSymbol` and
`\TrueNodeSymbol` by default typeset the current value of `\NorthNodeSymbol` with
a subscript M or T , as Ω_M and Ω_T .

Note that the default symbols for cusps assume an angular house system like that attributed to Placidus, in which the first house cusp is by definition equal to the ascendant, and so on. If you use a house system where that isn't true, and you will typeset symbols for the house cusps, then you must manually define appropriate symbols for them distinct from whatever symbols you are using for the actual angles. However, it's rare to need symbols for house cusps at all. Most wheel charts don't use them.

§2.5. Aspects.—When typesetting aspect webs, macros having names of the form `\horo<aspect>Symbol` will be used to print symbols identifying the aspects at the middle of each aspect line. These have the `\horo` prefix to keep them separated from the object symbols. If you define custom aspects, you should define corresponding symbols in the same pattern. The `marvosym` package does not define any aspect symbols (defaults to `textsym`), and `wasysym` only defines a few. The symbols for quintile and biquintile with `starfont`, and the one for trine with `wasysym`, are actually made from standard L^AT_EX symbols rather than using the named fonts. The table below also shows the default angles and orbs for the aspects; see Subsection 6.1.2.

macro	textsym	unicode	wasysym	starfont	angle
<code>\horoConjunctionSymbol</code>	Con	$\langle U+260C \rangle$	♌	♌	$0^\circ \pm 6^\circ$
<code>\horoOppositionSymbol</code>	Opp	$\langle U+260D \rangle$	♍	♍	$180^\circ \pm 6^\circ$
<code>\horoTrineSymbol</code>	Tri	$\langle U+25B3 \rangle$	△	△	$120^\circ \pm 5^\circ$
<code>\horoSquareSymbol</code>	Sqr	$\langle U+25A1 \rangle$	□	□	$90^\circ \pm 5^\circ$
<code>\horoQuintileSymbol</code>	Qnt			Q	$72^\circ \pm 2^\circ$
<code>\horoBiquintileSymbol</code>	Bqi			Q ²	$144^\circ \pm 2^\circ$
<code>\horoSextileSymbol</code>	Sex	$\langle U+26B9 \rangle$	✳	✳	$60^\circ \pm 4^\circ$
<code>\horoQuincunxSymbol</code>	Qcx	$\langle U+26BB \rangle$		⚡	$150^\circ \pm 3^\circ$
<code>\horoSemisextileSymbol</code>	Ssx	$\langle U+26BA \rangle$		⚡	$30^\circ \pm 3^\circ$
<code>\horoSemisquareSymbol</code>	Ssq	$\langle U+2220 \rangle$		∠	$45^\circ \pm 2^\circ$
<code>\horoSesquiquadrateSymbol</code>	Sqq	$\langle U+26BC \rangle$		⚡	$135^\circ \pm 2^\circ$

`\horodegrees` **§2.6. Text angle and direction symbols.**—Symbols for degrees, minutes, and
`\horominutes` seconds (`\horodegrees`, `\horominutes`, and `\horoseconds`) are provided for use
`\horoseconds` when an angle will be described numerically, as in $12^\circ 34' 56''$. As with other symbols, the macros can be redefined to change horoscop's behaviour. These three use standard L^AT_EX math symbols and do not require any special package support.

`\horoRetrogradeSymbol` The `\horoRetrogradeSymbol` macro is in a similar category: it will be used to typeset “retrograde” for labels that display that information.

Chapter Five

CALCULATING HOROSCOPES

h

1. ENTERING CHART DATA

Before erecting a chart we need the astrological locations of all the objects to be included. The horoscop package is not primarily a chart-computing system, but it can interface with other software to do the computation. The external interface takes the time and location data from macros named `\horocalcyear`, `\horocalcmonth`, `\horocalcday`, `\horocalctime`, `\horocalclon`, and `\horocalclat`, which should be defined to contain the year, month, day, time, longitude, and latitude respectively. Each one should be a decimal number, with only whole numbers for year and month; `\horocalctime` is the number of hours from midnight (so that a time like 4:30pm would be 16.5); and longitude and latitude are numbers of degrees. Use positive numbers for North and East and negative for South and West. Times and dates must be UTC. For example, the birth data for Uri Geller could be entered as follows.

`\horocalcyear`
`\horocalcmonth`
`\horocalcday`
`\horocalctime`
`\horocalclon`
`\horocalclat`

```
\renewcommand{\horocalcyear}{1946}
\renewcommand{\horocalcmonth}{12}
\renewcommand{\horocalcday}{20}
\renewcommand{\horocalctime}{0}
\renewcommand{\horocalclon}{34.76667}
\renewcommand{\horocalclat}{32.06667}
```

Setting all those macros one at a time is messy and inconvenient, so a simplified interface is available through the `\horocalcparms` macro, which sets them all at once and translates minutes and seconds, both of time and of arc.

`\horocalcparms`

```
\horocalcparms{1946}{12}{20}{0:0:0}{E34:46:0}{N32:4:0}
```

Although this macro is slightly more forgiving than the lower-level ones, one should nonetheless stick closely to the example format. In particular, do not omit the minutes and seconds even when they are zero.

2. CALCULATING OBJECT POSITIONS

`\horocalculate`

Once the chart data is defined, call `\horocalculate` to invoke the external calculation program and actually compute the object positions. As already mentioned, this requires `\write18` support in your \TeX interpreter, which is not the default and should not be turned on without caution.

The exact mechanics of chart calculation depend on the backend selected. With Swiss Ephemeris, which is the default, the package runs the “test” program

`swetest` once for each object and once more for all cusps, with the standard-output results directed into a `.hor` file which it will read in to get the data. With Astrolog, the `astrolog` program is run once and told to save its results directly to the `.hor` file as if to be read as options by a later invocation of Astrolog. Then the package reads that and parses the (undocumented) Astrolog command-line format to get the results.

Either way, the `\horoobjects` macro already described determines the list of objects whose positions will be calculated, and the positions go into the variables `Pos` (longitude), `DPos` (initially equal to longitude, but subject to later adjustment), and `Vel` (velocity). `Pos` and `DPos` are measured from $0^\circ\Upsilon$. `Vel` is degrees per day, positive for direct and negative for retrograde. Similar `Pos` and `DPos` values are calculated for the cusp pseudo-objects `CuspI`, `CuspII`, and so on (listed in `\horocusps`). In the case of Swiss Ephemeris, some extra pseudo-objects (`Ascendant`, `MC`, `ARMC`, and `Vertex`) are always calculated too.

The house system used in the calculation can be set by invoking a macro named like `\horo $\langle system \rangle$ Houses` before calling `\horocalculate`. The default for both backends is `\horoPlacidusHouses`. Astrolog supports systems called `Alcabitus`, `Campanus`, `Equal`, `EqualMC`, `Koch`, `Meridian`, `Morinus`, `NeoPorphyry`, `Placidus`, `PolichPage`, `Porphyry`, `Regiomontanus`, `Vedic`, and `Whole`. Swiss Ephemeris supports systems called `Alcabitus`, `Axial`, `Azimuthal`, `Campanus`, `Equal`, `Koch`, `Krusinski`, `Morinus`, `Placidus`, `PolichPage`, `Porphyry`, `Regiomontanus`, and `Vehlow`. Consult the external software’s documentation for details of these systems.

Please note that despite the inclusion of non-angular house systems like `Equal` among the choices for the Astrolog backend, their actual utility is severely limited because Astrolog overloads two objects it calls “`Asc`” and “`Mid`” to serve as both angles and house cusps; `Asc` can be the first house cusp, but then Astrolog will not write out the Ascendant, or it can be the Ascendant, but then Astrolog will not write out the first house cusp. A future version of this package will probably solve this issue by running Astrolog twice to get out both pieces of information. Another possibility would be to tell Astrolog to write out the angles, and then derive the first and tenth house cusps by calculating points opposite the seventh and fourth house cusps (which always seem to be house cusps rather than angles). Such a change might also be an opportunity to fix the annoying differences between the Astrolog and Swiss Ephemeris backends, as far as possible. In the mean time, the recommendation for Astrolog users who want equal houses is to select `Porphyry` (as a system that uses the Ascendant and Midheaven and does not break down near the poles) and then compute house cusps internally to `horoscop` as described below, rather than attempting to use Astrolog’s calculation. Note that the object names for the Ascendant and Midheaven will be `CuspI` and `CuspX`, not the Swiss Ephemeris names used in the examples.

`\horomakeequalcusps` It is possible for `horoscop` to calculate its own house cusps internally, to provide equal-sized houses starting from any position. Use `\horomakeequalcusps` to make twelve cusps spaced equally (30° apart) with `CuspI` set to the single argument. For instance, if using `swetest`, which calculates an object called `Ascendant` automatically, you can use `\horomakeequalcusps{\horoAscendantPos}` to make

equal houses starting from the Ascendant, simulating the “Equal” house system that actually is already provided anyway. That particular application may seem pointless; but the feature can also be used with any other object to create house systems unavailable from the backend. Someone who wanted to drop the concept of houses entirely, as for an unknown birth time, but still use a chart template that shows houses, might use `\horomakecusps{0}` to make houses equivalent to signs (Aries=first, Taurus=second, and so on).

The `\horomakeequalcusps` macro should be called after `\horocalculate` if it is to take effect, because they overwrite each others’ house cusp results. Any house system may be used for the original `\horocalculate` call because the results will be discarded anyway, except for the concern that a broken house calculation in `\horocalculate` (for instance, using polar latitudes with Placidus and without the `egrep` option) may cause typesetting of the whole document to fail.

`\horomakesigncusps`

A similar macro, `\horomakesigncusps`, makes equally-spaced cusps just like `\horomakeequalcusps` (in fact, using a call to it) but starting from the start of the sign that contains the argument value instead of starting at the argument value itself. This is useful for creating historical whole-sign systems. See also the use of `\horoshiftcusps` described in section 5.4 for ways to expand this feature to (for instance) simulate the Vehlow equal-house system, or base the system on the location of a cusp other than the first.

Note that in all cases horoscop follows the common astrological practice of computing house cusps based on the intersections of the house boundaries with the ecliptic and then placing objects into houses based solely on their longitude. In effect, the assumption is that all objects are located exactly on the ecliptic. For some house systems (with boundaries that do not follow lines of longitude) it can be argued that it would be better to assign house positions in a way that takes latitude into account. See the article “The Problems of House Division” by Deborah Houlding, available online at https://www.skyscript.co.uk/houprob_print.html, for more discussion of this issue. In some future version horoscop may be extended to provide more options for handling of this kind of thing.

Note also that in some house systems, the angles (A^{sc} , D^{sc} , M^{c} , and I^{c}) do not coincide with house cusps, and in such cases you may wish to compute and chart them as extra objects.

`\horoastrologopt`
`\horosweopt`

The macros `\horoastrologopt` and `\horosweopt` define extra command-line options for the external calculation programs (Astrolog and swetest respectively). These default to empty, but may be redefined to pass global configuration settings to these programs.

3. ENTERING POSITIONS MANUALLY

If it is not desired to compute the positions with horoscop’s external program interface, they can be entered manually instead. This approach might be useful if you use other software to do your calculations, or if you need to create \TeX source that will run on a system without the external calculation software installed. The macros that would be created by `\horocalculate` simply need to be created di-

rectly, as in the example below.

```
\def\horoSunPos{267.5002492}\def\horoSunVel{1.0185515}
\def\horoMoonPos{229.2067659}\def\horoMoonVel{11.9237313}
\def\horoMercuryPos{249.2355412}\def\horoMercuryVel{1.4004420}
\def\horoVenusPos{229.6610785}\def\horoVenusVel{0.4165367}
\def\horoMarsPos{271.9242409}\def\horoMarsVel{0.7571908}
\def\horoJupiterPos{228.1368358}\def\horoJupiterVel{0.1927896}
\def\horoSaturnPos{128.1056809}\def\horoSaturnVel{-0.0516070}
\def\horoUranusPos{79.4428694}\def\horoUranusVel{-0.0421386}
\def\horoNeptunePos{190.6482218}\def\horoNeptuneVel{0.0133648}
\def\horoPlutoPos{133.0955746}\def\horoPlutoVel{-0.0150157}
\def\horoNorthNodePos{70.7814892}\def\horoNorthNodeVel{-0.0529425}
\def\horoLilithPos{265.3475147}\def\horoLilithVel{0.1108368}
\def\horoCuspIPos{207.9120843}
\def\horoCuspIIPos{236.5553269}
\def\horoCuspIIIPos{267.6976404}
\def\horoCuspIVPos{300.5365877}
\def\horoCuspVPos{332.9973490}
\def\horoCuspVIPos{2.5292853}
\def\horoCuspVIIPos{27.9120843}
\def\horoCuspVIIIPos{56.5553269}
\def\horoCuspIXPos{87.6976404}
\def\horoCuspXPos{120.5365877}
\def\horoCuspXIIPos{152.9973490}
\def\horoCuspXIIIPos{182.5292853}
```

A few less obvious macros also need to be set for manually-entered positions to work correctly. Load `\horoobjects` with a list of the objects with manual positions, and set the `DPos` values to match the `Pos` values. The `\horocopyvar` macro can help with setting `DPos`. Finally, set the `\horocalculatedtrue` flag so that other parts of the system will know there is valid data in the variables. That flag is normally set by the calculation interface when it has read usable data from the backend.

`\horocalculatedtrue`

```
\def\horoobjects{Sun,Moon,Mercury,Venus,Mars,Jupiter,Saturn,%
  Uranus,Neptune,Pluto,NorthNode,Lilith}
\horocopyvar{\horoobjects}{Pos}{DPos}
\horocopyvar{\horocusps}{Pos}{DPos}
\horocalculatedtrue
```

4. SAVING AND MANIPULATING POSITIONS

In documents that contain multiple charts it may be useful to save the results of one calculation for use later. Something similar is also required for single charts that contain more than one person's objects, or for mixing the results of one calculation with another as when making synastry charts. The package provides macros for storing results in macros to handle these situations.

`\horosaveobjects` The `\horosaveobjects` macro takes one argument which is the name of a new macro to create (or to redefine, without warning, if it already exists). All the Pos and Vel values for objects in `\horoobjects`, and the current value of `\horoobjects` itself, are stored in the newly defined macro. When that macro is run later, it will restore all those values, and set DPos values to the restored Pos values. The effect is that `\horosaveobjects` creates a macro representing the object-related results of the last `\horocalculate`, so we can return to the current state later.

`\horosavecusps` The `\horosavecusps` macro does something similar, but instead of saving the objects it saves the current values of `\hororightcoord`, `\horocusps`, and the Pos of all cusps. This function is split into a separate macro to make it easy to create synastry charts: calculate one chart, save the objects, calculate the other chart, and restore the first chart's objects, to get one chart's objects in the other's houses.

`\hororotatechart` Some manipulations can be applied to calculated positions to alter the appearance of the resulting chart. First, the `\hororotatechart{object}{angle}` macro will rotate a chart to place a specified object at a specified place. The default is for CuspVII (the descendant) to be at the right. The angle must be specified in degrees according to the mathematical convention: 0 is to the right and it increases counterclockwise, so 90 is up, 180 is to the left, and 270 is down. For instance, `\hororotatechart{Sun}{90}` would place the Sun at the top. This macro works by manipulating an internal macro called `\hororightcoord`, mentioned above as one of the things saved by `\horosavecusps`. This rotation does not change the actual stored positions of the objects and cusps, only the way in which they will be plotted on the page.

`\horocalcharmonic` The `\horocalcharmonic{harmonic}` macro multiplies all the Pos values of objects in `\horoobjects` by its argument and sets the Pos and DPos values to the result. The harmonic should be an integer, and no more than 45 (because of \TeX 's limits on number magnitude). Higher harmonics, if they are not prime numbers, can be achieved by calling it twice; for instance, with 20 and 10 to get 200. The internal representation of Zodiac positions is only good to approximately 1/18 of a second of arc in the original position; that will be multiplied by the harmonic, limiting the precision of the result for high harmonics. Your original input data is probably even less good than that. The precision of positions in high-harmonic charts is inherently limited by the nature of high-harmonic charts, and users must understand that. Note also that this macro does not change the house cusp positions. The most popular current practices seem to be to leave cusps unchanged, or not to use them at all, in harmonic charts; and multiplying them like object positions could lead to problematic situations such as cusps ending up out of order, especially in higher harmonics.

`\horoshiftobjects`
`\horoshiftcusps` The macros `\horoshiftobjects` and `\horoshiftcusps` apply an additive shift, specified in decimal degrees as the single argument, to all the object or cusp Pos values respectively, and sets the Pos and DPos values to the results. These macros, unlike `\horochartrotate` above, do change the stored values of the variables; they can be seen as moving the planets instead of moving the chart. This kind of shift can be applied to object positions to create solar arc charts. With cusp positions, it can be used to create a variety of equal-house systems that may

or may not also be available directly from the calculation backends. For instance, assuming the `swetest` backend, this code will calculate whole-sign houses such that the midheaven is contained in the tenth house:

```
\horocalculate
\horomakesigncusps{\horoMCPos}
\horoshiftcusps{90}
```

That says: calculate the chart, including the implicit pseudo-object “MC”; create whole-sign houses with `CuspI` set to the start of the sign containing the MC pseudo-object and `CuspX` three signs (90°) earlier than that; and then shift the cusps forwards 90°, so that we end up with `CuspX` set to the start of the sign containing the MC object. A similar chart but with the midheaven actually on the tenth house cusp (instead of using whole signs), duplicating Astrolog’s `EqualMC` system, could be obtained by substituting `\horomakeequalcusps` for `\horomakesigncusps`.

In charts like these where the cusps do not necessarily coincide with the angles, one would probably also want to include MC and Ascendant in `\horoobjects` so that they will be plotted like objects on the chart. The angle-highlighting features of the standard templates remain associated with the angular cusps (`CuspI`, `CuspIV`, `CuspVII`, and `CuspX`) rather than the angle pseudo-objects like Ascendant and MC. It may be appropriate to turn off the highlighting features for charts in which the angular cusps are not actually the angles, but that is left to the user’s discretion.

5. PRINTING THE RESULTS AS TEXT

Instead of or in addition to printing a wheel chart, one may want to print the numerical values of object positions as text with astrological symbols. The `\horodsmstext` macro prints its argument as an astrological longitude with degree, (Zodiac) sign, minutes, and seconds. The argument can be a macro set by `\horocalculate` or a raw number in decimal degrees. For instance, `\horodsmstext{\horoVenusPos}` might print 19°♍,39’40”.

`\horodsmstext`

The `\horotimetext` has a similar function for times. Its argument should be hours after midnight, and it prints the value as hours, minutes, and seconds separated by colons. For instance, `\horotimetext{12.58222}` prints 12:34:56. This might typically be used with `\horocalctime`.

`\horotimetext`

For latitudes and longitudes, `\horolatlonstext` takes three arguments. The first is the number of degrees, which may be positive or negative, and it prints the number of degrees, minutes, and seconds followed by either the second or third argument depending on whether the angle was positive or negative. For instance, `\horolatlonstext{50}{N}{S}` and `\horolatlonstext{-40}{E}{W}` print 50°0’0” N and 40°0’0” W.

`\horolatlonstext`

Chapter Six

READY-MADE CHART TEMPLATES



1. BASIC WHEEL CHARTS

`horoscope` The package provides several ready-made chart templates for different purposes.
`\horowheelVancouver` The basic wheel chart, similar to the best ones produced by other software, is called Vancouver. It's easy to use in the simplest case: just open a horoscope environment and invoke `\horowheelVancouver`, as in the birth chart for John Lennon shown in Figure 1.

Please note that the standard templates are named after Canadian cities, but these names are only *thematic names for the visual styles of the charts*³ and the names of the templates have no connection with the geographic locations the charts refer to. You can plot a chart for any location in Vancouver style, and you can plot a chart that does happen to be for Vancouver (any of the multiple places in the world with that name) using any template. The geographic location for the chart must be specified by latitude and longitude, independently of the template name.

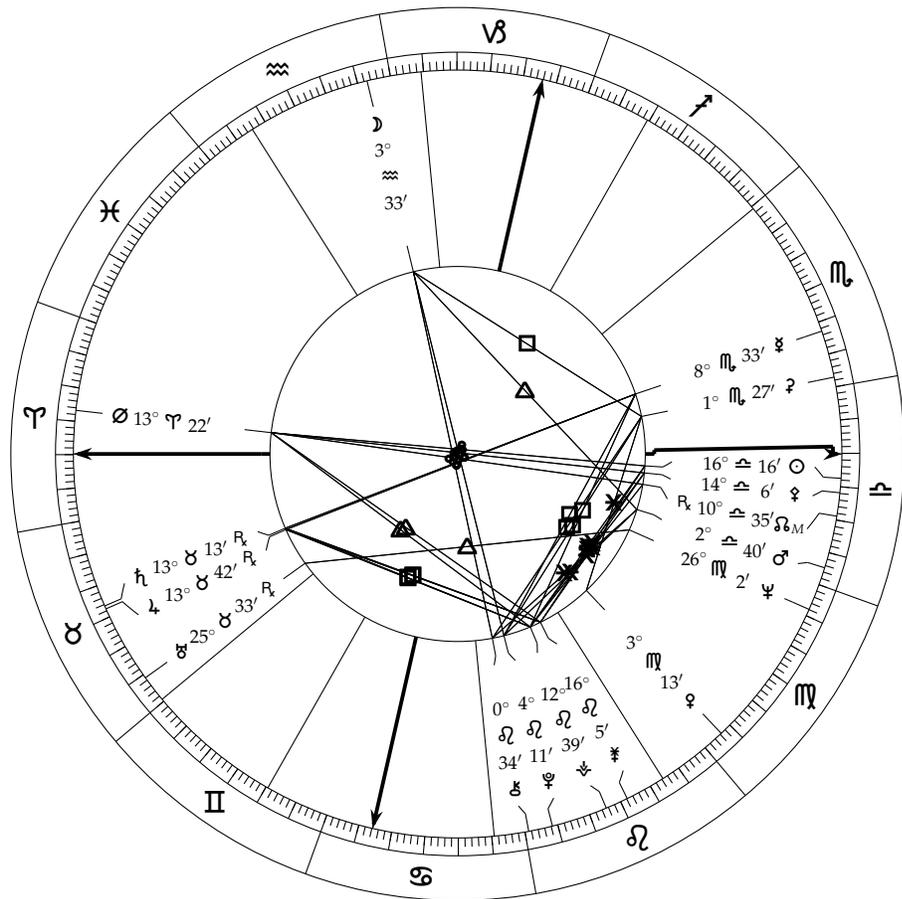
`\horounitlength` The horoscope environment is a picture environment, 100 unit-lengths square with the origin in the centre, so that coordinates range from $-50 \dots 50$ on both axes. The unit length is set to `\horounitlength`, which defaults to 0.00952 times `\textwidth`, so that a horoscope environment will be just slightly less than the full text width of the page. Change it to change the size of horoscope wheels, though if it is changed much the text size will need to be adjusted also.

`\horoULnote` **§1.1. Notes.**—Five macros are provided for typesetting notes in the corners or centre of a wheel chart. They are called `\horoULnote`, `\horoURnote`, `\horoLLnote`, `\horoLRnote`, and `\horoCnote`, for upper left, upper right, lower left, lower right, and centre respectively; each takes a single argument specifying the text to put there, which may include `\\` commands for line breaks. Typesetting is flushed into the corners for corner notes and centred for the centre note. If you're using a chart style that puts things of its own in the space where you want to put a note, then the note will collide with the chart, so depending on configuration it may not always make sense to use all five of these. Figure 2 demonstrates the use of these macros.

`\horoaspectwebtrue` **§1.2. Aspect webs.**—The pattern in the centre of the Vancouver wheel, showing which objects are or are not in aspect to each other, is called the *aspect web*. It can be turned on and off with `\horoaspectwebtrue` and `\horoaspectwebfalse`.

`\horoaspectobjectsa` By default, the aspect web will show aspects between any two objects from `\horoobjects`. That may not be desirable in charts with many minor objects; an aspect between two small asteroids may not be important enough to be worth

³I was inspired to use this naming theme by Ruskey and Weston's work on Venn diagrams; see <https://www.combinatorics.org/files/Surveys/ds5/ds5v3-2005/VennEJC.html>.



```

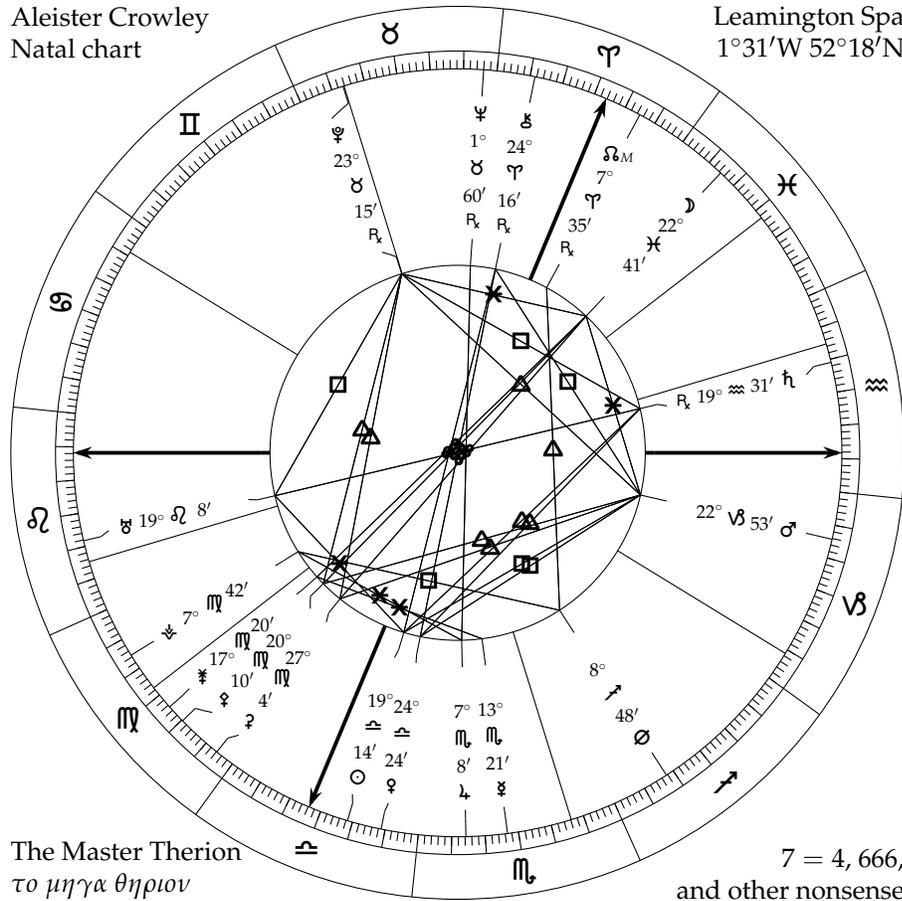
\horocalparms{1940}{10}{9}{17:30:0}{W2:55:0}{N53:25:0}
\horocalculate
\begin{horoscope}
  \horowheelVancouver
\end{horoscope}

```

Figure 1: Basic wheel chart (John Lennon).

Aleister Crowley
Natal chart

Leamington Spa
1°31'W 52°18'N



The Master Therion
το μηγα θηριον

7 = 4, 666,
and other nonsense

```

\horocalparms{1875}{10}{12}{23:30:0}{W1:31:0}{N52:18:0}
\horocalculate
\begin{horoscope}
\horowheelVancouver
\horoULnote{Aleister Crowley\\Natal chart}
\horoURnote{Leamington Spa\\
1\horodegrees31\horominutes W 52\horodegrees18\horominutes N}
\horoLLnote{The Master Therion\\
 $\tau \omicron \mu \eta \gamma \alpha \theta \eta \rho \iota \omicron \nu$ }
\horoLRnote{$7=4$; 666;\\and other nonsense}
\end{horoscope}

```

Figure 2: Corner notes.

displaying in the web. You can override the default to show only selected aspects by defining new values for `\horoaspectobjectsa` and `\horoaspectobjectsb`. Each should be a comma-separated list of object names; an aspect will be displayed if it goes between one object from one list and one from the other. Thus, if you set one to a list of objects to consider “major” while leaving the other set to all objects, you will get an aspect web showing only aspects that involve at least one major object, as in Figure 3.

`\horoaspects` The `\horoaspects` macro stores a list of aspects to include in aspect webs, much in the manner that `\horoobjects` lists objects to include in charts. The default is to include Opposition, Trine, Square, and Sextile. The other predefined aspects users can add to the list are all those for which Symbol macros were defined earlier: Conjunction, Quintile, Biquintile, Quincunx, Semisextile, Semisquare, and Sesquiquadrate. Other (user-defined) aspects may be added by defining `\horo<aspect>Symbol`, `\horo<aspect>Angle`, and `\horo<aspect>Orb` macros, with the angle and orb given in decimal degrees.

§1.3. House and angle markings.—The ready-made wheel templates also support several options for marking the houses and the angles. Exactly which options are supported depends on the template; the Vancouver template supports them all.

`\horoínhouselabelstrue`
`\horoínhouselabelsfalse`
`\horohouselabel` Internal house labels are numbers that appear inside the houses among the object labels, selected with the `\horoínhouselabelstrue` and `\horoínhouselabelsfalse` macros. The default is false, because they take up a fair bit of space and can lead to crowding. By default they are uppercase Roman numerals; you can make them be something else by redefining the `\horohouselabel` command to print the `horohouse` L^AT_EX counter in the desired style.

`\horoboldanglestrue`
`\horoboldanglesfalse`
`\horoanglecuspswidth` With `\horoboldanglestrue` and `\horoboldanglesfalse` the user can select whether to display angular house cusps as extra-bold lines. The default is true. The actual width to use is set by the L^AT_EX length `\horoanglecuspswidth`, which defaults to 1.44pt.

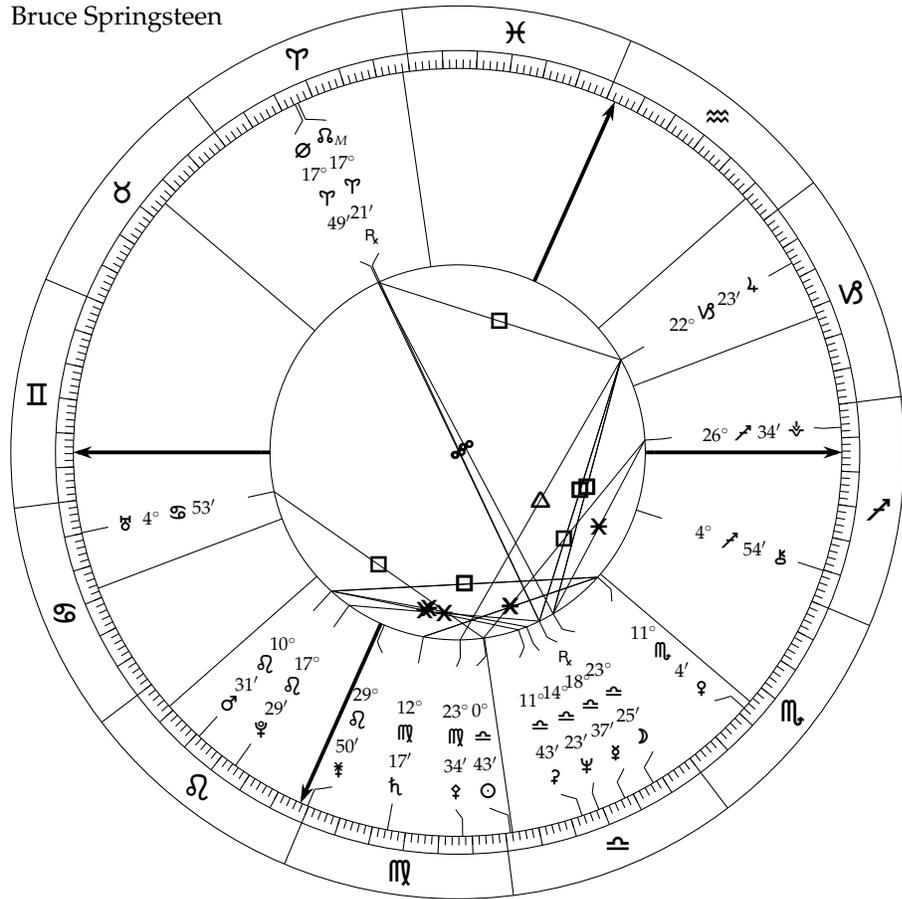
`\horoanglearrowstrue`
`\horoanglearrowsfalse` Angular cusps can also be highlighted by giving them arrowheads; this is selected with `\horoanglearrowstrue` and `\horoanglearrowsfalse`, default true. The size of the arrowheads is fixed as part of the template.

Figure 4 shows all the cusp-highlighting features set to the opposite from their defaults.

§1.4. Smart labels.—By default, the Vancouver chart template will display a label for each object containing its symbol, longitude down to the minute of arc, and the retrograde symbol if the object is retrograde. It will automatically adjust the sequence of the different parts of the label so that the longitude part will read in degree-sign-minute order in as close as possible to left-to-right, top-to-bottom order, with the object’s symbol at the outside near the rim of the wheel and the retrograde symbol on the inside near the hub. However, this labelling scheme can be customized in several ways.

First, the `\horowheelVancouver` command can take an optional argument which is a format string specifying what “chunks” should be typeset in each ob-

Bruce Springsteen



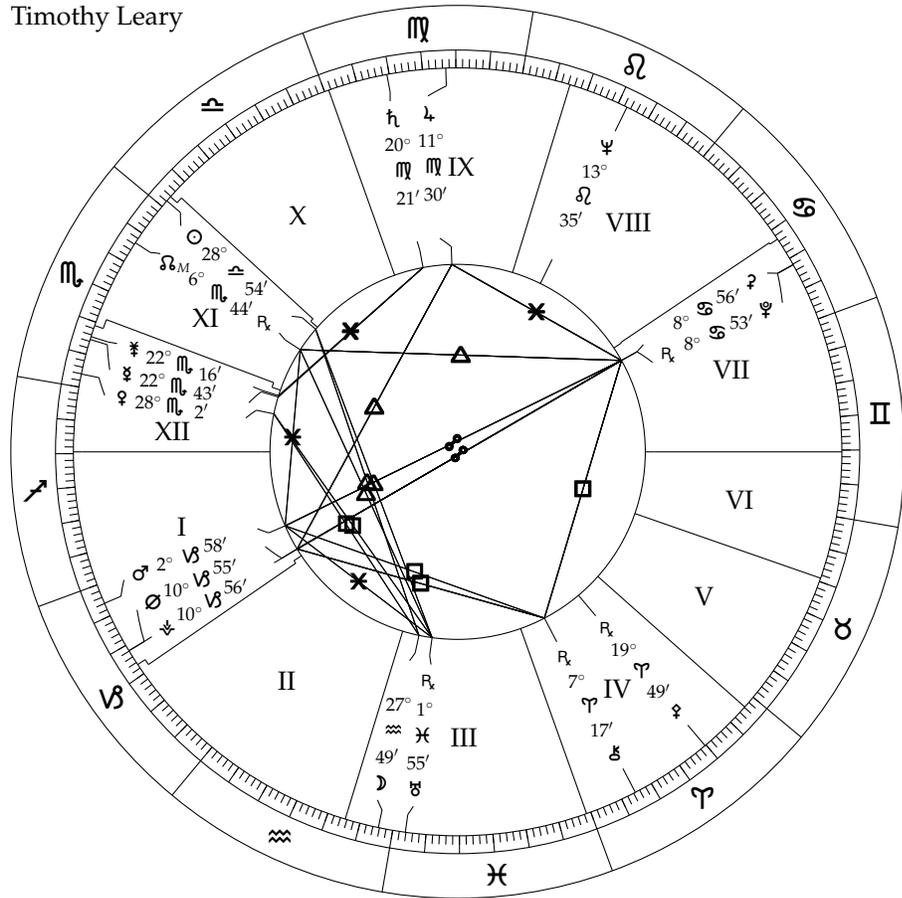
```

\horocalparms{1949}{9}{24}{2:50:0}{W74:17:0}{N40:16:0}
\horocalculate
\begin{horoscope}
  \renewcommand{\horoaspectobjectsa}%
    {Sun,Moon,Mercury,Venus,Mars,Jupiter,Saturn}
  \horowheelVancouver
  \horoULnote{Bruce Springsteen}
\end{horoscope}

```

Figure 3: Limiting the objects for the aspect web.

Timothy Leary



```

\horocalparms{1920}{10}{22}{14:45:0}{W72:35:0}{N42:06:0}
\horocalculate
\begin{horoscope}
  \horointhouselabelstrue
  \horoboldanglesfalse
  \horoanglearrowsfalse
  \horowheelVancouver
  \horoULnote{Timothy Leary}
\end{horoscope}

```

Figure 4: Cusp-highlighting features

ject label. The format string should be some sequence of the letters d, m, s, z, y, and r, which refer respectively to Degrees, Minutes, and Seconds of longitude, Zodiac sign symbol, object sYmbol, and possible Retrograde. If a label string like ydzmsr is specified it will be read in an inward direction; that example would typeset object symbols at the outside, then degrees, sign, minutes, seconds, and retrograde at successively smaller radii.

If two label strings are specified separated by a slash, then the first one will be used for all labels typeset above an imaginary line drawn at a 45° angle from lower left to upper right, and the second string will be used for labels below that line. The idea is that for the first class of labels, the preferred reading direction will be inward, and for the second class it will be outward, so by reversing the appropriate letters between the two strings one can keep sections of the label reading in the correct direction. The default format string is ydzmr/ymzdr, which implements the default behaviour described above.

`\horotextsize`

The Vancouver chart template automatically adjusts the size of text in its labels according to how much detail was selected in the labels. Formally, it counts the number of letters in the format string (the first one if two were specified) and uses that to index into the sequence of L^AT_EX type size commands `\Large`, `\large`, `\normalsize`, `\small`, `\scriptsize`, `\scriptsize` (twice—by trial and error, that seems to work best), `\tiny`. That results in larger text for shorter labels. Similar trickery changes the radii on which the different chunks are shown, to keep the labels reasonable-looking even when different levels of detail are chosen. However, this process is by no means foolproof and the result may not be the desired size. If necessary, the `\horotextsize` macro can be changed with `\renewcommand` to a positive or negative size adjustment. For instance, if it is set to 2, the labels will use text two sizes larger than the default.

Figure 5 is an example with lower-detail labels than default (showing longitude only down to the degree) and larger text. Rounding options described in section 7.1 are used because the default rounding assumes the labels will show minutes.

2. DIAL CHARTS

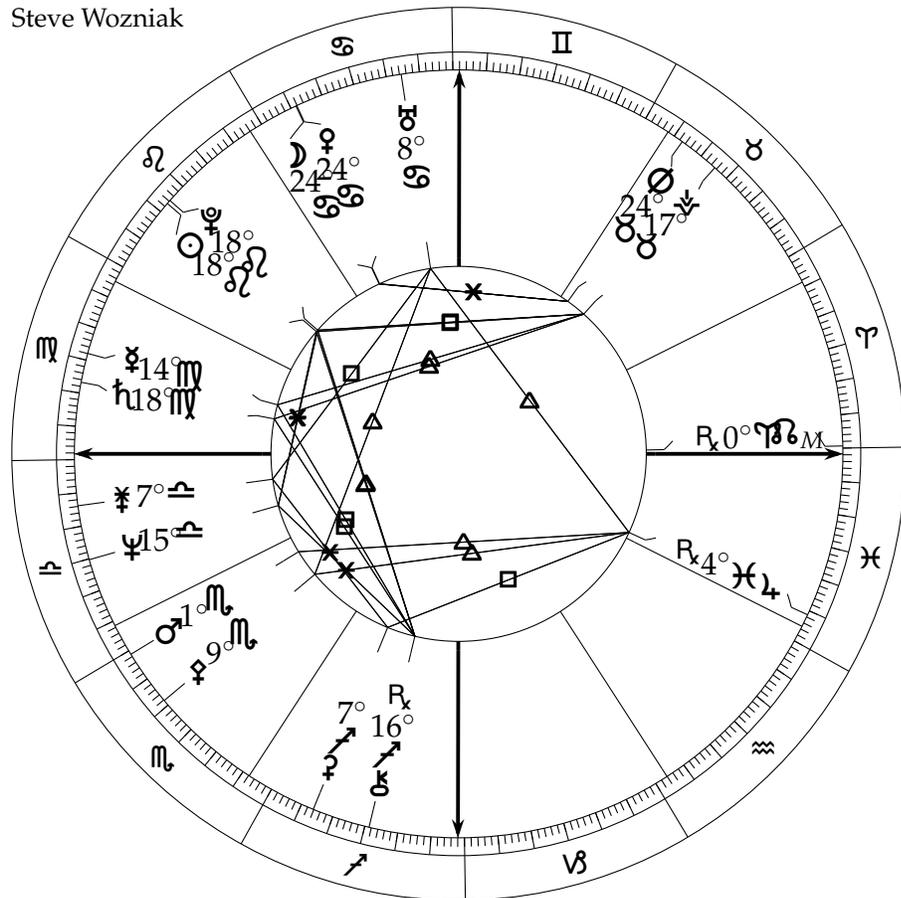
The Vancouver chart template is designed for a traditional style of wheel chart with house cusps featuring prominently. For some kinds of interpretation, you may want to focus more on the angles between objects without much reference to house cusps. The dial chart templates are designed to support that kind of view; they emphasize angles and do not include cusps. They are loosely based on the style of charts popular in Cosmobiology; they might also be worthwhile for charts where the birth time or location are unknown, so that house cusps cannot be meaningfully calculated. Templates are offered for comparing up to four sets of objects.

All dial charts are printed with the 0°♁ mark at the top. In the case of harmonic charts, that may coincide with other longitudes as well—for instance, 0°♁, 0°♂, and 0°♃ on a 90° dial.

`\horowheelqaluit`

Figure 6 shows the basic single 360° dial chart style, available through the

Steve Wozniak



```

\horocalparms{1950}{8}{11}{16:45:0}{W121:53:0}{N37:20:0}
\horocalculate
\begin{horoscope}
  \renewcommand{\horotextsize}{2}
  \hororoundautofalse
  \hororoundtodegkeepsign
  \horowheelVancouver[ydzt/yzdr]
  \horoULnote{Steve Wozniak}
\end{horoscope}

```

Figure 5: Custom label string and text size adjustment.

`\horowheelIqaluit` macro. This macro, like all the dial chart macros, has an optional first argument for a harmonic number. The popular 90° dial is available with `\horowheelIqaluit[4]`, as shown in Figure 7. Other harmonics may be chosen to give dials with other numbers of degrees.

All the dial templates support an aspect web, turned on by default. The aspect web will be based on whatever values are currently in the Pos variables when the chart is drawn, which will normally be the last ones calculated. Most users will probably prefer that those be the positions for the innermost dial in the case of a multi-dial chart, but by careful sequencing or use of `\horosaveobjects`, other selections are possible.

As seen in Figure 7, the aspects refer to the object longitudes *after* any harmonic transformation. For instance, Hefner’s Moon and Chiron are shown as in opposition in the aspect web because they are close to 180° apart on the chart. It is a fourth-harmonic chart, the objects are actually about 45° apart in the sky, and so the aspect is really a semisquare. His Jupiter and Saturn, on the other hand, really are square to each other in the sky, but appear as conjunct on the dial. Users who choose to use an aspect web with harmonic charts are assumed to understand these issues. If desired, the aspect symbols may be redefined to correspond to aspects in the sky (before harmonics) rather than on the chart (after harmonics).

If the aspect web is turned off, the dial chart will include a small cross at the centre of the dials, for centering the moveable pointer that some interpreters like to use. Note that that style of interpretation is not part of the author’s background; people who do want to do it are encouraged to comment on how the package’s features could be better tuned to their purposes.

`\horowheelIglolik`

The `\horowheelIglolik` command creates a double dial, as shown in Figure 8. It has two required arguments which should be saved object position macros created by the `\horosaveobjects` command, as illustrated by the example code. The first will be used for the inner dial and the second for the outer dial. An optional argument in square brackets may be added before the required ones (in the standard L^AT_EX usage) for a dial of less than 360°.

`\horowheelRankin`
`\horowheelResolute`

Triple and quadruple dials are also available, through `\horowheelRankin` and `\horowheelResolute` respectively. These are illustrated in Figures 9 and 10. Their operation is fundamentally the same as `\horowheelIglolik`, just extended to three or four sets of objects.

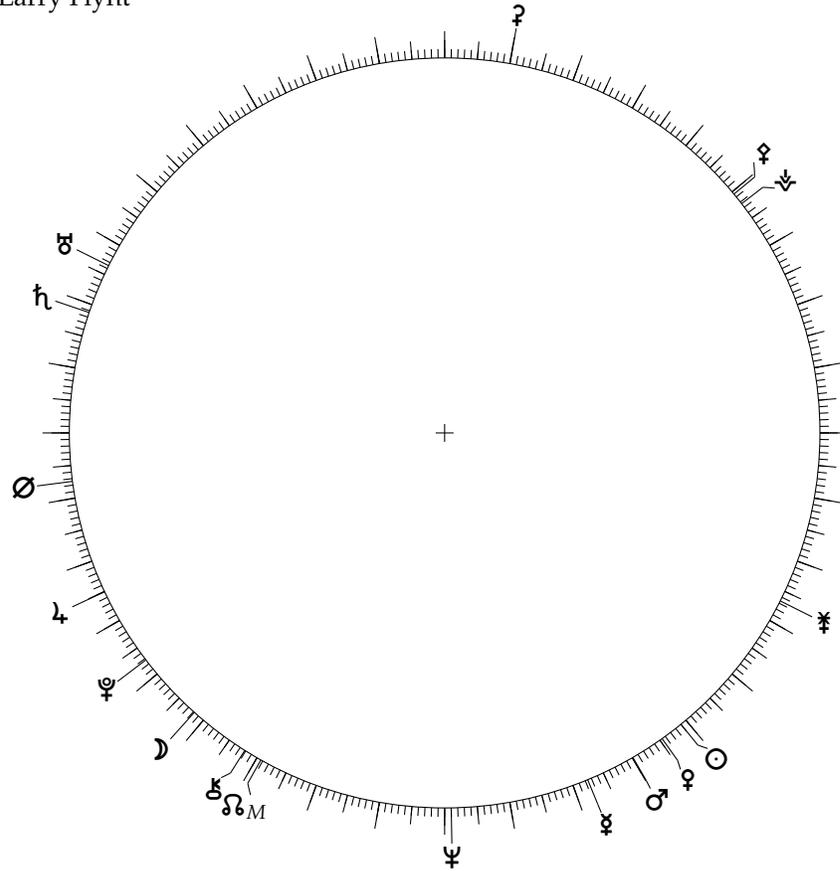
3. DECORATIVE WHEEL CHARTS

The standard wheel chart emphasizes house cusps, and the dial charts concentrate on longitudes without showing the houses. For other styles of interpretation it may be desirable to put more emphasis on the houses containing objects and less on the geometry of objects’ physical locations in the sky. The decorative wheel chart templates are designed to support that kind of emphasis, provide visually appealing designs similar to some historical chart styles, and demonstrate the possibilities for customized templates.

`\horowheelMontreal`

The `\horowheelMontreal` macro generates a chart like that shown in Figure 11. It shows labels for the angular cusps down to the minute of arc and for

Larry Flynt

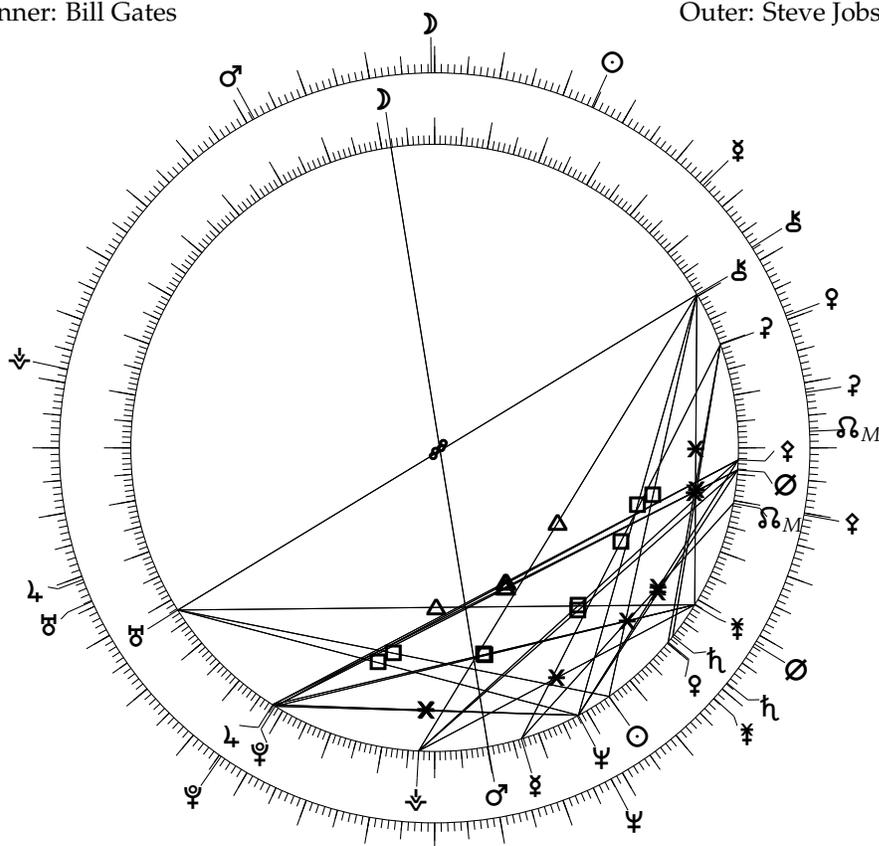


```
\horocalparms{1942}{11}{2}{2:10:0}{W83:04:0}{N37:45:0}  
\horocalculate  
\begin{horoscope}  
  \horoaspectwebfalse  
  \horowheelIqaluit  
  \horoULnote{Larry Flynt}  
\end{horoscope}
```

Figure 6: Single 360° dial (Iqaluit).

Inner: Bill Gates

Outer: Steve Jobs

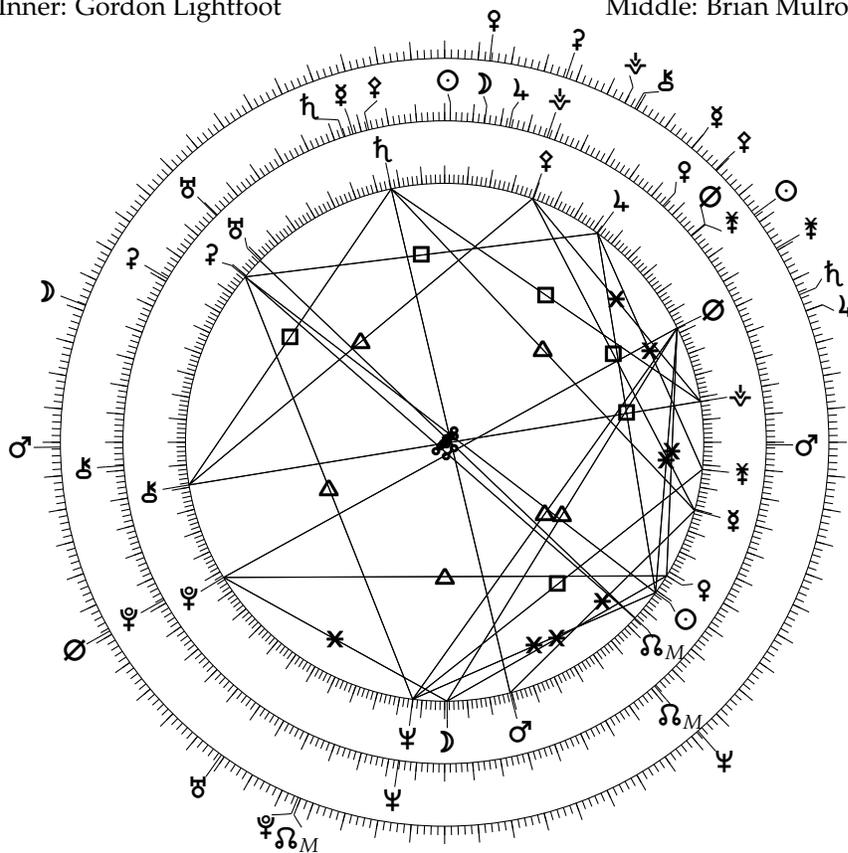


```
\horocalparms{1955}{2}{24}{15:0:0}{W122:25:0}{N37:46:0}  
\horocalculate\horosaveobjects{\SteveJobs}  
\horocalparms{1955}{10}{29}{6:0:0}{W122:20:0}{N47:36:0}  
\horocalculate\horosaveobjects{\BillGates}  
\begin{horoscope}  
  \horowheelIgllookik{\BillGates}{\SteveJobs}  
  \horoULnote{Inner: Bill Gates}  
  \horoURnote{Outer: Steve Jobs}  
\end{horoscope}
```

Figure 8: Double 360° dial (Igllookik).

Inner: Gordon Lightfoot

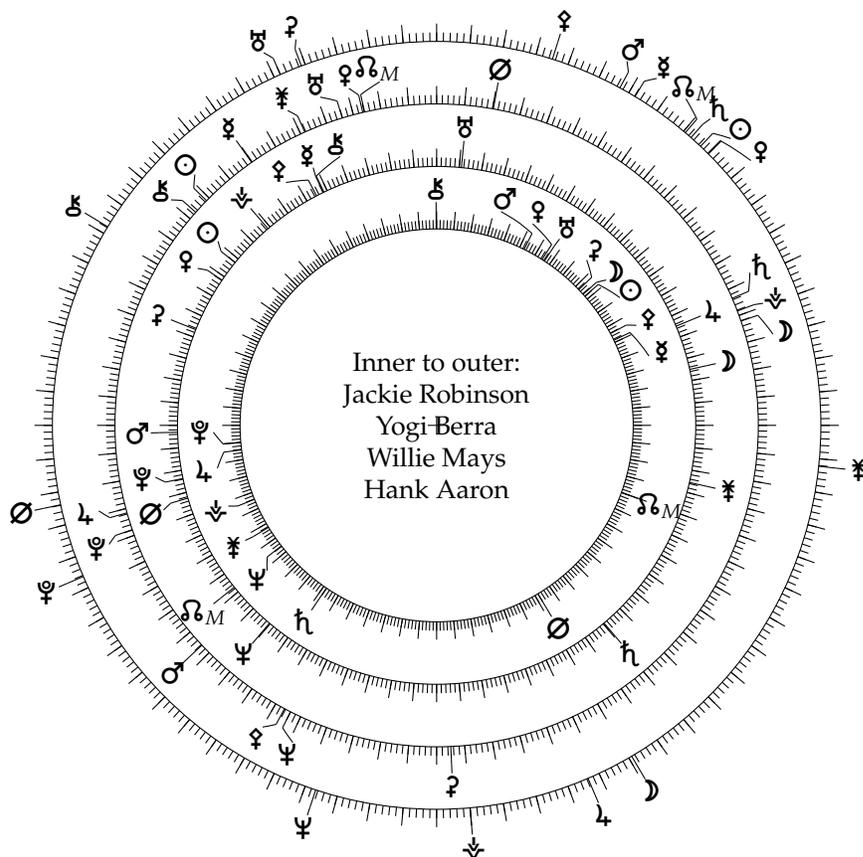
Middle: Brian Mulroney



Outer: Wayne Gretzky

```
\horocalparms{1961}{1}{26}{12:45:0}{W80:16:0}{N43:08:0}  
\horocalculate\horosaveobjects{\WayneGretzky}  
\horocalparms{1939}{3}{20}{12:47:0}{W68:30:0}{N49:13:0}  
\horocalculate\horosaveobjects{\BrianMulroney}  
\horocalparms{1938}{11}{17}{12:0:0}{W79:25:0}{N44:37:0}  
\horocalculate\horosaveobjects{\GordonLightfoot}  
\begin{horoscope}  
  \horowheelResolute{\GordonLightfoot}{\BrianMulroney}{\WayneGretzky}  
  \horoULnote{Inner: Gordon Lightfoot}  
  \horoURnote{Middle: Brian Mulroney}  
  \horoLLnote{Outer: Wayne Gretzky}  
\end{horoscope}
```

Figure 9: Triple 360° dial (Resolute).



```

\horocalparms{1934}{2}{6}{2:25:0}{W88:03:0}{N30:41:0}
\horocalculate\horosaveobjects{\HankAaron}
\horocalparms{1931}{5}{7}{4:30:0}{W86:55:0}{N33:28:0}
\horocalculate\horosaveobjects{\WillieMays}
\horocalparms{1925}{5}{12}{8:0:0}{W90:12:0}{N38:37:0}
\horocalculate\horosaveobjects{\YogiBerra}
\horocalparms{1919}{1}{31}{24:30:0}{W84:13:0}{N30:52:0}
\horocalculate\horosaveobjects{\JackieRobinson}
\begin{horoscope}
  \horoaspectwebfalse
  \horowheelRankin{\JackieRobinson}{\YogiBerra}%
    {\WillieMays}{\HankAaron}
  \horoCnote{Inner to outer:\\ Jackie Robinson\\
    Yogi Berra\\ Willie Mays\\ Hank Aaron}
\end{horoscope}

```

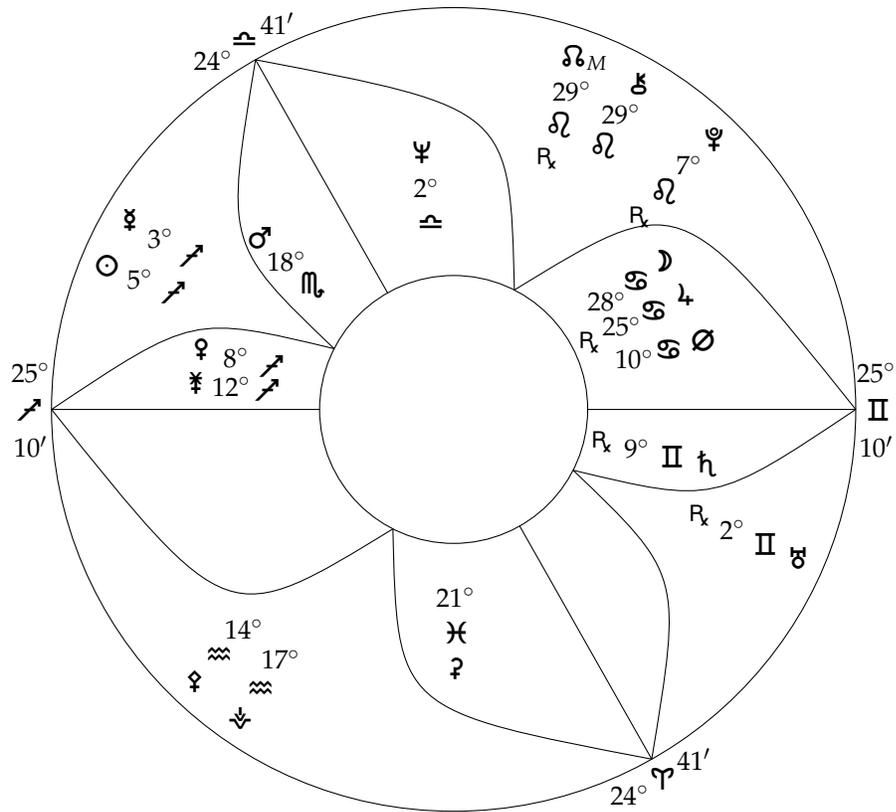
Figure 10: Quadruple 360° dial (Rankin).

objects showing object symbol, degree and sign, and possible retrograde. Most of the house cusps are drawn as curves, creating a floral effect. This template does not support an automatic aspect web (it wouldn't make sense because there are no object-location ticks) nor modification of the label content.

`\horowheelQuebecCity`

The `\horowheelQuebecCity` macro is another take on the decorative wheel chart concept: here eight houses are on the outside and four on the inside, in contrast to the eight on the inside and four around the outside of the Montreal template. As a result of the different layout it becomes possible to draw the cusps as straight lines rather than curves. The result is shown in Figure 12. This template style shows degree-sign-minute labels for all twelve cusps, and degree-sign-retrograde labels for objects. As with Montreal, the label detail is fixed and there is no provision for an aspect web (which would need to be stretched to fit the roughly square shape of the wheel's hub).

Jimi Hendrix



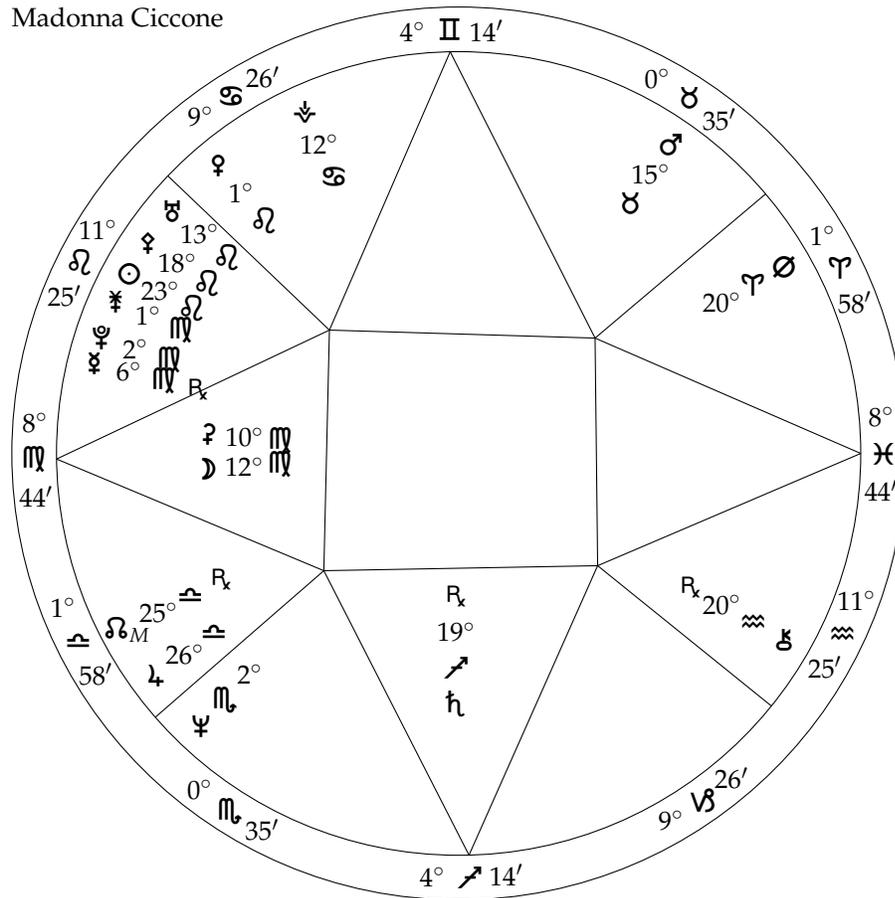
```

\horocalparms{1942}{11}{27}{17:15:0}{W122:20:0}{N47:36:0}
\horocalculate
\begin{horoscope}
  \horowheelMontreal
  \horoULnote{Jimi Hendrix}
\end{horoscope}

```

Figure 11: Decorative chart template (Montreal).

Madonna Ciccone



```

\horocalparms{1958}{8}{16}{12:05:0}{W83:54:0}{N43:36:0}
\horocalculate
\begin{horoscope}
  \horowheelQuebecCity
  \horoULnote{Madonna Ciccone}
\end{horoscope}

```

Figure 12: Decorative chart template (QuebecCity).

Chapter Seven

ADVANCED TOPICS



1. ROUNDING AND MIXED-BASE CONVERSION

§1.1. The rounding problem.—Astrologers traditionally describe Zodiac positions with a mixed-base number system in which a position might be written like $2^{\circ}\mathfrak{P}34'56''$. This format descends from Babylonian sexagesimal arithmetic, representing fractions in terms of multiples of powers of $1/60$. It is convenient for hand calculation and human analysis because it makes it easy for humans to recognize important boundaries (like sign cusps) and relationships (like aspects). Humans are generally good at doing arithmetic on small integers, which is the necessary skill for using this representation.

The mixed-base system is less convenient for computer arithmetic, however. Computers generally record positions in other formats, such as floating- or fixed-point degrees, radians, or “centiseconds” past the Pisces-Aries cusp; horoscop in particular uses degrees stored in \TeX length variables with a scaling of $1^{\circ}=1\text{pt}$. At that scale the inherent precision of a \TeX length means that the angles can be reproduced to an accuracy of $1^{\circ}/65536$, which is just under $1/18$ of a second of arc.

When an internal-format Zodiac position has to be displayed in human-readable form, some kind of rounding must necessarily occur. Each position needs a name. A name like $2^{\circ}\mathfrak{P}34'56''$ strictly speaking represents just one point on the Zodiac. There are an infinite number of points and only a finite number of possible names (about 1.3 million if we use whole seconds of arc as the precision level); so for any given point on the Zodiac, in general there will be no name exactly describing its location. How shall we assign names to points? Most astrological software has built-in arbitrary and undocumented assumptions on how to round positions for display; in horoscop, we attempt to do it in a more principled way, and expose the decisions to interested users.

Rounding is already a serious issue in general numerical computation, and people have developed a variety of solutions to serve varying purposes. Some of them are non-obvious—for instance, the “banker’s rounding” rule designed to reduce overall rounding error when taking the sum of a set of rounded numbers. Rounding for astrological purposes presents unique challenges because of the way humans will use the rounded results. In particular, sign and degree boundaries are important in astrology and otherwise-good rounding schemes may cause problems if they do not respect those boundaries.

For example, suppose some object in a horoscope has a Zodiac position 29.9999° past the Pisces-Aries cusp. It is before the Aries-Taurus cusp, but by less than half a second of arc. If we round it to the nearest second, we get $0^{\circ}\mathfrak{S}0'0''$. That misrepresents the position in an important way: the object has not yet en-

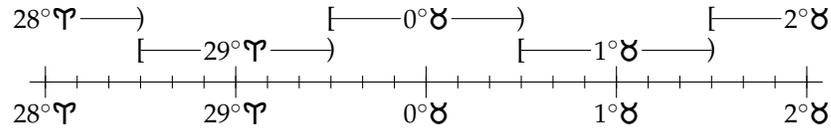


Figure 13: Pure rounding to nearest degree.

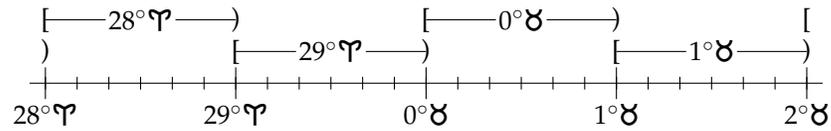


Figure 14: Truncation to degree.

tered Taurus, but the rounded position says it has. A similar issue shows up around the 10° and 20° boundaries if we are interested in decans; or around *all* the degree boundaries if we are interested in Sabian symbols.

This kind of issue also becomes worse when positions are rounded to larger units (minutes or degrees). With rounding to the nearest degree, a position can be represented as being in the next sign while actually being up to half a degree before the cusp. In the case of the Sun in particular, this kind of issue can only exacerbate existing public misunderstandings of what Sun-sign cusps actually mean. See Figure 13: the range of positions that will be labelled as a particular sign in a pure round-to-nearest-degree scheme actually starts half a degree before the start of the sign and ends half a degree before the end of the sign. Pure rounding seems to be a problem if we care about which signs things are really in.

One possible solution is to use pure truncation, as shown in Figure 14. This approach has the advantage of being very simple. Each name corresponds to an interval stretching from the named point to the next named point. All the intervals are the same size. It respects sign and degree boundaries, and smaller-unit boundaries when generalized to higher precision. The lower-precision truncated version of any position is always identical to the most significant few digits of the higher-precision truncated version. However, because the labels refer to the lower extremes of the rounding intervals, this approach maximizes the rounding error. A position truncated to the next lower degree may be as much as one degree away from its named position; and it may be much closer to the next degree than to its named position. Because this scheme is asymmetrical, truncating a batch of randomly chosen points to the degree level will tend to shift them backwards by an average of half a degree.

Another option might be to do rounding to nearest at the lowest level, but add extra rounding boundaries, and extra names, to solve the “signs start too

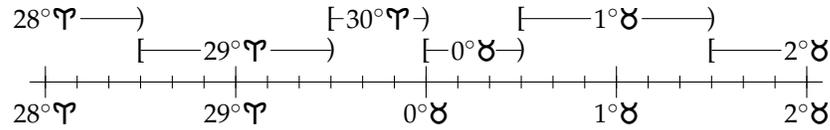


Figure 15: Rounding without carry.

early” problem. There are many equivalent ways of stating this scheme; one way to describe it is that we use truncation for all mixed-base digits except the least significant and then round the least significant digit to nearest without allowing it to carry into higher digits. The result is shown schematically in Figure 15.

In the rounding without carry scheme, a position very near the end of Aries might be named $30^\circ \♈$. Such a name may be upsetting to readers who expect every sign to contain 30 degrees named $0^\circ \dots 29^\circ$. It also creates strange exceptional degrees at either end of every sign: 0° and 30° are each $30'$ long, while $1^\circ \dots 29^\circ$ are $60'$ each. However, it has significant advantages. It makes the sign boundaries clear; it makes positions near sign boundaries stand out in an obvious way to informed readers; it keeps the maximum rounding error to half a degree, improving on truncation’s maximum error of one degree; and because of its symmetry, randomly chosen points do not tend to shift in a particular direction under this scheme. This scheme is used by some printed ephemerides, and is preferred by the author of horoscop.

Some astrological software uses another scheme, illustrated in Figure 16, where positions are in general rounded to nearest but the rule changes to truncation at the end of each sign. This is essentially the same as the previous rounding mode with the added rule that any digit given an out of range value (such as 30°) is changed to the next lower value. Swiss Ephemeris offers this as one of several options in the `swe_split_deg` library function, along with a similar mode that introduces truncation at the end of every degree. These rounding schemes obey sign (or degree) boundaries, and preserve the advantages of rounding to nearest elsewhere. However, this technique is asymmetrical, and it creates two different kinds of exceptional degrees around sign boundaries: 0° is $30'$ long, $1^\circ \dots 28^\circ$ are $60'$ each, and then 29° is $90'$. This technique is mentioned for completeness and because some users may want it; it is not particularly recommended. Its advantage over rounding without carry is that it will never produce confusing labels like 30° .

A philosophical issue exists regarding treatment of points that may happen to be exactly on sign boundaries. Is the exact equinoctial point, for instance, properly described as part of Aries, part of Pisces, neither, or both? This is essentially the same question as whether the exact moment of noon (in civil time) should be called “12:00 AM,” “12:00 PM,” or something special of its own. In mathematical terms, the question is whether signs are open or closed at their ends. As implied by the use of “[” and “)” in the figures in this section, horoscop assumes

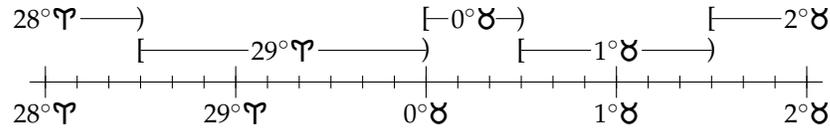


Figure 16: Rounding with truncation at end of sign.

that signs are half-open intervals closed at the beginning and open at the end, so that the equinoctial point is part of Aries and not part of Pisces. This approach is consistent with the convention that noon is 12:00 PM and midnight is 12:00 AM.

It is possible to argue using antiscion relationships that the Zodiac ought to be symmetric under a flip between retrograde and direct. The half-open interval scheme breaks that symmetry. Preserving it would require that cusps must be in both signs, or in neither—like saying that noon and midnight must always be called noon and midnight instead of AM or PM, or that “12:00 AM” and “12:00 PM” each refer to both moments. Such an approach has obvious problems for computer systems that try to assign one of twelve sign symbols to every position. Because of the limited precision both of computer arithmetic and the observations on which astrological calculations ultimately rest, it is not clear that we can ever really say a calculated position is *exactly* on a cusp anyway; there is always some amount of fuzz;⁴ so the decision on how to represent truly exact cusps seems not to be of much practical consequence anyway.

§1.2. Automatic rounding.—The default configuration of horoscope is to automatically choose a sensible rounding mode in each situation. As long as you stick to the ready-made wheel templates, don’t change the smart label strings, and don’t care about the arcane details described in the previous subsection, you don’t need to do anything about rounding and it will just work.

`\hororoundautofalse`
`\hororoundautotru`

In more detail: automatic rounding mode selection is on by default. It can be turned off and on with `\hororoundautofalse` and `\hororoundautotru` respectively. When this mode is active, the package will switch to `\hororoundtoseckkeepmin` for text-mode typeset positions from `\horodsmstext`, `\hororoundtruncate` for times and latitude-longitude coordinates, `\hororoundtominkeepdeg` for labels typeset in the Vancouver wheel template, and `\hororoundtodegkeepsign` for labels in the Montreal and Quebec City templates. These modes are described in more detail below. The rationale for their choice is that time and geographic coordinates would have been entered by the user to precision of seconds, and should always match what the user entered. For Zodiac positions as such, the automated choice is rounding without carry, as shown in Figure 15, generalized to the level

⁴The eminent Professor Doron Zeilberger has suggested an “ultrafinitist” interpretation under which the real number line itself has limited precision, so that points exactly on certain boundaries do not necessarily exist *even in theory*. He was probably joking, but so might I be. See “*Real Analysis is a Degenerate Case of Discrete Analysis*”, D. Zeilberger, <https://sites.math.rutgers.edu/~zeilberg/mamarim/mamarimhtml/real.html>.

of precision in the particular label.

If you design a template of your own, it would be a nice added feature to make it automatically choose an appropriate rounding mode when automatic rounding mode selection is in force. Test it with `\ifhororoundauto`; see the source code of the existing templates for examples of how this test can be used.

`\ifhororoundauto`

§1.3. Manual rounding modes.—When `\hororoundautofalse` is active, the user must choose rounding modes manually. There are 11 basic modes, and six of them can be modified by turning on clamping, for a total of 17 manually-selected rounding modes.

`\hororoundtruncate`

Truncation, as in Figure 14, is selected by `\hororoundtruncate`. This rounding mode is actually not pure truncation; it adds an offset of $1^\circ/65536$ before doing the truncation in order to compensate for precision lost in the internal representation of angles. Without the offset, values entered in degrees or hours, minutes, and integer seconds, converted to internal form, and then converted back, would usually end up one second less than the input value. The offset makes sure that exact whole-second values will survive a round trip conversion. However, if for some reason a really strict truncation with no offset is desired, that can be selected with `\hororoundstricttruncate`. It is not necessary to specify what boundaries to truncate to, because the digit values are the same; if you want less precision, just write out fewer of the mixed-base digits.

`\hororoundstricttruncate`

True rounding to the nearest unit, as shown for degrees in Figure 13, is not recommended for the reasons described in the previous subsection. However, it is available if desired, via the macros `\hororoundtosec`, `\hororoundtomin`, and `\hororoundtodeg`. Note that if you select rounding to a unit larger than seconds, then rounding will put nonsense values in the smaller-unit digits; so, for instance, if using the Vancouver template (which displays degrees and minutes by default) with rounding to the nearest degree, it is important to change the label string to only display degrees.

`\hororoundtosec`

`\hororoundtomin`

`\hororoundtodeg`

The remaining manual modes specify rounding to the nearest of one unit while keeping the boundaries of a larger unit intact, in a generalization of the scheme shown in Figure 15. The choices are `\hororoundtoseckeepsign`, `\hororoundtoseckeepdeg`, `\hororoundtoseckeepmin`, `\hororoundtominkeepsign`, `\hororoundtominkeepdeg`, and `\hororoundtodegkeepsign`. In general it is probably most useful to round to the smallest unit you will be displaying and keep the boundaries of the next larger unit (i.e. `-seckeepmin`, `-minkeepdeg`, or `-degkeepsign`), but the others are provided to cover some possibilities offered by other software. As with the pure rounding modes, these modes leave garbage in any digits smaller than the rounding unit and you should not display any digits less significant than the one you rounded to.

`\hororoundtoseckeepsign`

`\hororoundtoseckeepdeg`

`\hororoundtoseckeepmin`

`\hororoundtominkeepsign`

`\hororoundtominkeepdeg`

`\hororoundtodegkeepsign`

By default the “keep boundaries” modes can generate out-of-range digits like 30° . If you want to prevent that by switching to truncation (clamping the values) at the ends of higher-level units, as in Figure 16, turn on `\hororoundclamptrue`. This modification of the rounding algorithm is deprecated, but provided for compatibility. It can also be used to modify the automatic mode selection of `\hororoundautottrue`. Turn it off with `\hororoundclampfalse`.

`\hororoundclamptrue`

`\hororoundclampfalse`

2. ADDING CUSTOM OBJECTS

The Swiss Ephemeris calculation backend can potentially compute positions for a great many objects beyond the ones enabled by default. This section describes how to add support for a customized object, with the example of adding the dwarf planet 136199 Eris (formerly 2003 UB₃₁₃, once tentatively named Xena) to the birth chart of Lucy Lawless.

First, define a `\horo<object>SEOpt` macro specifying the `swetest` command-line option(s) for calculating the object's position. For Eris, those are `-ps` (planet to calculate is one of the "small" ones to be designated by number) and `-xs136199` (planet number 136199⁵). Other asteroid-like objects would be designated similarly; see the `swetest` documentation for how to select other kinds of objects.

```
\newcommand{\horoErisSEOpt}{-ps -xs136199}
```

The chart plotting system also needs a `\<object>Symbol` macro. The symbol macro can be as complicated as necessary; for the example we just use a letter X.

```
\newcommand{\ErisSymbol}{X}
```

Having defined those macros it only remains to add the new object to the `\horoobjects` list and proceed as with any other chart. The result is shown in Figure 17.

Adding customized objects for use with the Astrolog calculation backend may be possible, but less easy. Macros would have to be defined for two-way translation between the names used in horoscop and the abbreviations used in the Astrolog command line and output file. That is unsupported and so the macros involved have been given @ names to mark them as private. It is less useful in the case of Astrolog anyway because, unlike Swiss Ephemeris, current Astrolog does not support an arbitrarily growing set of calculable objects.

3. SPECIAL CHARTS

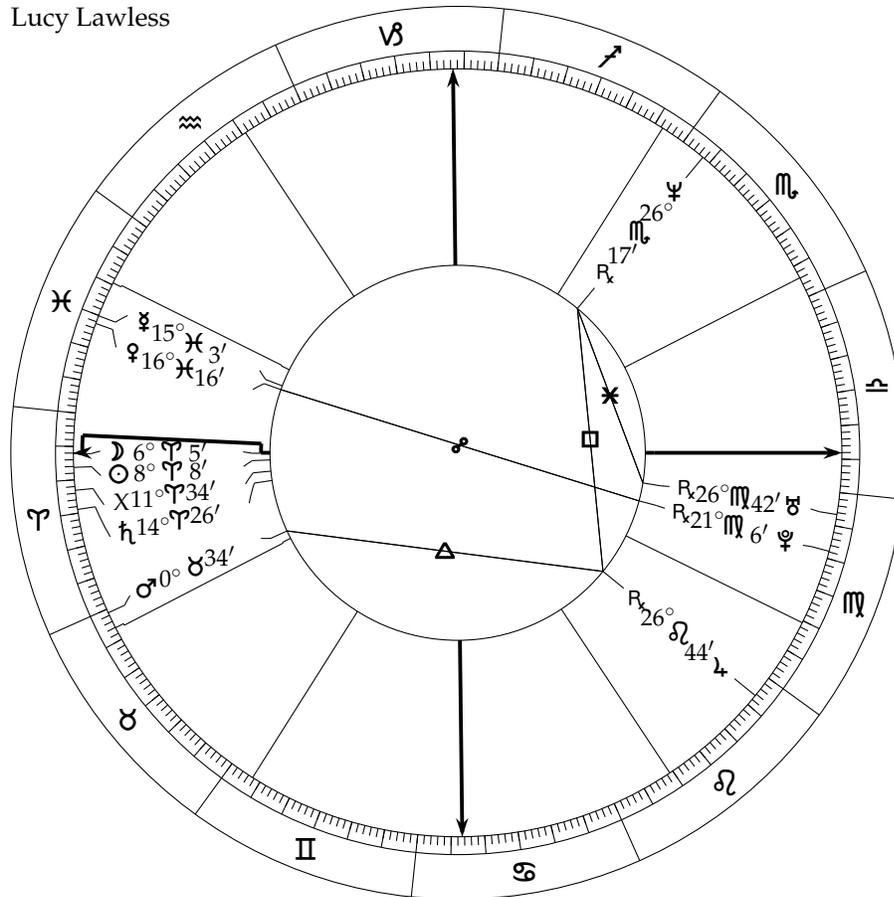
People sometimes want to compute and display special kinds of charts, other than the basic one representing the time and place of a native's birth. Here are some notes on those.

Relocation: A relocation chart is just the chart for a different geographic location at the native's moment of birth, so you can substitute the appropriate coordinates into the `\horocalcparms` command and proceed as with any other chart.

Transits, Solar returns, horary, etc.: These charts are also just ordinary charts for special times and locations, so they can be calculated normally given the right data. There is currently no special support for automatically calculating the data to use, for instance to get the exact time of a Solar return.

⁵For current versions of Swiss Ephemeris. Some versions published between the discovery of the planet and the assignment of its official number referred to it as object number 999001.

Lucy Lawless



```

\horocalparms{1968}{3}{28}{18:25:0}{E174:46:0}{S36:52:0}
\newcommand{\horoErisSEOpt}{-ps -xs136199}
\newcommand{\ErisSymbol}{X}
\renewcommand{\horoobjects}{Sun,Moon,Mercury,Venus,%
Mars,Jupiter,Saturn,Uranus,Neptune,Pluto,Eris}
\horocalculate
\begin{horoscope}
\renewcommand{\horotextsize}{1}
\horowheelVancouver
\horoULnote{Lucy Lawless}
\end{horoscope}

```

Figure 17: Adding a custom object.

Secondary progressions: A progressed chart is basically a standard birth chart with the birth time advanced (or retarded, in the case of converse progression) by a number of days equal to the number of years since the birth. The package can calculate these with no problem if you enter an appropriately modified birth date and time. However, there is no special support for computing the right data to enter. Some astrologers also use computation methods that are not equivalent to computing a standard chart for any real time and place—for instance, moving house cusps by a fixed angle while moving planets according to their actual motion in the sky. Some of these can be accomplished by computing a standard chart and then using `\horoshiftobjects` or `\horoshiftcusps` to move the objects or cusps by the appropriate angle, but the user must calculate for themselves the angles they wish to use.

Solar arc progression: This consists of computing an angle based on the Sun's progression and then adding it uniformly to all the object locations. The result does not represent the sky at any real time and place. This kind of chart can be calculated by computing the original natal chart and then using `\horoshiftobjects` to move the objects through the appropriate angle. There is no built-in support for computing the appropriate angle; the user must provide that,

Synastry: A synastry chart normally shows one person's objects in another person's houses. These can be typeset using the `\horosavecusps` command: compute one chart, use `\horosavecusps` to save the cusps to a macro, then compute the other chart and restore the cusps from the first one by calling the created macro before typesetting the synastry chart. See the next section for information on creating an aspect web between the objects of two different charts; though at present, there is no template for showing two sets of objects on a Vancouver-like traditional wheel chart.

Midpoint composite: No current support. Many things can go wrong with these charts in the worst case (for instance, house cusps can end up out of sequence if the two ascendants are near opposition), and they do not represent the actual sky at any real time and place. If necessary, they can be typeset by manually setting the Pos variables for the objects, then copying Pos to DPos, before calling the template macros.

Time-space midpoint (Davison): This type of midpoint chart does represent the actual sky at a real time and place (the midpoint of the birth times and locations of two people) so it can be obtained by entering the appropriate midpoint with `\horocalparms` and proceeding normally. There is no built-in support provided for calculating that midpoint, and supporting it would require a clearer definition of geographic "midpoint." Most people⁶ seem to do it by computing the numerical midpoint of the latitude and longitude, as angles, but that is not necessarily the same as the midpoint of the great-circle line between the two points, which might have more symbolic validity. This kind of issue seems to be beyond the intended scope of the current version of horoscop.

⁶For instance, based on an inspection of the source code this seems to be what Astrolog does.

4. ASPECTS BETWEEN TWO CHARTS

The aspect web in the standard chart templates normally shows aspects among the objects within a single chart. By clever hacking of the `\horoaspectobjectsa` and `\horoaspectobjectsb` macros, however, it is possible to make it display aspects between two different sets of objects, for instance between natal and transiting objects. See Figure 18 for an example.⁷

We start by calculating both sets of objects and saving them to macros named `\MarilynBirth` and `\MarilynDeath`. After calculating the first set we use the `\horocopyvar` macro to copy what is currently in the `Pos` variable for each object (the positions at death) into a new variable called `XPos`. Since this is not a standard variable used by the rest of the system, other macros will not touch it by default.

Then we redefine `\horoaspectobjectsa` to be the list of default objects with `X` appended to each of their names. That's the clever bit. When the aspect web attempts to look up the `Pos` of an object like the Sun, it will look in a macro called `\horoSunXPos`, which is the copied value of `Pos` from the Sun at the time we did the copying. That is the death location; meanwhile the `Pos` values will be for the inner chart, using the default value of `\horoaspectobjectsb`. The inner chart is the birth chart, so the aspect web shows aspects between birth objects and death objects.

5. LABEL ADJUSTMENTS

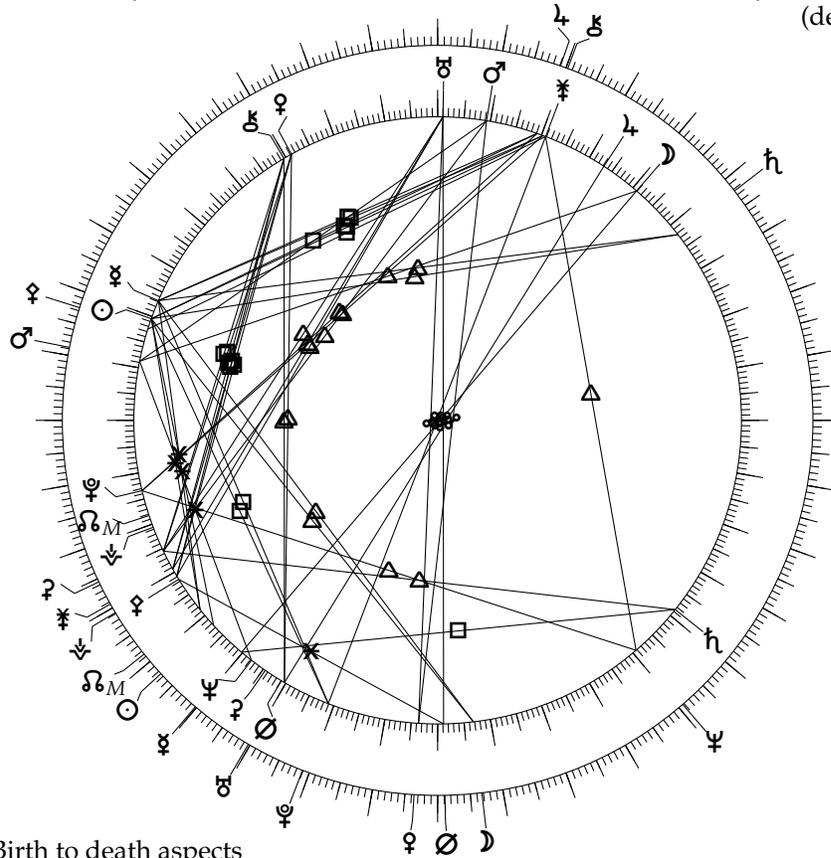
In general, it is desirable to put object labels on the chart in angular positions that correspond to their actual longitudes. However, when objects are close together in longitude, that can result in labels being printed on top of each other. The problem gets worse when other things are added to the chart, such as house cusps and internal house labels. One of the steps in typesetting a wheel chart involves an adjustment process that moves the labels around to keep them from interfering with each other and with house cusps. This process has a number of adjustable parameters, set by redefining internal macros. In general it should not be necessary to change these parameters while using the ready-made templates, but designers of new templates may need to set them, and some changes may also be needed if you modify the sizes of labels.

See Figures 19–21 for some examples of the effects of these options. In Figure 19 the adjustment process has been disabled by setting the minimum distances to zero, so labels are free to interfere. Internal house labels are turned on to exacerbate the crowding. In Figure 20, the distances have been increased to show how cusps will be modified to expand houses where necessary. Note the crowded eleventh house, which the system has expanded to make room for all the labels that must fit there. In Figure 21, the options have been set to prevent cusps from being modified, even at the cost of crowding a house. Note the differ-

⁷The figure shows the death of Marilyn Monroe, an event which happened at an uncertain time; 3:00 AM, when her housekeeper phoned the psychiatrist after discovering the body, was used for the calculation.

Inner: Marilyn Monroe (birth)

Outer: Marilyn Monroe
(death)



Birth to death aspects

```
\horocalparms{1962}{8}{5}{10:0:0}{W118:19:0}{N34:08:0}
\horocalculate\horosaveobjects{\MarilynDeath}
\horocopyvar{\horoobjects}{Pos}{XPos}
\horocalparms{1926}{6}{1}{17:30:0}{W118:15:0}{N34:04:0}
\horocalculate\horosaveobjects{\MarilynBirth}
\horocalculate
\begin{horoscope}
  \renewcommand\horoaspectobjects{a%
    {SunX,MoonX,MercuryX,VenusX,MarsX,JupiterX,SaturnX,%
      UranusX,NeptuneX,PlutoX,%
      NorthNodeX,LilithX,ChironX,CeresX,PallasX,JunoX,VestaX}
  \horowheelIglolik{\MarilynBirth}{\MarilynDeath}
  \horoULnote{Inner: Marilyn Monroe (birth)}
  \horoURnote{Outer: Marilyn Monroe (death)}
  \horoLLnote{Birth to death aspects}
\end{horoscope}
```

Figure 18: Aspects between two charts.

ence between this situation and Figure 19, visible for instance in the eighth and ninth houses: labels can still move, but cusps cannot.

`\horoomindist`
`\horocmindist` The minimum distance targets in degrees are set by `\horoomindist` (note triple o in name) and `\horocmindist`, for “\horo object-object minimum distance” and “\horo object-cusp minimum distance” respectively. The defaults are 6° object to object and 4° object to cusp. Any labels that are separated by less than this will be subject to movement.

`\horoposattobj`
`\horoposattcusp`
`\hororepulsion` The sizes of the adjustments (or strengths of the springs) are determined by three divisors applied to the distances between things and where they should be. Note that these must be integers, and they are *divisors*, so larger values mean weaker attraction or repulsion. The `\horoposattobj` macro sets the attraction between objects and their longitudes; the `\horoposattcusp` macro is, similarly, the attraction between cusps and their longitudes; and the `\hororepulsion` macro sets the repulsion between things (both cusps and objects) that are closer than their target distances. The defaults are 20, 7, and 3 respectively, so cusps spring toward their longitudes about three times as strongly as objects do (if cusps are allowed to move at all), and things that are crowded together spring apart a little more than twice as strongly as that.

Note that the spring strength settings are for adjusting the *relative* strengths of the different kind of adjustments. It will not work to increase or decrease all three uniformly to make the overall layout looser or tighter, because the system will simply make more or fewer iterations and end up with substantially the same solution. Overall looser-tighter control should instead be exercised by changing the minimum distances.

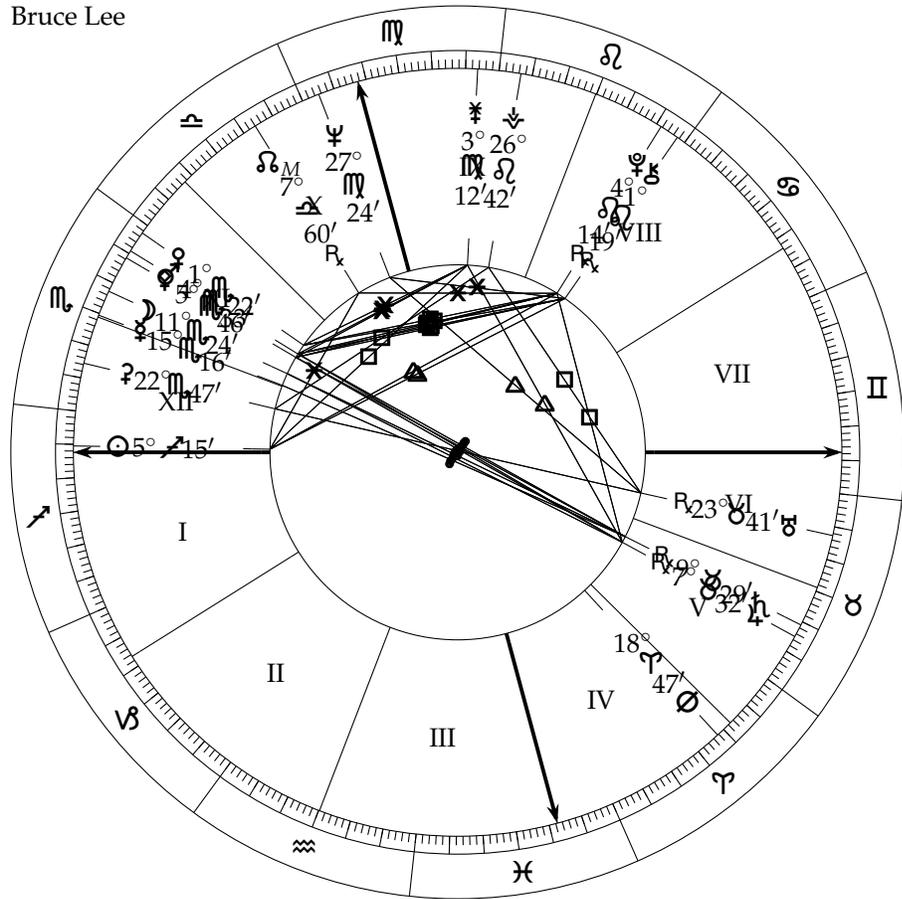
`\horosignificantadj` Label adjustment is done by an iterative processes that approximately simulates a system of springs. Each label is subject to tension when it is too close to its neighbours or too far from its longitude, and the system makes small adjustments to the label locations to reduce the tension. The usual way for the process to terminate is if the adjustments become so small as to make no visible difference. The `\horosignificantadj` macro sets the threshold in degrees for terminating the loop; it defaults to 0.1.

`\horoadjcycles` There is also a hard limit on how many cycles of adjustment the system will do, set by `\horoadjcycles`. The default is 30, which is almost never reached in practice because the `\horosignificantadj` terminating condition will normally trigger before that. However, the `\horoadjcycles` is for each overall attempt at convergence; if cusp adjustment, described next, happens to be triggered, then it will start over with a fresh set of 30 (or however many) iterations.

`\horocuspadjusttrigger` Adjustment will first try to find a converged solution without moving any house cusps. Once it does, it will compare the worst separation among objects and cusps actually achieved, against the configured target separation. If the result is less than `\horocuspadjusttrigger` as a percentage of the target, then adjustment without moving house cusps is considered to have failed, and the system will try again with house cusps permitted to move. The default is 65.

`\horoadjust` The ready-made templates will invoke `\horoadjust` automatically at the right time to adjust the locations of objects, but it can also be invoked manually in the context of user-created templates. Since most of the chart-drawing macros use

Bruce Lee



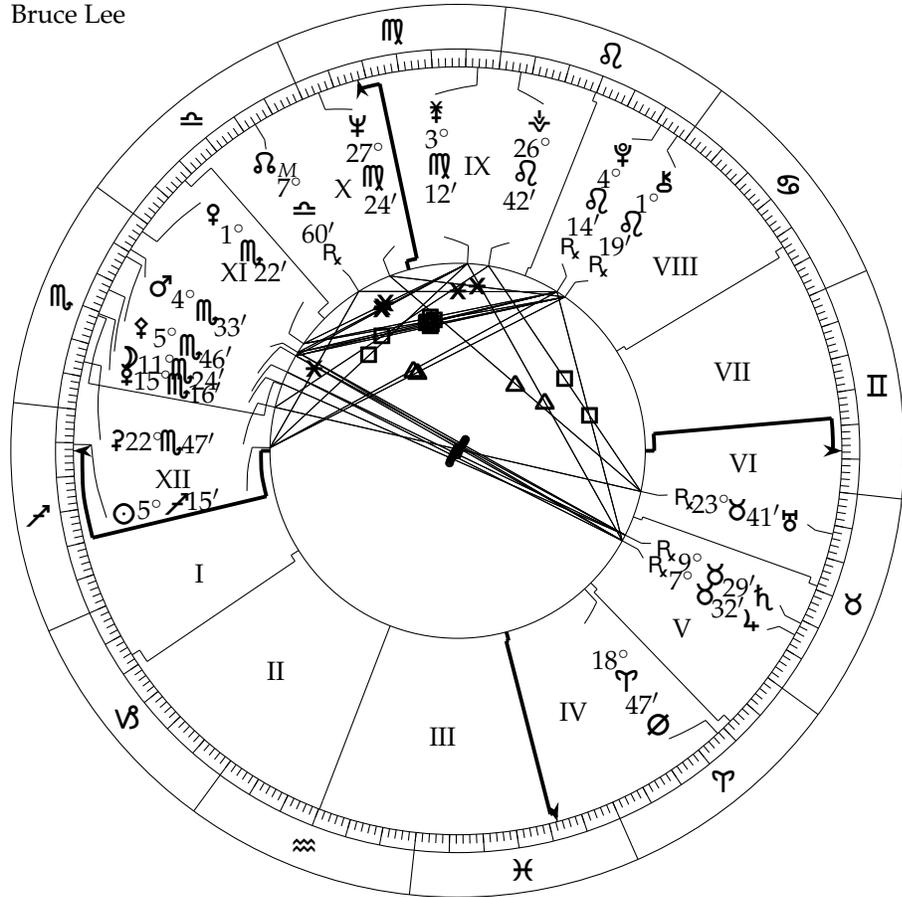
```

\horocalparms{1940}{11}{27}{15:12:0}{W122:25:0}{N37:47:0}
\horocalculate
\begin{horoscope}
  \horointhouselabelstrue
  \renewcommand{\horotextsize}{2}
  \renewcommand{\horoomindist}{0}
  \renewcommand{\horocmindist}{0}
  \horowheelVancouver
  \horoULnote{Bruce Lee}
\end{horoscope}

```

Figure 19: Labels printed at their longitudes without adjustment.

Bruce Lee



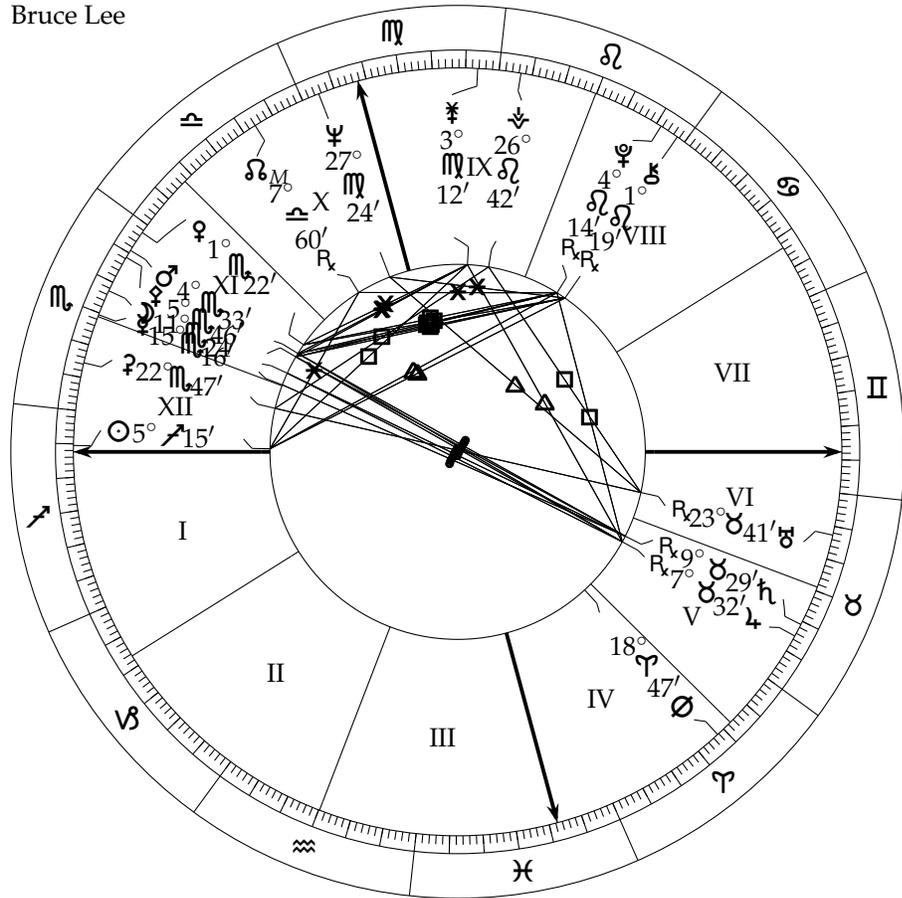
```

\horocalparms{1940}{11}{27}{15:12:0}{W122:25:0}{N37:47:0}
\horocalculate
\begin{horoscope}
  \horointhouselabelstrue
  \renewcommand{\horotextsize}{2}
  \renewcommand{\horoomindist}{15.0}
  \renewcommand{\horocmindist}{10.0}
  \renewcommand{\horomaxrepulsion}{15.0}
  \horowheelVancouver
  \horoULnote{Bruce Lee}
\end{horoscope}

```

Figure 20: Extra space between labels.

Bruce Lee



```

\horocalparms{1940}{11}{27}{15:12:0}{W122:25:0}{N37:47:0}
\horocalculate
\begin{horoscope}
  \horointhouselabelstrue
  \renewcommand{\horotextsize}{2}
  \renewcommand{\horocuspadjusttrigger}{1}
  \horowheelVancouver
  \horoULnote{Bruce Lee}
\end{horoscope}

```

Figure 21: Crowded houses permitted.

the current DPos values, it is important to invoke `\horoadjust` at the right time: after all things that should be drawn at the “true” Pos locations and before all things that should have their locations adjusted. Consult the source code, and the next section, for examples and information useful in deciding exactly when to run the adjustment.

6. DESIGNING NEW TEMPLATES

Many users will be content to simply use the provided horoscope wheel templates, but `horoscop` also provides a set of tools intended for more advanced users to design their own wheels. Interested users are encouraged to read the relevant parts of the source code and its comments for details on how the existing templates work, and imitate their approach. This section provides a brief summary of the user-visible macros provided.

§6.1. Ticks and keys.—Generally, a wheel template starts by drawing the basic furniture that will be shared by all wheels of that type—typically one or more circles, a key showing the twelve sign symbols, and perhaps some radial ticks that provide an angle scale. Circles can be drawn with the ordinary `\circle` command, bearing in mind that the horoscope environment is just a picture of fixed dimensions. Template authors are urged to design for a basic radius of 50 to match the existing templates.

`\horoputticks` The `\horoputticks{⟨r⟩}{⟨length⟩}{⟨interval⟩}` macro creates a set of ticks going around the circle starting at 0° \P and every $\langle interval \rangle^\circ$ thereafter, with inner radius $\langle r \rangle$ and length $\langle length \rangle$, so that the outer radius is $\langle r \rangle + \langle length \rangle$. All radii are in `\horounitlength` units.

`\horoputsignkey` The `\horoputsignkey{⟨r⟩}` macro creates a sign key, which consists of all twelve Zodiac sign symbols, each placed with its centre at radius $\langle r \rangle$ and longitude 15° into its own sign. One would typically use this along with a set of ticks at 30° intervals and spanning the radius of the sign key, to mark the boundaries between signs.

`\ifhorocalculated` **§6.2. Conditionals for use in templates.**—Macros that operate on the locations of things—especially `\horoadjust`—will generally take a long time to run or produce ugly results if they happen to run when there are no valid locations in the variables. To avoid that scenario, the ready-made templates wrap all such macros inside an `\ifhorocalculated... \fi` conditional; if the template is invoked without valid data, then the relevant parts of the chart will simply be blank in the typeset document. Similar measures are recommended for any newly-defined templates, because it is inevitable that users will eventually attempt to run the template without valid object positions.

`\ifhorodrawcusps` Some user-settable configuration flags are available through the conditional macros named `\ifhorodrawcusps`, `\ifhoroboldangles`, `\ifhoroanglearrows`, `\ifhoroathouselabels`, and `\ifhoroaspectweb`, and template authors may wish to use these to turn on or off special features in the template.

§6.3. Drawing sets of graphical elements.—Most chart designs include sets of lines drawn at angles determined by the positions of objects or cusps. The following macros draw things at the DPos values of sets of items. A template will typically invoke some of these to draw things like the true-location ticks corresponding to objects, then call `\horoadjust` to set the DPos values to keep labels from interfering, then draw the labels and call these macro again to draw things like the possibly-shifted house cusps that are subject to adjustment.

`\horoputradials` Most templates draw radial lines at angles corresponding to object or cusp longitudes, using `\horoputradials{⟨objects⟩}{⟨r⟩}{⟨length⟩}`. The `⟨objects⟩` argument should be a comma-separated list of things that will describe the longitudes for the radials. It might typically be `\horoobjects`, `\horocusps`, or `\horoangularcusps`. The inner radius is equal to `⟨r⟩`, and the length of each line is `⟨length⟩`. Radials will be drawn at the DPos values for all the listed objects.

`\horoputarrows` The `\horoputarrows` macro has the same syntax as `\horoputradials`. It places arrowheads (not actually complete arrows) pointing outward with their tips on the circle of radius `⟨r⟩`. The `⟨length⟩` value sets the size of the arrowheads: they will be sized to fit inside boxes of 3×2 this length. To draw complete arrows, draw a matching set of radial lines to connect with the arrowheads.

`\horoputinhouselabels` Use `\horoputinhouselabels{⟨radius⟩}` to draw a set of internal house labels. For this to work, `\ifhoroinhouselabels` must be true and `\horoadjust` must have been called; otherwise, the locations for the labels will not have been calculated. Each house label results in a call to `\horohouselabel`, which by default typesets the `horo`house L^AT_EX counter in uppercase Roman numerals. It can be redefined to get some other style. The centres of the labels are set at radius `⟨radius⟩`; they are attracted to the longitudes of the midpoints of the houses, but may be shifted somewhat by the adjustment process to avoid interfering with other labels.

§6.4. Single linework objects.—For finer control, template authors can also invoke macros to draw things on the chart one at a time at specified coordinates, instead of in sets. Object positions are available to use as coordinates, by invoking the relevant variable macros such as `\horoSunDPos`. It is generally preferable to use DPos rather than Pos so as to pick up the results of any adjustment.

`\horoputradial` Note this is distinct from the similarly-named `\horoputradials`. The `\horoputradial{⟨radius⟩}{⟨length⟩}{⟨theta⟩}` macro draws a radial line out from the point at radius `⟨radius⟩` and longitude `⟨theta⟩` for a length of `⟨length⟩`.

`\horoputline` The more general line-drawing macro, `\horoputline{⟨r1⟩}{⟨theta1⟩}{⟨r2⟩}{⟨theta2⟩}`, draws a line between any two points given in polar coordinates.

`\horoputarrowhead` The `\horoputarrowhead{⟨r⟩}{⟨theta⟩}{⟨size⟩}` macro creates an arrowhead just like the ones made by `\horoputarrows`, at the specified polar coordinates and with the size determined by `⟨size⟩`; the arrowhead fits into a box of 3×2 the value of `⟨size⟩`. It always points outward from the origin.

`\horoputcurve` The `\horoputcurve{⟨r1⟩}{⟨theta1⟩}{⟨r2⟩}{⟨theta2⟩}` draws a smooth curve connecting two points designated by their polar coordinates. If the points happen to be at the same radius and not too far apart, the curve will approximate a circular arc centred on the origin. If they happen to be at the same longitude

(theta coordinate) then the curve will be a straight line. In other cases it will be somewhere in between these. The main intended use is for the jogs drawn when a house cusp is displaced by adjustment in charts like Vancouver, although it was later applied to the floral shapes in Montreal as well. To draw a complete circle instead of an arc, use the existing L^AT_EX `\circle` macro.

<code>\horoforeach</code>	<p>§6.5. For-each and things to put in it.—To run a macro on every object in a comma-separated list, call <code>\horoforeach{<list>}{<macro>}</code>. The contents of <code><macro></code> will be invoked once for each comma-separated item in <code><list></code>, with the item (in curly braces) added to the end. For instance, an invocation of <code>\horoforeach{x,y,z}{\foo{a}}</code> would call <code>\foo{a}{x}</code>, <code>\foo{a}{y}</code>, and <code>\foo{a}{z}</code>. The single-item commands below are all designed to take an object or cusp name as their last argument, to make them easy to use with <code>\horoforeach</code>.</p>
<code>\horoconncurve</code>	<p>The <code>\horoconncurve{<r1>}{<r2>}{<object>}</code> macro draws a connecting curve showing the relationship between the Pos and DPos of an object. This would typically be used after <code>\horoadjust</code> to connect the label of an object, which might have been moved by adjustment, with the radial tick showing its true longitude. The curve goes from radius <code><r1></code> and the object's Pos, to radius <code><r2></code> and its DPos.</p>
<code>\horoputcusplabel</code>	<p>The <code>\horoputcusplabel{<radius>}{<spacing>}{<object>}</code> macro draws a style of label designed to show the longitude of an angle or cusp. This kind of label appears in the Montreal and QuebecCity chart templates. It consists of the degrees, Zodiac sign, and minutes of arc of the object's Pos, placed on the circle at radius <code><r></code> with the sign symbol at the object's DPos and the other two things arranged <code><spacing></code> degrees away from it on either side, ordered so that they will read degrees-sign-minutes as nearly as possible to left-to-right top-to-bottom.</p>
<code>\horoputobjsymbol</code>	<p>The <code>\horoputobjsymbol{<radius>}{<object>}</code> places an object's symbol at its DPos and the specified radius</p>
<code>\horoputobjdeglabel</code> <code>\horoputobjminlabel</code> <code>\horoputobjseclabel</code>	<p>There are three macros called <code>\horoputobjdeglabel</code>, <code>\horoputobjminlabel</code>, and <code>\horoputobjseclabel</code> for typesetting a label showing an object's longitude down to the degree, minute, or second of arc. The labels consist of two to four chunks showing the the object's Pos in degrees, sign, and possibly minutes and seconds, arranged radially on concentric circles and ordered to read in degree-sign-minute-second order as nearly as possible to left-to-right top-to-bottom. The macros each take three arguments <code>{<radius>}{<spacing>}{<object>}</code>, where <code><radius></code> is the radius on which to place the centre of the innermost label chunk and <code><spacing></code> is the spacing between successive chunks outward from there.</p>
<code>\horoputrxlabel</code>	<p>The <code>\horoputrxlabel{<radius>}{<object>}</code> macro typesets an optional retrograde label for the object at the object's DPos and radius <code><radius></code>: if the object's Vel is negative then the label will appear as the value of <code>\horoRetrogradeSymbol</code> and otherwise it will be blank. A similar function for use in text (possibly in tables showing numeric data) is provided by <code>\hororxttext{<object>}</code>.</p>
<code>\hororxttext</code> <code>\horoputsmartlabel</code>	<p>The <code>\horoputsmartlabel{<object>}</code> macro provides elaborately-configurable labels as seen in the Vancouver template. The macro itself simply plots the label for an object; it must be configured in advance using other macros.</p>
<code>\horoscanlabels</code>	<p>First, <code>\horoscanlabels{<string>}</code> takes one or two label format specifiers (sep-</p>

arated by a slash if there are two) just as documented for the Vancouver template. After calling that to set the format strings, use `\horosetsmartradii` $\{\langle outer \rangle\}\{\langle stepbase \rangle\}\{\langle stepadj \rangle\}$ to set the radii for the chunks of the label. The outermost chunk will be set with its centre at radius $\langle outer \rangle$. Successive chunks will be set inside that at a spacing of $\langle stepbase \rangle - n \times \langle stepadj \rangle$, where n is the number of chunks. For instance, with `\horosetsmartradii{30}{7}{1}` and four chunks, the radii will be 30, 27, 24, and 21. The reason for this apparently unusual design is that a simple even division of a fixed amount of space tends to produce excessively wide spacing when there are few chunks; with carefully chosen coefficients, this formula seems to produce more visually appealing results across the range of typical label lengths.

Label commands set things in the current text size. With smart labels it may be desirable to change the text size depending on how many chunks are in the label; `\horochoosetextsize` does that, according to the scheme documented for the Vancouver template.

§6.6. Drawing the aspect web.—Low-level macros used in drawing the aspect web are exposed primarily to make it easier to do more sophisticated aspect handling than the default, such as varying the orb based on the objects involved or more carefully selecting which individual aspects to show at all.

`\horoputaspect` $\{\langle radius \rangle\}\{\langle theta1 \rangle\}\{\langle theta2 \rangle\}\{\langle symbol \rangle\}$ draws a single aspect consisting of a line connecting the points at radius $\langle radius \rangle$ and longitudes $\langle theta1 \rangle$ and $\langle theta2 \rangle$. The contents of $\langle symbol \rangle$ will be typeset on the midpoint of the line.

`\horoautoaspect` To automatically find and draw aspects of a given type, use `\horoautoaspect-` $\{\langle list1 \rangle\}\{\langle list2 \rangle\}\{\langle angle \rangle\}\{\langle orb \rangle\}\{\langle radius \rangle\}\{\langle symbol \rangle\}$. This searches for all pairs of one object from $\langle list1 \rangle$ with one object from $\langle list2 \rangle$ whose Pos values are $\langle angle \rangle^\circ$ apart (in either direction) to within $\langle orb \rangle^\circ$. For each one it invokes `\horoputaspect` with the specified $\langle radius \rangle$ and $\langle symbol \rangle$.

`\horoautoaspects` Finally, `\horoautoaspects` $\{\langle radius \rangle\}$ (note plural) draws a complete aspect web of the kind demonstrated by the default templates. It loops through the aspects listed in `\horoaspects` running `\horoautoaspect` on each one and using the angle and orb information from the corresponding `\horo` $\langle aspect \rangle$ Angle and `\horo` $\langle aspect \rangle$ Orb macros.

Chapter Eight

IMPLEMENTATION



We start by declaring that this is the start of a L^AT_EX2 ϵ package and giving it a name.

```
1 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
2 \ProvidesPackage{horoscop}%
3 [2020/07/31 v1.01 Astrological chart macros by Matthew Skala]
```

1. INITIAL OPTION HANDLING

All the options correspond to Boolean flags created by `\newif`, which then get set appropriately by `\ProcessOptions`. The actual consequences of the options are implemented later.

§1.1. Symbol choices.—These can all be turned on and off independently.

```
4 \newif\ifhoro@textsym\horo@textsymfalse
5 \newif\ifhoro@unicode\horo@unicodetrue
6 \newif\ifhoro@wasysym\horo@wasysymfalse
7 \newif\ifhoro@marvosym\horo@marvosymfalse
8 \newif\ifhoro@starfont\horo@starfontfalse
9 \DeclareOption{textsym}{\horo@textsymtrue}
10 \DeclareOption{unicode}{\horo@unicodetrue}
11 \DeclareOption{wasysym}{\horo@wasysymtrue}
12 \DeclareOption{marvosym}{\horo@marvosymtrue}
13 \DeclareOption{starfont}{\horo@starfonttrue}
```

§1.2. Calculation backends.—At most one of these may be selected, so choosing either also turns off the other one's flag, and the `nocalc` option turns both flags off.

```
14 \newif\ifhoro@strolog\horo@strologfalse
15 \newif\ifhoro@swetest\horo@swetesttrue
16 \DeclareOption{nocalc}{\horo@strologfalse\horo@swetestfalse}
17 \DeclareOption{astrolog}{\horo@strologtrue\horo@swetestfalse}
18 \DeclareOption{swetest}{\horo@strologfalse\horo@swetesttrue}
```

Declare an `\if` and associated option for `egrep` filtering.

```
19 \newif\ifhoro@egrep\horo@egrepfalse
20 \DeclareOption{egrep}{\horo@egreptrue}
```

§1.3. Higher-level features.—Wheels (the general support for drawing wheels) and templates (the specific ready-made wheel designs) can each be turned off. Templates require wheels, so turning off wheels also turns off templates.

```
21 \newif\ifhoro@wheels\horo@wheelstrue
```

```

22 \newif\ifhoro@templates\horo@templatestrue
23 \DeclareOption{nowheels}{\horo@wheelsfalse\horo@templatesfalse}
24 \DeclareOption{notemplates}{\horo@templatesfalse}

```

§1.4. Processing the options.—Let L^AT_EX interpret the option settings.

```

25 \ProcessOptions\relax

```

If none of the symbol options were set, then pretend starfont was set. This is actually expected to be the most common case in actual use.

```

26 \ifhoro@textsym\else
27 \ifhoro@unicode\else
28 \ifhoro@wasysym\else
29 \ifhoro@marvosym\else
30 \horo@starfonttrue
31 \fi
32 \fi
33 \fi
34 \fi

```

Because `iftex` defines conditionals, we must load it unconditionally. Otherwise the parse of our style file (when it skims over not-taken conditional bodies) fails.

```

35 \RequirePackage{iftex}%

```

Load other packages as needed by the selected options. Since there are some name conflicts between macros defined by in different packages and we want to retain access to both versions, we save each package's versions immediately after loading the packages.

```

36 \ifhoro@wasysym
37 \RequirePackage{wasysym}%
38 \let\horow@sySquare\Square
39 \fi
40 \ifhoro@marvosym
41 \RequirePackage{marvosym}%
42 \let\horom@rvAries\Aries
43 \let\horom@rvTaurus\Taurus
44 \let\horom@rvGemini\Gemini
45 \let\horom@rvCancer\Cancer
46 \let\horom@rvLeo\Leo
47 \let\horom@rvVirgo\Virgo
48 \let\horom@rvLibra\Libra
49 \let\horom@rvScorpio\Scorpio
50 \let\horom@rvSagittarius\Sagittarius
51 \let\horom@rvCapricorn\Capricorn
52 \let\horom@rvAquarius\Aquarius
53 \let\horom@rvPisces\Pisces
54 \let\horom@rvSun\Sun
55 \let\horom@rvMoon\Moon
56 \let\horom@rvMercury\Mercury
57 \let\horom@rvVenus\Venus

```

```

58 \let\horom@rvMars\Mars
59 \let\horom@rvJupiter\Jupiter
60 \let\horom@rvSaturn\Saturn
61 \let\horom@rvUranus\Uranus
62 \let\horom@rvNeptune\Neptune
63 \let\horom@rvPluto\Pluto
64 \fi
65 \ifhoro@starfont\RequirePackage{starfont}%
66 \let\horost@rAries\Aries
67 \let\horost@rTaurus\Taurus
68 \let\horost@rGemini\Gemini
69 \let\horost@rCancer\Cancer
70 \let\horost@rLeo\Leo
71 \let\horost@rVirgo\Virgo
72 \let\horost@rLibra\Libra
73 \let\horost@rScorpio\Scorpio
74 \let\horost@rSagittarius\Sagittarius
75 \let\horost@rCapricorn\Capricorn
76 \let\horost@rAquarius\Aquarius
77 \let\horost@rPisces\Pisces
78 \let\horost@rSun\Sun
79 \let\horost@rMoon\Moon
80 \let\horost@rMercury\Mercury
81 \let\horost@rVenus\Venus
82 \let\horost@rMars\Mars
83 \let\horost@rJupiter\Jupiter
84 \let\horost@rSaturn\Saturn
85 \let\horost@rUranus\Uranus
86 \let\horost@rNeptune\Neptune
87 \let\horost@rPluto\Pluto
88 \let\horost@rSquare\Square
89 \fi
90 \ifhoro@wheels
91 \RequirePackage{pict2e}
92 \RequirePackage{trig}
93 \fi

```

2. SYMBOLS

§2.1. General symbols.—No special packages needed for these angle symbols; they’re all based on standard L^AT_EX symbols.

```

94 \def\horodegrees{\ensuremath{\^{\circ}}}
95 \def\horominutes{\ensuremath{'}}
96 \def\horoseconds{\ensuremath{''}}

```

`\Zodiac` Although `marvosym` and `starfont` both provide their own versions of this, we re-define it to use `horoscop`’s own abstraction layer so it can be applied to all the symbol sets.

```

97 \def\Zodiac#1{\ifcase#1

```

```

98 \or\AriesSymbol\or\TaurusSymbol\or\GeminiSymbol
99 \or\CancerSymbol\or\LeoSymbol\or\VirgoSymbol
100 \or\LibraSymbol\or\ScorpioSymbol\or\SagittariusSymbol
101 \or\CapricornSymbol\or\AquariusSymbol\or\PiscesSymbol\fi}

```

None of the packages define symbols specifically for mean versus true nodes, so this gets abstracted out and will use whatever the current setting for `\NorthNodeSymbol` might be.

```

102 \def\MeanNodeSymbol{\NorthNodeSymbol$_M$}%
103 \def\TrueNodeSymbol{\NorthNodeSymbol$_T$}%

```

§2.2. Text abbreviations.—These are used as defaults and to replace any symbols not provided by other packages. They’re all built into the `\horotextsym` command so that users who mix symbol sets will be able to switch *back* to this set after having defined other symbols.

`\horotextsym`

```

104 \newcommand{\horotextsym}{%

```

Signs of the Zodiac:

```

105 \gdef\AriesSymbol{Ar}%
106 \gdef\TaurusSymbol{Ta}%
107 \gdef\GeminiSymbol{Ge}%
108 \gdef\CancerSymbol{Cn}%
109 \gdef\LeoSymbol{Le}%
110 \gdef\VirgoSymbol{Vi}%
111 \gdef\LibraSymbol{Li}%
112 \gdef\ScorpioSymbol{Sc}%
113 \gdef\SagittariusSymbol{Sg}%
114 \gdef\CapricornSymbol{Cp}%
115 \gdef\AquariusSymbol{Aq}%
116 \gdef\PiscesSymbol{Pi}%

```

Traditional planets and luminaries:

```

117 \gdef\SunSymbol{Su}%
118 \gdef\MoonSymbol{Mo}%
119 \gdef\MercurySymbol{Me}%
120 \gdef\VenusSymbol{Ve}%
121 \gdef\MarsSymbol{Ma}%
122 \gdef\JupiterSymbol{Ju}%
123 \gdef\SaturnSymbol{Sa}%
124 \gdef\UranusSymbol{Ur}%
125 \gdef\NeptuneSymbol{Ne}%
126 \gdef\PlutoSymbol{Pl}%

```

Minor planets:

```

127 \gdef\ChironSymbol{Chi}%
128 \gdef\CeresSymbol{Cer}%
129 \gdef\PallasSymbol{Pal}%
130 \gdef\JunoSymbol{Jun}%
131 \gdef\VestaSymbol{Ves}%

```

Derived points:

```
132 \gdef\NorthNodeSymbol{No}%
133 \gdef\SouthNodeSymbol{SNo}%
134 \gdef\LilithSymbol{Lil}%
```

House cusps:

```
135 \gdef\CuspISymbol{ASC}%
136 \gdef\CuspIISymbol{$2^{nd}$}%
137 \gdef\CuspIIISymbol{$3^{rd}$}%
138 \gdef\CuspIVSymbol{IC}%
139 \gdef\CuspVSymbol{$5^{th}$}%
140 \gdef\CuspVISymbol{$6^{th}$}%
141 \gdef\CuspVIISymbol{DSC}%
142 \gdef\CuspIIISymbol{$8^{th}$}%
143 \gdef\CuspIXSymbol{$9^{th}$}%
144 \gdef\CuspXSymbol{MC}%
145 \gdef\CuspXISymbol{$11^{th}$}%
146 \gdef\CuspXIISymbol{$12^{th}$}%
```

Angles (in case these are distinct from house cusps):

```
147 \gdef\AscendantSymbol{ASC}%
148 \gdef\MCSymbol{MC}%
149 \gdef\VertexSymbol{Vtx}%
```

Direction:

```
150 \gdef\horoRetrogradeSymbol{Rx}%
```

Aspects:

```
151 \gdef\horoConjunctionSymbol{Con}%
152 \gdef\horoOppositionSymbol{Opp}%
153 \gdef\horoTrineSymbol{Tri}%
154 \gdef\horoSquareSymbol{Sqr}%
155 \gdef\horoQuintileSymbol{Qnt}%
156 \gdef\horoBiquintileSymbol{Bqi}%
157 \gdef\horoSextileSymbol{Sex}%
158 \gdef\horoQuincunxSymbol{Qcx}%
159 \gdef\horoSemisextileSymbol{Ssx}%
160 \gdef\horoSemisquareSymbol{Ssq}%
161 \gdef\horoSesquiquadrateSymbol{Sqq}%
162 }
```

Now invoke `\horotextsym` unconditionally to provide initial defaults.

```
163 \horotextsym
```

§2.3. Unicode astrological symbols.—As with `textsym`, the actual setting of the symbol macros is all inside a unicode-specific macro so that we can switch back to it easily after using other symbol sets. The macro `\horounicode@` contains the actual code points in caret notation, which traditional \TeX engines cannot process; this macro is called by `\horounicode` conditionally on its detecting a Unicode-compatible engine.

\horounicode@

164 \ifhoro@unicode\newcommand{\horounicode@}{%

Signs of the Zodiac:

165 \gdef\AriesSymbol{~~~~2648}%
166 \gdef\TaurusSymbol{~~~~2649}%
167 \gdef\GeminiSymbol{~~~~264a}%
168 \gdef\CancerSymbol{~~~~264b}%
169 \gdef\LeoSymbol{~~~~264c}%
170 \gdef\VirgoSymbol{~~~~264d}%
171 \gdef\LibraSymbol{~~~~264e}%
172 \gdef\ScorpioSymbol{~~~~264f}%
173 \gdef\SagittariusSymbol{~~~~2650}%
174 \gdef\CapricornSymbol{~~~~2651}%
175 \gdef\AquariusSymbol{~~~~2652}%
176 \gdef\PiscesSymbol{~~~~2653}%

Traditional planets and luminaries:

177 \gdef\SunSymbol{~~~~2609}%
178 \gdef\MoonSymbol{~~~~263d}%
179 \gdef\MercurySymbol{~~~~263f}%
180 \gdef\VenusSymbol{~~~~2640}%
181 \gdef\MarsSymbol{~~~~2642}%
182 \gdef\JupiterSymbol{~~~~2643}%
183 \gdef\SaturnSymbol{~~~~2644}%
184 \gdef\UranusSymbol{~~~~2645}%
185 \gdef\NeptuneSymbol{~~~~2646}%
186 \gdef\PlutoSymbol{~~~~2647}%

Minor planets:

187 \gdef\ChironSymbol{~~~~26b7}%
188 \gdef\CeresSymbol{~~~~26b3}%
189 \gdef\PallasSymbol{~~~~26b4}%
190 \gdef\JunoSymbol{~~~~26b5}%
191 \gdef\VestaSymbol{~~~~26b6}%

Derived points:

192 \gdef\NorthNodeSymbol{~~~~260a}%
193 \gdef\SouthNodeSymbol{~~~~260b}%
194 \gdef\LilithSymbol{~~~~26b8}%

Direction:

195 \gdef\horoRetrogradeSymbol{~~~~211e}%

Aspects:

196 \gdef\horoConjunctionSymbol{~~~~260c}%
197 \gdef\horoOppositionSymbol{~~~~260d}%
198 \gdef\horoTrineSymbol{~~~~25b3}%
199 \gdef\horoSquareSymbol{~~~~25a1}%
200 \gdef\horoSextileSymbol{~~~~26b9}%
201 \gdef\horoQuincunxSymbol{~~~~26bb}%
202 \gdef\horoSemisextileSymbol{~~~~26ba}%

```

203 \gdef\horoSemisquareSymbol{^^^^2220}%
204 \gdef\horoSesquiquadrateSymbol{^^^^26bc}%
205 }

```

This helper macro typesets a notation that describes a code point in text:

`\horounicode@point`

```

206 \newcommand{\horounicode@point}[1]{\ensuremath{\langle\mbox{U+#1}\rangle}}

```

The `\horounicode` driver can either call `\horounicode@` to really use Unicode, or display text-based descriptions of the Unicode code points it wants, depending on the engine. Factoring `\horounicode@` into a separate macro reduces repetition, since `\horounicode` wants to call it in two different places.

`\horounicode`

```

207 \newcommand{\horounicode}{%

```

Call `\horounicode@` if it is safe to do so:

```

208 \ifxetex\horounicode@\else
209 \ifluatex\horounicode@\else

```

Now define all the code points, as above, but in text notation.

Signs of the Zodiac:

```

210 \gdef\AriesSymbol{\horounicode@point{2648}}%
211 \gdef\TaurusSymbol{\horounicode@point{2649}}%
212 \gdef\GeminiSymbol{\horounicode@point{264A}}%
213 \gdef\CancerSymbol{\horounicode@point{264B}}%
214 \gdef\LeoSymbol{\horounicode@point{264C}}%
215 \gdef\VirgoSymbol{\horounicode@point{264D}}%
216 \gdef\LibraSymbol{\horounicode@point{264E}}%
217 \gdef\ScorpioSymbol{\horounicode@point{264F}}%
218 \gdef\SagittariusSymbol{\horounicode@point{2650}}%
219 \gdef\CapricornSymbol{\horounicode@point{2651}}%
220 \gdef\AquariusSymbol{\horounicode@point{2652}}%
221 \gdef\PiscesSymbol{\horounicode@point{2653}}%

```

Traditional planets and luminaries:

```

222 \gdef\SunSymbol{\horounicode@point{2609}}%
223 \gdef\MoonSymbol{\horounicode@point{263D}}%
224 \gdef\MercurySymbol{\horounicode@point{263F}}%
225 \gdef\VenusSymbol{\horounicode@point{2640}}%
226 \gdef\MarsSymbol{\horounicode@point{2642}}%
227 \gdef\JupiterSymbol{\horounicode@point{2643}}%
228 \gdef\SaturnSymbol{\horounicode@point{2644}}%
229 \gdef\UranusSymbol{\horounicode@point{2645}}%
230 \gdef\NeptuneSymbol{\horounicode@point{2646}}%
231 \gdef\PlutoSymbol{\horounicode@point{2647}}%

```

Minor planets:

```

232 \gdef\ChironSymbol{\horounicode@point{26B7}}%
233 \gdef\CeresSymbol{\horounicode@point{26B3}}%

```

```

234 \gdef\PallasSymbol{\horounicode@point{26B4}}%
235 \gdef\JunoSymbol{\horounicode@point{26B5}}%
236 \gdef\VestaSymbol{\horounicode@point{26B6}}%

```

Derived points:

```

237 \gdef\NorthNodeSymbol{\horounicode@point{260A}}%
238 \gdef\SouthNodeSymbol{\horounicode@point{260B}}%
239 \gdef\LilithSymbol{\horounicode@point{26B8}}%

```

Direction:

```

240 \gdef\horoRetrogradeSymbol{\horounicode@point{211E}}%

```

Aspects:

```

241 \gdef\horoConjunctionSymbol{\horounicode@point{260C}}%
242 \gdef\horoOppositionSymbol{\horounicode@point{260D}}%
243 \gdef\horoTrineSymbol{\horounicode@point{25B3}}%
244 \gdef\horoSquareSymbol{\horounicode@point{25A1}}%
245 \gdef\horoSextileSymbol{\horounicode@point{26B9}}%
246 \gdef\horoQuincunxSymbol{\horounicode@point{26BB}}%
247 \gdef\horoSemisextileSymbol{\horounicode@point{26BA}}%
248 \gdef\horoSemisquareSymbol{\horounicode@point{2220}}%
249 \gdef\horoSesquiquadrateSymbol{\horounicode@point{26BC}}%
250 \fi\fi
251 }

```

Now invoke `\horounicode` to make these symbols default at the start of the document, except for the few (house cusps, and quintiles) that don't exist in Unicode.

```

252 \horounicode
253 \fi

```

§2.4. Symbols from wasysym.—

`\horowasysym` These, too, are inside a macro specific to the package so that we can switch among multiple symbol sets when more than one is loaded.

```

254 \ifhoro@wasysym\newcommand{\horowasysym}{%

```

Signs of the Zodiac:

```

255 \gdef\AriesSymbol{\aries}%
256 \gdef\TaurusSymbol{\taurus}%
257 \gdef\GeminiSymbol{\gemini}%
258 \gdef\CancerSymbol{\cancer}%
259 \gdef\LeoSymbol{\leo}%
260 \gdef\VirgoSymbol{\virgo}%
261 \gdef\LibraSymbol{\libra}%
262 \gdef\ScorpioSymbol{\scorpio}%
263 \gdef\SagittariusSymbol{\sagittarius}%
264 \gdef\CapricornSymbol{\capricornus}%
265 \gdef\AquariusSymbol{\aquarius}%
266 \gdef\PiscesSymbol{\pisces}%

```

Traditional planets and luminaries:

```
267 \gdef\SunSymbol{\astrosun}%
268 \gdef\MoonSymbol{\rightmoon}%
269 \gdef\MercurySymbol{\mercury}%
270 \gdef\VenusSymbol{\venus}%
271 \gdef\MarsSymbol{\mars}%
272 \gdef\JupiterSymbol{\jupiter}%
273 \gdef\SaturnSymbol{\saturn}%
274 \gdef\UranusSymbol{\uranus}%
275 \gdef\NeptuneSymbol{\neptune}%
276 \gdef\PlutoSymbol{\pluto}%
```

Aspects:

```
277 \gdef\horoConjunctionSymbol{\conjunction}%
278 \gdef\horoOppositionSymbol{\opposition}%
279 \gdef\horoTrineSymbol{\ensuremath{\bigtriangleup}}%
280 \gdef\horoSquareSymbol{\horow@sySquare}%
281 \gdef\horoSextileSymbol{\hexstar}%
282 }
```

Invoke `\horowasysym` to set it as default:

```
283 \horowasysym
284 \fi
```

§2.5. Symbols from `marvosym`.—

`\horomarvosym` As above. Note that because of the macro name conflict between `marvosym` and `starfont`, we use the saved copies of `marvosym`'s symbols instead of the contested names.

```
285 \ifhoro@marvosym\newcommand{\horomarvosym}{%
```

Signs of the Zodiac:

```
286 \gdef\AriesSymbol{\horom@rvAries}%
287 \gdef\TaurusSymbol{\horom@rvTaurus}%
288 \gdef\GeminiSymbol{\horom@rvGemini}%
289 \gdef\CancerSymbol{\horom@rvCancer}%
290 \gdef\LeoSymbol{\horom@rvLeo}%
291 \gdef\VirgoSymbol{\horom@rvVirgo}%
292 \gdef\LibraSymbol{\horom@rvLibra}%
293 \gdef\ScorpioSymbol{\horom@rvScorpio}%
294 \gdef\SagittariusSymbol{\horom@rvSagittarius}%
295 \gdef\CapricornSymbol{\horom@rvCapricorn}%
296 \gdef\AquariusSymbol{\horom@rvAquarius}%
297 \gdef\PiscesSymbol{\horom@rvPisces}%
```

Traditional planets and luminaries:

```
298 \gdef\SunSymbol{\horom@rvSun}%
299 \gdef\MoonSymbol{\horom@rvMoon}%
300 \gdef\MercurySymbol{\horom@rvMercury}%
301 \gdef\VenusSymbol{\horom@rvVenus}%
302 \gdef\MarsSymbol{\horom@rvMars}%
```

```

303 \gdef\JupiterSymbol{\horom@rvJupiter}%
304 \gdef\SaturnSymbol{\horom@rvSaturn}%
305 \gdef\UranusSymbol{\horom@rvUranus}%
306 \gdef\NeptuneSymbol{\horom@rvNeptune}%
307 \gdef\PlutoSymbol{\horom@rvPluto}%
308 }

```

Invoke `\horomarvosym` to set it as default:

```

309 \horomarvosym
310 \fi

```

§2.6. Symbols from `starfont`.—

`\horostarfont` This is the preferred set of astrological symbols. Like the others, all the definitions are inside a symbol set selection macro; like `marvosym`, we use previously saved copies of the package macros instead of invoking the conflicting names directly.

```

311 \ifhoro@starfont\newcommand{\horostarfont}{%

```

Signs of the Zodiac:

```

312 \gdef\AriesSymbol{\horost@rAries}%
313 \gdef\TaurusSymbol{\horost@rTaurus}%
314 \gdef\GeminiSymbol{\horost@rGemini}%
315 \gdef\CancerSymbol{\horost@rCancer}%
316 \gdef\LeoSymbol{\horost@rLeo}%
317 \gdef\VirgoSymbol{\horost@rVirgo}%
318 \gdef\LibraSymbol{\horost@rLibra}%
319 \gdef\ScorpioSymbol{\horost@rScorpio}%
320 \gdef\SagittariusSymbol{\horost@rSagittarius}%
321 \gdef\CapricornSymbol{\horost@rCapricorn}%
322 \gdef\AquariusSymbol{\horost@rAquarius}%
323 \gdef\PiscesSymbol{\horost@rPisces}%

```

Traditional planets and luminaries:

```

324 \gdef\SunSymbol{\horost@rSun}%
325 \gdef\MoonSymbol{\horost@rMoon}%
326 \gdef\MercurySymbol{\horost@rMercury}%
327 \gdef\VenusSymbol{\horost@rVenus}%
328 \gdef\MarsSymbol{\horost@rMars}%
329 \gdef\JupiterSymbol{\horost@rJupiter}%
330 \gdef\SaturnSymbol{\horost@rSaturn}%
331 \gdef\UranusSymbol{\horost@rUranus}%
332 \gdef\NeptuneSymbol{\horost@rNeptune}%
333 \gdef\PlutoSymbol{\horost@rPluto}%

```

Asteroids:

```

334 \gdef\ChironSymbol{\Chiron}%
335 \gdef\CeresSymbol{\Ceres}%
336 \gdef\PallasSymbol{\Pallas}%
337 \gdef\JunoSymbol{\Juno}%
338 \gdef\VestaSymbol{\Vesta}%

```

```

Derived points:
339 \gdef\NorthNodeSymbol{\NorthNode}%
340 \gdef\SouthNodeSymbol{\SouthNode}%
341 \gdef\MeanNodeSymbol{\NorthNode$_M$}%
342 \gdef\TrueNodeSymbol{\NorthNode$_T$}%
343 \gdef\LilithSymbol{\Lilith}%

Cusps and angles:
344 \gdef\CuspISymbol{\ASC}%
345 \gdef\CuspIVSymbol{\IC}%
346 \gdef\CuspVIISymbol{\DSC}%
347 \gdef\CuspXSymbol{\MC}%
348 \gdef\AscendantSymbol{\ASC}%
349 \gdef\MCSymbol{\MC}%
350 \gdef\VertexSymbol{\Vertex}%

Direction:
351 \gdef\horoRetrogradeSymbol{\Retrograde}%

Aspects:
352 \gdef\horoConjunctionSymbol{\Conjunction}%
353 \gdef\horoOppositionSymbol{\Opposition}%
354 \gdef\horoTrineSymbol{\Trine}%
355 \gdef\horoSquareSymbol{\horost@rSquare}%
356 \gdef\horoQuintileSymbol{\mathsf{Q}}%
357 \gdef\horoBiquintileSymbol{\mathsf{Q}^2}%
358 \gdef\horoSextileSymbol{\Sextile}%
359 \gdef\horoQuincunxSymbol{\Quincunx}%
360 \gdef\horoSemisextileSymbol{\Semisextile}%
361 \gdef\horoSemisquareSymbol{\Semisquare}%
362 \gdef\horoSesquiquadrateSymbol{\Sesquiquadrate}%
363 }

Invoke \horostarfont to set it as default:
364 \horostarfont
365 \fi

```

3. UTILITIES

§3.1. Trimming spaces.—This code is from Michael Downes’s series of mailing list postings available at <http://ctan.math.utah.edu/ctan/tex-archive/info/aro-bend/answer.015>. The trick of changing the letter Q’s catcode to make an exotic *Q* that can’t occur in ordinary token lists, is used elsewhere in the horoscop code so we leave it in force after defining `\trimspaces`.

```

366 %% WATCH OUT! MAKING Q EXOTIC HERE!
367 \catcode'\Q=3
368 \def\horo@cue{Q}

```

```
\trimspaces
```

```
369 \def\trimspaces#1{%
```

```

370 \begingroup
371   \aftergroup\toks\aftergroup0\aftergroup{%
372   \expandafter\@trim\expandafter\noexpand#1Q Q}%
373   \edef#1{\the\toks0}%
374 }
375 \long\def\@trim#1 Q{\@trim#1Q}
376 \long\def\@trimc#1Q#2{\afterassignment\endgroup \vfuzz\the\vfuzz#1}

```

§3.2. Redefine after current group.—Sometimes we want to pass data currently in a `\defined` macro to the world outside the current group, but we don’t want to pollute the global namespace with a `\gdef`. This code provides a way to do that. It’s a bit nasty and should only be used with macros whose contents translate to nice well-behaved strings. In practice, we use it for macros whose contents are decimal numbers.

`\horo@fterdef` First, `\horo@fterdef` puts three tokens into the `\aftergroup` queue to be evaluated outside the current group. They are: `\horo@ft@a`, the name of the macro being passed out, and a newly-constructed token whose name is `\horo@@` followed by the *contents* of the macro being passed out.

```

377 \def\horo@fterdef#1{%
378   \aftergroup\horo@ft@a
379   \aftergroup#1\expandafter\aftergroup\csname horo@@#1\endcsname
380 }

```

`\horo@eathead` This will be used in a moment; it simply drops the next seven characters.

```

381 \def\horo@eathead#1#2#3#4#5#6#7{}

```

`\horo@ft@a` Now, watch carefully. When we left `\horo@fterdef`, there were three tokens queued to run after the group. Now the group has ended and it’s time to look at those tokens. The first one is `\horo@ft@a`, so this macro runs and the next two become its arguments. The first of those is the name of the macro we will redefine, and the second is macro whose name consists of `\horo@@` and then the contents of the original macro.

So the `\edef` expands its argument, which starts by deferring expansion of `\horo@eathead`, and then finding the string value of the `\horo@@...` token. That string value consists of seven characters spelling out “`\-h-o-r-o-@-`” followed by the data to pass. When the deferred `\horo@eathead` runs it eats those seven characters. What’s left in the `\edef` body is just the data, which gets assigned to the macro.

```

382 \def\horo@ft@a#1#2{%
383   \edef#1{\expandafter\horo@eathead\string#2}%
384 }

```

§3.3. Polar to Cartesian.—

`\horo@polarconvert` Positioning the items on a wheel chart generally makes more sense in polar than Cartesian coordinates, so this computes the Cartesian coordinates for given polar coordinates. Takes four arguments; #1 and #2 are numbers containing the polar

coordinates r and θ (in degrees), and #3 and #4 are dimension registers that will contain the result encoded into points. The coordinate θ is zero on the $+X$ axis and increases counterclockwise under the standard mathematical convention.

This macro calls `\TG@sin` from the `trig` package, which may be a slightly dangerous thing to do. It trashes `\dimen@`.

```
385 \def\horo@polarconvert#1#2#3#4{%
386   \dimen@=\nin@ty\p@\advance\dimen@-#2\p@\TG@sin
387   #3=#1\dimen@
388   \dimen@=#2\p@\TG@sin
389   #4=#1\dimen@
390 }
```

§3.4. For-each.—These macros allow applying another macro on each item in a comma-separated list. They use exotic \mathcal{Q} for parsing the list because it's convenient, but that's probably not necessary.

`\horo@fe` Internal for `\horoforeach`. It checks whether the next comma-separated item is \mathcal{Q} , and if not, it calls `\horo@fe@b` on it and then invokes itself tail-recursively to do the next one.

```
391 \def\horo@fe#1,{%
392   \def\horo@fe@c{#1}%
393   \ifx\horo@fe@c\horo@cue\relax
394   \else
395     \horo@fe@b{#1}%
396     \expandafter\horo@fe
397   \fi
398 }
```

`\horoforeach` No \mathcal{Q} in the name because a user could possibly want to use it and it's reasonably safe. First argument is the list of items, comma-separated. Second is the action to perform on them, which will be invoked with each successive items from the list as its first and only argument. The list of items gets expanded. The logic is simple: it just saves the arguments and calls `\horo@fe` with a list containing an extra \mathcal{Q} item at the end to terminate the recursion.

```
399 \def\horoforeach#1#2{%
400   \edef\horo@fe@a{#1}%
401   \def\horo@fe@b{#2}%
402   \expandafter\horo@fe\horo@fe@a,\mathcal{Q},%
403 }
```

§3.5. Double for-each.—This is two nested for-eaches in one; that's a common enough operation that it seems useful to have a special feature for it instead of trying to nest regular for-eaches. We think of the outer index as x and the inner index as y , so the list of items for the outer loop is the x -list and the list of items for the inner loop is the y -list.

`\horo@dfe@b` Inner loop. Logic is very similar to `\horo@fe`: it gets a new item from the y -list, checks whether it is the \mathcal{Q} terminator, and if not, applies `\horo@dfe@action` to the pair of `\horo@dfe@x` and `\horo@dfe@y` before tail-recursing.

```

404 \def\horo@dfe@b#1,{%
405   \def\horo@dfe@y{#1}%
406   \ifx\horo@dfe@y\horo@cue\relax
407   \else
408     \horo@dfe@action{\horo@dfe@x}{\horo@dfe@y}%
409     \expandafter\horo@dfe@b
410   \fi
411 }

```

`\horo@dfe@a` Outer loop. The same kind of tail-recursive iteration through the x -list, applying `\horo@dfe@b` to each item.

```

412 \def\horo@dfe@a#1,{%
413   \def\horo@dfe@x{#1}%
414   \ifx\horo@dfe@x\horo@cue\relax
415   \else
416     \expandafter\horo@dfe@b\horo@dfe@ylist,Q,\relax
417     \expandafter\horo@dfe@a
418   \fi
419 }

```

`\horo@dblforeach` Master entry point for double for-each. This does get an @ because it's sufficiently dangerous. Arguments are the x -list, the y -list, and the action. The code just saves those and invokes `\horo@dfe@a` with the appropriate terminating \mathcal{Q} .

```

420 \def\horo@dblforeach#1#2#3{%
421   \edef\horo@dfe@xlist{#1}\edef\horo@dfe@ylist{#2}\def\horo@dfe@action{#3}%
422   \expandafter\horo@dfe@a\horo@dfe@xlist,Q,\relax
423 }

```

4. BASIC ASTROLOGICAL CALCULATION ROUTINES

These handle stuff like angle arithmetic. They're included even in the `nocalc` state because some of them are needed by the chart graphics routines, or are things you might still want to do even with manually-entered coordinates.

The `\ifhorocalculated` flag keeps track of whether we expect there to be valid data in the object-position variables; that's useful to prevent wasted calculation when we're using the ephemeris interface but it isn't working (for instance, because of `\write18` being turned off), because in such a case the positions are unknown.

```
424 \newif\ifhorocalculated\horocalculatedfalse
```

§4.1. Cusp information.—These theoretically might be user-settable, but would seldom be changed in practice.

`\hororightcoord` The object, or more correctly, the astrological longitude, to put at the right of the chart corresponding to polar coordinate $\theta = 0$. This normally would be the Descendant, which in turn would normally be the seventh house cusp.

```
425 \def\hororightcoord{\ifhorocalculated\horoCuspVIIPos\else180\fi}%

```

`\horocusps` A list of all the house cusps.
426 `\def\horocusps{CuspI,CuspII,CuspIII,CuspIV,CuspV,CuspVI,%`
427 `CuspVII,CuspVIII,CuspIX,CuspX,CuspXI,CuspXII}%`

`\horoangularcusps` A list of which cusps correspond to angles, used for showing the angles more prominently in some ready-made wheels.
428 `\def\horoangularcusps{CuspI,CuspIV,CuspVII,CuspX}`

§4.2. Calculating harmonics.—

`\horocalcharmonic` Multiplies all the object positions from `\horoobjects` by its parameter. Straight-forward implementation: it just calls `\horo@calch` on each object. House cusps and objects not mentioned in `\horoobjects` will be unchanged. `Pos` is the variable that gets multiplied; `DPos` gets set to the new value of `Pos`.
429 `\def\horocalcharmonic#1{%`
430 `\horoforeach{\horoobjects}{\horo@calch{#1}}%`
431 `}`

`\horo@calch` Internal for `\horocalcharmonic`. Gets the object's `Pos` into `\dimen@`, multiplies it by the harmonic number, takes it modulo 360 degrees, and saves it back to `Pos` and `DPos`.
432 `\def\horo@calch#1#2{%`
433 `\dimen@=\csname horo#2Pos\endcsname\p@\relax`
434 `\multiply\dimen@ by #1\relax`
435 `\horo@fixdimen@`
436 `\expandafter\edef\csname horo#2Pos\endcsname{\TG@rem@pt\dimen@}%`
437 `\expandafter\edef\csname horo#2DPos\endcsname{\TG@rem@pt\dimen@}%`
438 `}`

§4.3. Shifting positions.—

`\horoshiftobjects` Adds an offset to all the object positions in `\horoobjects`, much like `\horocalcharmonic` above.
439 `\def\horoshiftobjects#1{%`
440 `\horoforeach{\horoobjects}{\horo@shift{#1}}%`
441 `}`

`\horoshiftcusps` Adds an offset to all the object positions in `\horocusps`, much like `\horocalcharmonic` above.
442 `\def\horoshiftcusps#1{%`
443 `\horoforeach{\horocusps}{\horo@shift{#1}}%`
444 `}`

`\horo@shift` Internal for `\horoshiftobjects` and `-cusps`. Much like `\horo@calch` except it adds instead of multiplying.
445 `\def\horo@shift#1#2{%`
446 `\dimen@=\csname horo#2Pos\endcsname\p@\relax`
447 `\advance\dimen@ by #1\p@\relax`

```

448 \horo@fixdimen@
449 \expandafter\edef\csname horo#2Pos\endcsname{\TG@rem@pt\dimen@}%
450 \expandafter\edef\csname horo#2DPos\endcsname{\TG@rem@pt\dimen@}%
451 }

```

§4.4. Calculating equal cusps.—

`\horomakeequalcusps` Creates a set of equal-house cusps starting from CuspI set to the argument. Implementation simply sets them to hardcoded values and then shifts.

```

452 \def\horomakeequalcusps#1{
453   \def\horoCuspIPos{0}%
454   \def\horoCuspIIPos{30}%
455   \def\horoCuspIIIPos{60}%
456   \def\horoCuspIVPos{90}%
457   \def\horoCuspVPos{120}%
458   \def\horoCuspVIPos{150}%
459   \def\horoCuspVIIPos{180}%
460   \def\horoCuspVIIIPos{210}%
461   \def\horoCuspIXPos{240}%
462   \def\horoCuspXPos{270}%
463   \def\horoCuspXIPos{300}%
464   \def\horoCuspXIIPos{330}%
465   \horoshiftcusps{#1}%
466 }

```

`\horomakesigncusps` Truncates the argument to the sign boundary and then uses it for `\horomakeequalcusps`. The magic number 1966080 is 30° measured in units of $1^\circ/65536$.

```

467 \def\horomakesigncusps#1{
468   \dimen@=#1\p@\relax
469   \edef\horo@savecount@{\the\count0}%
470   \count0=\dimen@\relax
471   \divide\count0 by 1966080\relax
472   \multiply\count0 by 30\relax
473   \expandafter\horomakeequalcusps{\the\count0}%
474   \count0=\horo@savecount@\relax
475 }

```

§4.5. Calculating midpoints.—At present this only includes a private macro used to compute the midpoint of two raw angles. A future feature might actually compute midpoint charts by the time-space and/or object-position methods.

`\horo@midpoint` Find the midpoint between two angles, going around the circle in the shorter direction. Input angles are the two arguments, output goes into `\dimen@`.

```

476 \def\horo@midpoint#1#2{%
477   \dimen@#1\p@\relax
478   \advance\dimen@ by-#2\p@\relax
479   \horo@fixdimen@diff
480   \divide\dimen@ by2\relax
481   \advance\dimen@ by#2\p@\relax
482   \horo@fixdimen@
483 }

```

§4.6. Angle adjustments.—These all have to do with fixing the angle currently in `\dimen@`, which is assumed to be encoded at one point per degree, one way or another to make it well-behaved.

`\horo@chartrotate` Subtract `\hororightcoord` to account for rotation of the entire chart.

```
484 \def\horo@chartrotate{%
485   \advance\dimen@-\hororightcoord\p@
486   \horo@fixdimen@
487 }
```

`\horo@fixdimen@` Takes `\dimen@` modulo 360° so it ends up in the range $0 \dots 360^\circ$. The magic number 23592960 is 65536 times 360.

```
488 \def\horo@fixdimen@{%
489   \edef\horo@savecount@{\the\count0}%
490   \count0=\dimen@\relax
491   \ifnum\count0<0\relax
492     \divide\count0 by 23592960\relax
493     \advance\count0 by -1\relax
494   \else
495     \divide\count0 by 23592960\relax
496   \fi
497   \multiply\count0 by -360\relax
498   \advance\dimen@ by\count0\p@\relax
499   \count0=\horo@savecount@\relax
500 }
```

`\horo@fixdimen@diff` Very similar to `\horo@fixdimen@` except that the output range is $-180 \dots 180^\circ$, which is useful when we want to find the difference between two locations (tells us whether they are clockwise or counterclockwise from each other).

```
501 \def\horo@fixdimen@diff{%
502   \horo@fixdimen@
503   \ifdim\dimen@>180\p@\advance\dimen@ by -360\p@\fi
504 }
```

§4.7. Chart rotation.—Rotates the chart so that object #1 appears at angle #2, which is expressed as a θ coordinate so that 0 is to the right, 90 is up, and so on. This works by changing the value of `\hororightcoord`, which is checked during the actual plotting. Note it doesn't happen globally, and you probably don't want it to; the global definition of `\hororightcoord` is a smart one that picks up the Descendant's value.

`\hororotatechart`

```
505 \def\hororotatechart#1#2{%
506   \begingroup
507     \dimen@=#2\p@\relax
508     \advance\dimen@ by-\csname horo#1DPos\endcsname\p@\relax
509     \multiply\dimen@ by -1\relax
510     \edef\hororightcoord{\TG@rem@pt\dimen@}%
511     \horo@fterdef\hororightcoord
```

```
512 \endgroup
513 }
```

§4.8. Variable copying and saving.—

`\horocopyvar` Copies one variable to another on some objects, such as setting all DPos to the value of the corresponding Pos. Arguments are the list of objects, the from variable, and the to variable.

```
514 \def\horocopyvar#1#2#3{%
515 \horoforeach{#1}{\horo@cv@{#2}{#3}}%
516 }
```

`\horo@cv` Internal for `\horocopyvar`. Does the actual copying on one object.

```
517 \def\horo@cv#1#2#3{%
518 \expandafter\edef\csname horo#3#2\endcsname{\csname horo#3#1\endcsname}%
519 }
```

`\horosaveobjects` Save all the object positions (Pos) and velocities (Vel) to a macro. When the macro is called it will set all the Pos and Vel values to the values from when it was created, and all the DPos values to match Pos. It also saves and restores the value of `\horoobjects` itself. This works by building up the appropriate `\def` in the `\aftergroup` queue, and using `\horo@fterdef` to put the definitions of the individual variables right inside the body of the `\def`. So it's `\defs` within `\defs` going into the queue. Several different approaches for this had to be tried before I found one that didn't cause \TeX to complain about memory.

```
520 \def\horosaveobjects#1{%
521 \begingroup
522 \aftergroup\def\aftergroup#1\aftergroup{%
523 \horo@fterdef\horoobjects
524 \horoforeach{\horoobjects}{\horo@svo@a}%
525 \aftergroup\horo@svo@b
526 \aftergroup}%
527 \endgroup
528 }
```

`\horo@svo@a` First internal: this adds definitions for Pos and Vel to the `\aftergroup` queue.

```
529 \def\horo@svo@a#1{%
530 \expandafter\horo@fterdef\csname horo#1Pos\endcsname
531 \expandafter\horo@fterdef\csname horo#1Vel\endcsname
532 }
```

`\horo@svo@b` Second internal: copies Pos to DPos on all objects. This gets invoked by the `restore-objects` command we are creating, after it has restored Pos on all objects.

```
533 \def\horo@svo@b{%
534 \horocopyvar{\horoobjects}{Pos}{DPos}%
535 }
```

`\horosavecusps` This is the same general deal as `\horosaveobjects`, but it saves `\horocusps`, `\hororightcoord`, and the Pos of all cusps in `\horocusps`. The DPos will be set to Pos on restore.

```
536 \def\horosavecusps#1{%
537   \begingroup
538     \aftergroup\def\aftergroup#1\aftergroup{%
539       \edef\hororightcoord{\hororightcoord}%
540       \horofterdef\horocusps\horofterdef\hororightcoord
541       \horoforeach{\horocusps}{\horo@svc@a}%
542       \aftergroup\horo@svc@b
543     \aftergroup}%
544 \endgroup
545 }
```

`\horo@svc@a` Save Pos of a cusp to the `\aftergroup` queue.

```
546 \def\horo@svc@a#1{%
547   \expandafter\horofterdef\csname horo#1Pos\endcsname
548 }
```

`\horo@svc@b` Copy Pos to DPos; the difference from `\horo@svo@b` is that here we do it to all cusps instead of all objects.

```
549 \def\horo@svc@b{%
550   \horocopyvar{\horocusps}{Pos}{DPos}%
551 }
```

§4.9. Sexagesimal arithmetic.—These routines handle a number of conversions between DMS (degree, minute, and second), DDMS (direction, degree, minute, and second), SDMS (Zodiac sign, degree, minute, second), HMS (hour, minute, second), and flat decimal (which is usually the number of hours or degrees). They also handle typesetting things in these kinds of formats.

We generally store flat decimal numbers in macro definitions. For arithmetic they get assigned to dimension registers encoded as 1 degree = 1 point. Then when they get transformed into count registers they end up encoded as 1 degree = 65536 counts. That gives 18.2044 counts per second of arc, coincidentally very close to the original IBM PC's timer tick rate of 18.2065 ticks per second of time, which was a convenient fraction of the NTSC colour burst frequency. We have just enough bits of precision to be reasonably sure of converting DMS format to this format, doing a little bit of arithmetic, and converting back while keeping the errors less than a second of arc. If you want to calculate insane harmonics to sub-second precision (I have read of people seriously attempting the 105th harmonic) then you're out of luck, but you shouldn't be doing that anyway because your native certainly didn't give you an accurate enough birth time for it to be valid.

The decimal number under consideration at the moment is often stored in `\horo@data`, or in the form of a number of points in `\dimen@`, although some of these macros take arguments instead. Output is returned in several ways. Generally the calling convention is determined by what's most convenient in the macros that will call these ones.

`\horo@twodig` Add a leading zero to a nonnegative integer to make it at least two digits; useful for the part after the colon in times like “12:03.”

```
552 \def\horo@twodig#1{\ifnum#1<10\relax\edef#1{0#1}\fi}
```

`\horo@ddms` Takes a direction (one letter from NSEW), degree, minute, and second and translates it to decimal. South and West are considered to be negative. Note the input format: direction is one argument, but degree, minute, and second are separated by colons, and an exotic \mathcal{Q} terminates. The output goes to redefine the last argument.

```
553 \def\horo@ddms#1#2:#3:#4Q#5{%
554   \def\horo@data{}%
555   \uppercase{\if#1S\def\horo@data{-}\fi\if#1W\def\horo@data{-}\fi}%
556   \dimen@=#4\p@
557   \divide\dimen@ by 60\relax
558   \advance\dimen@ by #3\p@
559   \divide\dimen@ by 60\relax
560   \advance\dimen@ by #2\p@
561   \edef#5{\horo@data\TG@rem@pt\dimen@}%
562   \trimspaces{#5}%
563 }
```

Now, some configuration for the smart-rounding macros. This is a bit complicated because of the diversity of rounding modes the user might want. The general idea is that we convert decimal to sexagesimal in several stages corresponding to the mixed-based digits of the result, most significant down to least significant (i.e. sign, degrees, minutes, seconds). At each stage we are rounding down, taking the floor function. However, at some stage we may add an offset to make the floor function round to nearest. For a pure round to nearest, that offset corresponds to half the size we’re rounding to, and we add right at the start. But if we’re trying to respect higher-digit boundaries, we’ll add the offset later, because the higher digits should always have the values they would have with truncation. In that case, it becomes possible for the rounded-to digit to have an out-of-range value; so there’s a flag for whether to expose that or round it down. Finally, because configuring all that is such a mess, we have some convenience macros that preset commonly-used modes, and a super-convenience mode that attempts to auto-select a reasonable rounding mode depending on context.

`\horo@r@offset` Amount to add to `\dimen@` when we’re ready to add the offset. Default is for “transparent truncate” mode.

```
564 \def\horo@r@offset{1sp}%
```

`\horo@r@offdig` Digit before which to apply the offset. May be Z for Zodiac sign, D for degrees, M for minutes, S for seconds, anything else will not add the offset at all. Default is add at the start.

```
565 \def\horo@r@offdig{Z}%
```

`\ifhororoundclamp` Boolean flag: should we clamp rounded digits to their expected range? Default no.

```
566 \newif\ifhororoundclamp\hororoundclampfalse
```

Preset rounding modes covering all the reasonable ones. First we offer basic truncation, both the transparent version (which is default) and the strict version.

```
567 \def\hororoundstrictruncate{\def\horo@r@offdigit{X}}
```

```
568 \def\hororoundtruncate{\def\horo@r@offset{1sp}\def\horo@r@offdigit{Z}}
```

Now, pure round to nearest whatever.

```
569 \def\hororoundtosec{\def\horo@r@offset{9sp}\def\horo@r@offdigit{Z}}
```

```
570 \def\hororoundtomin{\def\horo@r@offset{546sp}\def\horo@r@offdigit{Z}}
```

```
571 \def\hororoundtodeg{\def\horo@r@offset{0.5pt}\def\horo@r@offdigit{Z}}
```

Round to a lower digit but keep the boundaries of a higher digit. This may result in out-of-range lower digits if `\hororoundclampfalse` is in force.

```
572 \def\hororoundtoseckeepsign{\def\horo@r@offset{9sp}\def\horo@r@offdigit{D}}
```

```
573 \def\hororoundtoseckeepdeg{\def\horo@r@offset{546sp}\def\horo@r@offdigit{M}}
```

```
574 \def\hororoundtoseckeepmin{\def\horo@r@offset{0.5pt}\def\horo@r@offdigit{S}}
```

```
575 \def\hororoundtominkeepsign{\def\horo@r@offset{546sp}\def\horo@r@offdigit{D}}
```

```
576 \def\hororoundtominkeepdeg{\def\horo@r@offset{0.5pt}\def\horo@r@offdigit{M}}
```

```
577 \def\hororoundtodegkeepsign{\def\horo@r@offset{0.5pt}\def\horo@r@offdigit{D}}
```

`\ifhororoundauto` Boolean flag: should the system automatically choose a reasonable rounding mode when it's about to do rounding?

```
578 \newif\ifhororoundauto\hororoundautotru
```

`\horo@d@dms@` This is an internal macro used in truncation and rounding. The first argument is a macro name representing the digit to extract; the second is the greatest value to allow for that digit if `\hororoundclamptrue` is active. The current value of `\dimen@` gets truncated to an integer number of points, and clamped if appropriate. That integer gets stored as the `\horo@fterdef` definition of the macro whose name was passed in, and then it gets subtracted out of `\dimen@` and the remaining fractional part is multiplied by 60. The idea here is that we're extracting the sexagesimal digits of the number, going from decimal degrees (or hours) to integer degrees and decimal minutes; from decimal minutes to integer minutes and decimal seconds; or from decimal seconds to integer seconds and decimal treyf, which will be discarded.

```
579 \def\horo@d@dms@#1#2{%
```

```
580 \count0=\dimen@\relax
```

```
581 \divide\count0by65536\relax
```

```
582 \ifhororoundclamp\ifnum\count0>#2 \count0=#2\relax\fi\fi
```

```
583 \edef#1{\the\count0}%
```

```
584 \horo@fterdef#1%
```

```
585 \advance\dimen@-\count0\p@
```

```
586 \multiply\dimen@ 60\relax
```

```
587 }
```

`\horo@d@dms@` Do a decimal to sexagesimal conversion using simple truncation. This is used internally for angles passed to external software. The magic number 100000 is guaranteed to exceed any digit value, so digits will never be clamped. The digits

go into the `\horo@fterdef` queue, so this must be called in a group and the results appear after the group.

```
588 \def\horo@d@dms@{%
589   \advance\dimen@1sp\relax
590   \horo@d@dms@@\horo@d@deg{100000}%
591   \horo@d@dms@@\horo@d@min{100000}%
592   \horo@d@dms@@\horo@d@sec{100000}%
593 }
```

`\horo@d@sdms` Decimal to SDMS. The number of degrees past 0° , stored at one point per degree in `\dimen@`, is converted to macros of decimal integers representing Zodiac sign (1=Aries...12=Pisces), degrees, minutes, and seconds of arc in `\horo@d@sign`, `\horo@d@deg`, `\horo@d@min`, and `\horo@d@sec` respectively. It obeys all the complicated rounding instructions set by the above macros.

First, we open a prophylactic group, and add the offset if we're adding it before the sign.

```
594 \def\horo@d@sdms{%
595   \begingroup
596   \def\horo@tmp{Z}\ifx\horo@r@offdigit\horo@tmp
597     \advance\dimen@\horo@r@offset\relax
598   \fi
```

Find the sign by converting `\dimen@` to a count register, and dividing by 1966080, which is 30° measured in counts at 65536 counts per degree. We add one to the result, in a separate register, because signs are one-based, and save the result in `\horo@d@sign` with `\horo@fterdef` to preserve it outside the group.

```
599   \count0=\dimen@\relax
600   \divide\count0by1966080\relax
601   \count1=\count0\relax
602   \advance\count1by1\relax
603   \edef\horo@d@sign{\the\count1}%
604   \horo@fterdef\horo@d@sign
```

Subtract out the angle corresponding to the start of the sign, to leave just the degree part in `\dimen@`.

```
605   \multiply\count0by30\relax%
606   \advance\dimen@-\count0\p@
```

Degrees digit: add the offset if this is where we're adding it, then extract the degrees into `\horo@d@deg`.

```
607   \def\horo@tmp{D}\ifx\horo@r@offdigit\horo@tmp
608     \advance\dimen@\horo@r@offset\relax
609   \fi
610   \horo@d@dms@@\horo@d@deg{29}%
```

Minutes digit: add the offset if this is where we're adding it, then extract the minutes into `\horo@d@min`.

```
611   \def\horo@tmp{M}\ifx\horo@r@offdigit\horo@tmp
612     \advance\dimen@\horo@r@offset\relax
613   \fi
```

```

614 \horo@d@dms@@\horo@d@min{59}%
      Seconds digit: add the offset if this is where we're adding it, then extract the
      seconds into \horo@d@sec.
615 \def\horo@tmp{S}\ifx\horo@r@offdigit\horo@tmp
616   \advance\dimen@\horo@r@offset\relax
617   \fi
618   \horo@d@dms@@\horo@d@sec{59}%
619 \endgroup
620 }

```

`\horo@d@dms` Decimal to DMS. This is the same idea as `\horo@d@sdms` without the handling of signs.

```

621 \def\horo@d@dms{%
622   \begingroup
623     \def\horo@tmp{Z}\ifx\horo@r@offdigit\horo@tmp
624       \advance\dimen@ \horo@r@offset\relax
625       \fi
626     \def\horo@tmp{D}\ifx\horo@r@offdigit\horo@tmp
627       \advance\dimen@ \horo@r@offset\relax
628       \fi
629     \horo@d@dms@@\horo@d@deg{100000}%
630     \def\horo@tmp{M}\ifx\horo@r@offdigit\horo@tmp
631       \advance\dimen@ \horo@r@offset\relax
632       \fi
633     \horo@d@dms@@\horo@d@min{59}%
634     \def\horo@tmp{S}\ifx\horo@r@offdigit\horo@tmp
635       \advance\dimen@ \horo@r@offset\relax
636       \fi
637     \horo@d@dms@@\horo@d@sec{59}%
638   \endgroup
639 }

```

`\horodsmstext` Typeset DSMS text. This might be invoked by the user to print a text version of calculated positions. It calls `\horo@d@dms` and then typesets the results with the defined symbols. When `\hororoundauto` is active, this will automatically choose `\hororoundtoseckkeepmin` rounding.

```

640 \def\horodsmstext#1{%
641   \begingroup
642     \ifhororoundauto\hororoundtoseckkeepmin\fi
643     \dimen@=#1\p@
644     \horo@d@sdms
645     \horo@d@deg\horodegrees\Zodiac{\horo@d@sign}%
646     \horo@d@min\horominutes\horo@d@sec\horoseconds
647   \endgroup
648 }

```

`\horotimetext` Typeset DMS text as if it were a time, treating “degrees” as hours. The minutes and seconds are forced to two digits so that times like 12:03:04 will come out as

such instead of as 12:3:4. No provision for AM/PM; if you want that, you're on your own. When `\hororoundauto` is active, this will automatically choose `\hororoundtruncate` rounding, on the theory that the time was probably entered by the user earlier with integer seconds and should be preserved.

```
649 \def\horotimetext#1{%
650   \begingroup
651     \ifhororoundauto\hororoundtruncate\fi
652     \dimen@=#1\p@
653     \horo@d@dms
654     \horo@twodig\horo@d@min\horo@twodig\horo@d@sec
655     \horo@d@deg:\horo@d@min:\horo@d@sec
656   \endgroup
657 }
```

`\horo@nsew` This is an internal function for the next one: it strips the positive/negative (not Zodiac) sign from `\dimen@`, making it positive, but uses the sign to select one of the two arguments which will become the definition of `\horo@calc@b`. The intended use is for typesetting things like longitude that could be East or West.

```
658 \def\horo@nsew#1#2{%
659   \ifdim\dimen@<\z@\relax
660     \def\horo@calc@b{#2}\multiply\dimen@ by-1\relax
661   \else
662     \def\horo@calc@b{#1}%
663   \fi
664 }
```

`\horolatlon` User-callable wrapper for `horo@nsew`. Typesets the first argument as a number of degrees, minutes, and seconds, with the second and third arguments added at the end for positive or negative respectively. Intended for typesetting latitude or longitude. The default rounding is `\hororoundtruncate`, to preserve user input.

```
665 \def\horolatlon#1#2#3{%
666   \begingroup
667     \ifhororoundauto\hororoundtruncate\fi
668     \dimen@=#1\p@\horo@nsew{#2}{#3}%
669     \horo@d@dms
670     \horo@d@deg\horodegrees%
671     \horo@d@min\horominutes\horo@d@sec\horoseconds~\horo@calc@b
672   \endgroup
673 }
```

`\hororxtxt` Typeset a “retrograde” symbol if the specified object is retrograde, determined by whether its `Vel` is negative.

```
674 \def\hororxtxt#1{%
675   \begingroup
676     \dimen@\csname horo#1Vel\endcsname\p@
677     \ifdim\dimen@<\z@\relax
678       \horoRetrogradeSymbol
679     \fi
```

```
680 \endgroup
681 }
```

§4.10. Setting all data at once.—

`\horocalparms` This sets the year, month, day, time of day, latitude, and longitude all at once for user convenience, making use of the sexagesimal conversions.

```
682 \def\horocalparms#1#2#3#4#5#6{%
683 \edef\horocalyear{#1}\edef\horocalmonth{#2}\edef\horocalday{#3}%
684 \horo@ddms N#4Q\horocalctime
685 \horo@ddms#5Q\horocalclon
686 \horo@ddms#6Q\horocalclat
687 }
```

5. ASTROLOG CALCULATION BACK END

These macros handle the interface to Astrolog, if it was selected. We start by checking whether it *was* selected:

```
688 \ifhoro@strolog
```

§5.1. Name translation.—We define a bunch of macros that represent the short strings used on the Astrolog command line to identify different objects and cusps. We also define names in the reverse direction for translating the abbreviations from Astrolog’s format back to horoscop’s. Some of those contain illegal characters for \TeX command names, so we have to use `\expandafter\def\csname`.

```
689 \def\horoSun@strologname{Sun }\def\horo@stSunObj{Sun}
690 \def\horoMoon@strologname{Moo }\def\horo@stMooObj{Moon}
691 \def\horoMercury@strologname{Mer }\def\horo@stMerObj{Mercury}
692 \def\horoVenus@strologname{Ven }\def\horo@stVenObj{Venus}
693 \def\horoMars@strologname{Mar }\def\horo@stMarObj{Mars}
694 \def\horoJupiter@strologname{Jup }\def\horo@stJupObj{Jupiter}
695 \def\horoSaturn@strologname{Sat }\def\horo@stSatObj{Saturn}
696 \def\horoUranus@strologname{Ura }\def\horo@stUraObj{Uranus}
697 \def\horoNeptune@strologname{Nep }\def\horo@stNepObj{Neptune}
698 \def\horoPluto@strologname{Plu }\def\horo@stPluObj{Pluto}
699 \def\horoChiron@strologname{Chi }\def\horo@stChiObj{Chiron}
700 \def\horoCeres@strologname{Cer }\def\horo@stCerObj{Ceres}
701 \def\horoPallas@strologname{Pal }\def\horo@stPalObj{Pallas}
702 \def\horoJuno@strologname{Jun }\def\horo@stJunObj{Juno}
703 \def\horoVesta@strologname{Ves }\def\horo@stVesObj{Vesta}
704 \def\horoNorthNode@strologname{Nod }\def\horo@stNodObj{NorthNode}
705 \def\horoSouthNode@strologname{S.N }
706 \expandafter\def\csname horo@stS.NObj\endcsname{SouthNode}
707 \def\horoLilith@strologname{Lil }\def\horo@stLilObj{Lilith}
```

Zodiac signs and cusps only need to go in the reverse direction. The signs expand to the start of the sign in degrees.

```
708 \def\horo@stAscObj{CuspI}
709 \expandafter\def\csname horo@st2ndObj\endcsname{CuspII}
```

```

710 \expandafter\def\csname horo@st3rdObj\endcsname{CuspIII}
711 \def\horo@stNadObj{CuspIV}
712 \expandafter\def\csname horo@st5thObj\endcsname{CuspV}
713 \expandafter\def\csname horo@st6thObj\endcsname{CuspVI}
714 \def\horo@stDesObj{CuspVII}
715 \expandafter\def\csname horo@st8thObj\endcsname{CuspVIII}
716 \expandafter\def\csname horo@st9thObj\endcsname{CuspIX}
717 \def\horo@stMidObj{CuspX}
718 \expandafter\def\csname horo@st11tObj\endcsname{CuspXI}
719 \expandafter\def\csname horo@st12tObj\endcsname{CuspXII}
720 \def\horo@stAriSign{0}
721 \def\horo@stTauSign{30}
722 \def\horo@stGemSign{60}
723 \def\horo@stCanSign{90}
724 \def\horo@stLeoSign{120}
725 \def\horo@stVirSign{150}
726 \def\horo@stLibSign{180}
727 \def\horo@stScoSign{210}
728 \def\horo@stSagSign{240}
729 \def\horo@stCapSign{270}
730 \def\horo@stAquSign{300}
731 \def\horo@stPisSign{330}

```

§5.2. House systems.—And, similarly, a set of macros the user can use to choose a house system. These work by redefining `\horo@housenumber` to the integer codes Astrolog uses.

```

732 \def\horoPlacidusHouses{\def\horo@housenumber{0}}
733 \def\horoKochHouses{\def\horo@housenumber{1}}
734 \def\horoEqualHouses{\def\horo@housenumber{2}}
735 \def\horoCampanusHouses{\def\horo@housenumber{3}}
736 \def\horoMeridianHouses{\def\horo@housenumber{4}}
737 \def\horoRegiomontanusHouses{\def\horo@housenumber{5}}
738 \def\horoPorphyryHouses{\def\horo@housenumber{6}}
739 \def\horoMorinusHouses{\def\horo@housenumber{7}}
740 \def\horoPolichPageHouses{\def\horo@housenumber{8}}
741 \def\horoAlcabitusHouses{\def\horo@housenumber{9}}
742 \def\horoEqualMCHouses{\def\horo@housenumber{10}}
743 \def\horoNeoPorphyryHouses{\def\horo@housenumber{11}}
744 \def\horoWholeHouses{\def\horo@housenumber{12}}
745 \def\horoVedicHouses{\def\horo@housenumber{13}}

```

§5.3. Interface stuff.—Define a read for Astrolog’s output, and a name for the temporary file.

```

746 \newread\horo@tmpfile
747 \edef\horo@tmpfname{\jobname.hor}%

```

`\horoastrologopt` The `\horoastrologopt` macro is interpolated onto the Astrolog command line, so the user can set it if they have extra options to pass.

```

748 \def\horoastrologopt{}

```

`\horoobjects` Default list of objects to calculate.

```
749 \def\horoobjects{Sun,Moon,Mercury,Venus,Mars,Jupiter,Saturn,%  
750 Uranus,Neptune,Pluto,NorthNode,Lilith,%  
751 Chiron,Ceres,Pallas,Juno,Vesta}
```

Placidus houses are the default:

```
752 \horoPlacidusHouses
```

§5.4. The actual calculation.—

`\horocalculate` Okay, here's where the excitement happens. This macro takes the current chart data from the `\horocalcyear`, `\horocalcmonth`, `\horocalcday`, `\horocalctime`, `\horocalclat`, and `\horocalclong` macros, along with various ancillary configuration macros, and runs Astrolog.

First we open the definition, and a group which will be used to prevent macro and register pollution.

```
753 \def\horocalculate{%  
754 \begingroup
```

We want to build up a list of all the objects we'll compute, space separated, expressed in Astrolog's own abbreviations. That list will go on the Astrolog command line. Cusps always get computed. We build the list by the same trick used earlier in `\horosaveobjects`: building up a `\def` command in the `\aftergroup` queue while we run a loop inside a group to decide what goes there. The macro we are defining is `\horo@calc@a`. In the loop we execute `\horo@calc@c` on each object or cusp. It adds a space and the appropriate `\horo<object>@strologname` macro to the `\aftergroup` queue.

```
755 \begingroup  
756 \aftergroup\def\aftergroup\horo@calc@a\aftergroup{%  
757 \horoforeach{\horoobjects}{\horo@calc@c}%  
758 \aftergroup}%  
759 \endgroup
```

So at this point `\horo@calc@a` contains the list of objects.

Now we get the time of day, which is stored in decimal hours, into hours and minutes format and put it in `\horo@calc@d`.

```
760 \dimen@\horocalctime\p@\horo@d@dms@\horo@twodig\horo@d@min  
761 \edef\horo@calc@d{\horo@d@deg:\horo@d@min\space GMT\space}%
```

We process `\horocalclon` into degrees:minutes format with E or W for East or West and put it in `\horo@calc@e`. The macro `\horo@calc@b` is used to pass out the East/West letter.

```
762 \dimen@=\horocalclon\p@\horo@nsew EW%  
763 \horo@d@dms@  
764 \horo@twodig\horo@d@min  
765 \edef\horo@calc@e{\horo@d@deg:\horo@d@min\horo@calc@b\space}%
```

Similarly, `\horocalclat` goes into degrees:minutes format with N or S for North or South and put it in `\horo@calc@f`.

```
766 \dimen@=\horocalclat\p@\horo@nsew NS%
```

```

767 \horo@d@dms@
768 \horo@twodig\horo@d@min
769 \edef\horo@calc@f{\horo@d@deg:\horo@d@min\horo@calc@b\space}%

```

Now we run Astrolog with the set of options we've built.

```

770 \immediate\write18{%
771     astrolog\space
772     -o0 \horo@tmpfname\space
773     -c \horo@houzenumber\space
774     -qa \horocalcmonth\space
775         \horocalcday\space
776         \horocalcyear\space
777         \horo@calc@d\horo@calc@e\horo@calc@f
778     -R0 \horo@calc@a
779     \horoastrologopt
780 }%

```

The output will have gone into the file named \horo@tmpfname. We open it up and then call a parsing routine, which puts the results into \horo@fterdefs. The \ifhorocalculated flag gets set to false; it will be upgraded to true if we can successfully read anything out of the output file.

```

781 \openin\horo@tmpfile=\horo@tmpfname\relax
782 \aftergroup\horocalculatedfalse
783 \horo@calc@parse
784 \closein\horo@tmpfile

```

Then the group ends, all the \horo@fterdefs happen, and that's the end of \horocalculate.

```

785 \endgroup
786 }

```

\horo@calc@c Internal macro for building up the object list. This takes its argument, wraps it in \horo and @strologname, and adds the resulting token to the \aftergroup queue.

```

787 \def\horo@calc@c#1{%
788 \expandafter\aftergroup\csname horo#1@strologname\endcsname
789 }

```

\horo@calc@parse Parse the Astrolog output file. It has a few kinds of lines in it; the ones we are interested in all start with -YF;⁸ but every line contains at least one space. So the first stage of parsing is to iterate over all the lines by tail recursion and call \horo@calc@parse@b, which looks for -YF lines.

```

790 \def\horo@calc@parse{%
791 \read\horo@tmpfile to \horo@calc@parse@a
792 \ifeof\horo@tmpfile\else
793 \expandafter\horo@calc@parse@b\horo@calc@parse@a\space x\space Q%
794 \horo@calc@parse
795 \fi
796 }

```

⁸-YFing in designated areas only.

```

\horo@YF Helper for \horo@calc@parse@b: simply expands to -YF.
797 \def\horo@YF{-YF}

\horo@calc@parse@b First stage of parsing a line of Astrolog output. Checks whether the first space-
separated word of the line is -YF, and if it is, calls the second-stage parser
\horo@calc@parse@c on the rest of the line.
798 \def\horo@calc@parse@b#1 #2Q{%
799 \def\horo@calc@parse@b@{#1}%
800 \ifx\horo@calc@parse@b@\horo@YF\horo@calc@parse@c#2Q\fi
801 }

\horo@calc@parse@c Second stage of parsing a line of Astrolog output. We use TEX delimited argu-
ment parsing to find the fields on the line. They are a bit complicated but cor-
respond to object name; longitude in degree, sign, decimal minutes; the latitude
(which we ignore); the longitude speed in degrees per day; and the distance.
There's also some garbage at the end as a result of the safety stuff added by
\horo@calc@parse.
802 \def\horo@calc@parse@c#1: #2 #3 #4, #5 #6, #7 #8Q{%
    First convert from degrees, sign, and decimal minutes, to decimal degrees:
803 \dimen@=#4\p@
804 \divide\dimen@ by 60\relax
805 \advance\dimen@ by #2\p@
806 \advance\dimen@ by \csname horo@st#3Sign\endcsname\p@
    Save the result in Pos:
807 \expandafter\edef\csname
808 horo\csname horo@st#10bj\endcsname Pos\endcsname{%
809 \TG@rem@pt\dimen@}%
810 \expandafter\horo@fterdef\csname
811 horo\csname horo@st#10bj\endcsname Pos\endcsname
    Save it in DPos as well:
812 \expandafter\edef\csname
813 horo\csname horo@st#10bj\endcsname DPos\endcsname{%
814 \TG@rem@pt\dimen@}%
815 \expandafter\horo@fterdef\csname
816 horo\csname horo@st#10bj\endcsname DPos\endcsname
    Save the velocity, which is already nicely formatted in #7, to Vel:
817 \expandafter\def\csname
818 horo\csname horo@st#10bj\endcsname Vel\endcsname{#7}%
819 \expandafter\horo@fterdef\csname
820 horo\csname horo@st#10bj\endcsname Vel\endcsname
    Now we have at least a little bit of valid data, so things are probably cool.
821 \aftergroup\horocalculatedtrue
822 }

    End of the \ifhoro@strolog conditional:
823 \fi

```

6. SWISS EPHEMERIS CALCULATION BACK END

The interface to Swiss Ephemeris goes via the `swetest` command-line program. Unlike `Astrolog`, which gets invoked once to do all objects, we run `swetest` once for each object and once more for all cusps.

Although `\ifhoro@ppendtmp` is only used in the case where `swetest` is enabled, we have to define it outside the conditional because otherwise `TEX` will see the `\fi` that closes `\ifhoro@ppendtmp` as closing something else, with resulting entertaining results.

```
\ifhoro@ppendtmp
```

```
824 \newif\ifhoro@ppendtmp\horo@ppendtmpfalse
```

Then the rest of this section is conditional on the `swetest` option being enabled.

```
825 \ifhoro@swetest
```

§6.1. Command-line options.—One of these gets passed to `horoscop` according to the object currently being calculated. Users who want to add other objects will need to create the corresponding `\horo⟨object⟩SEOpt` macros.

```
826 \def\horoSunSEOpt{-p0}
827 \def\horoMoonSEOpt{-p1}
828 \def\horoMercurySEOpt{-p2}
829 \def\horoVenusSEOpt{-p3}
830 \def\horoMarsSEOpt{-p4}
831 \def\horoJupiterSEOpt{-p5}
832 \def\horoSaturnSEOpt{-p6}
833 \def\horoUranusSEOpt{-p7}
834 \def\horoNeptuneSEOpt{-p8}
835 \def\horoPlutoSEOpt{-p9}
836 \def\horoChironSEOpt{-pD}
837 \def\horoPholusSEOpt{-pE}
838 \def\horoCeresSEOpt{-pF}
839 \def\horoPallasSEOpt{-pG}
840 \def\horoJunoSEOpt{-pH}
841 \def\horoVestaSEOpt{-pI}
842 \def\horoMeanNodeSEOpt{-pm}
843 \def\horoTrueNodeSEOpt{-pt}
844 \def\horoLilithSEOpt{-pA}
```

These next few are fake options; the Ascendant, MC, ARMC, and Vertex are always calculated, and get their values through special handling instead of the usual calculation mechanism, but if the user includes them in `\horoobjects` we calculate the position of the Sun (overwritten later) so as not to have trouble with an unknown object having been listed.

```
845 \def\horoAscendantSEOpt{-p0}
846 \def\horoMCSEOpt{-p0}
847 \def\horoARMCSEOpt{-p0}
848 \def\horoVertexSEOpt{-p0}
```

`\horosweopt` Any extra options can be passed here.

```
849 \def\horosweopt{}
```

§6.2. House systems.—Gauquelin sectors aren't included here because they break the basic model of 12 houses; other things would have to change dramatically to use them properly, and it's not clear whether people who want to typeset wheel charts actually want to typeset them with 36 houses at all anyway. Objects called Ascendant, MC, ARMC, and Vertex are always calculated, and may be of interest (to manually add to `\horoobjects` after the calculation) if using a house system where they don't coincide with house cusps.

```
850 \def\horoAlcabitusHouses{\def\horo@houseletter{b}}
851 \def\horoAxialHouses{\def\horo@houseletter{x}}
852 \def\horoAzimuthalHouses{\def\horo@houseletter{h}}
853 \def\horoCampanusHouses{\def\horo@houseletter{c}}
854 \def\horoEqualHouses{\def\horo@houseletter{a}}
855 \def\horoKochHouses{\def\horo@houseletter{k}}
856 \def\horoKrusinskiHouses{\def\horo@houseletter{U}}
857 \def\horoMorinusHouses{\def\horo@houseletter{m}}
858 \def\horoPlacidusHouses{\def\horo@houseletter{p}}
859 \def\horoPolichPageHouses{\def\horo@houseletter{t}}
860 \def\horoPorphyryHouses{\def\horo@houseletter{o}}
861 \def\horoRegiomontanusHouses{\def\horo@houseletter{r}}
862 \def\horoVehlowHouses{\def\horo@houseletter{v}}
```

§6.3. Interface stuff.—Create a new read for the temporary file and save its name.

```
863 \newread\horo@tmpfile
864 \edef\horo@tmpfname{\jobname.hor}
```

`\horo@readdata` Read a line from the data file; flag a calculation failure if there's nothing to read. The result goes into `\horo@data`. The line of data is expected to be a decimal number with possible leading or trailing spaces, which we trim off before saving it.

```
865 \def\horo@readdata{%
866   \ifeof\horo@tmpfile
867     \def\horo@data{0.0}%
868     \horocalculatedfalse
869   \else
870     \begingroup
871     \let\do\@makeother\dospecials
872     \read\horo@tmpfile to \horo@data
873     \trimspaces\horo@data
874     \horo@fterdef\horo@data
875   \endgroup
876 \fi
877 }
```

`\horo@readpos` Read a line from the data file and put it into Pos and DPos of the specified object.

```

878 \def\horo@readpos#1{%
879   \horo@readdata
880   \expandafter\let\csname horo#1Pos\endcsname\horo@data
881   \expandafter\let\csname horo#1DPos\endcsname\horo@data
882 }

```

`\horo@readposvel` Read (D)Pos as above, and then also read the Vel of the same object.

```

883 \def\horo@readposvel#1{%
884   \horo@readpos{#1}%
885   \horo@readdata
886   \expandafter\let\csname horo#1Vel\endcsname\horo@data
887 }

```

`\horoobjects` Default object list for Swiss Ephemeris.

```

888 \def\horoobjects{Sun,Moon,Mercury,Venus,Mars,Jupiter,Saturn,%
889                 Uranus,Neptune,Pluto,MeanNode,Lilith,%
890                 Chiron,Ceres,Pallas,Juno,Vesta}

```

Placidus houses are default:

```
891 \horoPlacidusHouses
```

§6.4. Calculation routines.—

`\horo@calcobj@` Calculate the position *or* velocity of an object. The second argument is either “s” for position, or “l” for velocity; these are option letters for `swetest`’s format string. The results will go into the temporary file to be read later.

```

892 \def\horo@calcobj@#1#2{%
      Start by opening a prophylactic group:
893   \begingroup
      Convert the time of day to hours, minutes, and seconds:
894   \dimen@\horocalctime\p@\horo@d@dms@
895   \horo@twodig\horo@d@min\horo@twodig\horo@d@sec

```

Then do a shell escape, filling in all the appropriate options. The formats being used (passed in through #2) are such as to create lines of output containing only decimal numbers, with a few harmless stray spaces before or after.

Note the use of `\ifhoro@ppendtmp`. It starts out false, so only one `>` is used on the command line and the file gets overwritten.

```

896   \immediate\write18{%
897     swetest -f#2 -head\space
898     -b\horocalcday.\horocalcmonth.\horocalcyear\space
899     -ut\horo@d@deg:\horo@d@min:\horo@d@sec\space
900     \csname horo#1SEOpt\endcsname\space
901     \horosweopt\space
902     \ifhoro@egrep| egrep '[0-9.-]+' \space\fi
903     >\ifhoro@ppendtmp >\fi\space
904     \horo@tmpfname
905   }%

```

But after closing the prophylactic group...

```
906   \endgroup
... we set the flag true so that subsequent calls to swetest will append instead
of overwriting.
907   \horo@ppendtmptrue
908 }
```

`\horo@calcobj` Simply invokes `\horo@calcobj` twice to get both position and velocity.

```
909 \def\horo@calcobj#1{%
910   \horo@calcobj@{#1}-1}%
911   \horo@calcobj@{#1}-s}%
912 }
```

`\horocalculate` Main calculation macro. This starts by creating an option for cusp calculation, as if “Cusps” was the name of a legitimate object. It sets `\horo@ppendtmpfalse` to force the temporary file to be overwritten, and `\horocalculatedtrue` (to be possibly turned off if there is trouble reading the results). Then it uses a `\horoforeach` to call `\horo@calcobj` to get position and speed for all the regular objects. The internal function `\horo@calcobj@` is called as if it were going to compute the position of the Cusps object, but that actually writes 16 numbers to the temporary file.

```
913 \def\horocalculate{%
914   \edef\horoCuspsSEOpt{-house\horocalclon,\horocalclat,%
915                       \horo@houseletter\space-p}%
916   \horo@ppendtmpfalse
917   \horocalculatedtrue
918   \horoforeach{\horoobjects}{\horo@calcobj}%
919   \horo@calcobj@{Cusps}-1}%
```

Now with all the data in the temporary file, we read it in. Open the file, read position and velocity for all the regular objects, and then read the cusps (12 of them) and other things added by `swetest` (4 of those) with another loop.

```
920   \openin\horo@tmpfile=\horo@tmpfname\relax
921   \horoforeach{\horoobjects}{\horo@readposvel}%
922   \horoforeach{\horocusps,Ascendant,MC,ARMC,Vertex}{\horo@readpos}%
923   \closein\horo@tmpfile\relax
924 }
```

End the `\ifhoro@swetest` conditional:

```
925 \fi
```

7. LOW-LEVEL CHART GRAPHICS

These macros are a grab bag of utilities for drawing bits and pieces of wheel charts, used by the higher-level routines to actually draw complete charts.

§7.1. Configuration stuff.—

- `\ifhorodrawcusps` Setting for whether cusps should be drawn in ready-made wheel charts. The reason the user might not want that would be if they're trying to make do without a birth time. It has to be declared in this unconditional context for parsing reasons.
926 `\newif\ifhorodrawcusps\horodrawcusps>true`
- `\ifhoroboldangles` Similar setting for whether angular-house cusps should be drawn extra bold. A user who really doesn't want cusps must turn off all of `\ifhorodrawcusps`, `\ifhoroboldangles`, and `\ifhoroanglearrows`, as well as (probably) internal house labels; that is to allow drawing of angular cusps and not other cusps, should the user want such a thing.
927 `\newif\ifhoroboldangles\horoboldangle>true`
- `\ifhoroanglearrows` Setting for whether to draw angular house cusps as arrows instead of regular lines.
928 `\newif\ifhoroanglearrows\horoanglearrow>true`
- Now we can enter a conditional on wheel charts not being disabled by package option.
929 `\ifhoro@wheels`
- `\horounitlength` This determines the size of the horoscope environment, which is 100 unit-lengths square. The default value is a little less than 1/100 of `\textwidth` because we're going to want to put things with their centres on a circle of diameter 100, and the things may actually spill a little outside the circle. Exact value determined by trial and error, and depending on the user's application may need to be changed significantly.
930 `\newlength{\horounitlength}`
931 `\setlength{\horounitlength}{0.00952\textwidth}`
- `\horoanglecuswidth` Line width to use in drawing angular cusps when `\horoboldangles` is turned on.
932 `\newlength{\horoanglecuswidth}`
933 `\setlength{\horoanglecuswidth}{1.44pt}`

§7.2. Horoscope environment and chart labels.—

- `horoscope` This is the basic environment in which we'll draw charts. It's just a picture of size 100×100 units with the origin in the middle.
934 `\newenvironment{horoscope}{%`
935 `\setlength{\unitlength}{\horounitlength}%`
936 `\begin{picture}(100,100)(-50,-50)%`
937 `}{%`
938 `\end{picture}%`
939 `}`

We also have macros to be used inside a horoscope to typeset notes for the whole chart at the centre or any of the four corners of the chart, in what might otherwise be wasted space.

```

\horoCnote
940 \newcommand{\horoCnote}[1]{%
941   \put(0,0){\makebox(0,0){\parbox{75\unitlength}{\centering #1}}}%
942 }

\horoULnote
943 \newcommand{\horoULnote}[1]{%
944   \put(-50,50){\makebox(40,0)[t]{\parbox[t]{40\unitlength}{\raggedright #1}}}%
945 }

\horoURnote
946 \newcommand{\horoURnote}[1]{%
947   \put(10,50){\makebox(40,0)[t]{\parbox[t]{40\unitlength}{\raggedleft #1}}}%
948 }

\horoLLnote
949 \newcommand{\horoLLnote}[1]{%
950   \put(-50,-50){\makebox(40,0)[b]{\parbox[t]{40\unitlength}{\raggedright #1}}}%
951 }

\horoLRnote
952 \newcommand{\horoLRnote}[1]{%
953   \put(10,-50){\makebox(40,0)[b]{\parbox[t]{40\unitlength}{\raggedleft #1}}}%
954 }

```

§7.3. Scratch dimension registers.—We define our own aliases for eight dimension registers. \LaTeX or other packages probably use these same registers, but we always use them within a group in such a way as to not conflict.

```

955 \dimendef\horo@dim@x=1\relax
956 \dimendef\horo@dim@y=2\relax
957 \dimendef\horo@dim@xa=3\relax
958 \dimendef\horo@dim@ya=4\relax
959 \dimendef\horo@dim@xb=5\relax
960 \dimendef\horo@dim@yb=6\relax
961 \dimendef\horo@dim@xc=7\relax
962 \dimendef\horo@dim@yc=8\relax

```

§7.4. Polar-coordinate puts.—These are the simplest graphics operations in polar coordinates: putting an arbitrary graphics object at a polar-coordinate location in a picture environment, drawing a radial (constant- θ) line segment, or drawing a line segment between two arbitrary points. These are designed for internal use and may expand arguments in a less than friendly way, so although they work well for our higher-level macros they perhaps should not be used incautiously.

`\horo@putpolar` Arguments are r , θ , and the object to be `\put` at those coordinates. The centre of the object goes at the specified coordinates. Implementation is straightforward: we just expand and save the polar coordinates to temporary macros, run `\horo@polarconvert` to convert to Cartesian, and then do a `\put`.

```

963 \def\horoputpolar#1#2#3{%
964   \begingroup
965     \edef\p@one{#1}\edef\p@two{#2}%
966     \horopolarconvert\p@one\p@two\horodim@x\horodim@y
967     \put(\TG@rem@pt\horodim@x,\TG@rem@pt\horodim@y){\makebox(0,0){#3}}%
968   \endgroup
969 }

```

`\horoputradial` Draw a radial; that is, a line segment at a constant θ coordinate. Arguments are the inner radius, the length, and θ . Implementation is straightforward. One small gotcha is that instead of adding the inner radius and length to get the outer radius, we treat $(length, \theta)$ as a vector and do the addition on the Cartesian side, because that saves some shuffling between macro definitions and dimension registers.

```

970 \def\horoputradial#1#2#3{%
971   \begingroup
972     \edef\p@one{#1}\edef\p@two{#2}\edef\p@three{#3}%
973     \horopolarconvert\p@one\p@three\horodim@xa\horodim@ya
974     \horopolarconvert\p@two\p@three\horodim@xb\horodim@yb
975     \advance\horodim@xb by\horodim@xa
976     \advance\horodim@yb by\horodim@ya
977     \Line(\TG@rem@pt\horodim@xa,\TG@rem@pt\horodim@ya)%
978         (\TG@rem@pt\horodim@xb,\TG@rem@pt\horodim@yb)%
979   \endgroup
980 }

```

`\horoputline` This is the more general line-segment primitive: it takes two (r, θ) pairs and draws a line between them.

```

981 \def\horoputline#1#2#3#4{%
982   \begingroup
983     \edef\p@one{#1}\edef\p@two{#2}\edef\p@three{#3}\edef\p@four{#4}%
984     \horopolarconvert\p@one\p@two\horodim@xa\horodim@ya
985     \horopolarconvert\p@three\p@four\horodim@xb\horodim@yb
986     \Line(\TG@rem@pt\horodim@xa,\TG@rem@pt\horodim@ya)%
987         (\TG@rem@pt\horodim@xb,\TG@rem@pt\horodim@yb)%
988   \endgroup
989 }

```

§7.5. Arrowheads.—Although L^AT_EX can draw arrows (which it calls “vectors”), the implementation has annoying limitations that make it unsuitable for use in horoscop’s charts. Here we define our own arrowhead-drawing routine, with a different set of annoying limitations.

`\horoputarrowhead` Draw an arrowhead with its tip at $(r = \#1, \theta = \#2)$, pointing outward from the origin. The arrowhead is an appropriately scaled and rotated version of the nonconvex polygon with Cartesian vertex coordinates $(0, 1), (1, 0), (0, -1), (3, 0)$. The $\#3$ argument is equal to the unit length of that coordinate system, measured in horoscope length units.

Implementation starts by opening a group and saving the arguments in macros.

```

990 \def\horoputarrowhead#1#2#3{%
991   \begingroup
992     \edef\horo@pa@r{#1}\edef\horo@pa@th{#2}%
993     \edef\horo@pa@sc@le{#3}%

```

Convert tip coordinates to Cartesian.

```

994   \horo@polarconvert\horo@pa@r\horo@pa@th\horo@dim@x\horo@dim@y

```

Compute a vector in the direction of the tip, used for offsetting its coordinates to find the coordinates of the other vertices. This vector corresponds to a unit vector in the +X direction of the coordinate system mentioned above.

```

995   \horo@polarconvert\horo@pa@sc@le\horo@pa@th\horo@dim@xa\horo@dim@ya

```

Compute the Cartesian coordinates of the two vertices on either side; each is 3 units back from the tip in the -X direction and ±1 unit off to the side.

```

996   \horo@dim@xb=\horo@dim@x\relax\horo@dim@yb=\horo@dim@y\relax
997   \advance\horo@dim@xb by-3\horo@dim@xa\relax
998   \advance\horo@dim@yb by-3\horo@dim@ya\relax
999   \advance\horo@dim@xb by-\horo@dim@ya\relax
1000  \advance\horo@dim@yb by\horo@dim@xa\relax
1001  \horo@dim@xc=\horo@dim@x\relax\horo@dim@yc=\horo@dim@y\relax
1002  \advance\horo@dim@xc by-3\horo@dim@xa\relax
1003  \advance\horo@dim@yc by-3\horo@dim@ya\relax
1004  \advance\horo@dim@xc by\horo@dim@ya\relax
1005  \advance\horo@dim@yc by-\horo@dim@xa\relax

```

Compute the Cartesian coordinates of the final vertex, at the back of the arrowhead. The sequencing is careful because of not wanting to use an extra pair of registers; the @xa series used to hold the offset vector and will now hold the vertex coordinates.

```

1006  \multiply\horo@dim@xa by-2\relax\multiply\horo@dim@ya by-2\relax
1007  \advance\horo@dim@xa by\horo@dim@x\relax
1008  \advance\horo@dim@ya by\horo@dim@y\relax

```

Plot the arrowhead, using pict2e's \polygon* macro.

```

1009  \polygon*%
1010  (\TG@rem@pt\horo@dim@x,\TG@rem@pt\horo@dim@y)%
1011  (\TG@rem@pt\horo@dim@xb,\TG@rem@pt\horo@dim@yb)%
1012  (\TG@rem@pt\horo@dim@xa,\TG@rem@pt\horo@dim@ya)%
1013  (\TG@rem@pt\horo@dim@xc,\TG@rem@pt\horo@dim@yc)%
1014  (\TG@rem@pt\horo@dim@x,\TG@rem@pt\horo@dim@y)%
1015  \endgroup
1016 }

```

§7.6. Polar-coordinate curves.—Some chart types need to draw curves that are neither radial line segments nor origin-centric circular arcs. For that matter, they need origin-centric circular arcs, which are non-trivial. The curves defined here

are cubic Bézier splines with two endpoints specified in polar coordinates and two control points both set to the value of this vector formula:

$$\mathbf{c} = 2\mathbf{m} - \frac{1}{2}(\mathbf{a} + \mathbf{b})$$

where \mathbf{c} is the control point, \mathbf{a} and \mathbf{b} are the endpoints, and \mathbf{m} is their *polar* midpoint: the point whose r coordinate is the average of the r coordinates of \mathbf{a} and \mathbf{b} , and whose θ coordinate is midway between theirs, going the shorter direction around the circle.

This formula was derived by educated trial and error. If the endpoints happen to have the same r , it is a reasonable approximation of the origin-centric arc as long as they're within about 30° of each other, and it looks halfway acceptable even at larger distances. Interesting special effects are obtained if the r values do not match.

`\horoputcurve` Draw a smooth curve connecting $(r = \#1, \theta = \#2)$ to $(r = \#3, \theta = \#4)$. Implementation starts by opening a group and saving all the arguments in macros.

```

1017 \def\horoputcurve#1#2#3#4{%
1018   \begingroup
1019     \edef\horo@pc@r@ne{#1}\edef\horo@pc@th@ne{#2}%
1020     \edef\horo@pc@rtw@{#3}\edef\horo@pc@thtw@{#4}%
      Both  $\theta$  coordinates are subjected to \horo@chartrotate.
1021     \dimen@=\horo@pc@th@ne\p@\relax\horo@chartrotate
1022     \edef\horo@pc@th@ne{\TG@rem@pt\dimen@}%
1023     \dimen@=\horo@pc@thtw@\p@\relax\horo@chartrotate
1024     \edef\horo@pc@thtw@{\TG@rem@pt\dimen@}%
      Convert first endpoint to Cartesian.
1025     \horo@polarconvert\horo@pc@r@ne\horo@pc@th@ne\horo@dim@xa\horo@dim@ya
      Convert second endpoint to Cartesian.
1026     \horo@polarconvert\horo@pc@rtw@\horo@pc@thtw@\horo@dim@xb\horo@dim@yb
      Compute the polar-coordinate midpoint. Radius is simple average, but angle
      must go through \horo@midpoint to handle cases like wrapping around  $360^\circ$ .
1027     \dimen@=\horo@pc@r@ne\p@\relax
1028     \advance\dimen@ by\horo@pc@rtw@\p@\relax
1029     \divide\dimen@ by2\relax
1030     \edef\horo@pc@rmid{\TG@rem@pt\dimen@}%
1031     \horo@midpoint\horo@pc@th@ne\horo@pc@thtw@
1032     \edef\horo@pc@thmid{\TG@rem@pt\dimen@}%
      Convert midpoint to Cartesian.
1033     \horo@polarconvert\horo@pc@rmid\horo@pc@thmid\horo@dim@x\horo@dim@y
      Compute the control point, which basically means moving the control point
      away from the polar midpoint by an amount equal to and opposite from the
      difference between it and the Cartesian midpoint. Recall that epic's spline curves
      are based on the Chaikin technique of cutting corners off a polyline until it looks

```

like a smooth curve. With the switch to `pict2e` we are no longer using classic L^AT_EX splines, but the underlying concept of a cubic spline should remain the same. The initial polyline looks like two sides of a triangle, with the remaining side being the segment directly connecting the two endpoints. If we place the control point *twice* as far away from that direct segment as the polar midpoint would be, then after we're finished cutting corners the middle of the remaining curve should end up pretty close to half its original (control point) distance from the Cartesian midpoint, and *the same distance* as the polar midpoint would be—thus passing through the polar midpoint. We want it to pass through the polar midpoint at least in the important special case of drawing an origin-centred arc. That's the education behind the educated guess of the control point formula. In actual fact it doesn't work perfectly because the cut ratio isn't right, but the result looks good anyway.

```

1034 \multiply\horo@dim@x by -4\relax
1035 \advance\horo@dim@x by \horo@dim@xa\relax
1036 \advance\horo@dim@x by \horo@dim@xb\relax
1037 \divide\horo@dim@x by -2\relax
1038 \multiply\horo@dim@y by -4\relax
1039 \advance\horo@dim@y by \horo@dim@ya\relax
1040 \advance\horo@dim@y by \horo@dim@yb\relax
1041 \divide\horo@dim@y by -2\relax

```

With all the relevant coordinates computed, plot the actual curve and end.

```

1042 \cbezier(\TG@rem@pt\horo@dim@xa,\TG@rem@pt\horo@dim@ya)%
1043          (\TG@rem@pt\horo@dim@x,\TG@rem@pt\horo@dim@y)%
1044          (\TG@rem@pt\horo@dim@x,\TG@rem@pt\horo@dim@y)%
1045          (\TG@rem@pt\horo@dim@xb,\TG@rem@pt\horo@dim@yb)%
1046 \endgroup
1047 }

```

`\horoconncurve` This is the most common case of using `\horoputcurve`: drawing a little connector to show the relationship between an object label (plotted at the object's DPos) and a radial that shows the object's real location (plotted at Pos). It draws a curve from radius #1 and the Pos of object #3, to radius #2 and the DPos of object #3. Object name last to make it easy to call inside a `\horoforeach`.

```

1048 \def\horoconncurve#1#2#3{%
1049 \horoputcurve{#1}{\csname horo#3Pos\endcsname}%
1050             {#2}{\csname horo#3DPos\endcsname}%
1051 }

```

§7.7. Locating objects.—These two simple macros are used to find where a given object should be plotted, which is a frequent operation in the ready-made wheels.

`\horo@getobjdpos` Gets the object's DPos and rotates it into chart coordinates.

```

1052 \def\horo@getobjdpos#1{%
1053 \dimen@\csname horo#1DPos\endcsname\p@
1054 \horo@chartrotate
1055 }

```

`\horo@getobjsdms` Gets the object's Pos (not DPos) and converts it into SDMS, which is what you want if you're going to print a label saying where the object is in the sky. Then also gets the DPos so you know where to print the label, which might be different from the actual sky location if the object has been adjusted.

```
1056 \def\horo@getobjsdms#1{%
1057   \expandafter\dimen@\csname horo#1Pos\endcsname\p@
1058   \horo@d@sdms
1059   \horo@getobjdpos{#1}%
1060 }
```

§7.8. Cusps, ticks, and sign keys.—These display some landmark kinds of things that people like to have in charts. Cusps are just lines radiating out from the centre at locations determined by the chart data. Ticks are similar radial lines that occur at regular intervals to create a sort of angular ruler around the circle, allowing visual measurement of angles. A sign key shows the locations of the Zodiac signs; it's simply the twelve sign symbols arranged regularly around a circle. In this subsection we also describe cusp labels, which show the degree, sign, and minute of a house cusp (or, conceivably, something else) written around the circle at a constant radius.

`\horo@pr` Helper for `\horoputradials`: draw a radial at the DPos of object #3.

```
1061 \def\horo@pr#1#2#3{%
1062   \horo@getobjdpos{#3}%
1063   \horoputradial{#1}{#2}{\TG@rem@pt\dimen@}%
1064 }
```

`\horoputradials` Iterate through a list of objects drawing radials for each. Normally you'd use this to draw the radial lines separating houses.

```
1065 \def\horoputradials#1#2#3{%
1066   \horoforeach{#1}{\horo@pr{#2}{#3}}%
1067 }
```

`\horo@pt` Helper for `\horoputticks`. Puts one tick, advances `\dimen@`, and then tail-recurses until we've gone around the whole circle.

```
1068 \def\horo@pt{%
1069   \ifdim\dimen@<360\p@
1070     {\horo@chartrotate
1071       \horoputradial{\horo@pta}{\horo@ptb}{\TG@rem@pt\dimen@}}%
1072     \advance\dimen@ by \horo@ptc\p@
1073     \expandafter\horo@pt
1074   \fi
1075 }
```

`\horoputticks` Repeats `\putradial{#1}{#2}` every #3 degrees around the circle to create a regular pattern. Implementation saves all the arguments into macros, sets `\dimen@` to zero, and calls `\horo@pt` to do the actual plotting.

```
1076 \def\horoputticks#1#2#3{%
1077   \begingroup
```

```

1078 \edef\horo@pta{#1}\edef\horo@ptb{#2}\edef\horo@ptc{#3}%
1079 \dimen@=\z@
1080 \horo@pt
1081 \endgroup
1082 }

```

`\horo@pa` Helper for `\horoputarrows`: draw an arrow, pointing outward, at the DPos of object #3.

```

1083 \def\horo@pa#1#2#3{%
1084 \horo@getobjdpos{#3}%
1085 \horoputararrowhead{#1}{\TG@rem@pt\dimen@}{#2}%
1086 }

```

`\horoputarrows` Iterate through a list of objects drawing arrows for each. Normally you'd use this to draw the arrowheads for angular house cusps.

```

1087 \def\horoputarrows#1#2#3{%
1088 \horoforeach{#1}{\horo@pa{#2}{#3}}%
1089 }

```

`\horo@psk` Helper for `\horoputsignkey`. This is another tail-recursive loop with `\count0` as the loop counter, going from 0 to 11.⁹ For each sign we calculate the θ coordinate (of the middle of the sign, the 15° mark) and print the appropriate Zodiac symbol at that angle and the chosen radius.

```

1090 \def\horo@psk{
1091 \ifnum\count0<12\relax
1092 \dimen@=30pt\relax
1093 \dimen@\count0\dimen@\relax
1094 \advance\dimen@ by 15pt\relax
1095 \horo@chartrotate
1096 \advance\count0 by 1%
1097 \horo@putpolar{\horo@radius}{\TG@rem@pt\dimen@}%
1098 {\Zodiac{\the\count0}}%
1099 \expandafter\horo@psk
1100 \fi
1101 }

```

`\horoputsignkey` Driver for plotting a sign key. It just saves the argument, sets the loop counter to zero, and invokes `\horo@psk`.

```

1102 \def\horoputsignkey#1{%
1103 \begingroup
1104 \def\horo@radius{#1}%
1105 \count0=0\relax
1106 \horo@psk
1107 \endgroup
1108 }

```

⁹Well, it's one louder, isn't it?

`\horoputcuslabel` Make a label for a house cusp or (astrological) angle. This is a preview of the techniques used in the object label code later on. The arguments are the radius, the angular spacing in degrees between the three elements of the label, and the object to label.

First we open a prophylactic group and get the object's location data, both its Pos in SDMS (which is what the label will *say*) and its DPos (which is *where* the label will appear).

```
1109 \def\horoputcuslabel#1#2#3{%
1110   \begingroup
1111   \horo@getobjsdms{#3}%
```

The sign part of the label is straightforwardly set at DPos and the specified radius.

```
1112   \horo@putpolar{#1}{\TG@rem@pt\dimen@}{\Zodiac{\horo@d@sign}}%
```

Determining reading direction: the label is intended to be read in a circular direction that points away from the 135° mark, which is at the upper left of the chart. That way it will come as close as possible to reading either left to right or top to bottom. That means if the label's centre location θ coordinate is less than 135° or more than 315°, then the reading direction will be turnwise and otherwise it will be widdershins. The `\horo@pcl@s` macro is set to T or W accordingly.

```
1113   \def\horo@pcl@s{W}%
1114   \ifdim\dimen@<135\p@\def\horo@pcl@s{T}\fi
1115   \ifdim\dimen@>315\p@\def\horo@pcl@s{T}\fi
```

Typeset the part of the label that goes #2 degrees turnwise of the sign symbol. If the reading direction is turnwise, then this will be the end of the label—the minutes part. Otherwise it will be the degrees part. This happens inside a group so that `\dimen@` will be restored to the θ of the sign symbol afterward.

```
1116   \begingroup
1117   \advance\dimen@ by -#2\p@\horo@fixdimen@
1118   \horo@putpolar{#1}{\TG@rem@pt\dimen@}{%
1119     \expandafter\if\horo@pcl@s T%
1120     \horo@d@min\horominutes%
1121     \else
1122     \horo@d@deg\horodegrees%
1123     \fi
1124   }%
1125   \endgroup
```

Typeset the part of the label that goes #2 degrees widdershins of the sign symbol. Essentially the same logic as the other part above, except that this time we don't need to save `\dimen@` because it won't be used again.

```
1126   \advance\dimen@ by #2\p@\horo@fixdimen@
1127   \horo@putpolar{#1}{\TG@rem@pt\dimen@}{%
1128     \expandafter\if\horo@pcl@s T%
1129     \horo@d@deg\horodegrees%
1130     \else
1131     \horo@d@min\horominutes%
1132     \fi
```

```

1133 }%
1134 \endgroup
1135 }

```

§7.9. Object labels.—In general, an object label is printed in a radial direction with up to six chunks (object symbol, sign symbol, degrees, minutes, seconds, retrograde symbol) at equally spaced radii; the θ of all of them is determined by the DPos, and the sequence in which they’re printed (which chunk is closest to the centre and which is furthest) is determined by θ to create a consistent reading direction. We offer a few variations to provide varying levels of control over the details.

`\horoputobjsymbol` Put the symbol for object #2 at its DPos and radius #1.

```

1136 \def\horoputobjsymbol#1#2{%
1137 \begingroup
1138 \horo@getobjdpos{#2}%
1139 \horo@putpolar{#1}{\TG@rem@pt\dimen@}{\csname #2Symbol\endcsname}%
1140 \endgroup
1141 }

```

`\horo@pol@i` Support for the smarter object-labelling macros. Opens a group, sets `\dimen1` to #1+#2, and gets the DPos and SDMS of Pos for object #3. Based on DPos, the macro `\horo@pol@s` is set to indicate reading direction: H for hubward or R for rimward to make the reading direction be as far from the 135° mark¹⁰ as possible.

```

1142 \def\horo@pol@i#1#2#3{%
1143 \begingroup
1144 \dimen1=#1\p@\advance\dimen1by#2\p@
1145 \horo@getobjsdms{#3}%
1146 \def\horo@pol@s{H}%
1147 \ifdim\dimen@<45\p@\def\horo@pol@s{R}\fi
1148 \ifdim\dimen@>225\p@\def\horo@pol@s{R}\fi
1149 }

```

`\horoputobjdeglabel` Typeset a label consisting of the object’s degree and sign. Invokes `\horo@pol@i`, and then typesets the sign symbol and degree in an order determined by reading direction so that degree will read first. Arguments are radius of the innermost chunk, radius step size between the two chunks, and object name.

```

1150 \def\horoputobjdeglabel#1#2#3{%
1151 \horo@pol@i{#1}{#2}{#3}%
1152 \expandafter\if\horo@pol@s H%
1153 \horo@putpolar{#1}{\TG@rem@pt\dimen@}{\Zodiac{\horo@d@sign}}%
1154 \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen@}%
1155 {\horo@d@deg\horodegrees}%
1156 \else
1157 \horo@putpolar{#1}{\TG@rem@pt\dimen@}{\horo@d@deg\horodegrees}%
1158 \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen@}%
1159 {\Zodiac{\horo@d@sign}}%

```

¹⁰Great A’Tuin’s left rear flipper.

```

1160 \fi
1161 \endgroup
1162 }

```

`\horoputobjminlabel` Much the same as `\horoputobjdegreelabel` but makes a three-chunk label that reads degrees, sign, minutes.

```

1163 \def\horoputobjminlabel#1#2#3{%
1164 \horo@pol@i{#1}{#2}{#3}%
1165 \expandafter\if\horo@pol@s H%
1166 \horo@putpolar{#1}{\TG@rem@pt\dimen0}%
1167 {\horo@d@min\horominutes}%
1168 \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen0}%
1169 {\Zodiac{\horo@d@sign}}%
1170 \advance\dimen1by#2\p@
1171 \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen0}%
1172 {\horo@d@deg\horodegrees}%
1173 \else
1174 \horo@putpolar{#1}{\TG@rem@pt\dimen0}%
1175 {\horo@d@deg\horodegrees}%
1176 \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen0}%
1177 {\Zodiac{\horo@d@sign}}%
1178 \advance\dimen1by#2\p@
1179 \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen0}%
1180 {\horo@d@min\horominutes}%
1181 \fi
1182 \endgroup
1183 }

```

`\horoputobjseclabel` Typesets a four-chunk label reading degrees, sign, minutes, seconds.

```

1184 \def\horoputobjseclabel#1#2#3{%
1185 \horo@pol@i{#1}{#2}{#3}%
1186 \expandafter\if\horo@pol@s H%
1187 \horo@putpolar{#1}{\TG@rem@pt\dimen0}%
1188 {\horo@d@sec\horoseconds}%
1189 \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen0}%
1190 {\horo@d@min\horominutes}%
1191 \advance\dimen1by#2\p@
1192 \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen0}%
1193 {\Zodiac{\horo@d@sign}}%
1194 \advance\dimen1by#2\p@
1195 \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen0}%
1196 {\horo@d@deg\horodegrees}%
1197 \else
1198 \horo@putpolar{#1}{\TG@rem@pt\dimen0}%
1199 {\horo@d@deg\horodegrees}%
1200 \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen0}%
1201 {\Zodiac{\horo@d@sign}}%
1202 \advance\dimen1by#2\p@
1203 \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen0}%
1204 {\horo@d@min\horominutes}%

```

```

1205     \advance\dimen1by#2\p@
1206     \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen@}%
1207         {\horo@d@sec\horoseconds}%
1208     \fi
1209 \endgroup
1210 }

```

`\horo@putrxlabel` Checks the Vel of object #2. If it is negative, typesets `\horoRetrogradeSymbol` at radius #1 to indicate “retrograde.”

```

1211 \def\horo@putrxlabel#1#2{%
1212   \begingroup
1213     \dimen@\csname horo#2Vel\endcsname\p@
1214     \ifdim\dimen@<\z@\relax
1215       \horo@getobjdpos{#2}%
1216       \horo@putpolar{#1}{\TG@rem@pt\dimen@}{\horoRetrogradeSymbol}%
1217     \fi
1218   \endgroup
1219 }

```

`\horo@putsmartlabel` This is the do-everything label macro. It takes one argument which is the object name, but it’s sensitive to the definitions of several other macros, most notably `\horo@lblone` and `\horo@lbltwo`. Each of those is a string consisting of some combination of the letters (d, m, s, z, y, r) for Degrees, Minutes, Seconds, Zodiac sign symbol, object sYmbol, and possible Retrograde, respectively. They represent the sequence of chunks to typeset reading in an *inward* direction, for cases where the preferred reading direction is inward for `\horo@lblone` or outward for `\horo@lbltwo`. The reason to do it that way is that one typically wants the order of some chunks to change with reading direction but not others. A typical setting would be `\horo@lblone` equal to “ydzmr” and `\horo@lbltwo` equal to “ymzdr”. Then the degrees, sign, and minute will always read in that order as closely as possible to top to bottom and left to right, but the object symbol will always be on the outside and the optional retrograde symbol always on the inside. Proper handling and configuration of reading direction is the major complication in this macro and responsible for much of the complexity of its support macros.

The implementation at this high level is fairly simple: it wraps everything in a group to prevent pollution, gets the DPos and SDMS Pos of the object, and figures out the reading direction into `\horo@pol@s`. The `\dimen1` register is set to `\horo@outerrad`, which is the radius at which we’ll place the outermost chunk. Then depending on the reading direction it calls `\horo@ps1` with either `\horo@lblone` or `\horo@lbltwo` and a \mathcal{Q} (which is still exotic, remember) to aid in parsing.

```

1220 \def\horo@putsmartlabel#1{%
1221   \begingroup
1222     \horo@getobjsdms{#1}%
1223     \def\horo@ps1@o{#1}%
1224     \def\horo@pol@s{H}%
1225     \ifdim\dimen@<45\p@\def\horo@pol@s{R}\fi
1226     \ifdim\dimen@>225\p@\def\horo@pol@s{R}\fi

```

```

1227 \dimen1=\horo@outerrad\p@\relax
1228 \expandafter\if\horo@pol@s H%
1229 \expandafter\horo@psl\horo@lblone Q%
1230 \else
1231 \expandafter\horo@psl\horo@lbltwo Q%
1232 \fi
1233 \endgroup
1234 }

```

`\horo@psl` Main chunk-putting code for `\horoputsmartlabel`. This is a tail-recursive loop that iterates through the format string typesetting one chunk per letter until it hits the terminating `Q`. For each chunk it calls `\horo@psl@` to actually typeset the chunk, and subtracts `\horo@delta` from `\dimen1`, which is where we're storing the current radius, initially `\horo@outerrad`.

```

1235 \def\horo@psl#1{%
1236 \def\horo@psl@a{#1}%
1237 \ifx\horo@psl@a\horo@cue\relax
1238 \else
1239 \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen@}%
1240 {\horo@psl@#1}%
1241 \advance\dimen1by-\horo@delta\p@\relax
1242 \expandafter\horo@psl
1243 \fi
1244 }

```

`\horo@psl@` Typeset a single chunk. This is simply a bunch of `\ifs` that identify the chunk type and typeset the appropriate data.

```

1245 \def\horo@psl@#1{%
1246 \if#1d\relax\horo@d@deg\horodegrees\fi
1247 \if#1m\relax\horo@d@min\horominutes\fi
1248 \if#1s\relax\horo@d@sec\horoseconds\fi
1249 \if#1z\relax\Zodiac{\horo@d@sign}\fi
1250 \if#1y\relax\csname \horo@psl@o Symbol\endcsname\fi
1251 \if#1r\relax
1252 \beginngroup
1253 \dimen@\csname \horo@psl@o Vel\endcsname\p@
1254 \ifdim\dimen@<\z@\relax\horoRetrogradeSymbol\fi
1255 \endgroup
1256 \fi
1257 }

```

`\horoscanlabels` User interface to set up `\horo@lblone` and `\horo@lbltwo`. The argument may be a format string in the format described above, or *two* of them separated by a slash. If one, then both formats are set to it. If two, then the first goes into `\horo@lblone` and the second into `\horo@lbltwo`. The implementation works by calling the helper `\horo@scanlabels@` with two copies of `#1` and some terminating stuff so that it can read the first two slash-terminated arguments and get either the user's two strings, or the user's one string repeated twice.

```

1258 \def\horoscanlabels#1{%
1259   \horo@scanlabels@ #1/#1/xQ%
1260 }

```

`\horo@scanlabels@` Helper for `\horoscanlabels`. Looks for two slash-terminated arguments and then throws out everything up to the `Q`. The arguments go into `\horo@lblone` and `\horo@lbltwo`, but we also take the opportunity to scan `#1` for length, using the helper macro `\horo@scanlabels@@` which searches for a `Q` bumping the `\count0` register for every letter it sees. The result of that scan goes into `\horoblilen`, and may be used to automatically adjust radius step and text size.

```

1261 \def\horo@scanlabels@#1/#2/#3Q{%
1262   \def\horo@lblone{#1}%
1263   \def\horo@lbltwo{#2}%
1264   \begingroup
1265     \count0=0\relax
1266     \horo@scanlabels@@#1Q%
1267     \edef\horoblilen{\the\count0}%
1268     \horo@fterdef\horoblilen
1269   \endgroup
1270 }

```

`\horo@scanlabels@@` Helper macro for `\horo@scanlabels@`: tail-recursive loop to count letters in a `Q`-terminated string.

```

1271 \def\horo@scanlabels@@#1{%
1272   \def\horo@sils@@a{#1}%
1273   \ifx\horo@sils@@a\horo@cue\relax
1274   \else
1275     \advance\count0by1\relax
1276     \expandafter\horo@scanlabels@@
1277   \fi
1278 }

```

`\horosetsmartradii` Sets up the radii for `\horoputsmartradii` in a semi-intelligent way, sensitive to the label format string length from the previous macro. The outer radius is set to `#1`. The step size (distance between chunks) is set to `#2 - n · #3` where `n` is the number of chunks specified in the (inward-reading) label format string. The idea is that with more chunks, you want them to be closer together. However, simply allocating a fixed amount of space and dividing by the number of chunks is suboptimal because with few chunks they end up too far apart; it's better, if the user selects fewer chunks than the chart was originally designed for, to spread them out a little but also leave substantial extra space on the inside side. A more rigorous solution might involve using TeX's fancy variable-stretchability glue to create something that could be thought of as a "radial list" comparable to horizontal and vertical lists.

```

1279 \def\horosetsmartradii#1#2#3{
1280   \def\horo@outerrad{#1}%
1281   \begingroup
1282     \dimen@=#3\p@\relax

```

```

1283 \multiply\dimen@ by -\horoblilen\relax
1284 \advance\dimen@ by #2\p@\relax
1285 \edef\horo@delta{\TG@rem@pt\dimen@}%
1286 \horo@fterdef\horo@delta
1287 \endgroup
1288 }

```

8. ASPECT WEB

These routines typeset a web of lines that indicate aspects, with symbols on their midpoints to clarify which aspects are which. The aspects are automatically recognized, with some limited support for orbs and such, but it's up to the user to do it manually if they want better control over exactly which aspects will be shown.

At this point in the code we no longer need to use exotic Q for parsing, and we're going to need regular Q for use in spelling things like "Quincunx," so we have to make Q mundane again. We drop temporarily out of the `\ifhoro@wheels` conditional to do it so that things will not be left in a screwed-up state in the event `nowheels` has been selected.

```

1289 \fi
1290 %% MAKING Q MUNDANE HERE!
1291 \catcode'\Q=11
1292 \ifhoro@wheels

```

§8.1. Configuration settings.—

- `\horoaspectobjectsa` List of objects that can appear on one side of an aspect relation. By default, all objects.
- ```
1293 \def\horoaspectobjectsa{\horoobjects}
```
- `\horoaspectobjectsb` List of objects that can appear on the other side of an aspect relation. This is separated from `\horoaspectobjectsa` so that you can set one to all objects and the other to only "major" objects, to prevent counting aspects that involve only "minor" objects.
- ```
1294 \def\horoaspectobjectsb{\horoobjects}
```
- `\horoaspects` This lists the aspect types that will be recognized. Note conjunctions are not normally listed because they don't need to be marked in the aspect web, though if one wanted to add conjunction symbols on top of the ticks to really flag conjunctions, then that would be possible.
- ```
1295 \def\horoaspects{Opposition,Trine,Square,Sextile}
```
- `\ifhoroaspectweb` Configuration flag for the ready-made wheels: whether they should or shouldn't put an aspect web in the middle of the wheel.
- ```
1296 \newif\ifhoroaspectweb\horoaspectwebtrue
```

§8.2. Per-aspect-type configuration.—First we state the angular separation that defines each aspect. Only longitude aspects are supported; parallel/antiparallel aren't shown.

```

1297 \def\horoConjunctionAngle{0}
1298 \def\horoOppositionAngle{180}
1299 \def\horoTrineAngle{120}
1300 \def\horoSquareAngle{90}
1301 \def\horoQuintileAngle{72}
1302 \def\horoBiquintileAngle{144}
1303 \def\horoSextileAngle{60}
1304 \def\horoQuincunxAngle{150}
1305 \def\horoSemisextileAngle{30}
1306 \def\horoSemisquareAngle{45}
1307 \def\horoSesquiquadrateAngle{135}

```

Then the orb for each aspect. This package supports only a very simple model for orb, in which it's determined solely by the aspect type. Advanced users can do clever things with redrawing a couple of aspect webs for different objects, to get orbs depending on the objects involved or on other variables.

```

1308 \def\horoConjunctionOrb{6}
1309 \def\horoOppositionOrb{6}
1310 \def\horoTrineOrb{5}
1311 \def\horoSquareOrb{5}
1312 \def\horoQuintileOrb{2}
1313 \def\horoBiquintileOrb{2}
1314 \def\horoSextileOrb{4}
1315 \def\horoQuincunxOrb{3}
1316 \def\horoSemisextileOrb{3}
1317 \def\horoSemisquareOrb{2}
1318 \def\horoSesquiquadrateOrb{2}

```

§8.3. Drawing the aspect web.—

`\horoputaspect` Draw a single aspect. The arguments are #1 radius of the endpoints, #2 and #3 the θ coordinates of the endpoints, and #4 the symbol to display at the midpoint of the aspect. Implementation starts by opening a group and saving the arguments to macros.

```

1319 % radius theta1 theta2 symbol
1320 \def\horoputaspect#1#2#3#4{%
1321   \begingroup
1322     \edef\p@one{#1}\edef\p@two{#2}\edef\p@three{#3}%

```

Apply chart rotation to both θ coordinates:

```

1323   \dimen@=\p@two\p@\relax\horo@chartrotate\edef\p@two{\TG@rem@pt\dimen@}%
1324   \dimen@=\p@three\p@\relax\horo@chartrotate\edef\p@three{\TG@rem@pt\dimen@}%

```

Convert endpoints to polar and draw the line segment representing the aspect:

```

1325   \horo@polarconvert\p@one\p@two\horo@dim@xa\horo@dim@ya
1326   \horo@polarconvert\p@one\p@three\horo@dim@xb\horo@dim@yb

```

```

1327 \Line(\TG@rem@pt\horo@dim@xa,\TG@rem@pt\horo@dim@ya)%
1328 (\TG@rem@pt\horo@dim@xb,\TG@rem@pt\horo@dim@yb)%
      Compute the midpoint by doing an average on the Cartesian coordinates:
1329 \advance\horo@dim@xa by\horo@dim@xb
1330 \advance\horo@dim@ya by\horo@dim@yb
1331 \divide\horo@dim@xa by2\divide\horo@dim@ya by2%
      Put the symbol at the midpoint and end.
1332 \put(\TG@rem@pt\horo@dim@xa,\TG@rem@pt\horo@dim@ya){\makebox(0,0){#4}}%
1333 \endgroup
1334 }

```

`\horo@aa` Helper for more automated aspect drawing: this checks whether there is an aspect (with angular separation of $\#1 \pm \#2$) between objects $\#5$ and $\#6$, and if so, draws an aspect between them with endpoint radius $\#3$ and symbol $\#4$. The two object names come last because this is meant to be used in a `\horo@dblforeach`.

```

1335 \def\horo@aa#1#2#3#4#5#6{%
1336 \dimen@ \csname horo#5Pos\endcsname \p@
1337 \advance\dimen@ by -\csname horo#6Pos\endcsname \p@
1338 \horo@fixdimen@diff
1339 \ifdim\dimen@ < \z@ \relax \multiply\dimen@ by -1 \relax \fi
1340 \advance\dimen@ by -#1 \p@
1341 \horo@fixdimen@diff
1342 \ifdim\dimen@ < \z@ \relax \multiply\dimen@ by -1 \relax \fi
1343 \ifdim\dimen@ < #2 \p@ \relax
1344 \horoputaspect{#3}{\csname horo#5Pos\endcsname}%
1345 \csname horo#6Pos\endcsname}{#4}%
1346 \fi
1347 }

```

`\horoautoaspect` Wrapper for `\horo@aa`: runs the `\horo@dblforeach` inside a group. This does all the checking and drawing for one aspect type.

```

1348 \def\horoautoaspect#1#2#3#4#5#6{%
1349 \begingroup
1350 \horo@dblforeach{#1}{#2}{\horo@aa{#3}{#4}{#5}{#6}}%
1351 \endgroup
1352 }

```

`\horo@aa` Helper for `\horoautoaspects`: calls `\horoautoaspect` to do one aspect type, using configured settings. Argument $\#1$ is radius and $\#2$ is the aspect type name. Note that this does every pair of one object from `\horoaspectobjectsa` and one from `\horoaspectobjectsb`, which will quite possibly hit a given pair twice, once in each direction (e.g. Sun/Moon and Moon/Sun) Normally that will not be a problem; it only means typesetting an identical aspect twice, one on top of the other.

```

1353 \def\horo@aa#1#2{%
1354 \horoautoaspect{\horoaspectobjectsa}{\horoaspectobjectsb

```

```

1356   {#1}{\csname horo#2Symbol\endcsname}%
1357 }

```

`\horoautoaspects` Do the entire aspect web by iterating over all configured aspect type names. Most configuration settings come in through defined macros; the sole argument is the radius.

```

1358 \def\horoautoaspects#1{%
1359   \horoforeach{\horoaspects}{\horo@as{#1}}%
1360 }

```

9. INTERNAL HOUSE LABELS

These support adding labels to the middles of houses. The labels are treated (for the purposes of the adjustment code) as objects that take up space in the house, so they make it much more likely for a house to end up overfilled and having to expand. However, they also make it easier to identify houses at a glance.

Much of the support for these is actually incorporated into the “adjustment” code below. At this point we only define configuration data and the routines to actually print the labels.

`\ifhorointhouselabels` Define an `\if` for controlling whether to use this feature.

```

1361 \newif\ifhorointhouselabels\horointhouselabelsfalse

```

Allocate a \LaTeX counter for which house we’re in.

```

1362 \newcounter{horohouse}

```

`\horohouselabel` The default form of a label is the counter value in upper-case Roman, but this can be redefined if the user wants something else.

```

1363 \newcommand{\horohouselabel}{%
1364   \Roman{horohouse}%
1365 }

```

`\horo@houses` List of the pseudo-objects used for internal house labels. This is the only tricky bit in this section of code: the internal house labels are treated as objects with names like “CuspIM.” Then we have macros with names including strings like “CuspIMPos,” and that can be treated as either the Pos of CuspIM, or the MPos of CuspI. Exploiting that ambiguity allows for simpler use of the existing variable-copying code to do some useful things with label positions.

```

1366 \def\horo@houses{CuspIM,CuspIIM,CuspIIIM,CuspIVM,CuspVM,CuspVIM,%
1367   CuspVIIM,CuspVIIIM,CuspIXM,CuspXM,CuspXIM,CuspXIIM}

```

`\horo@findcuspamid` Finds the midpoint of a house, given (as arguments) the names of the house’s cusp and of the next house’s cusp. Looks at the DPos of those two cusps, puts the result into the MPos for the house (see above about the Pos/MPos trick).

```

1368 \def\horo@findcuspamid#1#2{%
1369   \horo@midpoint{\csname horoCusp#1DPos\endcsname}%
1370   {\csname horoCusp#2DPos\endcsname}%
1371   \expandafter\edef\csname horoCusp#1MPos\endcsname{\TG@rem@pt\dimen@}%
1372 }

```

`\horo@findcuspmids` Repeat `\horo@findcuspmid` for all twelve houses. This is done explicitly because of the wrap-around between XII and I, which makes it non-trivial to write a nice looping structure to do it.

```
1373 \def\horo@findcuspmids{%
1374 \horo@findcuspmid{I}{II}\horo@findcuspmid{II}{III}%
1375 \horo@findcuspmid{III}{IV}\horo@findcuspmid{IV}{V}%
1376 \horo@findcuspmid{V}{VI}\horo@findcuspmid{VI}{VII}%
1377 \horo@findcuspmid{VII}{VIII}\horo@findcuspmid{VIII}{IX}%
1378 \horo@findcuspmid{IX}{X}\horo@findcuspmid{X}{XI}%
1379 \horo@findcuspmid{XI}{XII}\horo@findcuspmid{XII}{I}%
1380 }
```

`\horo@pihl` Prints an internal house label given the radius at which to print as #1 and the name of the midpoint (like CuspIM) as #2. Also steps the `horohouse` counter, which is probably being used to decide how the labels look.

```
1381 \def\horo@pihl#1#2{%
1382 \begingroup
1383 \horo@getobjdpos{#2}%
1384 \horo@putpolar{#1}{\TG@rem@pt\dimen@}{\horohouselabel}%
1385 \stepcounter{horohouse}%
1386 \endgroup
1387 }
```

`\horoputinhouselabels` Typesets a complete ring of twelve house labels. The sole argument is the radius at which to put them.

```
1388 \def\horoputinhouselabels#1{%
1389 \setcounter{horohouse}{1}%
1390 \horo@foreach{\horo@houses}{\horo@pihl{#1}}%
1391 }
```

10. OBJECT AND CUSP ADJUSTMENT

These macros implement a full-featured system for adjusting the location of objects on the chart to prevent them from interfering with each other. The basic concept involves two variables on each object and cusp: Pos and DPos. Pos represents the true position of the object in the sky, normally its longitude. That is also in some sense the “preferred” position for the object or cusp on the chart. However, if necessary to avoid interference, the object or cusp can be shifted to a new location. DPos is the position where it will actually be displayed.

Adjustment is done by an iterative spring-tensioning algorithm. Everything starts out with DPos equal to Pos. Then in successive loops, we examine whether any rules (such as “no two objects within so many degrees of each other”) are broken, and if so we shift things in such a way as to reduce the rule violations. Things that are too close to each other are mutually repelled away from each other. Things that are not in their preferred locations are attracted towards their preferred locations.

There are limits on how much these forces can affect the location of an object, to prevent the system from running away or oscillating. We hope (without formal

proof, but it seems to be true in practice and in principle it could probably be proved formally) that in successive iterations the adjustments will get smaller and smaller until the system settles down into a reasonably good solution. There are a few tricks implemented to encourage that. If two successive iterations give results that are almost the same (to the point that the difference would not be visible to the viewer) then we figure it has converged and stop the loop. There is also a limit on the absolute number of iterations that will be allowed, in case it does run away.

Ideally we would have a solution that doesn't involve moving any cusps, because moved cusps end up having unappealing "jogs" when rendered. That may or may not be possible. The system will first attempt a solution without moving any cusps, but if it converges with no cusp movement and the result isn't good enough, then it will unlock the cusps and continue iterating.

§10.1. Configuration settings.—

`\horosignificantadj` Number of degrees of adjustment that will be considered "significant" and cause another iteration. This should be small enough that it won't be visible in the output.

```
1392 \def \horosignificantadj{0.1}
```

`\horocuspadjusttrigger` This represents the percentage of the normal "minimum" distance between objects and objects, or objects and cusps, below which houses will be considered too cramped. If the converged solution without moving cusps results in any distances becoming shorter than this percentage, which will normally imply all the distances in that house are similarly cramped, then cusp adjustment will be triggered.

```
1393 \def \horocuspadjusttrigger{65}
```

`\horoadjcycles` Maximum number of cycles (total) to permit. If convergence is not detected after this many, then it'll just go ahead with the result of the last one. The default of 30 is generous.

```
1394 \def \horoadjcycles{30}
```

`\horoposattobj` Strength of attraction between an object and its desired Pos value. This is given as a divisor, so the default of 20 means that the adjustment to bring an object towards its Pos will be equal to 1/20 of the amount by which it's currently displaced from there.

```
1395 \def \horoposattobj{20}
```

`\horoposattcusp` Strength of attraction between a cusp and its desired Pos value. The default of 7 means cusps are attracted to their Pos locations about three times as much as objects; but note that that's if we allow cusps to move at all, which only happens if we're forced into it.

```
1396 \def \horoposattcusp{7}
```

`\hororepulsion` Basic divisor for repulsion between all pairs of things that are repelled from each other.

```
1397 \def\hororepulsion{3}
```

`\horomaxrepulsion` Interference distance (in degrees) at which maximum repulsion is achieved. If two objects interfere by more than this amount, they will be repelled only as much as if they interfered this much.¹¹ The case this is intended to cover is the one where an object somehow happens to be completely on the other side of the chart from where it should be, and out of order with everything else. If repulsion were unlimited, then everything would be kicked around to the point of creating many more constraint violations and the whole thing would take many iterations to resolve itself. Limiting repulsion per pair of objects gives more chance to resolve such situations reasonably.

```
1398 \def\horomaxrepulsion{5.0}
```

`\horoomindist` Object/object minimum distance. If objects are more than this distance apart (and in the correct sequence with each other) then they will not repel. When they're closer, including being on the wrong side of each other, then repulsion increases linearly with the amount of interference up to the maximum set by the previous macro. Note that this one is for *object/object* relationships only.

```
1399 \def\horoomindist{6.0}
```

`\horoo mindist` Object/cusp minimum distance.

```
1400 \def\horoo mindist{4.0}
```

§10.2. Internal variables.—These are used to keep track of where we are in the iteration and whether we want to continue. Note dropping temporarily out of `\ifhoro@wheels` because doing `\newif` in a conditional context causes parsing problems.

```
1401 \fi
```

```
1402 \newif\ifhoro@djusted\relax
```

```
1403 \newif\ifhoro@djustcusps\relax
```

```
1404 \ifhoro@wheels
```

```
1405 \countdef\horo@i=1\relax
```

§10.3. Support macros for adjustment iterations.—These do low-level tasks needed within a single cycle of adjusting objects or cusps.

¹¹Don't try this at home, kids. Sharp cutoffs like this one tend to introduce nonlinearity into the underlying system of partial differential equations, which in turn tends to create catastrophic instability. The famous Tacoma Narrows Bridge disaster is blamed on a similar effect: due to nonlinearity in the actual behaviour of the system, the linear analysis performed by the designers of the bridge was woefully inaccurate. The decision to use sharp cutoffs in the particular case of horoscop's object adjustment system is supported by very careful testing, the fact that the discontinuities are only in the second and higher derivatives, and the basic harmlessness of the situation: in the worst imaginable failure mode you'd just get a visually unappealing astrological chart. It should not be taken as an endorsement of sharp-cutoff designs in general.

`\horo@findpdiff` Find the distance in Pos between two objects or cusps named by the arguments. Note we can also get the distance in DPos by tacking a D onto the end of an object name. Result goes into `\dimen@`.

```
1406 \def\horo@findpdiff#1#2{%
1407   \dimen@=\csname horo#1Pos\endcsname\p@
1408   \multiply\dimen@ by -1\relax
1409   \advance\dimen@ by\csname horo#2Pos\endcsname\p@
1410   \horo@fixdimen@diff
1411 }
```

`\horo@plycorrection` Apply a correction (which should be in `\dimen@`) to the DPos of the specified object or cusp.

```
1412 \def\horo@plycorrection#1{%
1413   \advance\dimen@ by \csname horo#1DPos\endcsname\p@
1414   \horo@fixdimen@
1415   \expandafter\edef\csname horo#1DPos\endcsname{\TG@rem@pt\dimen@}%
1416 }
```

`\horo@djo@` Adjust an object or cusp against another. Arguments are #1 the minimum distance, #2 the thing to adjust, and #3 the thing to adjust it against. Implementation starts by finding, and saving, the distance in Pos between the two objects or cusps, which gives us a sanity check on whether they should affect each other at all, and tells us on which side of each other they should appear.

```
1417 \def\horo@djo@#1#2#3{%
1418   \horo@findpdiff{#3}{#2}%
1419   \horo@dim@x=\dimen@
```

If the distance in Pos is more than 45° in either direction, then we skip the rest of this. That prevents some bad nonconverging cases if objects happen to get into very bad locations on the wrong side of the chart; they'll only interact with the objects that they want to be near.

```
1420   \ifdim\horo@dim@x<45\p@\relax
1421     \ifdim\horo@dim@x>-45\p@\relax
```

Now find the difference in *display* positions as opposed to sky positions, and flip its sign depending on the saved Pos difference. The result in `\dimen@` is a number that tells us how far apart the objects are in the direction they're supposed to be, so it is negative if they are on the wrong side of each other.

```
1422     \horo@findpdiff{#3D}{#2D}%
1423     \ifdim\horo@dim@x<\z@\multiply\dimen@ by -1\relax\fi
```

Subtract out the minimum distance. The result is positive if the objects are on the correct side and separated by more than the minimum distance—in which case we set the adjustment to zero. Otherwise it's a negative number saying how far they are interfering. We hard-limit it by `\horomaxrepulsion` on the other side, and then divide by `\hororepulsion` to get the adjustment for the current cycle.

```
1424     \advance\dimen@ by -#1\p@
1425     \ifdim\dimen@>\z@
1426       \dimen@=\z@
```

```

1427 \else
1428 \ifdim\dimen@<-\horomaxrepulsion\p@\relax
1429 \dimen@=-\horomaxrepulsion\p@%
1430 \fi
1431 \divide\dimen@ by\hororepulsion\relax
1432 \fi

```

If we flipped the sign earlier, flip it back. Then `\dimen@` will contain the actual adjustment to apply to `\horo<#2>\DPos`.

```

1433 \ifdim\horo@dim@x>\z@\multiply\dimen@ by -1\relax\fi

```

Apply the correction and end.

```

1434 \horo@pplycorrection{#2}%
1435 \fi
1436 \fi
1437 }

```

`\horo@djoo` Adjust an object against an object. This is just the small amount of additional intelligence needed on top of `\horo@djo@` to handle objects versus objects. We check the arguments to make sure we aren't adjusting the same object against itself, because that is handled separately later. Also, the value of `\horoomindist` is picked up and passed into `\horo@djo@`.

```

1438 \def\horo@djoo#1#2{%
1439 \edef\horo@tmpa{#1}\edef\horo@tmpb{#2}%
1440 \ifx\horo@tmpa\horo@tmpb\else
1441 \horo@djo@{\horoomindist}{#1}{#2}%
1442 \fi
1443 }

```

`\horo@djoo@same` The special case of object or cusp against itself: we find the difference between its Pos and DPos, divide that by #1 (which should be the spring divisor, different for objects or cusps), and apply the result as a correction.

```

1444 \def\horo@djoo@same#1#2{%
1445 \horo@findpdiff{#2D}{#2}%
1446 \divide\dimen@ by#1\relax
1447 \horo@pplycorrection{#2}%
1448 }

```

`\horo@djcheckmovement` Check for whether an object or cusp has moved significantly in the current cycle. That is, if DPos differs from SPos (which was set to DPos before the current cycle) by more than `\horosignificantadj`, then set `\horo@djustedtrue`.

```

1449 \def\horo@djcheckmovement#1{%
1450 \horo@findpdiff{#1D}{#1S}%
1451 \ifdim\dimen@<0\p@\relax\multiply\dimen@ by -1\relax\fi
1452 \ifdim\dimen@>\horosignificantadj\p@\relax\horo@djustedtrue\fi
1453 }

```

§10.4. Single adjustment cycles.—These each do one complete cycle of adjusting objects or objects and cusps. There are two versions because including cusps or not makes a big enough difference to warrant a separate implementation.

`\horo@djobjcycle` One cycle of adjusting objects against each other and against cusps. Start by showing the user the current cycle number as a progress indicator, and setting the `\ifhoro@djusted` flag to false because we haven't moved anything yet this cycle.

```
1454 \def\horo@djobjcycle{%
1455   \message{\the\horo@i}%
1456   \horo@djustedfalse
```

Copy all the current DPos values to a new variable called SPos (“saved position”) so we’ll be able to check whether anything has moved.

```
1457 \horocopyvar{\horoobjects}{DPos}{SPos}%
```

Adjust objects against objects.

```
1458 \horo@dblforeach{\horoobjects}{\horoobjects}{\horo@djo}%
```

Adjust objects against cusps. This is one-sided: objects move to accommodate cusps, but cusps do not move to accommodate objects. Since (we assume) nothing is both an object and a cusp, we can call the low-level `\horo@djo@` directly instead of needing a wrapper like `\horo@adjoo`.

```
1459 \horo@dblforeach{\horoobjects}{\horocusps}{\horo@djo@\horoo mindist}}%
```

Adjust each object against itself; that is, handle its attraction to Pos, its “correct” location in the sky.

```
1460 \horoforeach{\horoobjects}{\horo@djo@same{\horoposattobj}}%
```

Check for movement. This turns on the `\ifhoro@djusted` flag if the net result of the current cycle has been to move anything a “significant” amount from the position we saved in SPos.

```
1461 \horoforeach{\horoobjects}{\horo@djcheckmovement}%
```

If the flag has remained false, then we have convergence, and can blow out of the loop by setting the counter to its terminating value.

```
1462 \ifhoro@djusted\else\horo@i=\horoadjcycles\relax\fi
```

Loop test. If we aren’t on the last loop (which could have happened either naturally or because of the exception in the last line), then advance the loop counter and schedule another tail-recursive iteration.

```
1463 \ifnum\horoadjcycles>\horo@i\relax
1464   \advance\horo@i by1\relax
1465   \expandafter\horo@djobjcycle
1466 \fi
1467 }
```

`\horo@djcuspcycle` Adjust all objects and cusps against each other. As in `\horo@djobjcycle`, we start by displaying the iteration number, setting the `\ifhoro@djusted` flag false, and saving all DPos values to SPos.

```
1468 \def\horo@djcuspcycle{%
1469   \message{\the\horo@i}%
1470   \horo@djustedfalse
1471   \horocopyvar{\horoobjects,\horocusps}{DPos}{SPos}%
```

Do the adjustments. We use `\horo@djo@` for adjusting cusps against objects and objects against cusps because nothing appears on both lists. For cusps against cusps and objects against objects we use `\horo@djoo` because it has the added handling for not adjusting anything against itself.

```
1472 \horo@dblforeach{\horocusps}{\horoobjects}{\horo@djo@{\horoocmindist}}%
1473 \horo@dblforeach{\horocusps}{\horocusps}{\horo@djoo}%
1474 \horo@dblforeach{\horoobjects}{\horoobjects}{\horo@djoo}%
1475 \horo@dblforeach{\horoobjects}{\horocusps}{\horo@djo@{\horoocmindist}}%
```

Adjust objects and cusps towards their desired positions (Pos values). Note that the logic is the same but the divisor passed in is `\horoposattobj` or `\horoposattcusp` depending on whether we are considering objects or cusps.

```
1476 \horoforeach{\horoobjects}{\horo@djoo@same{\horoposattobj}}%
1477 \horoforeach{\horocusps}{\horo@djoo@same{\horoposattcusp}}%
```

Check for movement and handle the tail-recursive loop, just like in `\horo@dj-objcycle` except that we examine cusps as well as objects.

```
1478 \horoforeach{\horoobjects,\horocusps}{\horo@djcheckmovement}%
1479 \ifhoro@djusted\else\horo@i=\horoadjcycles\relax\fi
1480 \ifnum\horoadjcycles>\horo@i\relax
1481   \advance\horo@i by1\relax
1482   \expandafter\horo@djcuspcycle
1483 \fi
1484 }
```

§10.5. Support macros for main loop.—

`\horo@djsavedpos` Use `\horo@fterdef` to pass the DPos of an object or cusp outside the prophylactic group.

```
1485 \def\horo@djsavedpos#1{%
1486   \expandafter\horo@fterdef\csname horo#1DPos\endcsname
1487 }
```

`\horo@djcheckcusps` This checks a pair of objects or cusps for excessive stress, to determine, after we have a converged solution with only objects moving, whether we need to try moving cusps too. It makes sure not to compare something against itself, finds the absolute value of the distance in degrees between the two things, and checks whether that is less than `\horocuspadjusttrigger` percentage of the minimum distance which was passed in through #1. If so, there's too much stress, and the `\ifhoro@djustcusps` flag gets turned on.

```
1488 \def\horo@djcheckcusps#1#2#3{%
1489   \edef\horo@tmpa{#2}\edef\horo@tmpb{#3}%
1490   \ifx\horo@tmpa\horo@tmpb
1491     \else
1492       \horo@findpdiff{#2D}{#3D}%
1493       \ifdim\dimen@<\z@\multiply\dimen@ by-1\relax\fi
1494       \ifdim\dimen@<45\p@\relax
1495         \multiply\dimen@ by 100\relax
1496         \divide\dimen@ by \horocuspadjusttrigger\relax
```

```

1497     \ifdim\dimen@<#1\p@\relax\horo@djustcusptrue\fi
1498     \fi
1499     \fi
1500 }

```

§10.6. Main loop.—

`\horoadjust` This is the user-callable adjustment macro that does the whole task of adjusting objects and possibly cusps to make them look good. The whole works is conditional on the `\ifhorocalculated` flag; if there's no good data in the initial Pos and DPos values, then the adjustment may not converge and will be wasted effort in any case. An earlier version actually used multiple passes to do the external program calls (much like the way BibTeX works), so `\horocalculatedfalse` was guaranteed to occur on the first pass; now, assuming `\write18` support in the interpreter, it's less probable, but still needs to be caught because people *will* attempt to run without the required `\write18` support.

```

1501 \def\horoadjust{%
1502   \ifhorocalculated

```

Support for internal house labels: if they're turned on, then find the house midpoints and add them as objects to `\horoobjects`, saving its old value to be later restored.

```

1503     \ifhoroinhouselabels
1504     \horo@findcuspmids
1505     \horocopyvar{\horocusps}{MPos}{MDPos}%
1506     \let\horo@savobjlist\horoobjects
1507     \edef\horoobjects{\horoobjects,\horo@houses}%
1508     \fi

```

Get ready to run the loop. Prints an opening parenthesis as a message to the user, to give some progress indication, and sets the loop counter to 1. A session of `\horoadjust` will give a message something like “(1 2 3 4 5 6 7 C 1 2 3)” showing each cycle through the loop, and the decision to adjust cusps. That way (since this requires a fair bit of processing) the user won't be left hanging, wondering what TeX is doing.

```

1509     \begingroup
1510     \message{ }%
1511     \horo@i=1\relax

```

Do the actual adjustment cycles for objects only.

```

1512     \horo@djobcycle

```

Now to decide whether we need to adjust cusps as well. Start by setting the flag `false`. Then check objects against objects, and objects against cusps, for excessive stress. If we find any, then cusp adjustment will be triggered. This assumes that cusps against cusps will never be an issue, but that's probably reasonable: cusps only fall right on top of each other when the house system is misbehaving (e.g. Placidus at high latitudes), and even then it will only be a visual problem if there are objects trapped in the resulting tiny houses, because the cusps themselves take up negligible angular space.

```

1513 \horo@djustcuspsfalse
1514 \horo@dblforeach{\horoobjects}{\horoobjects}%
1515     {\horo@djcheckcusps{\horoocmindist}}%
1516 \horo@dblforeach{\horoobjects}{\horocusps}%
1517     {\horo@djcheckcusps{\horoocmindist}}%

```

If we do want to adjust cusps: give the user a “C” to let them know, then set the loop counter back to 1 and do the cusp adjustment cycle.

```

1518 \ifhoro@djustcusps
1519     \message{C}%
1520     \horo@i=1\relax
1521     \horo@djcuspcycle
1522 \fi

```

At this point everything is converged, or as close as we were able to get. Print a terminating parenthesis for the user message and pass all the DPos values we calculated out of the prophylactic group.

```

1523     \message{)}%
1524 \horoforeach{\horocusps, \horoobjects}{\horo@djsavedpos}%
1525 \endgroup

```

Finally, restore the \horoobjects list if we tampered with it earlier.

```

1526 \ifhoro@inhouselabels\let\horoobjects\horo@savobjlist\fi
1527 \fi
1528 }

```

At this point we end the \ifhoro@wheels conditional.

```
1529 \fi
```

11. SUPPORT MACROS FOR READY-MADE WHEELS

These provide some low-level operations specific to the ready-made wheel templates.

At this point we open a conditional so that users can turn off this support if they won’t be using it.

```
1530 \ifhoro@templates
```

§11.1. Recognizing houses.—Some of the templates treat objects differently depending on in which house, or which kind of house, the objects are located. These macros are all designed to be called in the conditional part of an \if; they start out by expanding TT (which the \if recognizes as unconditional true), then open a new \if that does the actual conditioning.

`\horo@isclockwise` Conditional, true if #1 is clockwise of #2. Implementation finds the difference in Pos, inside a group to prevent pollution, and checks its sign.

```

1531 \def\horo@isclockwise#1#2{%
1532 TT\fi
1533 \begingroup
1534     \horo@findpdiff{#1}{#2}%
1535     \edef\horo@data{\the\dimen0}%

```

```

1536 \horo@fterdef\horo@data
1537 \endgroup
1538 \ifdim\horo@data>\z@\relax
1539 }

```

`\horo@anghouse` Conditional, true if #1 is in an angular house; that is, I, IV, VII, or X. Works by comparing the object's position against the two cusps bounding the house, for each of the four angular houses.

```

1540 \def\horo@anghouse#1{%
1541 TT\fi
1542 \begingroup
1543 \def\horo@angh@{F}%
1544 \if\horo@isclockwise{CuspI}{#1}\if\horo@isclockwise{#1}{CuspII}%
1545 \def\horo@angh@{T}%
1546 \fi\fi
1547 \if\horo@isclockwise{CuspIV}{#1}\if\horo@isclockwise{#1}{CuspV}%
1548 \def\horo@angh@{T}%
1549 \fi\fi
1550 \if\horo@isclockwise{CuspVII}{#1}\if\horo@isclockwise{#1}{CuspVIII}%
1551 \def\horo@angh@{T}%
1552 \fi\fi
1553 \if\horo@isclockwise{CuspX}{#1}\if\horo@isclockwise{#1}{CuspXI}%
1554 \def\horo@angh@{T}%
1555 \fi\fi
1556 \horo@fterdef\horo@angh@
1557 \endgroup
1558 \if\horo@angh@ T\relax
1559 }

```

`\horo@succhouse` Conditional, true if #1 is in a succedent house; that is, II, V, VIII, or XI. Implementation basically the same as `\horo@anghouse`.

```

1560 \def\horo@succhouse#1{%
1561 TT\fi
1562 \begingroup
1563 \def\horo@succh@{F}%
1564 \if\horo@isclockwise{CuspII}{#1}\if\horo@isclockwise{#1}{CuspIII}%
1565 \def\horo@succh@{T}%
1566 \fi\fi
1567 \if\horo@isclockwise{CuspV}{#1}\if\horo@isclockwise{#1}{CuspVI}%
1568 \def\horo@succh@{T}%
1569 \fi\fi
1570 \if\horo@isclockwise{CuspVIII}{#1}\if\horo@isclockwise{#1}{CuspIX}%
1571 \def\horo@succh@{T}%
1572 \fi\fi
1573 \if\horo@isclockwise{CuspXI}{#1}\if\horo@isclockwise{#1}{CuspXII}%
1574 \def\horo@succh@{T}%
1575 \fi\fi
1576 \horo@fterdef\horo@succh@
1577 \endgroup

```

```
1578 \if\horo@succh@ T\relax
1579 }
```

§11.2. Support for Montreal template.—

`\horo@montrealcurve` Draw one of the special curves that typify the Montreal chart template. This goes from the midpoint of a succedent house, based on DPos of the starting and ending cusps, to the Pos of an angular house’s cusp. There are a total of eight of them to be drawn. The inner and outer radii are fixed at 15 and 45. The implementation just calls `\horo@midpoint` to compute the inner end, and then `\horo@putcurve` to draw the curve. This is not one of the nice special cases for `\horo@putcurve` (constant r or constant θ) but the result looks good in context anyway.

```
1580 \def\horo@montrealcurve#1#2#3{%
1581 \horo@midpoint{\csname horoCusp#2DPos\endcsname}%
1582             {\csname horoCusp#3DPos\endcsname}%
1583 \horoputcurve{45}{\csname horoCusp#1Pos\endcsname}{15}{\TG@rem@pt\dimen@}%
1584 }
```

`\horo@putmontrealobj` Typeset an object label for the Montreal template. Objects in succedent houses get labels on a larger radius than those in angular or cadent houses.

```
1585 \def\horo@putmontrealobj#1{%
1586 \if\horo@succhouse{#1}%
1587 \horoputobjsymbol{42}{#1}%
1588 \horoputobjdeglabel{34}{4}{#1}%
1589 \horoputrxlabel{30}{#1}%
1590 \else
1591 \horoputobjsymbol{29}{#1}%
1592 \horoputobjdeglabel{21}{4}{#1}%
1593 \horoputrxlabel{17}{#1}%
1594 \fi
1595 }
```

§11.3. Support for Quebec City template.—

`\horo@putqcobj` Typeset an object label for the Quebec City template. Objects in angular houses get labels on a smaller radius than those in succedent or cadent houses.

```
1596 \def\horo@putqcobj#1{%
1597 \if\horo@anghouse{#1}%
1598 \horoputobjsymbol{28}{#1}%
1599 \horoputobjdeglabel{20}{4}{#1}%
1600 \horoputrxlabel{16}{#1}%
1601 \else
1602 \horoputobjsymbol{42}{#1}%
1603 \horoputobjdeglabel{34}{4}{#1}%
1604 \horoputrxlabel{30}{#1}%
1605 \fi
1606 }
```

`\horo@qcline` Draw a straight line as part of the Quebec City template. This goes from radius #1 and angle #2 to radius #3 and angle #4. The angles are specified as cusp number

concatenated with variable, as in “IIMPos,” because that’s the most convenient form for the calling macro.

```
1607 \def\horo@qcline#1#2#3#4{%
1608   \begingroup
1609     \dimen@=\csname horoCusp#2\endcsname\p@\horo@chartrotate
1610     \dimen1=\dimen@\relax
1611     \dimen@=\csname horoCusp#4\endcsname\p@\horo@chartrotate
1612     \horoputline{#1}{\TG@rem@pt\dimen1}%
1613             {#3}{\TG@rem@pt\dimen@}%
1614   \endgroup
1615 }
```

§11.4. Support for dial templates.—The dial templates all follow much the same pattern, so most of the parts that are the same regardless of the number of dials are factored into these two macros.

`\horo@dialstart` Start drawing a dial template. The first argument is the radius of the innermost dial, and the second is the harmonic—specially supported because people who use dial charts seem to also often want harmonics on them, and the degree scale should expand according to the harmonic.

Implementation starts by calculating the harmonic positions of the objects:

```
1616 \def\horo@dialstart#1#2{%
1617   \horocalcharmonic{#2}
```

To print a nice degree scale we want ticks of differing sizes every one, five, and ten degrees - but with the spacing adjusted according to the harmonic. So we open a group and compute #2 times one, five, and ten, and save the results in the macros `\horo@dtone`, `\horo@dtfive`, and `\horo@dtten` respectively.

```
1618   \begingroup
1619     \count0=#2\relax
1620     \edef\horo@dtone{\the\count0}%
1621     \multiply\count0 by 5\relax
1622     \edef\horo@dtfive{\the\count0}%
1623     \multiply\count0 by 2\relax
1624     \edef\horo@dtten{\the\count0}%
1625     \horo@fterdef\horo@dtone
1626     \horo@fterdef\horo@dtfive
1627     \horo@fterdef\horo@dtten
1628   \endgroup
```

Force the right coordinate to 270°, which means that the 0° mark will be at the top of the chart; that seems to be what dial-chart users want.

```
1629   \def\hororightcoord{270}%
```

If there’s to be an aspect web (which may not be standard usage for dial charts, but costs little to support) then typeset it. Otherwise, just make a little cross in the centre of the dial so that the Cosmobiologists know where to place their angle-finding instruments.

```
1630   \ifhoroaspectweb
```

```

1631   \horoautoaspects{#1}%
1632   \else
1633     \Line(-1,0)(1,0)%
1634     \Line(0,-1)(0,1)%
1635   \fi
1636 }

```

`\horo@dialwheel` Draw one wheel of a potentially multi-wheel dial chart. The argument #1 is the `\horoomindist` setting (because the outer wheels may be less cramped) and #2 the diameter (not radius) of the basic circle. Then #3 through #6 are the radii of various things: start of object-Pos ticks, start of connector curves between object Pos and object DPos, end of connector curves, and object symbols. The last argument, #7, is the length of object-Pos ticks.

Implementation starts by setting object-object minimum distance according to the argument, and object-cusp minimum distance to -90° , which effectively means that object-cusp interference won't be meaningful to the adjustment algorithm. Dial charts don't have cusps. Then we draw a circle for the basic dial, and set the right coordinate to 270° .

```

1637 \def\horo@dialwheel#1#2#3#4#5#6#7{%
1638   \def\horoomindist{#1}\def\horocmindist{-90.0}%
1639   \put(0,0){\circle{#2}}%
1640   \def\hororightcoord{270}%

```

The rest is only meaningful if there is calculated data. We plot a set of radials to show objects' Pos values.

```

1641   \ifhorocalculated
1642     \horoputradials{\horoobjects}{#3}{#7}%

```

Do `\horoadjust` to come up with reasonable DPos values.

```

1643   \horoadjust

```

Then plot the connecting curves and the object symbols and end (closing off the conditional).

```

1644   \horoforeach{\horoobjects}{\horoconncurve{#4}{#5}}%
1645   \horoforeach{\horoobjects}{\horoputobjsymbol{#6}}%
1646   \fi
1647 }

```

§11.5. Choosing text size.—

`\horotextsize` Adjustment to automatically-determined label text size in charts like Vancouver, where the text size depends on the length of the smart-label format string. Positive numbers make the text bigger, negative make it smaller.

```

1648 \def\horotextsize{0}

```

`\horochoosetextsize` This makes some attempt to choose a reasonable text size based on `\horobl1len`, so that labels with more chunks will be set in smaller type. We start out by setting `\count@` to the configuration setting `\horotextsize` (actually its negative) and adding the detected value of `\horobl1len`.

```

1649 \def\horochosetextsize{%
1650   \count@=\horotextsize\relax
1651   \multiply\count@ by-1\relax
1652   \advance\count@ by\horoblilen\relax
      Based on the result, we choose a text size from \Large down to \tiny.
1653   \ifcase\the\count@\or\Large\or\large\or\normalsize\or\small\or
1654     \scriptsize\or\scriptsize\or\tiny\fi
1655 }

```

12. READY-MADE WHEEL TEMPLATES

These are all designed to be used inside a horoscope environment.

§12.1. Basic wheel.—This is a standard wheel of the sort typically produced by a lot of astrological software; it probably resembles Astrolog’s more than anything else. It has a sign key and angle scale around the outside, and space for a centre label or aspect web in the middle. Objects have ticks showing their true locations against the angle scale and aspect web, and curves connecting those to their displayed locations within the houses. House cusps can jog out of their true positions in order to accomodate adjusted labels.

`\horowheelVancouver` The optional argument is the label format string, which will be passed into the smart labels code. Default value shows degrees and minutes of object position.

```

1656 \newcommand{\horowheelVancouver}[1][ydzmr/ymzdr]{%

```

The graphic elements that make up the wheel are arranged at the following radii:

r	element
0–21	aspect web or centre label
21	circle bounding aspect web from houses
21–22	radials showing true cusp positions
22	arcs connecting cusp radials to displaced cusps
21–23	radials showing true object positions
23–24	curves connecting object labels to radials
22–42	possibly-displaced cusps
25	approximate innermost object label chunk
32	centres of internal house labels
38	centres of outermost label chunks
40–41	curves connecting object labels to radials
41–43	radials showing true object positions
42	arcs connecting cusp radials to displaced cusps
42–43	radials showing true cusp positions
43	circle on inside of angle scale
43–44	1° ticks in angle scale
43–45	5° ticks in angle scale
45	circle bounding angle scale from sign key
45–50	30° ticks between signs
47.5	centres of sign symbols in key
50	circle around entire chart

The object label chunk radii are set by `\horosetsmartradii` to spacing that depends on the number of chunks, but the rough guideline is that they occupy radii 25–38. In fact, scanning the label string and setting the radii are the first things `\horowheelVancouver` does.

```
1657 \horoscanlabels{#1}%
1658 \horosetsmartradii{38}{4.666}{0.333}%
```

Set the default rounding mode, which is round to minutes keep degrees, if `\hororoundauto` is active.

```
1659 \ifhororoundauto\hororoundtominkeepdeg\fi
```

Draw the bounding circles (given by their diameter, which are twice their radii).

```
1660 \put(0,0){\circle{100}}%
1661 \put(0,0){\circle{90}}%
1662 \put(0,0){\circle{86}}%
1663 \put(0,0){\circle{42}}%
```

Draw the angle scale with its 1° and 5° ticks, and the sign key.

```
1664 \horoputticks{45}{5}{30}%
1665 \horoputticks{43}{1}{1}%
1666 \horoputticks{43}{2}{5}%
1667 \horoputsignkey{47.5}%
```

At this point `DPos=Pos`. Draw all the radials that go in true positions: two each for objects and cusps, and two more bold ones for the cusps.

```

1668 \ifhorocalculated
1669   \horoputradials{\horoobjects}{41}{2}%
1670   \horoputradials{\horoobjects}{21}{2}%
1671   \ifhorodrawcusps
1672     \horoputradials{\horocusps}{21}{1}%
1673     \horoputradials{\horocusps}{42}{1}%
1674   \fi
1675   \ifhoroboldangles
1676     \begingroup
1677       \thicklines\linethickness{\horoanglecuspswidth}%
1678       \horoputradials{\horoangularcusps}{21}{1}%
1679       \ifhoroanglearrows
1680         \horoputradials{\horoangularcusps}{42}{0.5}%
1681       \else
1682         \horoputradials{\horoangularcusps}{42}{1}%
1683       \fi
1684     \endgroup
1685   \fi
1686   \ifhoroanglearrows
1687     \horoputradials{\horoangularcusps}{43}{0.7}%
1688   \fi
1689 \fi

```

Compute the adjusted DPos values.

```
1690 \horoadjust
```

Now draw the linework that goes in, or uses, adjusted positions. That includes connecting curves for objects and cusps, and the main chunks of the cusps themselves.

```

1691 \ifhorocalculated
1692   \horoforeach{\horoobjects}{\horoconncurve{41}{40}}%
1693   \horoforeach{\horoobjects}{\horoconncurve{23}{24}}%
1694   \ifhorodrawcusps
1695     \horoforeach{\horocusps}{\horoconncurve{42}{42}}%
1696     \horoforeach{\horocusps}{\horoconncurve{22}{22}}%
1697     \horoputradials{\horocusps}{22}{20}%
1698   \fi
1699   \ifhoroboldangles
1700     \thicklines\linethickness{\horoanglecuspswidth}%
1701     \horoforeach{\horoangularcusps}{\horoconncurve{42}{42}}%
1702     \horoforeach{\horoangularcusps}{\horoconncurve{22}{22}}%
1703     \horoputradials{\horoangularcusps}{22}{20}%
1704     \thinlines
1705   \fi

```

Finally, the smart labels for objects, the aspect web, and the internal house labels.

```

1706   {\horochoosetextsize
1707   \horoforeach{\horoobjects}{\horoputsmartlabel}}%
1708   \ifhoroaspectweb\horoautoaspects{21}\fi
1709   \ifhoroínhouselabels\horoputínhouselabels{32}\fi

```

```
1710 \fi
1711 }
```

§12.2. Harmonic multi-dial charts.—These templates are intended to be similar to some of those used in Cosmobiology; they’re also handy for comparing multiple charts. One, two, three, or four concentric dials are supported. There’s also built-in support for calculating harmonics to create a 90° wheel, or a 45° wheel, or whatever. The dials have ticks on them indicating multiples of 1° and 5° (also 10° for the single and double dials), and the ticks expand with the harmonic to make it easy to measure angles visually. Labels for objects consist of just the object symbols, to save radial space (especially in the many-wheels case).

All these take an optional first argument which is the harmonic number; use 4 for a 90° dial. Their implementations are straightforward, based on the helper macros already defined.

```
\horowheelIqaluit Single dial with the main circle at radius 42.
1712 \newcommand{\horowheelIqaluit}[1][1]{%
1713 \horo@dialstart{42}{#1}%
1714 \horoputticks{42}{1}{\horo@dtone}%
1715 \horoputticks{42}{2}{\horo@dtfive}%
1716 \horoputticks{42}{3}{\horo@dtten}%
1717 \horo@dialwheel{5.0}{84}{42}{45}{46}{47.5}{3}%
1718 }}

\horowheelIgloolik Double dial, radii 34 and 42. The optional first argument is the harmonic; second
and third should be two sets of object positions saved with \horosaveobjects,
corresponding to the positions to display in the inner and outer dials respectively.
1719 \newcommand{\horowheelIgloolik}[3][1]{%
1720 \horo@dialstart{34}{#1}%
1721 #3\horocalcharmonic{#1}%
1722 \horoputticks{42}{1}{\horo@dtone}%
1723 \horoputticks{42}{2}{\horo@dtfive}%
1724 \horoputticks{42}{3}{\horo@dtten}%
1725 \horo@dialwheel{5.0}{84}{42}{45}{46}{47.5}{3}%
1726 #2\horocalcharmonic{#1}%
1727 \horoputticks{34}{1}{\horo@dtone}%
1728 \horoputticks{34}{2}{\horo@dtfive}%
1729 \horoputticks{34}{3}{\horo@dtten}%
1730 \horo@dialwheel{6.0}{68}{34}{37}{38}{39.5}{3}%
1731 }}

\horowheelResolute Triple dial, radii 29, 36, and 43. Optional first argument is harmonic; remaining
arguments are sets of saved object positions for the dials ordered from inner to
outer.
1732 \newcommand{\horowheelResolute}[4][1]{%
1733 \horo@dialstart{29}{#1}%
1734 #4\horocalcharmonic{#1}%
1735 \horoputticks{43}{1}{\horo@dtone}%

```

```

1736 \horoputticks{43}{2}{\horo@dtfive}%
1737 \horo@dialwheel{5.0}{86}{43}{45}{46}{47.5}{2}%
1738 #3\horocalcharmonic{#1}%
1739 \horoputticks{36}{1}{\horo@dtone}%
1740 \horoputticks{36}{2}{\horo@dtfive}%
1741 \horo@dialwheel{6.0}{72}{36}{38}{39}{40.5}{2}%
1742 #2\horocalcharmonic{#1}%
1743 \horoputticks{29}{1}{\horo@dtone}%
1744 \horoputticks{29}{2}{\horo@dtfive}%
1745 \horo@dialwheel{7.0}{58}{29}{31}{32}{33.5}{2}%
1746 }}

```

`\horowheelRankin` Quadruple dial, radii 22, 29, 36, and 43. Optional first argument is harmonic; remaining arguments are sets of saved object positions for the dials ordered from inner to outer.

```

1747 \newcommand{\horowheelRankin}[5][1]{%
1748 \horo@dialstart{22}{#1}%
1749 #5\horocalcharmonic{#1}%
1750 \horoputticks{43}{1}{\horo@dtone}%
1751 \horoputticks{43}{2}{\horo@dtfive}%
1752 \horo@dialwheel{5.0}{86}{43}{45}{46}{47.5}{2}%
1753 #4\horocalcharmonic{#1}%
1754 \horoputticks{36}{1}{\horo@dtone}%
1755 \horoputticks{36}{2}{\horo@dtfive}%
1756 \horo@dialwheel{6.0}{72}{36}{38}{39}{40.5}{2}%
1757 #3\horocalcharmonic{#1}%
1758 \horoputticks{29}{1}{\horo@dtone}%
1759 \horoputticks{29}{2}{\horo@dtfive}%
1760 \horo@dialwheel{7.0}{58}{29}{31}{32}{33.5}{2}%
1761 #2\horocalcharmonic{#1}%
1762 \horoputticks{22}{1}{\horo@dtone}%
1763 \horoputticks{22}{2}{\horo@dtfive}%
1764 \horo@dialwheel{10.0}{44}{22}{24}{25}{26.5}{2}%
1765 }}

```

§12.3. Decorative wheel charts.—These were created partly to demonstrate some of the possibilities of putting the pieces together to create interesting variations on the basic chart, and partly in an effort to imitate historical chart styles. The Montreal design was an attempt to create something similar to old-fashioned square charts with triangular houses that I’ve seen; but at the time I wasn’t remembering clearly how those charts were laid out, and I misunderstood which direction the triangles should point. The result is not much like historical charts at all, but was retained because it’s visually interesting anyway, and demonstrates creative abuse of spline curves. The Quebec City design is a second attempt at imitating historical charts; although circular instead of square, it uses the triangular-house layout, in a design that puts emphasis on the angles and house cusps. Both are designed to emphasize the houses things are in and de-emphasize the precise longitude relationships that the Vancouver chart focuses on. That may or may not accommodate a different style of interpretation.

`\horowheelMontreal` This chart style puts some labels at small radii, so the minimum distances for label adjustment have to be relatively large angles to prevent interference.

```
1766 \newcommand{\horowheelMontreal}{%
1767   \def\horoomindist{10.0}\def\horoomindist{6.0}%
      Set the default rounding mode, which is round to degrees keep sign, if
      \hororoundauto is active.
```

```
1768   \ifhororoundauto\hororoundtodegkeepsign\fi
      Draw circles at radii 15 and 45.
```

```
1769   \put(0,0){\circle{90}}%
1770   \put(0,0){\circle{30}}%
```

Put circular-reading labels at radius 47.5 for the four angles.

```
1771   \ifhorocalculated
1772     \ifhorodrawcusps
1773       \horoforeach{\horoangularcusps}{\horoputcusplabel{47.5}{5}}%
1774       \fi
1775   \fi
```

Do adjustment to find non-conflicting locations for everything else.

```
1776   \horoadjust
```

Now draw cusps separating succedent houses from the others. Note that each call of `\horo@montrealcurve` draws a curve from the *middle* of one house, on the inner circle, to the *cusp* of another, on the outer circle. For instance, `\horo@montrealcurve{I}{II}{III}` connects the first house cusp (that is the boundary between the twelfth and first houses; the ascendant) on the outside, to the midpoint of the second and third house cusps, which is the middle of the second house, on the inside. In general it's midpoints of succedent houses and cusps of angular houses, for a total of eight connecting curves around the circle. Also note that we use adjusted positions (DPos) on the inside and true positions (Pos) on the outside.

```
1777   \ifhorocalculated
1778     \ifhorodrawcusps
1779       \horo@montrealcurve{I}{II}{III}%
1780       \horo@montrealcurve{IV}{II}{III}%
1781       \horo@montrealcurve{IV}{V}{VI}%
1782       \horo@montrealcurve{VII}{V}{VI}%
1783       \horo@montrealcurve{VII}{VIII}{IX}%
1784       \horo@montrealcurve{X}{VIII}{IX}%
1785       \horo@montrealcurve{X}{XI}{XII}%
1786       \horo@montrealcurve{I}{XI}{XII}%
```

Draw cusps separating cadent from angular houses. The end result looks like a sort of four-petalled flower, with two houses inside each petal and one more in each of the spaces between.

```
1787     \horoforeach{\horoangularcusps}{\horoconncurve{45}{15}}%
1788   \fi
```

Finally, draw object labels. The code inside `\horo@putmontrealobj` places the labels at different radii depending on the house type. In general, the cusp adjustments mean that labels should not interfere with the cusp curves, but it's imaginable in the case of heavily overfilled houses.

```
1789 \horoforeach{\horoobjects}{\horo@putmontrealobj}%
1790 \fi
1791 }}
```

`\horowheelQuebecCity` As with the Montreal template, this one requires a fair bit of angular space between object labels.

```
1792 \newcommand{\horowheelQuebecCity}{%
1793 \def\horoomindist{10.0}\def\horooomindist{6.0}%
```

Set the default rounding mode, which is round to degrees keep sign, if `\horo-roundauto` is active.

```
1794 \ifhoro-roundauto\horo-roundto-deg-keep-sign\fi
```

Draw two circles for the outside; the cusp labels will go between them.

```
1795 \put(0,0){\circle{100}}%
1796 \put(0,0){\circle{90}}%
```

Do adjustment to find nice spacing for everything.

```
1797 \horo-adjust
```

Find the midpoints of all the houses and rotate the chart so that the triangle forming the seventh house will point exactly to the right, which means the triangle forming the first house should be very close to pointing exactly to the left.

```
1798 \ifhoro-calculated
1799 \horo@find-cusp-mids
1800 \let\horo-right-coord=\horo-cusp-VII-IMP-Pos
```

The layout here is that there's a (approximate) square in the middle, with triangles pointing out of its four sides. Those form the angular houses. The spaces between them are each bisected by a radial, to form the succedent and cadent houses. We draw those radials first; they are exactly on the cusps, and go from radius 20 (which circumscribes the near-square) to 45 (the inside of the two rim circles).

```
1801 \ifhoro-draw-cusps
1802 \horo-put-radials{CuspIII,CuspVI,CuspIX,CuspXII}{20}{25}%
```

Now the eight lines connecting the square corners to the rim to form the triangles; these go between angular house midpoints on the outside, and cadent house cusps (which are the square corners) on the inside.

```
1803 \horo@qcline{45}{IMP-Pos}{20}{III-ID-Pos}%
1804 \horo@qcline{45}{IVM-Pos}{20}{III-ID-Pos}%
1805 \horo@qcline{45}{IVM-Pos}{20}{VI-ID-Pos}%
1806 \horo@qcline{45}{VI-IMP-Pos}{20}{VI-ID-Pos}%
1807 \horo@qcline{45}{VI-IMP-Pos}{20}{IX-ID-Pos}%
1808 \horo@qcline{45}{XMP-Pos}{20}{IX-ID-Pos}%
1809 \horo@qcline{45}{XMP-Pos}{20}{XII-ID-Pos}%
1810 \horo@qcline{45}{IMP-Pos}{20}{XII-ID-Pos}%
```

Draw the square itself by connecting cusps of cadent houses.

```
1811     \horo@qccline{20}{IIIDPos}{20}{VIDPos}%
1812     \horo@qccline{20}{VIDPos}{20}{IXDPos}%
1813     \horo@qccline{20}{IXDPos}{20}{XIIDPos}%
1814     \horo@qccline{20}{XIIDPos}{20}{IIIDPos}%
1815 \fi
```

Now label the cusps. These labels go at midpoints of the houses, so we need to copy the MPos of the houses to their DPos.

```
1816     \horocopyvar{\horocusps}{MPos}{DPos}%
1817     \ifhorodrawcusps
1818         \horoforeach{\horocusps}{\horoputcusplabel{47.5}{5}}%
1819     \fi
```

Finally, add the labels for the objects.

```
1820     \horoforeach{\horoobjects}{\horo@putqcobj}%
1821 \fi
1822}}
```

To end the file: we close the conditional, and set default values for the calculation data so users can start typesetting charts immediately and not worry about whether they are *defining* or *redefining* macros. The default chart data is the author's.

```
1823 \fi
1824 \horocalparms{1976}{8}{1}{17:22:19}{W123:20:38}{N48:25:53}
```

CHANGE HISTORY



<p>v0.9</p> <p style="padding-left: 20px;">General: Initial (beta) version 1</p> <p>v0.91</p> <p style="padding-left: 20px;">\horo@aa: Fixed</p> <p style="padding-left: 40px;">only-checking-one-direction</p> <p style="padding-left: 40px;">bug 107</p> <p style="padding-left: 20px;">\horomakeequalcusps: New macro 73</p> <p style="padding-left: 20px;">\horomakesigncusps: New macro 73</p> <p style="padding-left: 20px;">\horoshiftcusps: New macro . . . 72</p> <p style="padding-left: 20px;">\horoshiftobjects: New macro . 72</p> <p style="padding-left: 20px;">General: egrep filtering option . . . 11</p> <p style="padding-left: 40px;">Added smart rounding features 40</p> <p style="padding-left: 40px;">Added support for Krusinski</p> <p style="padding-left: 60px;">houses with Swiss Ephemeris . 88</p> <p style="padding-left: 20px;">Documentation on equal-house</p> <p style="padding-left: 40px;">systems 18</p> <p style="padding-left: 20px;">Documentation on new shift</p> <p style="padding-left: 40px;">macros 21</p> <p style="padding-left: 20px;">Fake SEOpt macros for</p> <p style="padding-left: 40px;">always-calculated angles 87</p> <p style="padding-left: 20px;">Label example to use rounding . 29</p> <p>v0.92</p> <p style="padding-left: 20px;">\horo@dialstart: Replace</p> <p style="padding-left: 40px;">\drawline with \Line 120</p> <p style="padding-left: 20px;">\horoputararrowhead: Replace</p> <p style="padding-left: 40px;">\blacken\path with \polygon* 94</p> <p style="padding-left: 20px;">\horoputaspect: Replace</p> <p style="padding-left: 40px;">\drawline with \Line 106</p> <p style="padding-left: 20px;">\horoputcurve: Replace \spline</p> <p style="padding-left: 40px;">with \cbezier 96</p>	<p style="padding-left: 20px;">\horoputline: Replace \drawline</p> <p style="padding-left: 40px;">with \Line 93</p> <p style="padding-left: 20px;">\horoputradial: Replace</p> <p style="padding-left: 40px;">\drawline with \Line 93</p> <p style="padding-left: 20px;">\horowheelVancouver: Replace</p> <p style="padding-left: 40px;">\allinethickness with</p> <p style="padding-left: 40px;">\thicklines\linethickness</p> <p style="padding-left: 40px;">. 123, 124</p> <p style="padding-left: 20px;">General: Update links for font</p> <p style="padding-left: 40px;">packages 8</p> <p style="padding-left: 40px;">Use pict2e for graphics support . . 8</p> <p style="padding-left: 40px;">Use official MPC number for Eris 45</p> <p>v1.0</p> <p style="padding-left: 20px;">\horounicode: New macro 64</p> <p style="padding-left: 20px;">General: unicode option . 8, 10, 58, 62</p> <p style="padding-left: 20px;">Correct macro names for manual</p> <p style="padding-left: 40px;">position example 20</p> <p style="padding-left: 20px;">Emphasize that template names</p> <p style="padding-left: 40px;">do not determine geographic</p> <p style="padding-left: 40px;">locations of charts 23</p> <p style="padding-left: 20px;">General updates to intro after</p> <p style="padding-left: 40px;">seven years 5</p> <p style="padding-left: 20px;">In memoriam Axel 6</p> <p style="padding-left: 20px;">Make page after title page page 2</p> <p style="padding-left: 40px;">for clearer numbering in PDF . . 2</p> <p style="padding-left: 20px;">New links for my own Web sites . 6</p> <p style="padding-left: 20px;">Require iftex 59</p> <p style="padding-left: 20px;">Upgrade links to HTTPS . 8, 19, 43</p> <p>v1.01</p> <p style="padding-left: 20px;">General: Correct macro names for</p> <p style="padding-left: 40px;">manual position example, again 20</p>
--	---

INDEX



Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols		
<code>\@makeother</code> 871	<code>\catcode</code> 367, 1291	<code>\ensuremath</code>
<code>\@trimb</code> 372, 375	<code>\cbezier</code> 1042	... 94–96, 206, 279
<code>\@trimc</code> 375, 376	<code>\centering</code> 941	environments:
	<code>\Ceres</code> 335	horoscope <u>934</u>
	<code>\CeresSymbol</code>	
	. . 128, 188, 233, 335	
A	<code>\Chiron</code> 334	G
<code>\afterassignment</code> .. 376	<code>\ChironSymbol</code>	<code>\Gemini</code> 44, 68
<code>\aftergroup</code> ... 371,	. . 127, 187, 232, 334	<code>\gemini</code> 257
378, 379, 522,	<code>\circ</code> 94	<code>\GeminiSymbol</code>
525, 526, 538,	<code>\circle</code> 1639,	... 98, 107, 167,
542, 543, 756,	1660–1663, 1769,	212, 257, 288, 314
758, 782, 788, 821	1770, 1795, 1796	
<code>\Aquarius</code> 52, 76	<code>\closein</code> 784, 923	H
<code>\aquarius</code> 265	<code>\Conjunction</code> 352	<code>\hexstar</code> 281
<code>\AquariusSymbol</code> ...	<code>\conjunction</code> 277	<code>\horo@aa</code> <u>1335</u> , 1350
. . 101, 115, 175,	<code>\count@</code> 1650–1653	<code>\horo@aa</code> ... <u>1353</u> , 1359
220, 265, 296, 322	<code>\countdef</code> 1405	<code>\horo@angh@</code> .. 1543,
<code>\Aries</code> 42, 66	<code>\CuspIISymbol</code> 137	1545, 1548, 1551,
<code>\aries</code> 255	<code>\CuspIISymbol</code> 136	1554, 1556, 1558
<code>\AriesSymbol</code>	<code>\CuspISymbol</code> ... 135, 344	<code>\horo@anghouse</code> <u>1540</u> , 1597
... 98, 105, 165,	<code>\CuspIVSymbol</code> .. 138, 345	<code>\horo@calc@a</code> ... 756, 778
210, 255, 286, 312	<code>\CuspIXSymbol</code> 143	<code>\horo@calc@b</code> ... 660,
<code>\ASC</code> 344, 348	<code>\CuspVIIISymbol</code> ... 142	662, 671, 765, 769
<code>\AscendantSymbol</code> 147, 348	<code>\CuspVIISymbol</code> . 141, 346	<code>\horo@calc@c</code> ... 757, <u>787</u>
<code>\astrosun</code> 267	<code>\CuspVISymbol</code> 140	<code>\horo@calc@d</code> ... 761, 777
	<code>\CuspVSymbol</code> 139	<code>\horo@calc@e</code> ... 765, 777
B	<code>\CuspXIIISymbol</code> 146	<code>\horo@calc@f</code> ... 769, 777
<code>\begin</code> 936	<code>\CuspXISymbol</code> 145	<code>\horo@calc@parse</code> 783, <u>790</u>
<code>\bigtriangleup</code> 279	<code>\CuspXSymbol</code> ... 144, 347	<code>\horo@calc@parse@a</code>
	 791, 793
C	D	<code>\horo@calc@parse@b</code>
<code>\Cancer</code> 45, 69	<code>\DeclareOption</code> .. 9– 793, <u>798</u>
<code>\cancer</code> 258	13, 16–18, 20, 23, 24	<code>\horo@calc@parse@b@</code>
<code>\CancerSymbol</code>	<code>\dimendef</code> 955–962 799, 800
... 99, 108, 168,	<code>\do</code> 871	<code>\horo@calc@parse@c</code>
213, 258, 289, 315	<code>\dospecials</code> 871 800, <u>802</u>
<code>\Capricorn</code> 51, 75	<code>\DSC</code> 346	<code>\horo@calch</code> ... 430, <u>432</u>
<code>\CapricornSymbol</code> ..		<code>\horo@calcobj</code> .. <u>909</u> , 918
. . 101, 114, 174,	E	<code>\horo@calcobj@</code>
219, 264, 295, 321	<code>\end</code> 938	. . <u>892</u> , 910, 911, 919
<code>\capricornus</code> 264		

\horo@chartrotate .	\horo@delta	1008, 1012, 1025,
. <u>484</u> ,	. . . 1241, 1285, 1286	1039, 1042, 1325,
1021, 1023, 1054,	\horo@dfe@a . . . <u>412</u> , 422	1327, 1330–1332
1070, 1095, 1323,	\horo@dfe@b . . . <u>404</u> , 416	\horo@dim@yb . . . 960,
1324, 1609, 1611	\horo@dfe@ction <u>408</u> , 421	974, 976, 978,
\horo@cue . . 368, 393,	\horo@dfe@x 408, 413, 414	985, 987, 996,
406, 414, 1237, 1273	\horo@dfe@xlist 421, 422	998, 1000, 1011,
\horo@cv <u>517</u>	\horo@dfe@y 405, 406, 408	1026, 1040, 1045,
\horo@cv@ 515, 517	\horo@dfe@ylist 416, 421	1326, 1328, 1330
\horo@d@deg . . . 590,	\horo@dialstart . . .	\horo@dim@yc
610, 629, 645,	. . . <u>1616</u> , 1713,	. . . 962, 1001,
655, 670, 761,	1720, 1733, 1748	1003, 1005, 1013
765, 769, 899,	\horo@dialwheel . . .	\horo@djcheckcusps
1122, 1129, 1155,	. . . <u>1637</u> , 1717,	. . . <u>1488</u> , 1515, 1517
1157, 1172, 1175,	1725, 1730, 1737,	\horo@djcheckmovement
1196, 1199, 1246	1741, 1745, 1752,	. . . <u>1449</u> , 1461, 1478
\horo@d@dms . . . <u>621</u> , 669	1756, 1760, 1764	\horo@djcuspcycle .
\horo@d@dms@ . . . <u>588</u> ,	\horo@dim@x 955, 966, <u>1468</u> , 1521
760, 763, 767, 894	967, 994, 996,	\horo@djo@ <u>1417</u> , 1441,
\horo@d@dms@@	1001, 1007, 1010,	1459, 1472, 1475
. <u>579</u> , 590–	1014, 1033–1037,	\horo@djobjcycle . .
592, 610, 614,	1043, 1044, 1419– <u>1454</u> , 1512
618, 629, 633, 637	1421, 1423, 1433	\horo@djoo . . . <u>1438</u> ,
\horo@d@min . . . 591,	\horo@dim@xa	1458, 1473, 1474
614, 633, 646,	. . 957, 973, 975,	\horo@djoo@same <u>1444</u> ,
654, 655, 671,	977, 984, 986,	1460, 1476, 1477
760, 761, 764,	995, 997, 1000,	\horo@djsavedpos . .
765, 768, 769,	1002, 1005–1007, <u>1485</u> , 1524
895, 899, 1120,	1012, 1025, 1035,	\horo@djustcuspsfalse
1131, 1167, 1180,	1042, 1325, 1327, 1513
1190, 1204, 1247	1329, 1331, 1332	\horo@djustcuspstrue
\horo@d@sdms	\horo@dim@xb . . . 959, 1497
. <u>594</u> , 644, 653, 1058	974, 975, 978,	\horo@djustedfalse
\horo@d@sec 592, 618,	985, 987, 996, 1456, 1470
637, 646, 654,	997, 999, 1011,	\horo@djustedtrue 1452
655, 671, 895,	1026, 1036, 1045,	\horo@dtfive
899, 1188, 1207, 1248	1326, 1328, 1329	1622, 1626, 1715,
\horo@d@sign 603, 604,	\horo@dim@xc	1723, 1728, 1736,
645, 1112, 1153, 961, 1001,	1740, 1744, 1751,
1159, 1169, 1177,	1002, 1004, 1013	1755, 1759, 1763
1193, 1201, 1249	\horo@dim@y 956, 966,	\horo@dtone
\horo@data . 554, 555,	967, 994, 996,	1620, 1625, 1714,
561, 867, 872–	1001, 1008, 1010,	1722, 1727, 1735,
874, 880, 881,	1014, 1033, 1038–	1739, 1743, 1750,
886, 1535, 1536, 1538	1041, 1043, 1044	1754, 1758, 1762
\horo@dblforeach . .	\horo@dim@ya	\horo@dtten
. . . . <u>420</u> , 1350,	. . 958, 973, 976, 1624, 1627,
1458, 1459, 1472–	977, 984, 986,	1716, 1724, 1729
1475, 1514, 1516	995, 998, 999,	\horo@eathead . . <u>381</u> , 383
\horo@ddms . <u>553</u> , 684–686	1003, 1004, 1006,	\horo@egrepfalse . . . 19

\horo@egreptrue	20	\horo@lbtwo .	1231, 1263	\horo@psl@a .	1236, 1237
\horo@fe	<u>391</u> , 402	\horo@marvosymfalse .	7	\horo@psl@o	1223, 1250, 1253
\horo@fe@a	400, 402	\horo@marvosymtrue .	12	\horo@pt	<u>1068</u> , 1080
\horo@fe@b	395, 401	\horo@midpoint	476, 1031, 1369, 1581	\horo@pta	1071, 1078
\horo@fe@c	392, 393	\horo@montrealcurve	<u>1580</u> , 1779–1786	\horo@ptb	1071, 1078
\horo@findcuspmid	<u>1368</u> , 1374–1379	\horo@nsew	658, 668, 762, 766	\horo@ptc	1072, 1078
\horo@findcuspmids	<u>1373</u> , 1504, 1799	\horo@outerrad	1227, 1280	\horo@putmontrealobj	<u>1585</u> , 1789
\horo@findpdiff <u>1406</u> ,	1418, 1422, 1445, 1450, 1492, 1534	\horo@pa	<u>1083</u> , 1088	\horo@putpolar	<u>963</u> , 1097, 1112, 1118, 1127, 1139, 1153, 1154, 1157, 1158, 1166, 1168, 1171, 1174, 1176, 1179, 1187, 1189, 1192, 1195, 1198, 1200, 1203, 1206, 1216, 1239, 1384
\horo@fixdimen@	435, 448, 482, 486, <u>488</u> , 502, 1117, 1126, 1414	\horo@pa@r	992, 994	\horo@putqcbj <u>1596</u> , 1820	
\horo@fixdimen@diff	<u>479</u> , <u>501</u> , 1338, 1341, 1410	\horo@pa@sc@le	993, 995	\horo@qcline	1607, 1803–1814
\horo@ft@a	378, <u>382</u>	\horo@pa@th	992, 994, 995	\horo@r@offdig	<u>565</u>
\horo@fterdef	<u>377</u> , 511, 523, 530, 531, 540, 547, 584, 604, 810, 815, 819, 874, 1268, 1286, 1486, 1536, 1556, 1576, 1625–1627	\horo@pc@r@ne	1019, 1025, 1027	\horo@r@offdigit	567–577, 596, 607, 611, 615, 623, 626, 630, 634
\horo@getobjdpos	<u>1052</u> , 1059, 1062, 1084, 1138, 1215, 1383	\horo@pc@r@mid	1030, 1033	\horo@r@offset	<u>564</u> , 568–577, 597, 608, 612, 616, 624, 627, 631, 635
\horo@getobjsdms	<u>1056</u> , 1111, 1145, 1222	\horo@pc@r@tw@	1020, 1026, 1028	\horo@radius	1097, 1104
\horo@houseletter	850–862, 915	\horo@pc@th@ne	1019, 1021, 1022, 1025, 1031	\horo@readdata	865, 879, 885
\horo@housenumber	732–745, 773	\horo@pc@th@mid	1032, 1033	\horo@readpos	878, 884, 922
\horo@houses	<u>1366</u> , 1390, 1507	\horo@pc@th@tw@	1020, 1023, 1024, 1026, 1031	\horo@readposvel <u>883</u> , 921	
\horo@i	1405, 1455, 1462–1464, 1469, 1479–1481, 1511, 1520	\horo@pcl@os	1113–1115, 1119, 1128	\horo@savecount@	469, 474, 489, 499
\horo@isclockwise	<u>1531</u> , 1544, 1547, 1550, 1553, 1564, 1567, 1570, 1573	\horo@pihl	<u>1381</u> , 1390	\horo@savobjlist	1506, 1526
\horo@lblone	1229, 1262	\horo@pol@i	<u>1142</u> , 1151, 1164, 1185	\horo@scanlabels@	1259, <u>1261</u>
		\horo@pol@os	1146–1148, 1152, 1165, 1186, 1224–1226, 1228	\horo@scanlabels@@	1266, <u>1271</u>
		\horo@polarconvert	<u>385</u> , 966, 973, 974, 984, 985, 994, 995, 1025, 1026, 1033, 1325, 1326	\horo@shift	440, 443, <u>445</u>
		\horo@ppendtmpfalse	824, 916	\horo@s@ls@a	1272, 1273
		\horo@ppendtmptrue	907	\horo@stAquSign	730
		\horo@pplycorrection	<u>1412</u> , 1434, 1447	\horo@starfontfalse	8
		\horo@pr	<u>1061</u> , 1066		
		\horo@psk	<u>1090</u> , 1106		
		\horo@psl	1229, 1231, <u>1235</u>		
		\horo@psl@	1240, <u>1245</u>		

\horo@starfonttrue	\horo@templatesfalse	\horoaspectwebtrue 1296
..... 13, 30 23, 24	\horoastrologo 748, 779
\horo@stAriSign ... 720	\horo@templatestrue 22	\horoautoaspect ...
\horo@stAscObj 708	\horo@textsymfalse .. 4 1348, 1354
\horo@stCanSign ... 723	\horo@textsymtrue ... 9	\horoautoaspects ..
\horo@stCapSign ... 729	\horo@tmp 596,	... 1358, 1631, 1708
\horo@stCerObj 700	607, 611, 615,	\horoAxialHouses .. 851
\horo@stChiObj 699	623, 626, 630, 634	\horoAzimuthalHouses 852
\horo@stDesObj 714	\horo@tmpa ... 1439,	\horoBiquintileAngle
\horo@stGemSign ... 722	1440, 1489, 1490 1302
\horo@stJunObj 702	\horo@tmpb ... 1439,	\horoBiquintileOrb 1313
\horo@stJupObj 694	1440, 1489, 1490	\horoBiquintileSymbol
\horo@stLeoSign ... 724	\horo@tmpfile 156, 357
\horo@stLibSign ... 726	.. 746, 781, 784,	\horoBoldanglestrue 927
\horo@stLilObj 707	791, 792, 863,	\horocalcday 683, 775, 898
\horo@stMarObj 693	866, 872, 920, 923	\horocalcharmonic .
\horo@stMerObj 691	\horo@tmpfname 429, 1617,
\horo@stMidObj 717 747, 772,	1721, 1726, 1734,
\horo@stMooObj 690	781, 864, 904, 920	1738, 1742, 1749,
\horo@stNadObj 711	\horo@twodig 552, 654,	1753, 1757, 1761
\horo@stNepObj 697	760, 764, 768, 895	\horocalclat 686, 766, 914
\horo@stNodObj 704	\horo@unicodedefalse .. 5	\horocalclon 685, 762, 914
\horo@stPalObj 701	\horo@unicodetrue .. 10	\horocalcmmonth
\horo@stPisSign ... 731	\horo@wasysymfalse .. 6 683, 774, 898
\horo@stPluObj 698	\horo@wasysymtrue .. 11	\horocalcparms 682, 1824
\horo@strologfalse	\horo@wheelsfalse .. 23	\horocalctime
..... 14, 16, 18	\horo@wheelstrue ... 21 684, 760, 894
\horo@strologtrue .. 17	\horo@YF 797, 800	\horocalculate . 753, 913
\horo@stSagSign ... 728	\horoadjcycles	\horocalculatedfalse
\horo@stSatObj 695 1394, 1462, 424, 782, 868
\horo@stScoSign ... 727	1463, 1479, 1480	\horocalculatedtrue
\horo@stSunObj 689	\horoadjust 821, 917
\horo@stTauSign ... 721 1501, 1643,	\horocalcyear
\horo@stUraObj 696	1690, 1776, 1797 683, 776, 898
\horo@stVenObj 692	\horoAlcabitushouses	\horoCampanushouses
\horo@stVesObj 703 741, 850 735, 853
\horo@stVirSign ... 725	\horoanglearrowstrue 928	\horoCeres@strologname
\horo@succh@ .. 1563,	\horoanglecuspswidth 700
1565, 1568, 1571,	... 932, 1677, 1700	\horoCeresSEOpt ... 838
1574, 1576, 1578	\horoangularcusps .	\horoChiron@strologname
\horo@succhouse ...	428, 1678, 1680, 699
..... 1560, 1586	1682, 1687, 1701-	\horoChironSEOpt .. 836
\horo@svc@a ... 541, 546	1703, 1773, 1787	\horochoosetextsize
\horo@svc@b ... 542, 549	\horoARMCSEOpt 847 1649, 1706
\horo@svo@a ... 524, 529	\horoAscendantSEOpt 845	\horoCnote
\horo@svo@b ... 525, 533	\horoaspectobjects a 940
\horo@swetestfalse 1293, 1354	\horoConjunctionAngle
..... 16, 17	\horoaspectobjects b 1297
\horo@swetesttrue 15, 18 1294, 1354	\horoConjunctionOrb 1308
	\horoaspects . 1295, 1359	

\horoConjunctionSymbol	1524, 1644, 1645,	\horomakeequalcusps 452, 473
..... 151,	1692, 1693, 1695,	\horomakesigncusps	467
196, 241, 277, 352	1696, 1701, 1702,	\horoMars@strologname 693
\horoconncurve	1707, 1773, 1787,	\horoMarsSEOpt	830
1048, 1644, 1692,	1789, 1818, 1820	\horomarvosym	285
1693, 1695, 1696,	\horoouselabel ...	\horomaxrepulsion 1398, 1428, 1429
1701, 1702, 1787 1363, 1384	\horoMCSEOpt	846
\horoCopyvar ... 514,	\horoInhouseLabelsfalse	\horoMeanNodeSEOpt	842
534, 550, 1457, 1361	\horoMercury@strologname 691
1471, 1505, 1816	\horoJuno@strologname	\horoMercurySEOpt .	828
\horoCuspAdjustTrigger 702	\horoMeridianHouses	736
..... 1393, 1496	\horoJunoSEOpt	\horominutes 95,	646, 671, 1120,
\horoCuspIIIPos ... 455	\horoJupiter@strologname	1131, 1167, 1180,	1190, 1204, 1247
\horoCuspIIPos 454 694	\horoMoon@strologname 690
\horoCuspIVPos 456	\horoJupiterSEOpt .	\horoMoonSEOpt	827
\horoCuspIXPos 461	831	\horoMorinusHouses 739, 857
\horoCusps 426,	\horoKochHouses 733, 855	\horoNeoPorphyryHouses 743
443, 540, 541,	\horoKrusinskiHouses 856	\horoNeptune@strologname 697
550, 922, 1459,	\horoLatlonText ... 665	\horoNeptuneSEOpt .	834
1471-1473, 1475,	\horoLbllen .. 1267,	\horoNorthNode@strologname 704
1477, 1478, 1505,	1268, 1283, 1652	\horoobjects 430, 440,	523, 524, 534,
1516, 1524, 1672,	\horoLilith@strologname	749, 757, 888,	918, 921, 1293,
1673, 1695- 707	1294, 1457-1461,	1471, 1472, 1474-
1697, 1816, 1818	\horoLilithSEOpt ..	1476, 1478, 1506,	1507, 1514, 1516,
\horoCuspsSEOpt ... 914	844	1524, 1526, 1642,	1644, 1645, 1669,
\horoCuspVIIIPos .. 460	\horoLNote	1670, 1692, 1693,	1707, 1789, 1820
\horoCuspVIIPos . 1800	949	\horoocmindist 1400, 1459,
\horoCuspVIIPos 425, 459	\horoLRnote	1472, 1475, 1517,	1638, 1767, 1793
\horoCuspVIPos 458	952	\horoocmindist 1399, 1441, 1515,
\horoCuspVPos	\horoM@rvAquarius 52, 296	1638, 1767, 1793	
457	\horoM@rvAries .. 42, 286		
\horoCuspXIIPos ... 464	\horoM@rvCancer . 45, 289		
\horoCuspXIIPos 463	\horoM@rvCapricorn		
\horoCuspXIPos 462 51, 295		
\horoDegrees	\horoM@rvGemini . 44, 288		
... 94, 645, 670,	\horoM@rvJupiter 59, 303		
1122, 1129, 1155,	\horoM@rvLeo 46, 290		
1157, 1172, 1175,	\horoM@rvLibra .. 48, 292		
1196, 1199, 1246	\horoM@rvMars ... 58, 302		
\horoDrawCuspstrue 926	\horoM@rvMercury 56, 300		
\horoDsmstext	\horoM@rvMoon ... 55, 299		
640	\horoM@rvNeptune 62, 306		
\horoEqualHouses 734, 854	\horoM@rvPisces . 53, 297		
\horoEqualMCHouses 742	\horoM@rvPluto .. 63, 307		
\horoForeach ... 399,	\horoM@rvSagittarius		
430, 440, 443, 50, 294		
515, 524, 541,	\horoM@rvSaturn . 60, 304		
757, 918, 921,	\horoM@rvScorpio 49, 293		
922, 1066, 1088,	\horoM@rvSun 54, 298		
1359, 1390, 1460,	\horoM@rvTaurus . 43, 287		
1461, 1476-1478,	\horoM@rvUranus . 61, 305		
	\horoM@rvVenus .. 57, 301		
	\horoM@rvVirgo .. 47, 291		

<code>\horoOppositionAngle</code>	<code>\horoPutrxLabel</code> ...	<code>\horoRoundtoseckkeepdeg</code>
..... 1298 1211 , 1589, 573
<code>\horoOppositionOrb</code> 1309	1593, 1600, 1604	<code>\horoRoundtoseckkeepmin</code>
<code>\horoOppositionSymbol</code>	<code>\horoPutrsignkey</code> 574, 642
..... 152, 1102 , 1667	<code>\horoRoundtoseckkeepsign</code>
197, 242, 278, 353	<code>\horoPutsmartlabel</code> 572
<code>\horoPallas@strologname</code> 1220 , 1707	<code>\horoRoundtruncate</code>
..... 701	<code>\horoPutticks</code> 568, 651, 667
<code>\horoPallasSEOpt</code> .. 839	... 1076 , 1664–	<code>\horoRxtext</code>
<code>\horoPholusSEOpt</code> .. 837	1666, 1714–	674
<code>\horoPlacidusHouses</code>	1716, 1722–	<code>\horoSaturn@strologname</code>
.. 732, 752, 858, 891	1724, 1727– 695
<code>\horoPluto@strologname</code>	1729, 1735, 1736,	<code>\horoSaturnSEOpt</code> .. 832
..... 698	1739, 1740, 1743,	<code>\horosavecusps</code> 536
<code>\horoPlutoSEOpt</code> ... 835	1744, 1750, 1751,	<code>\horosaveobjects</code> .. 520
<code>\horoPolichPageHouses</code>	1754, 1755, 1758,	<code>\horoscanlabels</code> ...
..... 740, 859	1759, 1762, 1763 1258 , 1657
<code>\horoPorphyryHouses</code>	<code>\horoQuincunxAngle</code> 1304	horoscope (environ-
..... 738, 860	<code>\horoQuincunxOrb</code> . 1315	ment)
<code>\horoposattcusp</code> ...	<code>\horoQuincunxSymbol</code>	934
..... 1396 , 1477	.. 158, 201, 246, 359	<code>\horoseconds</code> . 96, 646,
<code>\horoposattobj</code>	<code>\horoQuintileAngle</code> 1301	671, 1188, 1207, 1248
... 1395 , 1460, 1476	<code>\horoQuintileOrb</code> . 1312	<code>\horoSemisextileAngle</code>
<code>\horoPutarrowhead</code> .	<code>\horoQuintileSymbol</code> 1305
..... 990 , 1085 155, 356	<code>\horoSemisextileOrb</code> 1316
<code>\horoPutarrows</code> 1087 , 1687	<code>\horoRegiomontanusHouses</code>	<code>\horoSemisextileSymbol</code>
<code>\horoPutaspect</code> 1319 , 1344 737, 861	.. 159, 202, 247, 360
<code>\horoPutcurve</code>	<code>\horoRepulsion</code> 1397 , 1431	<code>\horoSemisquareAngle</code>
... 1017 , 1049, 1583	<code>\horoRetrogradeSymbol</code> 1306
<code>\horoPutcusplabel</code> .	.. 150, 195, 240,	<code>\horoSemisquareOrb</code> 1317
... 1109 , 1773, 1818	351, 678, 1216, 1254	<code>\horoSemisquareSymbol</code>
<code>\horoPutinhouseLabels</code>	<code>\horoRightCoord</code> 160, 203, 248, 361
..... 1388 , 1709 425 , 485,	<code>\horoSesquiquadrateAngle</code>
<code>\horoPutline</code> .. 981 , 1612	510, 511, 539, 1307
<code>\horoPutobjdeglabel</code>	540, 1629, 1640, 1800	<code>\horoSesquiquadrateOrb</code>
..... 1150 , 1588,	<code>\horoRotatchart</code> .. 505 1318
1592, 1599, 1603	<code>\horoRoundautottrue</code> 578	<code>\horoSesquiquadrateSymbol</code>
<code>\horoPutobjminlabel</code> 1163	<code>\horoRoundclampfalse</code> 566	.. 161, 204, 249, 362
<code>\horoPutobjseclabel</code> 1184	<code>\horoRoundstrictruncate</code>	<code>\horoSetsmartradii</code>
<code>\horoPutobjsymbol</code> 567 1279 , 1658
1136 , 1587, 1591,	<code>\horoRoundtodeg</code> ... 571	<code>\horoSextileAngle</code> 1303
1598, 1602, 1645	<code>\horoRoundtodegkeepsign</code>	<code>\horoSextileOrb</code> .. 1314
<code>\horoPutradial</code> 577, 1768, 1794	<code>\horoSextileSymbol</code>
... 970 , 1063, 1071	<code>\horoRoundtomin</code> ... 570 157,
<code>\horoPutradials</code> ...	<code>\horoRoundtominkeepdeg</code>	200, 245, 281, 358
1065 , 1642, 1669, 576, 1659	<code>\horoshiftcusps</code> 442 , 465
1670, 1672, 1673,	<code>\horoRoundtominkeepsign</code>	<code>\horoshiftobjects</code> . 439
1678, 1680, 1682, 575	<code>\horosignificantadj</code>
1697, 1703, 1802	<code>\horoRoundtosec</code> ... 569 1392 , 1452
		<code>\horoSouthNode@strologname</code>
	 705
		<code>\horoSquareAngle</code> . 1300

<code>\Libra</code>	48, 72	<code>\newread</code>	746, 863	<code>\Quincunx</code>	359
<code>\libra</code>	261	<code>\nin@ty</code>	386		
<code>\LibraSymbol</code>		<code>\noexpand</code>	372	R	
.. 100, 111, 171,		<code>\normalsize</code>	1653	<code>\raggedleft</code> ...	947, 953
216, 261, 292, 318		<code>\NorthNode</code> .	339, 341, 342	<code>\raggedright</code> ...	944, 950
<code>\Lilith</code>	343	<code>\NorthNodeSymbol</code> ..		<code>\rangle</code>	206
<code>\LilithSymbol</code> 102, 103,		<code>\read</code>	791, 872
.. 134, 194, 239, 343		132, 192, 237, 339		<code>\RequirePackage</code> ...	
<code>\Line</code>	977,			35, 37, 41, 65, 91, 92	
986, 1327, 1633, 1634		O		<code>\Retrograde</code>	351
<code>\linethickness</code> 1677, 1700		<code>\openin</code>	781, 920	<code>\rightmoon</code>	268
<code>\long</code>	375, 376	<code>\Opposition</code>	353	<code>\Roman</code>	1364
		<code>\opposition</code>	278		
M		<code>\or</code> ..	98–101, 1653, 1654	S	
<code>\makebox</code> 941, 944, 947,				<code>\Sagittarius</code>	50, 74
950, 953, 967, 1332		P		<code>\sagittarius</code>	263
<code>\Mars</code>	58, 82	<code>\p@four</code>	983, 985	<code>\SagittariusSymbol</code>	
<code>\mars</code>	271	<code>\p@one</code>	965, 966,	.. 100, 113, 173,	
<code>\MarsSymbol</code> 121, 181,		972, 973, 983,		218, 263, 294, 320	
226, 271, 302, 328		984, 1322, 1325, 1326		<code>\Saturn</code>	60, 84
<code>\mathsf</code>	356, 357	<code>\p@three</code> 972–974, 983,		<code>\satur</code>	273
<code>\mbox</code>	206	985, 1322, 1324, 1326		<code>\SaturnSymbol</code>	
<code>\MC</code>	347, 349	<code>\p@two</code>	965, 966, 123, 183,	
<code>\MCSymbol</code>	148, 349	972, 974, 983,		228, 273, 304, 330	
<code>\MeanNodeSymbol</code> 102, 341		984, 1322, 1323, 1325		<code>\Scorpio</code>	49, 73
<code>\Mercury</code>	56, 80	<code>\Pallas</code>	336	<code>\scorpio</code>	262
<code>\mercury</code>	269	<code>\PallasSymbol</code>		<code>\ScorpioSymbol</code>	
<code>\MercurySymbol</code> 129, 189, 234, 336		.. 100, 112, 172,	
..... 119, 179,		<code>\parbox</code>	941,	217, 262, 293, 319	
224, 269, 300, 326		944, 947, 950, 953		<code>\scriptsize</code>	1654
<code>\message</code> . 1455, 1469,		<code>\Pisces</code>	53, 77	<code>\Semisextile</code>	360
1510, 1519, 1523		<code>\pisces</code>	266	<code>\Semisquare</code>	361
<code>\Moon</code>	55, 79	<code>\PiscesSymbol</code>		<code>\Sesquiquadrate</code> ...	362
<code>\MoonSymbol</code> 118, 178,		.. 101, 116, 176,		<code>\setcounter</code>	1389
223, 268, 299, 325		221, 266, 297, 323		<code>\setlength</code> .	931, 933, 935
		<code>\Pluto</code>	63, 87	<code>\Sextile</code>	358
N		<code>\pluto</code>	276	<code>\small</code>	1653
<code>\NeedsTeXFormat</code>	1	<code>\PlutoSymbol</code> 126, 186,		<code>\SouthNode</code>	340
<code>\Neptune</code>	62, 86	231, 276, 307, 333		<code>\SouthNodeSymbol</code> ..	
<code>\neptune</code>	275	<code>\polygon</code>	1009	.. 133, 193, 238, 340	
<code>\NeptuneSymbol</code>		<code>\ProcessOptions</code>	25	<code>\space</code>	761, 765,
..... 125, 185,		<code>\ProvidesPackage</code>	2	769, 771–776,	
230, 275, 306, 332		<code>\put</code>	941, 944,	793, 897–903, 915	
<code>\newcounter</code>	1362	947, 950, 953,		<code>\Square</code>	38, 88
<code>\newif</code> ... 4–8, 14, 15,		967, 1332, 1639,		<code>\stepcounter</code>	1385
19, 21, 22, 424,		1660–1663, 1769,		<code>\string</code>	383
566, 578, 824,		1770, 1795, 1796		<code>\Sun</code>	54, 78
926–928, 1296,				<code>\SunSymbol</code> .	117, 177,
1361, 1402, 1403		Q		222, 267, 298, 324	
<code>\newlength</code>	930, 932	<code>\Q</code>	367, 1291		

T			
<code>\Taurus</code>	43, 67		
<code>\taurus</code>	256		
<code>\TaurusSymbol</code>		1206, 1216, 1239,	
... 98, 106, 166,		1285, 1323, 1324,	
211, 256, 287, 313		1327, 1328, 1332,	
<code>\textwidth</code>	931	1371, 1384, 1415,	
<code>\TG@sin</code>	386, 388	1583, 1612, 1613	
<code>\TG@rem@pt</code>	436,	<code>\thicklines</code> . 1677, 1700	
437, 449, 450,		<code>\thinlines</code>	
510, 561, 809,		<code>\tiny</code>	
814, 967, 977,		<code>\toks</code>	
978, 986, 987,		<code>\trimspaces</code> <u>369</u> , 562, 873	
1010–1014, 1022,		<code>\Trine</code>	
1024, 1030, 1032,		<code>\TrueNodeSymbol</code> 103, 342	
1042–1045, 1063,		U	
1071, 1085, 1097,		<code>\unitlength</code> 935, 941,	
1112, 1118, 1127,		944, 947, 950, 953	
1139, 1153, 1154,		<code>\uppercase</code>	
1157, 1158, 1166,		<code>\Uranus</code>	
1168, 1171, 1174,		<code>\uranus</code>	
1176, 1179, 1187,		<code>\UranusSymbol</code>	
1189, 1192, 1195,	 124, 184,	
1198, 1200, 1203,		229, 274, 305, 331	
		V	
		<code>\Venus</code>	57, 81
		<code>\venus</code>	270
		<code>\VenusSymbol</code> 120, 180,	
		225, 270, 301, 327	
		<code>\Vertex</code>	350
		<code>\VertexSymbol</code> ..	149, 350
		<code>\Vesta</code>	338
		<code>\VestaSymbol</code>	
		.. 131, 191, 236, 338	
		<code>\vfuzz</code>	376
		<code>\Virgo</code>	47, 71
		<code>\virgo</code>	260
		<code>\VirgoSymbol</code>	
		... 99, 110, 170,	
		215, 260, 291, 317	
		W	
		<code>\write</code>	770, 896
		Z	
		<code>\Zodiac</code>	<u>97</u> , 645,
			1098, 1112, 1153,
			1159, 1169, 1177,
			1193, 1201, 1249