

TYPESETTING

ASTROLOGY

with

horoscop



MATTHEW SKALA
mskala@ansuz.sooke.bc.ca

Version 0.91, August 30, 2008

CONTENTS



1	Introduction	4
2	Prerequisites and Warnings	6
2.1	Font Support	6
2.2	Chart Graphics	6
2.3	External Programs	6
3	Package Options	8
3.1	Astrological Symbol Options	8
3.2	Calculation Backend Options	8
3.3	Optional Package Components	9
4	General Concepts	10
4.1	Objects and Variables	10
4.1.1	Objects	10
4.1.2	Cusps	10
4.1.3	Variables	11
4.2	Symbols	11
4.2.1	Naming scheme	11
4.2.2	Zodiac signs	12
4.2.3	Luminaries and traditional planets	12
4.2.4	Other objects, cusps, and angles	12
4.2.5	Aspects	13
4.2.6	Text angle and direction symbols	14
5	Calculating Horoscopes	15
5.1	Entering Chart Data	15
5.2	Calculating Object Positions	15
5.3	Entering Positions Manually	17
5.4	Saving and Manipulating Positions	18
5.5	Printing the Results as Text	20
6	Ready-Made Chart Templates	21
6.1	Basic Wheel Charts	21
6.1.1	Notes	21
6.1.2	Aspect webs	21
6.1.3	House and angle markings	25
6.1.4	Smart labels	25
6.2	Dial Charts	27
6.3	Decorative Wheel Charts	31

7	Advanced Topics	37
7.1	Rounding and Mixed-Base Conversion	37
7.1.1	The rounding problem	37
7.1.2	Automatic rounding	40
7.1.3	Manual rounding modes	40
7.2	Adding Custom Objects	41
7.3	Special Charts	42
7.4	Aspects Between Two Charts	44
7.5	Label Adjustments	46
7.6	Designing New Templates	50
7.6.1	Ticks and keys	50
7.6.2	Conditionals for use in templates	50
7.6.3	Drawing sets of graphical elements	51
7.6.4	Single linework objects	51
7.6.5	For-each and things to put in it	52
7.6.6	Drawing the aspect web	53
8	Implementation	54
8.1	Initial Option Handling	54
8.1.1	Symbol choices	54
8.1.2	Calculation backends	54
8.1.3	Higher-level features	54
8.1.4	Processing the options	55
8.2	Symbols	56
8.2.1	General symbols	56
8.2.2	Text abbreviations	56
8.2.3	Symbols from wasysym	58
8.2.4	Symbols from marvosym	59
8.2.5	Symbols from starfont	59
8.3	Utilities	61
8.3.1	Trimming spaces	61
8.3.2	Redefine after current group	61
8.3.3	Polar to Cartesian	62
8.3.4	For-each	62
8.3.5	Double for-each	63
8.4	Basic Astrological Calculation Routines	63
8.4.1	Cusp information	64
8.4.2	Calculating harmonics	64
8.4.3	Shifting positions	64
8.4.4	Calculating equal cusps	65
8.4.5	Calculating midpoints	65
8.4.6	Angle adjustments	66
8.4.7	Chart rotation	66
8.4.8	Variable copying and saving	67
8.4.9	Sexagesimal arithmetic	68
8.4.10	Setting all data at once	73
8.5	Astrolog Calculation Back End	73
8.5.1	Name translation	73
8.5.2	House systems	74
8.5.3	Interface stuff	75

8.5.4	The actual calculation	75
8.6	Swiss Ephemeris Calculation Back End	78
8.6.1	Command-line options	78
8.6.2	House systems	79
8.6.3	Interface stuff	79
8.6.4	Calculation routines	80
8.7	Low-Level Chart Graphics	81
8.7.1	Configuration stuff	81
8.7.2	Horoscope environment and chart labels	82
8.7.3	Scratch dimension registers	83
8.7.4	Polar-coordinate puts	83
8.7.5	Arrowheads	84
8.7.6	Polar-coordinate curves	85
8.7.7	Locating objects	87
8.7.8	Cusps, ticks, and sign keys	87
8.7.9	Object labels	90
8.8	Aspect Web	94
8.8.1	Configuration settings	95
8.8.2	Per-aspect-type configuration	95
8.8.3	Drawing the aspect web	96
8.9	Internal House Labels	97
8.10	Object and Cusp Adjustment	98
8.10.1	Configuration settings	99
8.10.2	Internal variables	100
8.10.3	Support macros for adjustment iterations	101
8.10.4	Single adjustment cycles	102
8.10.5	Support macros for main loop	104
8.10.6	Main loop	104
8.11	Support Macros for Ready-Made Wheels	106
8.11.1	Recognizing houses	106
8.11.2	Support for Montreal template	107
8.11.3	Support for Quebec City template	108
8.11.4	Support for dial templates	108
8.11.5	Choosing text size	110
8.12	Ready-Made Wheel Templates	110
8.12.1	Basic wheel	110
8.12.2	Harmonic multi-dial charts	112
8.12.3	Decorative wheel charts	114

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.
- 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 multi-dials 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 \mathfrak{V} or \mathfrak{Z} 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 \LaTeX

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.

This 0.91 release fixes some bugs and addresses some non-bug issues brought up by testers. A few remain, so there will probably be at least one more beta release before a more final really-usable release happens.

This package, the earlier starfont package (which horoscop looks for by default), and other material on typesetting astrology with L^AT_EX, are available from <http://ansuz.sooke.bc.ca/astrology/astro-latex.php>.

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. Although the `starfont` package itself is in CTAN, it requires a font file not distributed by CTAN but available from the package author’s Web page. Instead of using an astrological font at all, the `textsym` option may be used to substitute abbreviations written in \LaTeX ’s ordinary text font.

Sources for astrological font packages:

- `marvosym`:
<http://tug.ctan.org/cgi-bin/ctanPackageInformation.py?id=marvosym>
- `starfont` (author’s Web site):
<http://ansuz.sooke.bc.ca/astrology/astro-latex.php>
- `starfont` (CTAN):
<http://tug.ctan.org/cgi-bin/ctanPackageInformation.py?id=starfont>
- `wasysym`:
<http://tug.ctan.org/cgi-bin/ctanPackageInformation.py?id=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 `epic`, `eepic`, and `trig`; all these *are* typically included in \LaTeX installations by default. Use of `eepic` has an important consequence: you can’t typeset charts properly with `pdflatex` because it doesn’t support the “specials” `eepic` uses to draw large circles and arbitrary-slope lines. The current workaround is to use `dvips` and translate the resulting file to PDF. A future version may be able to work with `pdflatex` more directly, but it won’t be trivial; large circles and arbitrary-slope lines really are absolutely necessary for most wheel chart designs.

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: <http://www.astrolog.org/astrolog.htm>

- Swiss Ephemeris: <http://www.astro.com/swissep/>

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`, `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.

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

The macros `\horotextsym`, `\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 `\horotextsym` which 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 horoscop 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.

`\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

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.

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 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

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

from other fonts or for instance if it's desired to use $\text{\textcircled{z}}$ instead of $\text{\textcircled{v}}$ 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 $\text{\langle object \rangle Symbol}$ 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 $\text{\Zodiac}\{\langle sign \rangle\}$ 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 \Zodiac calls the appropriate macro from $\text{\AriesSymbol}\dots\text{\PiscesSymbol}$, allowing the user to redefine individual sign symbols without needing to redefine all of \Zodiac . Note that all use of sign symbols by horoscop (for instance, in chart wheel labels) goes through \Zodiac , so it is possible to redefine the entire zodiac by redefining \Zodiac instead of redefining individual symbols.

macro	textsym	wasysym	marvosym	starfont
\AriesSymbol	Ar	$\text{\textcircled{a}}$	$\text{\textcircled{A}}$	$\text{\textcircled{A}}$
\TaurusSymbol	Ta	$\text{\textcircled{t}}$	$\text{\textcircled{T}}$	$\text{\textcircled{T}}$
\GeminiSymbol	Ge	$\text{\textcircled{g}}$	$\text{\textcircled{G}}$	$\text{\textcircled{G}}$
\CancerSymbol	Cn	$\text{\textcircled{c}}$	$\text{\textcircled{C}}$	$\text{\textcircled{C}}$
\LeoSymbol	Le	$\text{\textcircled{l}}$	$\text{\textcircled{L}}$	$\text{\textcircled{L}}$
\VirgoSymbol	Vi	$\text{\textcircled{v}}$	$\text{\textcircled{V}}$	$\text{\textcircled{V}}$
\LibraSymbol	Li	$\text{\textcircled{L}}$	$\text{\textcircled{L}}$	$\text{\textcircled{L}}$
\ScorpioSymbol	Sc	$\text{\textcircled{s}}$	$\text{\textcircled{S}}$	$\text{\textcircled{S}}$
$\text{\SagittariusSymbol}$	Sg	$\text{\textcircled{S}}$	$\text{\textcircled{S}}$	$\text{\textcircled{S}}$
\CapricornSymbol	Cp	$\text{\textcircled{c}}$	$\text{\textcircled{C}}$	$\text{\textcircled{V}}$
\AquariusSymbol	Aq	$\text{\textcircled{a}}$	$\text{\textcircled{A}}$	$\text{\textcircled{A}}$
\PiscesSymbol	Pi	$\text{\textcircled{p}}$	$\text{\textcircled{P}}$	$\text{\textcircled{P}}$

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

macro	textsym	wasysym	marvosym	starfont
\SunSymbol	Su	$\text{\textcircled{S}}$	$\text{\textcircled{S}}$	$\text{\textcircled{S}}$
\MoonSymbol	Mo	$\text{\textcircled{M}}$	$\text{\textcircled{M}}$	$\text{\textcircled{M}}$
\MercurySymbol	Me	$\text{\textcircled{m}}$	$\text{\textcircled{m}}$	$\text{\textcircled{m}}$
\VenusSymbol	Ve	$\text{\textcircled{v}}$	$\text{\textcircled{v}}$	$\text{\textcircled{v}}$
\MarsSymbol	Ma	$\text{\textcircled{M}}$	$\text{\textcircled{M}}$	$\text{\textcircled{M}}$
\JupiterSymbol	Ju	$\text{\textcircled{J}}$	$\text{\textcircled{J}}$	$\text{\textcircled{J}}$
\SaturnSymbol	Sa	$\text{\textcircled{S}}$	$\text{\textcircled{S}}$	$\text{\textcircled{S}}$
\UranusSymbol	Ur	$\text{\textcircled{U}}$	$\text{\textcircled{U}}$	$\text{\textcircled{U}}$
\NeptuneSymbol	Ne	$\text{\textcircled{N}}$	$\text{\textcircled{N}}$	$\text{\textcircled{N}}$
\PlutoSymbol	Pl	$\text{\textcircled{P}}$	$\text{\textcircled{P}}$	$\text{\textcircled{P}}$

§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. Cusp symbols are listed here too, although they gen-

erally do not appear in the default templates and will seldom be used in actual practice.

macro	textsym	starfont
<code>\ChironSymbol</code>	Chi	♄
<code>\CeresSymbol</code>	Cer	♆
<code>\PallasSymbol</code>	Pal	♃
<code>\JunoSymbol</code>	Jun	♅
<code>\VestaSymbol</code>	Ves	♁
<code>\NorthNodeSymbol</code>	No	♁
<code>\SouthNodeSymbol</code>	SNo	♁
<code>\LilithSymbol</code>	Lil	♁
<code>\CuspISymbol</code>	ASC	A ^{SC}
<code>\CuspIISymbol</code>	2 rd	
<code>\CuspIIISymbol</code>	3 rd	
<code>\CuspIVSymbol</code>	IC	I ^C
<code>\CuspVSymbol</code>	5 th	
<code>\CuspVISymbol</code>	6 th	
<code>\CuspVIISymbol</code>	DSC	D ^{SC}
<code>\CuspVIIISymbol</code>	8 th	
<code>\CuspIXSymbol</code>	9 th	
<code>\CuspXSymbol</code>	MC	M ^C
<code>\CuspXISymbol</code>	11 th	
<code>\CuspXIISymbol</code>	12 th	
<code>\AscendantSymbol</code>	ASC	A ^{SC}
<code>\MCSymbol</code>	MC	M ^C
<code>\VertexSymbol</code>	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 (Ω) 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`
`\TrueNodeSymbol`

There are no well-agreed standard symbols for “mean” as opposed to “true” (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 de-

fine 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	wasysym	starfont	angle
<code>\horoConjunctionSymbol</code>	Con	♌	♌	0° ± 6°
<code>\horoOppositionSymbol</code>	Opp	♌ ^o	♌ ^o	180° ± 6°
<code>\horoTrineSymbol</code>	Tri	△	△	120° ± 5°
<code>\horoSquareSymbol</code>	Sqr	□	□	90° ± 5°
<code>\horoQuintileSymbol</code>	Qnt		Q	72° ± 2°
<code>\horoBiquintileSymbol</code>	Bqi		Q ²	144° ± 2°
<code>\horoSextileSymbol</code>	Sex	*	✳	60° ± 4°
<code>\horoQuincunxSymbol</code>	Qcx		⌘	150° ± 3°
<code>\horoSemisextileSymbol</code>	Ssx		⌞	30° ± 3°
<code>\horoSemisquareSymbol</code>	Ssq		⌟	45° ± 2°
<code>\horoSesquiquadrateSymbol</code>	Sqq		⌞	135° ± 2°

`\horodegrees` **§2.6. Text angle and direction symbols.**—Symbols for degrees, minutes, and seconds (`\horodegrees`, `\horominutes`, and `\horoseconds`) are provided for use when an angle will be described numerically, as in 12°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



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 whole whole numbers only 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<system>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 http://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 T_EX 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 directly, 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\horoCuspI{207.9120843}
\def\horoCuspII{236.5553269}
\def\horoCuspIII{267.6976404}
\def\horoCuspIV{300.5365877}
\def\horoCuspV{332.9973490}
\def\horoCuspVI{2.5292853}
\def\horoCuspVII{27.9120843}
\def\horoCuspVIII{56.5553269}
\def\horoCuspIX{87.6976404}
\def\horoCuspX{120.5365877}
\def\horoCuspXI{152.9973490}
\def\horoCuspXII{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.

```

\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 T_EX'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 fea-

tures 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 $19^{\circ}\text{♍}39'40''$.

`\horodsmstext`

`\horotimetext`

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`.

`\horolatlonstext`

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 $50^{\circ}0'0''$ N and $40^{\circ}0'0''$ W.

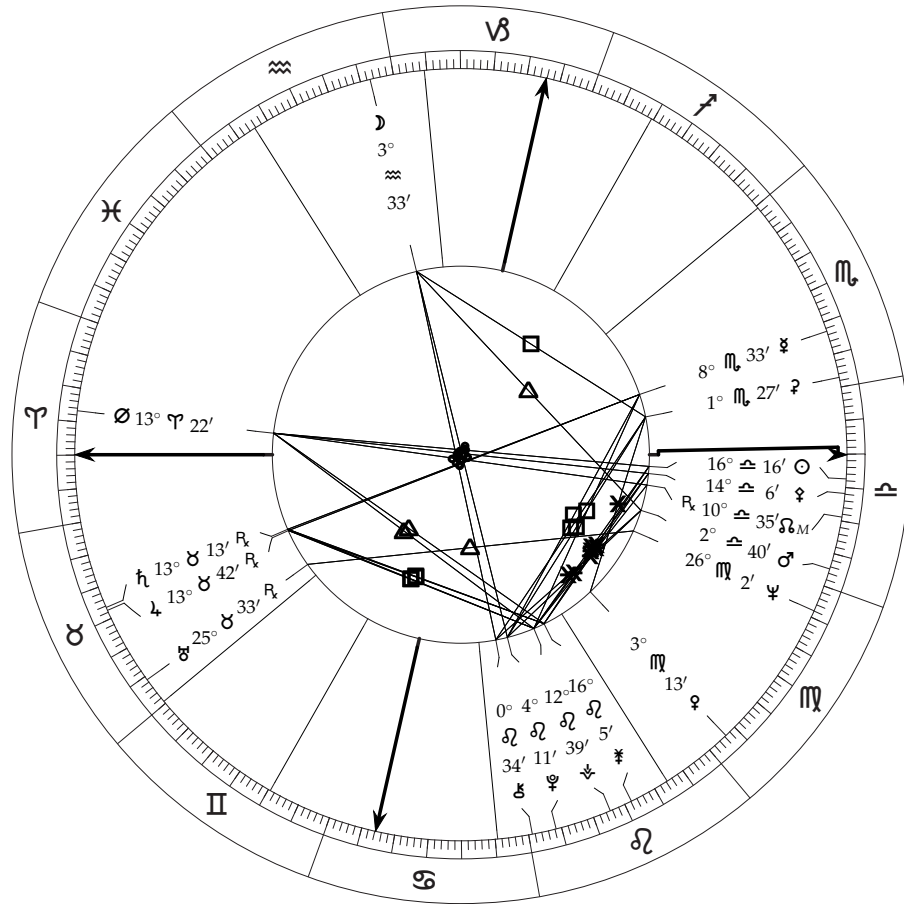
Chapter Six

READY-MADE CHART TEMPLATES



1. BASIC WHEEL CHARTS

<code>horoscope</code>	The package provides several ready-made chart templates for different purposes.
<code>\horowheelVancouver</code>	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 <code>\horowheelVancouver</code> , as in the birth chart for John Lennon shown in Figure 1.
<code>\horounitlength</code>	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 <code>\horounitlength</code> , which defaults to 0.00952 times <code>\textwidth</code> , 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.
<code>\horoULnote</code>	§1.1. Notes. —Five macros are provided for typesetting notes in the corners or centre of a wheel chart. They are called <code>\horoULnote</code> , <code>\horoURnote</code> , <code>\horoLLnote</code> , <code>\horoLRnote</code> , and <code>\horoCnote</code> , 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 <code>\\</code> 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.
<code>\horoURnote</code>	
<code>\horoLLnote</code>	
<code>\horoLRnote</code>	
<code>\horoCnote</code>	
<code>\horoaspectwebtrue</code>	§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 <i>aspect web</i> . It can be turned on and off with <code>\horoaspectwebtrue</code> and <code>\horoaspectwebfalse</code> .
<code>\horoaspectwebfalse</code>	
<code>\horoaspectobjectsa</code>	By default, the aspect web will show aspects between any two objects from <code>\horoobjects</code> . 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 displaying in the web. You can override the default to show only selected aspects by defining new values for <code>\horoaspectobjectsa</code> and <code>\horoaspectobjectsb</code> . 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.
<code>\horoaspectobjectsb</code>	
<code>\horoaspects</code>	The <code>\horoaspects</code> macro stores a list of aspects to include in aspect webs, much in the manner that <code>\horoobjects</code> 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



```

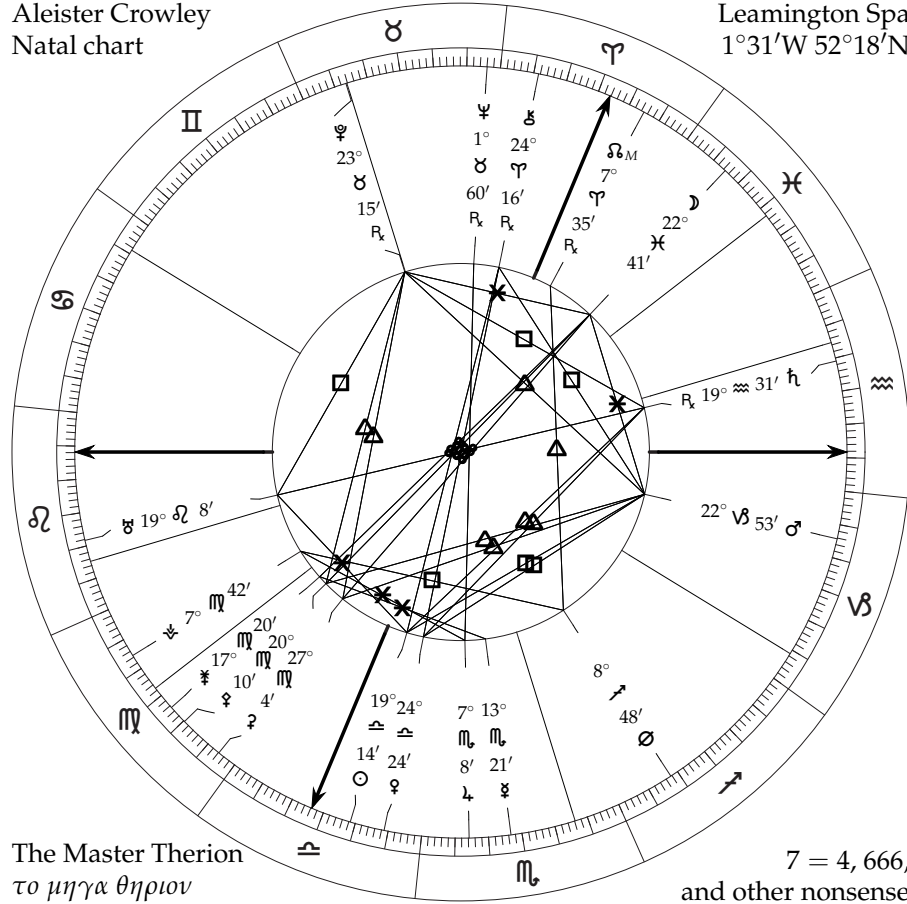
\horocalparms{1940}{10}{9}{17:30:0}{W2:55:0}{N53:25:0}
\horocalculate
\begin{horoscope}
\horowheelVancouver
\end{horoscope}

```

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

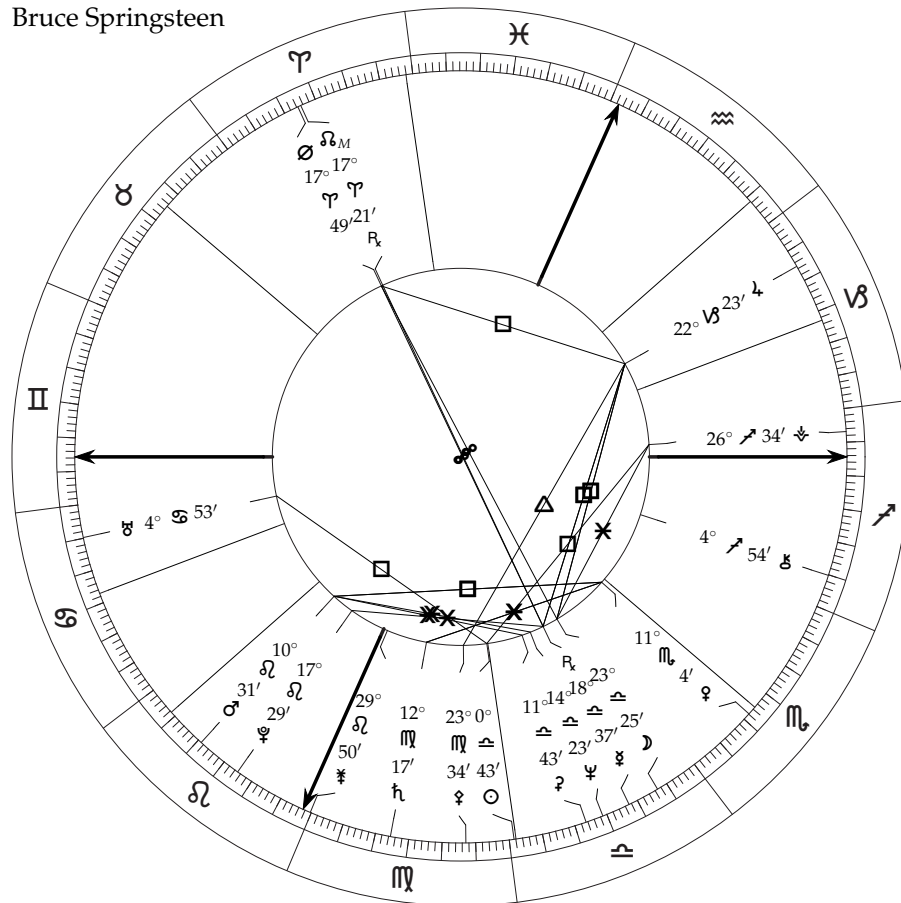
```

\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 o \mu eta \gamma ma \alpha \ \theta eta \eta rho \iota ota o \nu $}
\horoLRnote{$7=4$; 666; \\and other nonsense}
\end{horoscope}

```

Figure 2: Corner notes.

Bruce Springsteen



```

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

```

Figure 3: Limiting the objects for the aspect web.

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.

`\horoinhouselabelstrue`
`\horoinhouselabelsfalse`
`\horohouselabel`

Internal house labels are numbers that appear inside the houses among the object labels, selected with the `\horoinhouselabelstrue` and `\horoinhouselabelsfalse` 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`
`\horoanglecuswidth`

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 `\horoanglecuswidth`, 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.

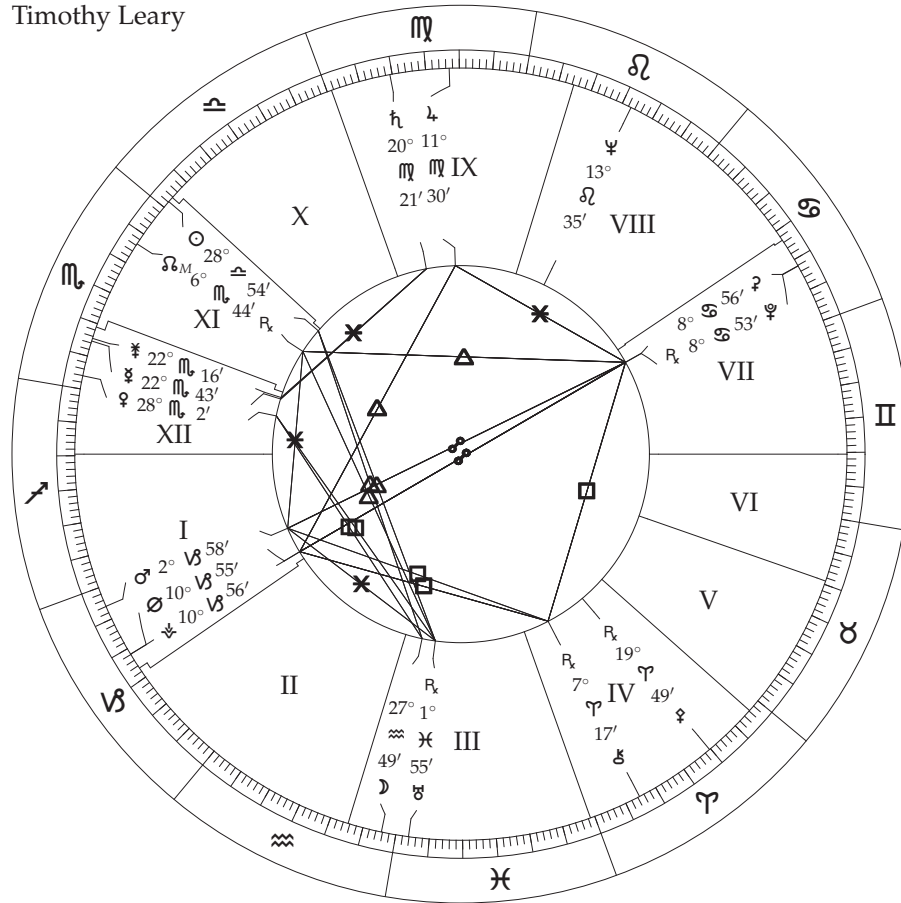
First, the `\horowheelVancouver` command can take an optional argument which is a format string specifying what “chunks” should be typeset in each object label. The format string should be some sequence of the letters `d`, `m`, `s`, `z`, `y`, and `r`, which refer respectively to Degrees, Minutes, and Seconds of longitude, Zodiac sign symbol, object `sYmbol`, and possible Retrograde. If a label string like `yzmsr` 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 `yzmzr/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

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

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.

`\horowheelIqaluit`

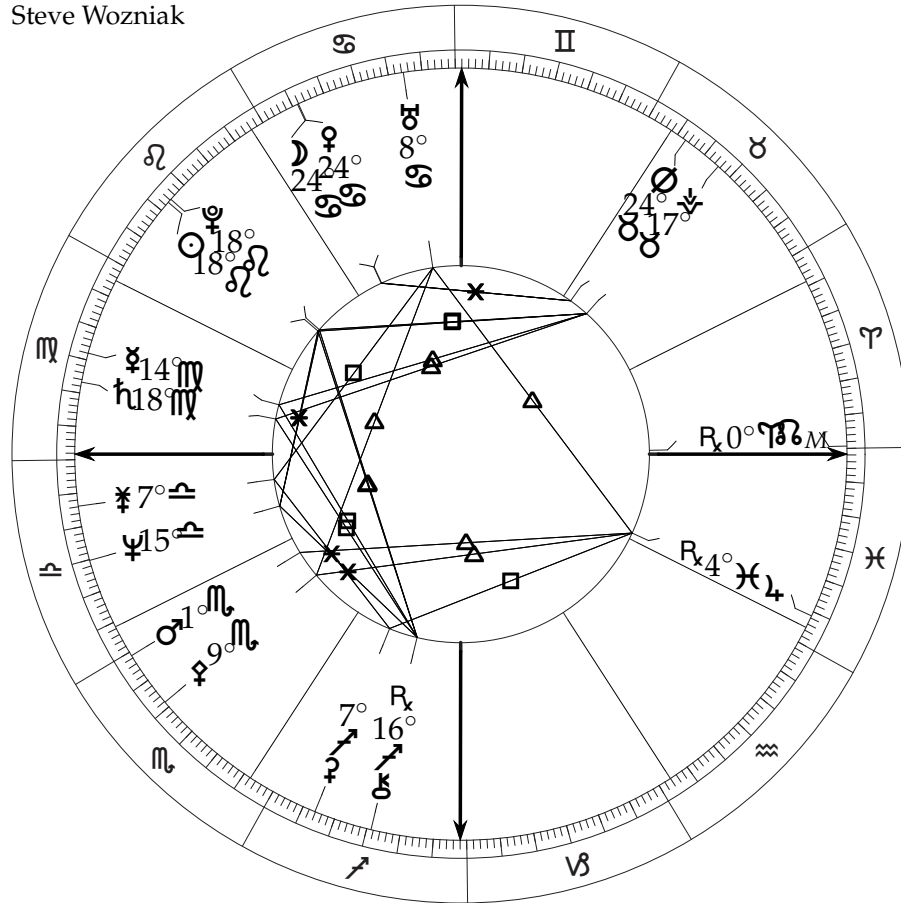
Figure 6 shows the basic single 360° dial chart style, available through the `\horowheelIqaluit` macro. This macro, like all the dial chart macros, has an optional first argument for a harmonic number. The popular 90° dial is available with `\horowheelIqaluit[4]`, as shown in Figure 7. Other harmonics 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

Steve Wozniak



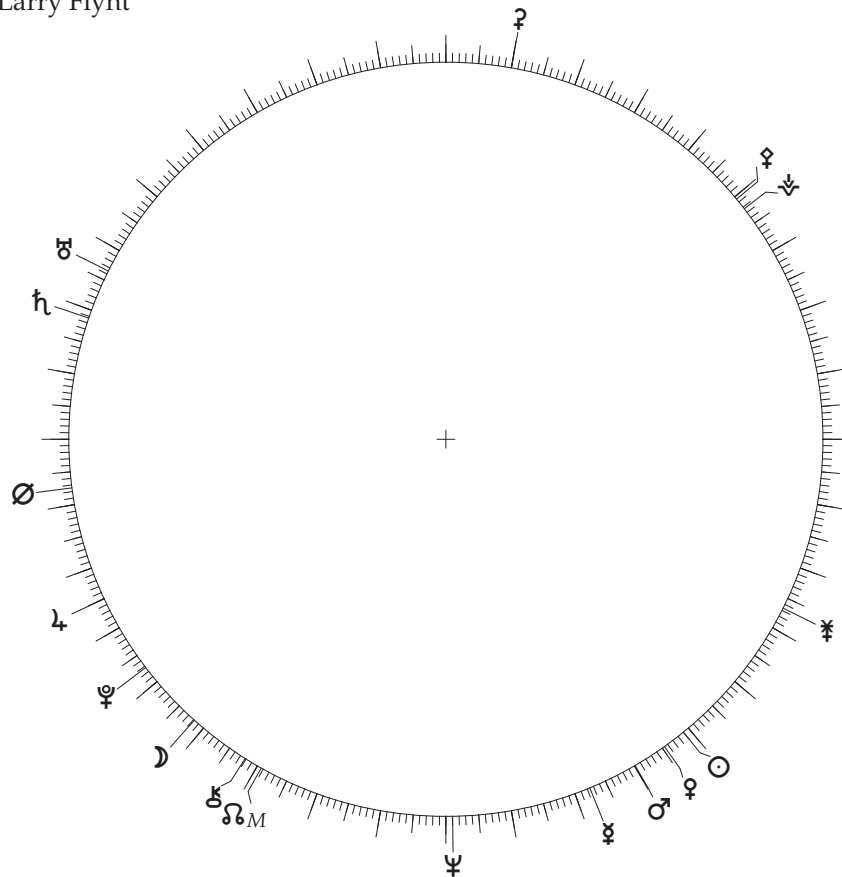
```

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

```

Figure 5: Custom label string and text size adjustment.

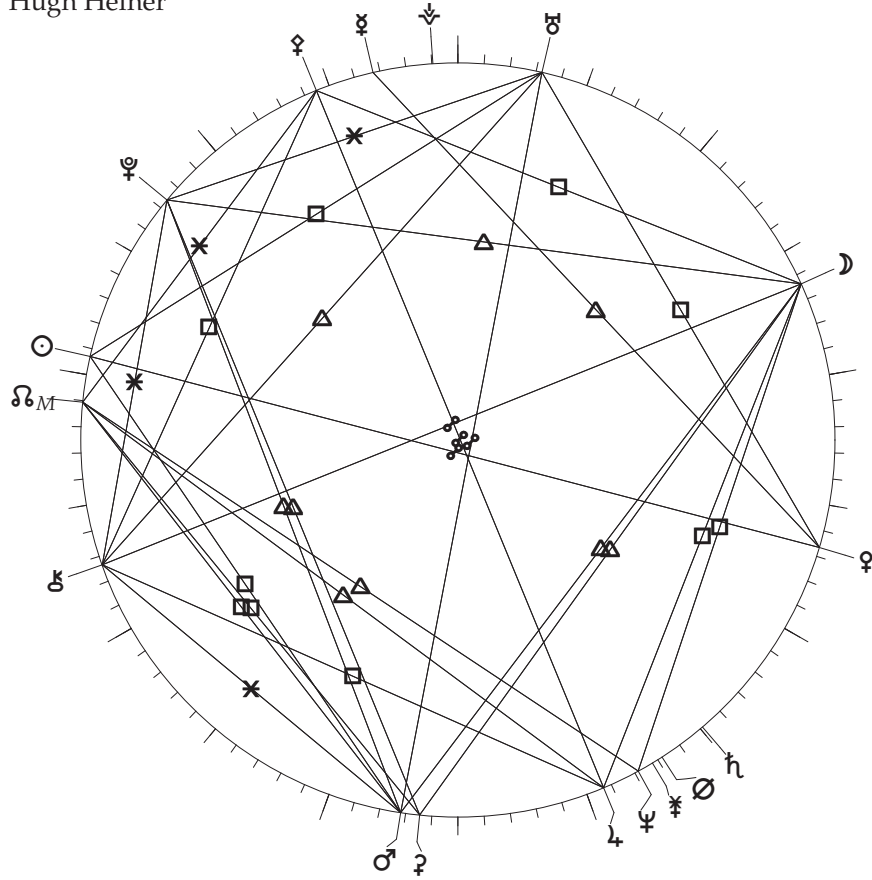
Larry Flynt



```
\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).

Hugh Hefner



```
\horocalparms{1926}{4}{9}{22:20:0}{W87:39:0}{N41:52:0}
\horocalculate
\begin{horoscope}
  \horowheelIqaluit[4]
  \horoULnote{Hugh Hefner}
\end{horoscope}
```

Figure 7: Single 90° dial (Iqaluit).

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`

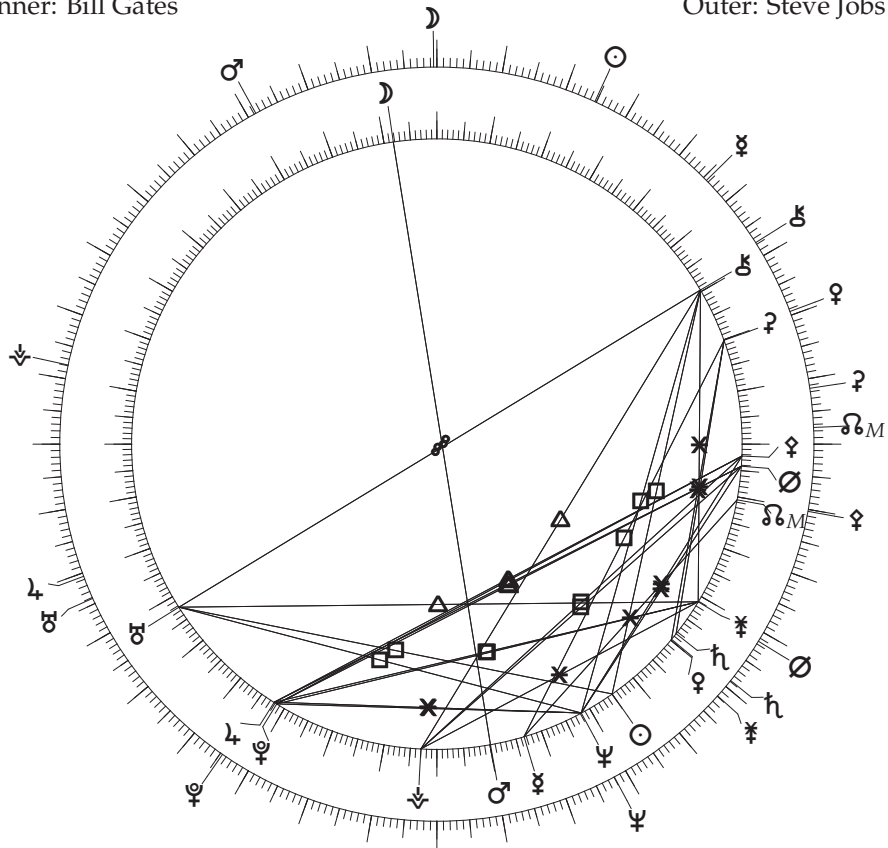
The `\horowheelMontreal` macro generates a chart like that shown in Figure 11. It shows labels for the angular cusps down to the minute of arc and for objects showing object symbol, degree and sign, and possible retrograde. Most of the house cusps are drawn as curves, creating a floral effect. This template 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).

Inner: Bill Gates

Outer: Steve Jobs

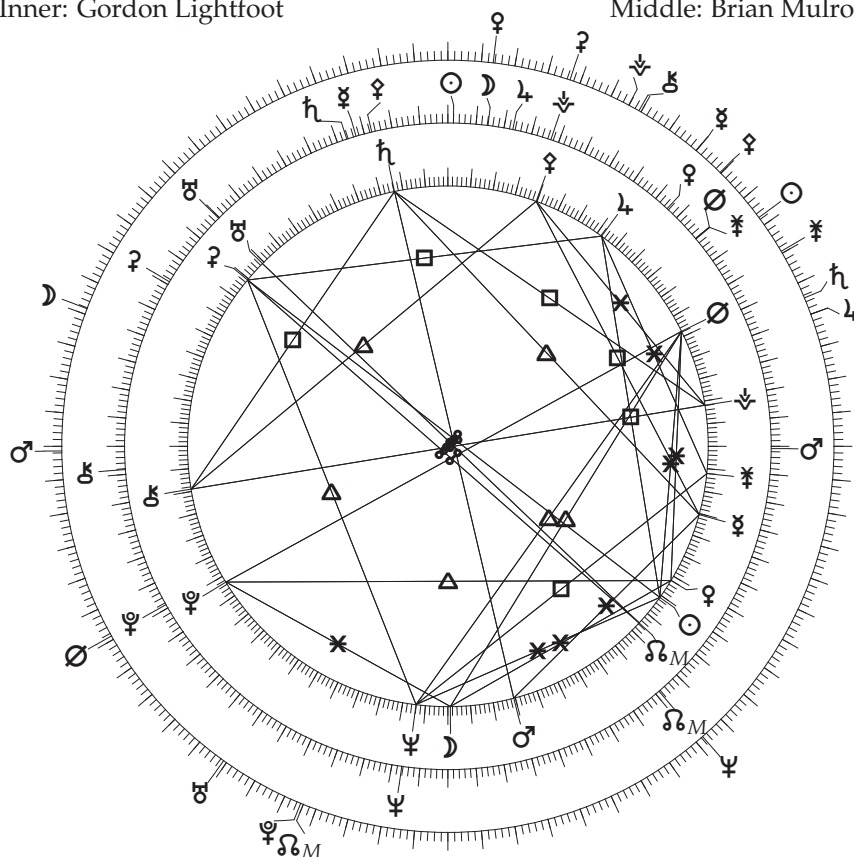


```
\horocalparms{1955}{2}{24}{15:0:0}{W122:25:0}{N37:46:0}  
\horocalculate\horosaveobjects{\SteveJobs}  
\horocalparms{1955}{10}{29}{6:0:0}{W122:20:0}{N47:36:0}  
\horocalculate\horosaveobjects{\BillGates}  
\begin{horoscope}  
  \horowheelIglolik{\BillGates}{\SteveJobs}  
  \horoULnote{Inner: Bill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

Figure 8: Double 360° dial (Iglolik).

Inner: Gordon Lightfoot

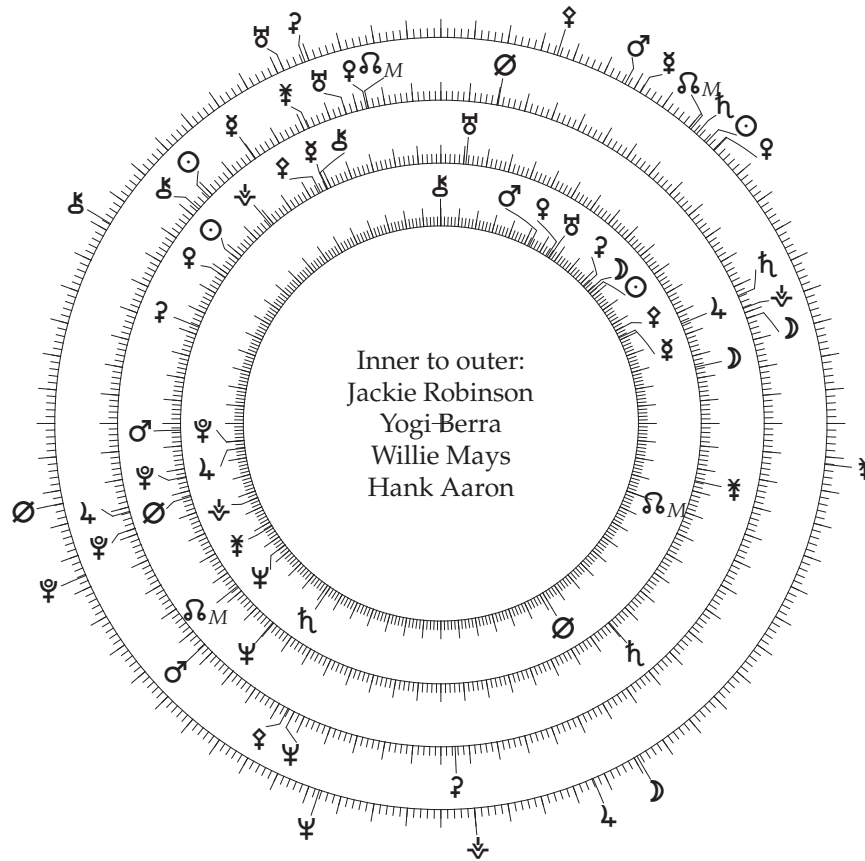
Middle: Brian Mulroney



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).



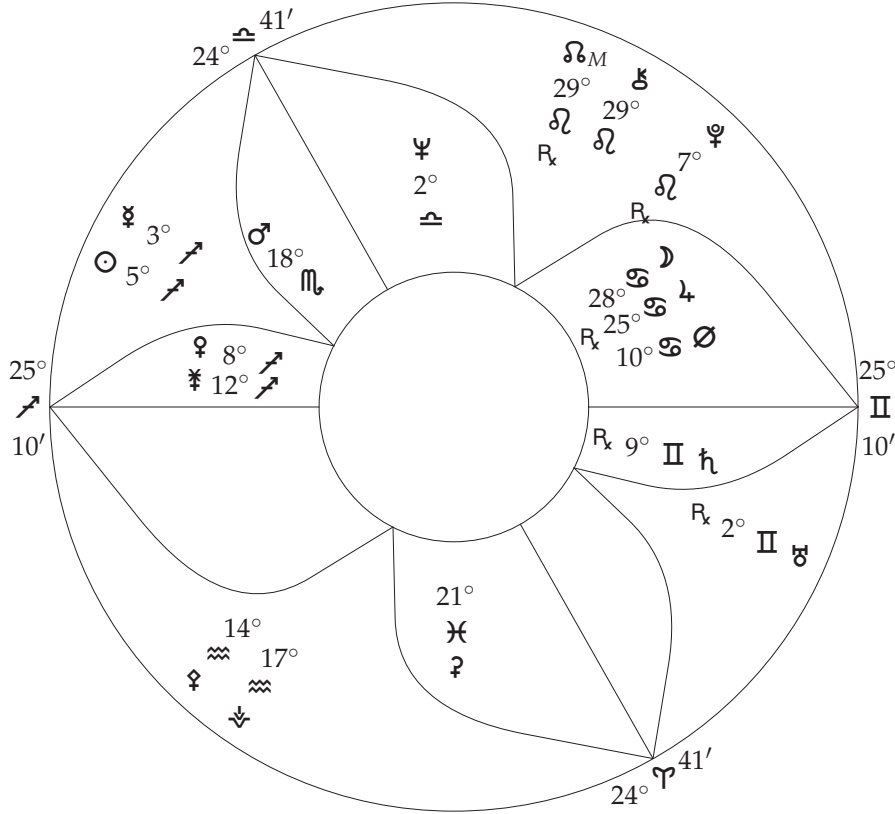
```

\horocalcparms{1934}{2}{6}{2:25:0}{W88:03:0}{N30:41:0}
\horocalculate\horosaveobjects{\HankAaron}
\horocalcparms{1931}{5}{7}{4:30:0}{W86:55:0}{N33:28:0}
\horocalculate\horosaveobjects{\WillieMays}
\horocalcparms{1925}{5}{12}{8:0:0}{W90:12:0}{N38:37:0}
\horocalculate\horosaveobjects{\YogiBerra}
\horocalcparms{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).

Jimi Hendrix



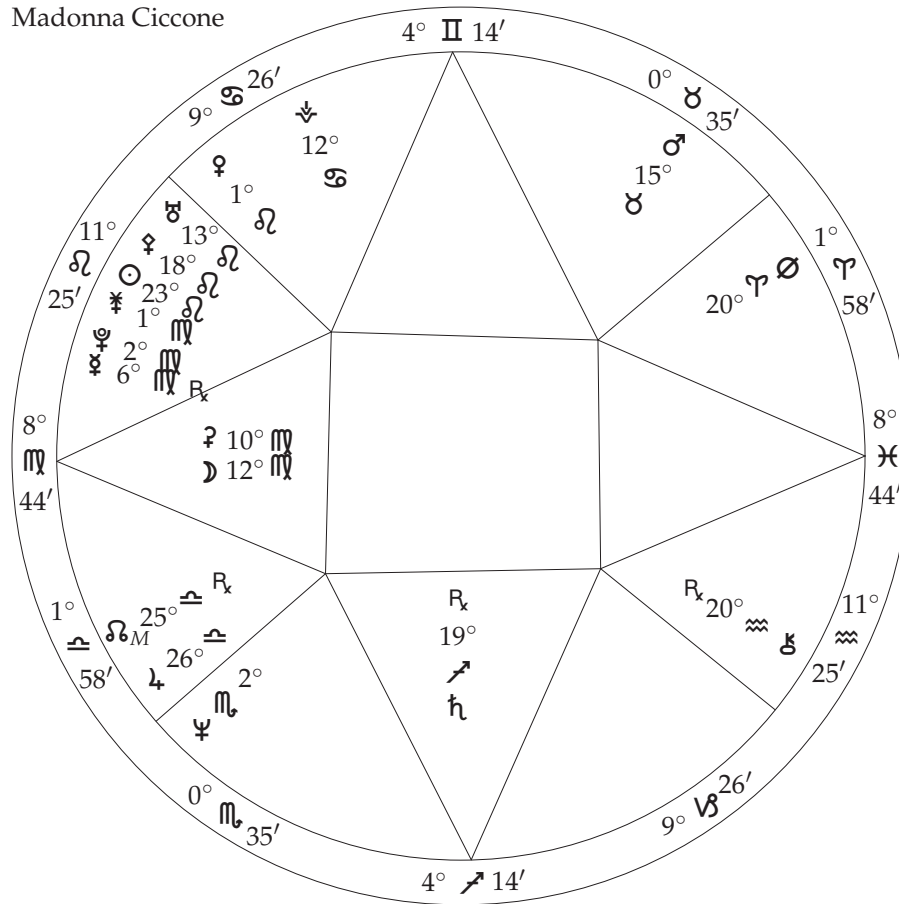
```

\horocalcparms{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



```

\horocalcparms{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 $2^{\circ}\mathcal{P}34'56''$. 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 $2^{\circ}\mathcal{P}34'56''$ 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 $0^{\circ}\mathcal{S}0'0''$. That misrepresents the position in an important way: the object has not yet entered 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

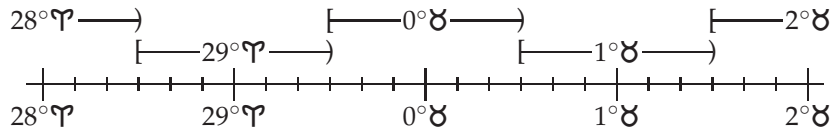


Figure 13: Pure rounding to nearest degree.

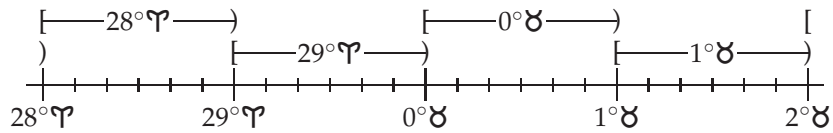


Figure 14: Truncation to degree.

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 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

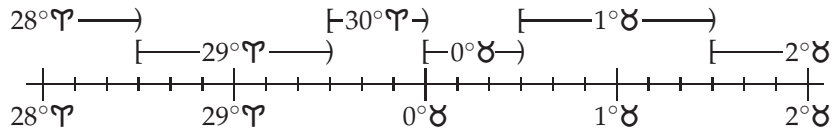


Figure 15: Rounding without carry.

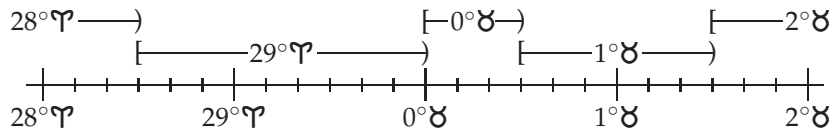


Figure 16: Rounding with truncation at end of sign.

$1^\circ \dots 29^\circ$ 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 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;³ 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 horoscop 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 of precision in the particular label.

`\ifhororoundauto` 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.

§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. How-

³The 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, <http://www.math.rutgers.edu/~zeilberg/mamarim/mamarimhtml/real.html>.

`\hororoundstrictruncate`

ever, if for some reason a really strict truncation with no offset is desired, that can be selected with `\hororoundstrictruncate`. 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.

`\hororoundtosec`

`\hororoundtomin`

`\hororoundtodeg`

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.

`\hororoundtoseckeepsign`

`\hororoundtoseckeepdeg`

`\hororoundtoseckeepmin`

`\hororoundtominkeepsign`

`\hororoundtominkeepdeg`

`\hororoundtodegkeepsign`

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.

`\hororoundclamptrue`

`\hororoundclampfalse`

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 `\hororoundautotru`. Turn it off with `\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 `-xs999001` (planet number 999001⁴). Other asteroid-like objects would be designated similarly; see the `swetest` documentation for how to select other kinds of objects.

```
\newcommand{\horoErisSEOpt}{-ps -xs999001}
```

⁴For Swiss Ephemeris version 1.70 as installed on the author’s system; in some future version they will probably switch to using the MPC number 136199 now that one has been assigned.

The chart plotting system also needs a `\langle object \rangle 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 `\horobjects` 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 `\horocalparms` command and proceed as with any other chart.

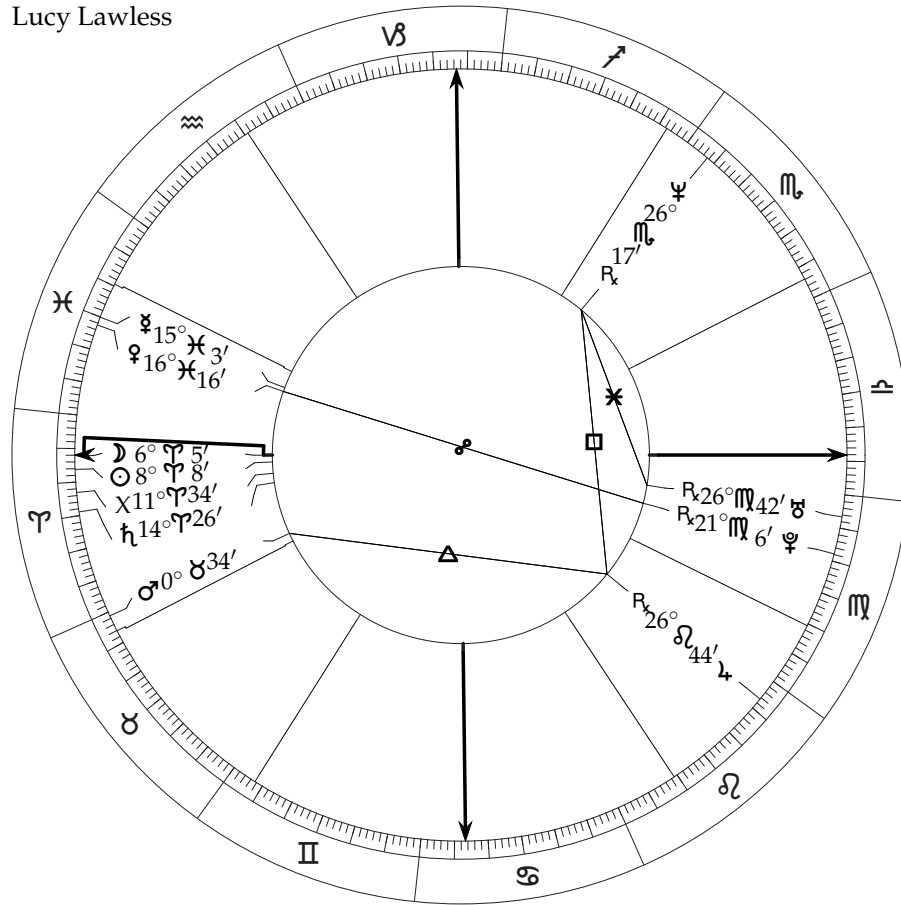
Transits, Solar returns, horary, etc.: These charts are also just ordinary charts for special times and locations, so they can be calculated normally given the right data. There is currently no special support for automatically calculating the data to use, for instance to get the exact time of a Solar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

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

Lucy Lawless



```

\horocalparms{1968}{3}{28}{18:25:0}{E174:46:0}{S36:52:0}
\newcommand{\horoErisSEOpt}{-ps -xs999001}
\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.

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 `\horocalcparms` 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 people⁵ 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.

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.⁶

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.

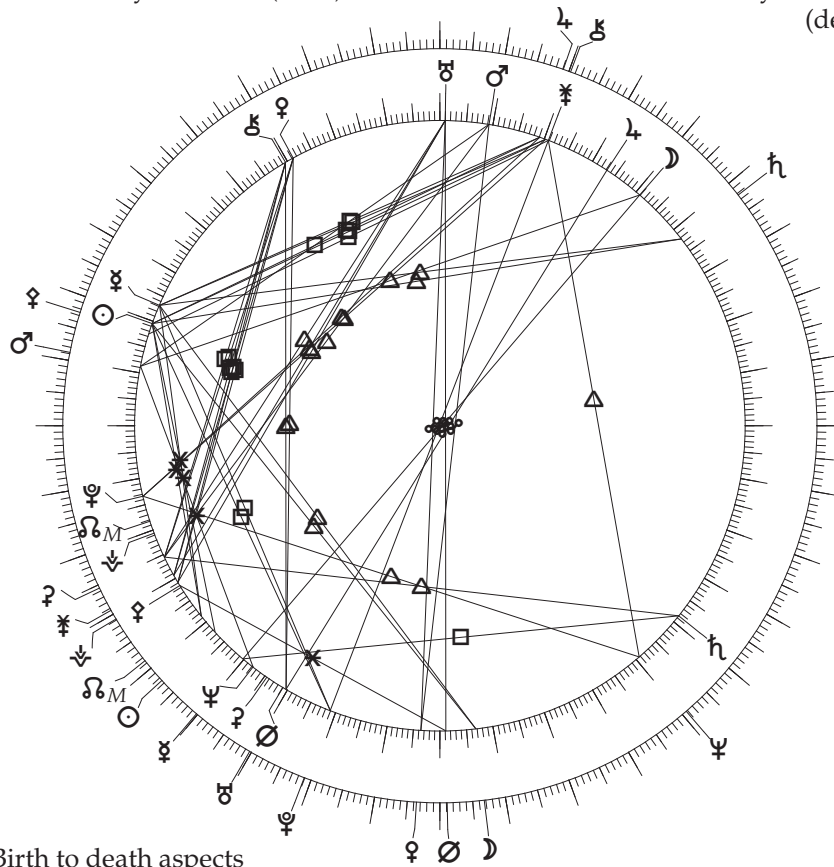
Then we redefine `\horoaspectobjectsa` to be the list of default objects with X appended to each of their names. That’s the clever bit. When the aspect web attempts to look up the Pos of an object like the Sun, it will look in a macro called `\horoSunXPos`, which is the copied value of Pos from the Sun 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.

⁵For instance, based on an inspection of the source code this seems to be what Astrolog does.

⁶The 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{\horooobjects}{Pos}{XPos}
\horocalparms{1926}{6}{1}{17:30:0}{W118:15:0}{N34:04:0}
\horocalculate\horosaveobjects{\MarilynBirth}
\horocalculate
\begin{horoscope}
  \renewcommand\horoaspectobjects%
    {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.

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

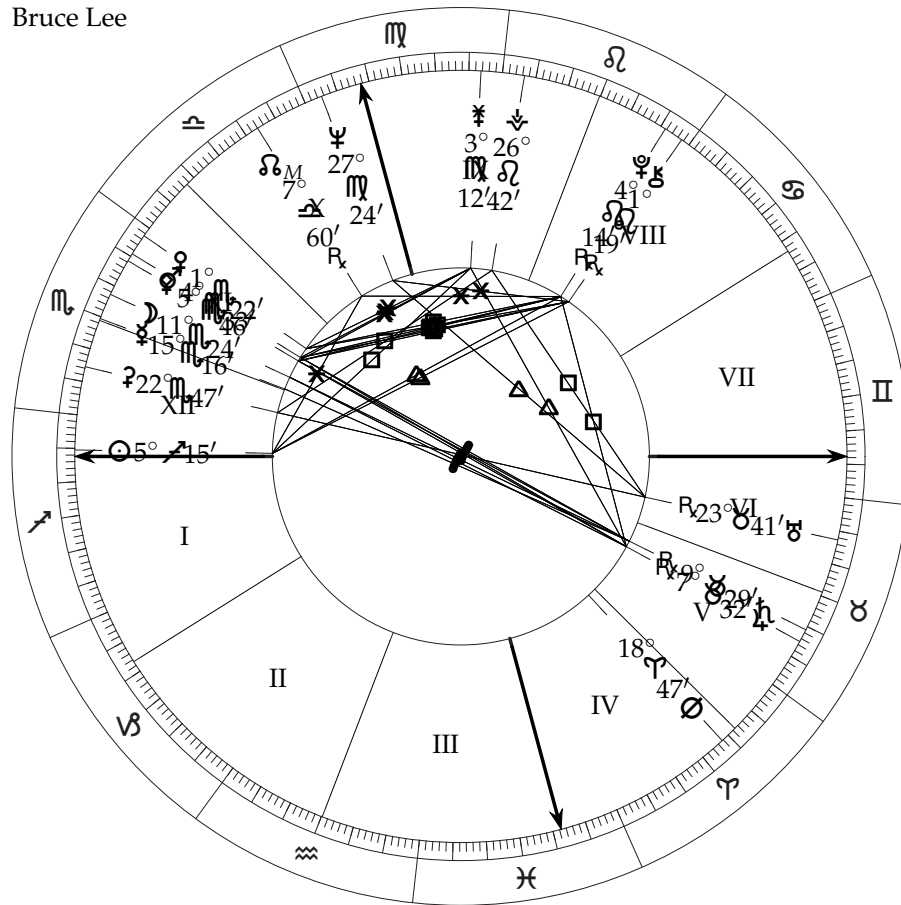
`\horoomindist` The minimum distance targets in degrees are set by `\horoomindist` (note
`\horocmindist` triple o in name) and `\horocmindist`, for “\horo object-object minimum distance” and “\horo object-cusp minimum distance” respectively. The defaults are 6° object to object and 4° object to cusp.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

`\horoposattobj` The sizes of the adjustments (or strengths of the springs) are determined by
`\horoposattcusp` three divisors applied to the distances between things and where they should
`\hororepulsion` 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

Bruce Lee



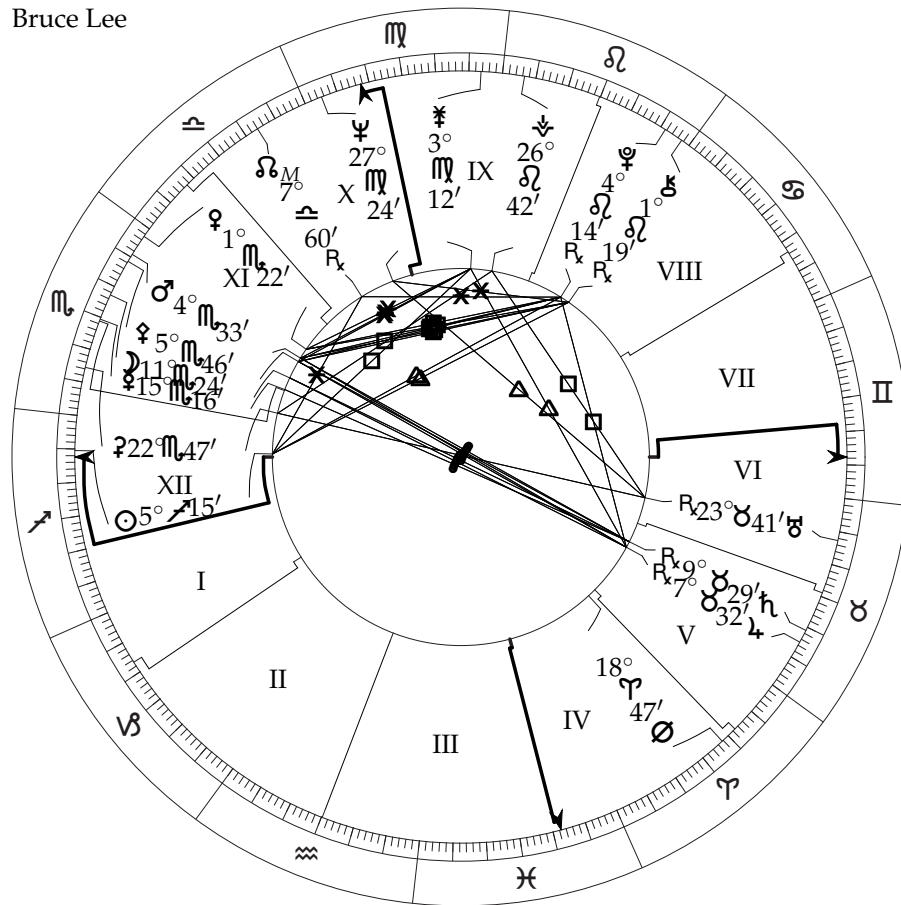
```

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

```

Figure 19: Labels printed at their longitudes without adjustment.

Bruce Lee



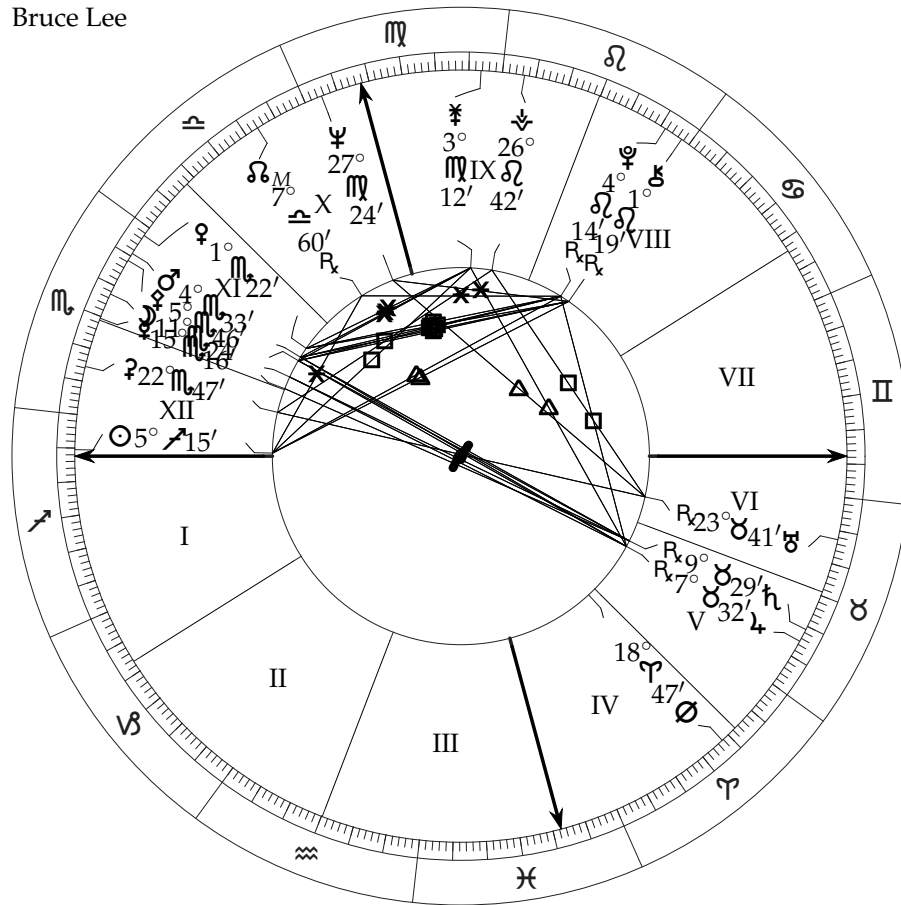
```

\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{\horooomindist}{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.

`\horosignificantadjust` 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 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° 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`, `\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, `\ifhoroathouselabels` 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 `horohouse` 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⟩}`-

$\langle\theta_2\rangle$, draws a line between any two points given in polar coordinates.

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

`\horoputcurve` The `\horoputcurve{\langle r1\rangle}\langle\theta_1\rangle\langle r2\rangle\langle\theta_2\rangle` 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.

`\horoforeach` **§6.5. For-each and things to put in it.**—To run a macro on every object in a comma-separated list, call `\horoforeach{\langle list\rangle}\langle macro\rangle`. The contents of $\langle macro\rangle$ will be invoked once for each comma-separated item in $\langle list\rangle$, with the item (in curly braces) added to the end. For instance, an invocation of `\horoforeach{x,y,z}\foo{a}` would call `\foo{a}{x}`, `\foo{a}{y}`, and `\foo{a}{z}`. 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 `\horoforeach`.

`\horoconncurve` The `\horoconncurve{\langle r1\rangle}\langle r2\rangle\langle object\rangle` macro draws a connecting curve showing the relationship between the Pos and DPos of an object. This would typically be used after `\horoadjust` 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 $\langle r1\rangle$ and the object's Pos, to radius $\langle r2\rangle$ and its DPos.

`\horoputcusplabel` The `\horoputcusplabel{\langle radius\rangle}\langle spacing\rangle\langle object\rangle` 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 $\langle r\rangle$ with the sign symbol at the object's DPos and the other two things arranged $\langle spacing\rangle$ 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.

`\horoputobjsymbol` The `\horoputobjsymbol{\langle radius\rangle}\langle object\rangle` places an object's symbol at its DPos and the specified radius

`\horoputobjdeglabel`
`\horoputobjminlabel`
`\horoputobjseclabel` There are three macros called `\horoputobjdeglabel`, `\horoputobjminlabel`, and `\horoputobjseclabel` 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 $\{\langle radius\rangle\}\langle spacing\rangle\{\langle object\rangle\}$, where $\langle radius\rangle$ is the radius on which to place the centre of the innermost label chunk and $\langle spacing\rangle$ is the spacing between successive chunks outward from there.

`\horoputrxlabel` The `\horoputrxlabel{\langle radius\rangle}\langle object\rangle` macro typesets an optional retrograde label for the object at the object's DPos and radius $\langle radius\rangle$: if the object's Vel is negative then the label will appear as the value of `\horoRetrogradeSymbol` and otherwise it will be blank. A similar function for use in text (possibly in

`\hororxtext` tables showing numeric data) is provided by `\hororxtext{⟨object⟩}`.

`\horoputsmartlabel` The `\horoputsmartlabel{⟨object⟩}` 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.

`\horoscanlabels` First, `\horoscanlabels{⟨string⟩}` takes one or two label format specifiers (separated by a slash if there are two) just as documented for the Vancouver template.

`\horosetsmartradii` After calling that to set the format strings, use `\horosetsmartradii{⟨outer⟩}{⟨stepbase⟩}{⟨stepadj⟩}` to set the radii for the chunks of the label. The outermost chunk will be set with its centre at radius $⟨outer⟩$. Successive chunks will be set inside that at a spacing of $⟨stepbase⟩ - n \times ⟨stepadj⟩$, 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` The macro `\horoputaspect {⟨radius⟩}{⟨theta1⟩}{⟨theta2⟩}{⟨symbol⟩}` draws a single aspect consisting of a line connecting the points at radius $⟨radius⟩$ and longitudes $⟨theta1⟩$ and $⟨theta2⟩$. The contents of $⟨symbol⟩$ will be typeset on the midpoint of the line.

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

`\horoautoaspects` Finally, `\horoautoaspects{⟨radius⟩}` (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⟨aspect⟩Angle` and `\horo⟨aspect⟩Orb` macros.

Chapter Eight

IMPLEMENTATION



We start by declaring that this is the start of a $\text{\LaTeX}2\epsilon$ package and giving it a name.

```
1 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
2 \ProvidesPackage{horoscop}%
3   [2008/08/30 v0.91 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@wasysym\horo@wasysymfalse
6 \newif\ifhoro@marvosym\horo@marvosymfalse
7 \newif\ifhoro@starfont\horo@starfontfalse
8 \DeclareOption{textsym}{\horo@textsymtrue}
9 \DeclareOption{wasysym}{\horo@wasysymtrue}
10 \DeclareOption{marvosym}{\horo@marvosymtrue}
11 \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.

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

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

```
17 \newif\ifhoro@egrep\horo@egrepfalse
18 \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.

```
19 \newif\ifhoro@wheels\horo@wheelstrue
20 \newif\ifhoro@templates\horo@templatestrue
21 \DeclareOption{nowheels}{\horo@wheelsfalse\horo@templatesfalse}
22 \DeclareOption{notemplates}{\horo@templatesfalse}
```

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

```
23 \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.

```
24 \ifhoro@textsym\else
25   \ifhoro@wasysym\else
26     \ifhoro@marvosym\else
27       \horo@starfonttrue
28     \fi
29   \fi
30 \fi
```

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.

```
31 \ifhoro@wasysym
32   \RequirePackage{wasysym}%
33   \let\horow@sySquare\Square
34 \fi
35 \ifhoro@marvosym
36   \RequirePackage{marvosym}%
37   \let\horom@rvAries\Aries
38   \let\horom@rvTaurus\Taurus
39   \let\horom@rvGemini\Gemini
40   \let\horom@rvCancer\Cancer
41   \let\horom@rvLeo\Leo
42   \let\horom@rvVirgo\Virgo
43   \let\horom@rvLibra\Libra
44   \let\horom@rvScorpio\Scorpio
45   \let\horom@rvSagittarius\Sagittarius
46   \let\horom@rvCapricorn\Capricorn
47   \let\horom@rvAquarius\Aquarius
48   \let\horom@rvPisces\Pisces
49   \let\horom@rvSun\Sun
50   \let\horom@rvMoon\Moon
51   \let\horom@rvMercury\Mercury
52   \let\horom@rvVenus\Venus
53   \let\horom@rvMars\Mars
54   \let\horom@rvJupiter\Jupiter
55   \let\horom@rvSaturn\Saturn
56   \let\horom@rvUranus\Uranus
57   \let\horom@rvNeptune\Neptune
58   \let\horom@rvPluto\Pluto
59 \fi
60 \ifhoro@starfont\RequirePackage{starfont}%
61   \let\horost@rAries\Aries
62   \let\horost@rTaurus\Taurus
63   \let\horost@rGemini\Gemini
64   \let\horost@rCancer\Cancer
65   \let\horost@rLeo\Leo
66   \let\horost@rVirgo\Virgo
67   \let\horost@rLibra\Libra
```



```

68 \let\horost@rScorpio\Scorpio
69 \let\horost@rSagittarius\Sagittarius
70 \let\horost@rCapricorn\Capricorn
71 \let\horost@rAquarius\Aquarius
72 \let\horost@rPisces\Pisces
73 \let\horost@rSun\Sun
74 \let\horost@rMoon\Moon
75 \let\horost@rMercury\Mercury
76 \let\horost@rVenus\Venus
77 \let\horost@rMars\Mars
78 \let\horost@rJupiter\Jupiter
79 \let\horost@rSaturn\Saturn
80 \let\horost@rUranus\Uranus
81 \let\horost@rNeptune\Neptune
82 \let\horost@rPluto\Pluto
83 \let\horost@rSquare\Square
84 \fi
85 \ifhoro@wheels
86 \RequirePackage{epic}
87 \RequirePackage{eepic}
88 \RequirePackage{trig}
89 \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.

```

90 \def\horodegrees{\ensuremath{^\circ}}
91 \def\horominutes{\ensuremath{'}}
92 \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.

```

93 \def\Zodiac#1{\ifcase#1
94 \or\AriesSymbol\or\TaurusSymbol\or\GeminiSymbol
95 \or\CancerSymbol\or\LeoSymbol\or\VirgoSymbol
96 \or\LibraSymbol\or\ScorpioSymbol\or\SagittariusSymbol
97 \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.

```

98 \def\MeanNodeSymbol{\NorthNodeSymbol$_M$}%
99 \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
100 \newcommand{\horotextsym}{%

```

Signs of the Zodiac:

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

Traditional planets and luminaries:

```
113 \gdef\SunSymbol{Su}%
114 \gdef\MoonSymbol{Mo}%
115 \gdef\MercurySymbol{Me}%
116 \gdef\VenusSymbol{Ve}%
117 \gdef\MarsSymbol{Ma}%
118 \gdef\JupiterSymbol{Ju}%
119 \gdef\SaturnSymbol{Sa}%
120 \gdef\UranusSymbol{Ur}%
121 \gdef\NeptuneSymbol{Ne}%
122 \gdef\PlutoSymbol{Pl}%
```

Minor planets:

```
123 \gdef\ChironSymbol{Chi}%
124 \gdef\CeresSymbol{Cer}%
125 \gdef\PallasSymbol{Pal}%
126 \gdef\JunoSymbol{Jun}%
127 \gdef\VestaSymbol{Ves}%
```

Derived points:

```
128 \gdef\NorthNodeSymbol{No}%
129 \gdef\SouthNodeSymbol{SNo}%
130 \gdef\LilithSymbol{Lil}%
```

House cusps:

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

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

```
143 \gdef\AscendantSymbol{ASC}%
144 \gdef\MCSymbol{MC}%
145 \gdef\VertexSymbol{Vtx}%
```

Direction:

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

Aspects:

```
147 \gdef\horoConjunctionSymbol{Con}%
```

```
148 \gdef\horoOppositionSymbol{Opp}%
```

```
149 \gdef\horoTrineSymbol{Tri}%
```

```
150 \gdef\horoSquareSymbol{Sqr}%
```

```
151 \gdef\horoQuintileSymbol{Qnt}%
```

```
152 \gdef\horoBiquintileSymbol{Bqi}%
```

```
153 \gdef\horoSextileSymbol{Sex}%
```

```
154 \gdef\horoQuincunxSymbol{Qcx}%
```

```
155 \gdef\horoSemisextileSymbol{Ssx}%
```

```
156 \gdef\horoSemisquareSymbol{Ssq}%
```

```
157 \gdef\horoSesquiquadrateSymbol{Sqq}%
```

```
158 }
```

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

```
159 \horotextsym
```

§2.3. Symbols from `wasysym`.—

`\horowasysym` As with `textsym`, the actual setting of the symbol macros is all inside a `wasysym`-specific macro so that we can switch back to it easily after using other symbol sets.

```
160 \ifhoro@wasysym\newcommand{\horowasysym}{%
```

Signs of the Zodiac:

```
161 \gdef\AriesSymbol{\aries}%
```

```
162 \gdef\TaurusSymbol{\taurus}%
```

```
163 \gdef\GeminiSymbol{\gemini}%
```

```
164 \gdef\CancerSymbol{\cancer}%
```

```
165 \gdef\LeoSymbol{\leo}%
```

```
166 \gdef\VirgoSymbol{\virgo}%
```

```
167 \gdef\LibraSymbol{\libra}%
```

```
168 \gdef\ScorpioSymbol{\scorpio}%
```

```
169 \gdef\SagittariusSymbol{\sagittarius}%
```

```
170 \gdef\CapricornSymbol{\capricornus}%
```

```
171 \gdef\AquariusSymbol{\aquarius}%
```

```
172 \gdef\PiscesSymbol{\pisces}%
```

Traditional planets and luminaries:

```
173 \gdef\SunSymbol{\astrosun}%
```

```
174 \gdef\MoonSymbol{\rightmoon}%
```

```
175 \gdef\MercurySymbol{\mercury}%
```

```
176 \gdef\VenusSymbol{\venus}%
```

```
177 \gdef\MarsSymbol{\mars}%
```

```
178 \gdef\JupiterSymbol{\jupiter}%
```

```
179 \gdef\SaturnSymbol{\saturn}%
```

```
180 \gdef\UranusSymbol{\uranus}%
```

```
181 \gdef\NeptuneSymbol{\neptune}%
```

```
182 \gdef\PlutoSymbol{\pluto}%
```

Aspects:

```
183 \gdef\horoConjunctionSymbol{\conjunction}%
```

```
184 \gdef\horoOppositionSymbol{\opposition}%
```

```

185 \gdef\horoTrineSymbol{\ensuremath{\bigtriangleup}}%
186 \gdef\horoSquareSymbol{\horow@sySquare}%
187 \gdef\horoSextileSymbol{\hexstar}%
188 }

```

Invoke `\horowasysym` to set it as default:

```

189 \horowasysym
190 \fi

```

§2.4. 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.

```

191 \ifhoro@marvosym\newcommand{\horomarvosym}{%

```

Signs of the Zodiac:

```

192 \gdef\AriesSymbol{\horom@rvAries}%
193 \gdef\TaurusSymbol{\horom@rvTaurus}%
194 \gdef\GeminiSymbol{\horom@rvGemini}%
195 \gdef\CancerSymbol{\horom@rvCancer}%
196 \gdef\LeoSymbol{\horom@rvLeo}%
197 \gdef\VirgoSymbol{\horom@rvVirgo}%
198 \gdef\LibraSymbol{\horom@rvLibra}%
199 \gdef\ScorpioSymbol{\horom@rvScorpio}%
200 \gdef\SagittariusSymbol{\horom@rvSagittarius}%
201 \gdef\CapricornSymbol{\horom@rvCapricorn}%
202 \gdef\AquariusSymbol{\horom@rvAquarius}%
203 \gdef\PiscesSymbol{\horom@rvPisces}%

```

Traditional planets and luminaries:

```

204 \gdef\SunSymbol{\horom@rvSun}%
205 \gdef\MoonSymbol{\horom@rvMoon}%
206 \gdef\MercurySymbol{\horom@rvMercury}%
207 \gdef\VenusSymbol{\horom@rvVenus}%
208 \gdef\MarsSymbol{\horom@rvMars}%
209 \gdef\JupiterSymbol{\horom@rvJupiter}%
210 \gdef\SaturnSymbol{\horom@rvSaturn}%
211 \gdef\UranusSymbol{\horom@rvUranus}%
212 \gdef\NeptuneSymbol{\horom@rvNeptune}%
213 \gdef\PlutoSymbol{\horom@rvPluto}%
214 }

```

Invoke `\horomarvosym` to set it as default:

```

215 \horomarvosym
216 \fi

```

§2.5. 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.

```

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

```

Signs of the Zodiac:

```

218 \gdef\AriesSymbol{\horost@rAries}%

```

219 \gdef\TaurusSymbol{\horost@rTaurus}%
 220 \gdef\GeminiSymbol{\horost@rGemini}%
 221 \gdef\CancerSymbol{\horost@rCancer}%
 222 \gdef\LeoSymbol{\horost@rLeo}%
 223 \gdef\VirgoSymbol{\horost@rVirgo}%
 224 \gdef\LibraSymbol{\horost@rLibra}%
 225 \gdef\ScorpioSymbol{\horost@rScorpio}%
 226 \gdef\SagittariusSymbol{\horost@rSagittarius}%
 227 \gdef\CapricornSymbol{\horost@rCapricorn}%
 228 \gdef\AquariusSymbol{\horost@rAquarius}%
 229 \gdef\PiscesSymbol{\horost@rPisces}%

Traditional planets and luminaries:

230 \gdef\SunSymbol{\horost@rSun}%
 231 \gdef\MoonSymbol{\horost@rMoon}%
 232 \gdef\MercurySymbol{\horost@rMercury}%
 233 \gdef\VenusSymbol{\horost@rVenus}%
 234 \gdef\MarsSymbol{\horost@rMars}%
 235 \gdef\JupiterSymbol{\horost@rJupiter}%
 236 \gdef\SaturnSymbol{\horost@rSaturn}%
 237 \gdef\UranusSymbol{\horost@rUranus}%
 238 \gdef\NeptuneSymbol{\horost@rNeptune}%
 239 \gdef\PlutoSymbol{\horost@rPluto}%

Asteroids:

240 \gdef\ChironSymbol{\Chiron}%
 241 \gdef\CeresSymbol{\Ceres}%
 242 \gdef\PallasSymbol{\Pallas}%
 243 \gdef\JunoSymbol{\Juno}%
 244 \gdef\VestaSymbol{\Vesta}%

Derived points:

245 \gdef\NorthNodeSymbol{\NorthNode}%
 246 \gdef\SouthNodeSymbol{\SouthNode}%
 247 \gdef\MeanNodeSymbol{\NorthNode\$_M\$}%
 248 \gdef\TrueNodeSymbol{\NorthNode\$_T\$}%
 249 \gdef\LilithSymbol{\Lilith}%

Cusps and angles:

250 \gdef\CuspISymbol{\ASC}%
 251 \gdef\CuspIVSymbol{\IC}%
 252 \gdef\CuspVIISymbol{\DSC}%
 253 \gdef\CuspXSymbol{\MC}%
 254 \gdef\AscendantSymbol{\ASC}%
 255 \gdef\MCSymbol{\MC}%
 256 \gdef\VertexSymbol{\Vertex}%

Direction:

257 \gdef\horoRetrogradeSymbol{\Retrograde}%

Aspects:

258 \gdef\horoConjunctionSymbol{\Conjunction}%
 259 \gdef\horoOppositionSymbol{\Opposition}%
 260 \gdef\horoTrineSymbol{\Trine}%
 261 \gdef\horoSquareSymbol{\horost@rSquare}%
 262 \gdef\horoQuintileSymbol{\\$ \mathsf{Q} \\$}%

```

263 \gdef\horoBiquintileSymbol{\mathsf{Q}^2}%
264 \gdef\horoSextileSymbol{\Sextile}%
265 \gdef\horoQuincunxSymbol{\Quincunx}%
266 \gdef\horoSemisextileSymbol{\Semisextile}%
267 \gdef\horoSemisquareSymbol{\Semisquare}%
268 \gdef\horoSesquiquadrateSymbol{\Sesquiquadrate}%
269 }

```

Invoke `\horostarfont` to set it as default:

```

270 \horostarfont
271 \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 horoscopus code so we leave it in force after defining `\trimspaces`.

```

272 %% WATCH OUT! MAKING Q EXOTIC HERE!
273 \catcode'\Q=3
274 \def\horo@cue{Q}

```

`\trimspaces`

```

275 \def\trimspaces#1{%
276   \begingroup
277   \aftergroup\toks\aftergroup0\aftergroup{%
278   \expandafter\@trimb\expandafter\noexpand#1Q Q}%
279   \edef#1{\the\toks0}%
280 }
281 \long\def\@trimb#1 Q{\@trimc#1Q}
282 \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.

```

283 \def\horo@fterdef#1{%
284   \aftergroup\horo@ft@a
285   \aftergroup#1\expandafter\aftergroup\csname horo@@#1\endcsname
286 }

```

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

```

287 \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.

```
288 \def\horo@ft@a#1#2{%
289   \edef#1{\expandafter\horo@eathead\string#2}%
290 }
```

§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@`.

```
291 \def\horo@polarconvert#1#2#3#4{%
292   \dimen@=\nin@ty\p@advance\dimen@-#2\p@\TG@@sin
293   #3=#1\dimen@
294   \dimen@=#2\p@\TG@@sin
295   #4=#1\dimen@
296 }
```

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

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

```
297 \def\horo@fe#1,{%
298   \def\horo@fe@c{#1}%
299   \ifx\horo@fe@c\horo@cue\relax
300   \else
301     \horo@fe@b{#1}%
302     \expandafter\horo@fe
303   \fi
304 }
```

`\horo@foreach` No `@` 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 Q item at the end to terminate the recursion.

```
305 \def\horoforeach#1#2{%
306   \edef\horo@fe@a{#1}%
307   \def\horo@fe@b{#2}%
308   \expandafter\horo@fe\horo@fe@a,Q,%
309 }
```

§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 Q terminator, and if not, applies `\horo@dfe@action` to the pair of `\horo@dfe@x` and `\horo@dfe@y` before tail-recursing.

```
310 \def\horo@dfe@b#1,{%
311   \def\horo@dfe@y{#1}%
312   \ifx\horo@dfe@y\horo@cue\relax
313   \else
314     \horo@dfe@action{\horo@dfe@x}{\horo@dfe@y}%
315     \expandafter\horo@dfe@b
316   \fi
317 }
```

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

```
318 \def\horo@dfe@a#1,{%
319   \def\horo@dfe@x{#1}%
320   \ifx\horo@dfe@x\horo@cue\relax
321   \else
322     \expandafter\horo@dfe@b\horo@dfe@ylist,Q,\relax
323     \expandafter\horo@dfe@a
324   \fi
325 }
```

`\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 Q .

```
326 \def\horo@dblforeach#1#2#3{%
327   \edef\horo@dfe@xlist{#1}\edef\horo@dfe@ylist{#2}\def\horo@dfe@action{#3}%
328   \expandafter\horo@dfe@a\horo@dfe@xlist,Q,\relax
329 }
```

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.

```
330 \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.

```
331 \def\hororightcoord{\ifhorocalculated\horocuspVIIPos\else180\fi}%
```

`\horocusps` A list of all the house cusps.

```
332 \def\horocusps{CuspI,CuspII,CuspIII,CuspIV,CuspV,CuspVI,%
333 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.

```
334 \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.

```
335 \def\horocalcharmonic#1{%
336 \horoforeach{\horoobjects}{\horo@calch{#1}}%
337 }
```

`\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.

```
338 \def\horo@calch#1#2{%
339 \dimen@=\csname horo#2Pos\endcsname\p@\relax
340 \multiply\dimen@ by #1\relax
341 \horo@fixdimen@
342 \expandafter\edef\csname horo#2Pos\endcsname{\TG@rem@pt\dimen@}%
343 \expandafter\edef\csname horo#2DPos\endcsname{\TG@rem@pt\dimen@}%
344 }
```

§4.3. Shifting positions.—

`\horoshiftobjects` Adds an offset to all the object positions in `\horoobjects`, much like `\horocalcharmonic` above.

```
345 \def\horoshiftobjects#1{%
346 \horoforeach{\horoobjects}{\horo@shift{#1}}%
347 }
```

`\horoshiftcusps` Adds an offset to all the object positions in `\horocusps`, much like `\horocalcharmonic` above.

```
348 \def\horoshiftcusps#1{%
349   \horoforeach{\horocusps}{\horo@shift{#1}}%
350 }
```

`\horo@shift` Internal for `\horoshiftobjects` and `-cusps`. Much like `\horo@calch` except it adds instead of multiplying.

```
351 \def\horo@shift#1#2{%
352   \dimen@=\csname horo#2Pos\endcsname\p@\relax
353   \advance\dimen@ by #1\p@\relax
354   \horo@fixdimen@
355   \expandafter\edef\csname horo#2Pos\endcsname{\TG@rem@pt\dimen@}%
356   \expandafter\edef\csname horo#2DPos\endcsname{\TG@rem@pt\dimen@}%
357 }
```

§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.

```
358 \def\horomakeequalcusps#1{
359   \def\horoCuspIPos{0}%
360   \def\horoCuspIIPos{30}%
361   \def\horoCuspIIIPos{60}%
362   \def\horoCuspIVPos{90}%
363   \def\horoCuspVPos{120}%
364   \def\horoCuspVIPos{150}%
365   \def\horoCuspVIIPos{180}%
366   \def\horoCuspVIIIPos{210}%
367   \def\horoCuspIXPos{240}%
368   \def\horoCuspXPos{270}%
369   \def\horoCuspXIPos{300}%
370   \def\horoCuspXIIPos{330}%
371   \horoshiftcusps{#1}%
372 }
```

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

```
373 \def\horomakesigncusps#1{
374   \dimen@=#1\p@\relax
375   \edef\horo@savecount@{\the\count0}%
376   \count0=\dimen@\relax
377   \divide\count0 by 1966080\relax
378   \multiply\count0 by 30\relax
379   \expandafter\horomakeequalcusps{\the\count0}%
380   \count0=\horo@savecount@\relax
381 }
```

§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@`.

```

382 \def\horo@midpoint#1#2{%
383   \dimen@#1\p@\relax
384   \advance\dimen@ by-#2\p@\relax
385   \horo@fixdimen@diff
386   \divide\dimen@ by2\relax
387   \advance\dimen@ by#2\p@\relax
388   \horo@fixdimen@
389 }

```

§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.

```

390 \def\horo@chartrotate{%
391   \advance\dimen@-\hororightcoord\p@
392   \horo@fixdimen@
393 }

```

`\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.

```

394 \def\horo@fixdimen@{%
395   \edef\horo@savecount@{\the\count0}%
396   \count0=\dimen@\relax
397   \ifnum\count0<0\relax
398     \divide\count0 by 23592960\relax
399     \advance\count0 by -1\relax
400   \else
401     \divide\count0 by 23592960\relax
402   \fi
403   \multiply\count0 by -360\relax
404   \advance\dimen@ by\count0\p@\relax
405   \count0=\horo@savecount@\relax
406 }

```

`\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).

```

407 \def\horo@fixdimen@diff{%
408   \horo@fixdimen@
409   \ifdim\dimen@>180\p@\advance\dimen@ by -360\p@\fi
410 }

```

§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`

```

411 \def\hororotatechart#1#2{%
412   \begingroup

```

```

413 \dimen@=#2\p@\relax
414 \advance\dimen@ by-\csname horo#1DPos\endcsname\p@\relax
415 \multiply\dimen@ by -1\relax
416 \edef\hororightcoord{\TG@rem@pt\dimen@}%
417 \horo@fterdef\hororightcoord
418 \endgroup
419 }

```

§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.

```

420 \def\horocopyvar#1#2#3{%
421 \horoforeach{#1}{\horo@cv@{#2}{#3}}%
422 }

```

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

```

423 \def\horo@cv#1#2#3{%
424 \expandafter\edef\csname horo#3#2\endcsname{\csname horo#3#1\endcsname}%
425 }

```

`\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.

```

426 \def\horosaveobjects#1{%
427 \begingroup
428 \aftergroup\def\aftergroup#1\aftergroup{%
429 \horo@fterdef\horoobjects
430 \horoforeach{\horoobjects}{\horo@svo@a}%
431 \aftergroup\horo@svo@b
432 \aftergroup}%
433 \endgroup
434 }

```

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

```

435 \def\horo@svo@a#1{%
436 \expandafter\horo@fterdef\csname horo#1Pos\endcsname
437 \expandafter\horo@fterdef\csname horo#1Vel\endcsname
438 }

```

`\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.

```

439 \def\horo@svo@b{%
440 \horocopyvar{\horoobjects}{Pos}{DPos}%
441 }

```

`\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.

```
442 \def\horosavecusps#1{%
443   \begingroup
444     \aftergroup\def\aftergroup#1\aftergroup{%
445       \edef\hororightcoord{\hororightcoord}%
446       \horo@fterdef\horocusps\horo@fterdef\hororightcoord
447       \horoforeach{\horocusps}{\horo@svc@a}%
448       \aftergroup\horo@svc@b
449       \aftergroup}%
450   \endgroup
451 }
```

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

```
452 \def\horo@svc@a#1{%
453   \expandafter\horo@fterdef\csname horo#1Pos\endcsname
454 }
```

`\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.

```
455 \def\horo@svc@b{%
456   \horocopyvar{\horocusps}{Pos}{DPos}%
457 }
```

§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.”

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

`\horo@dms` 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 `Q` terminates. The output goes to redefine the last argument.

```

459 \def\horo@dms#1#2:#3:#4Q#5{%
460   \def\horo@data{ }%
461   \uppercase{\if#1S\def\horo@data{-}\fi\if#1W\def\horo@data{-}\fi}%
462   \dimen@=#4\p@
463   \divide\dimen@ by 60\relax
464   \advance\dimen@ by #3\p@
465   \divide\dimen@ by 60\relax
466   \advance\dimen@ by #2\p@
467   \edef#5{\horo@data\TG@rem@pt\dimen@}%
468   \trimspaces{#5}%
469 }

```

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.

```
470 \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.

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

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

```
472 \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.

```
473 \def\hororoundstricttruncate{\def\horo@r@offdigit{X}}
```

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

Now, pure round to nearest whatever.

```
475 \def\hororoundtosec{\def\horo@r@offset{9sp}\def\horo@r@offdigit{Z}}
476 \def\hororoundtomin{\def\horo@r@offset{546sp}\def\horo@r@offdigit{Z}}
477 \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.

```
478 \def\hororoundtoseckeepsign{\def\horo@r@offset{9sp}\def\horo@r@offdigit{D}}
479 \def\hororoundtoseckeepdeg{\def\horo@r@offset{546sp}\def\horo@r@offdigit{M}}
480 \def\hororoundtoseckeepmin{\def\horo@r@offset{0.5pt}\def\horo@r@offdigit{S}}
481 \def\hororoundtominkeepsign{\def\horo@r@offset{546sp}\def\horo@r@offdigit{D}}
482 \def\hororoundtominkeepdeg{\def\horo@r@offset{0.5pt}\def\horo@r@offdigit{M}}
483 \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?

```
484 \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.

```
485 \def\horo@d@dms@@#1#2{%
486   \count0=\dimen@\relax
487   \divide\count0by65536\relax
488   \ifhororoundclamp\ifnum\count0>#2 \count0=#2\relax\fi\fi
489   \edef#1{\the\count0}%
490   \horo@fterdef#1%
491   \advance\dimen@-\count0\p@
492   \multiply\dimen@ 60\relax
493 }
```

`\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.

```
494 \def\horo@d@dms@{%
495   \advance\dimen@1sp\relax
496   \horo@d@dms@@\horo@d@deg{100000}%
497   \horo@d@dms@@\horo@d@min{100000}%
498   \horo@d@dms@@\horo@d@sec{100000}%
499 }
```

`\horo@d@sdms` Decimal to SDMS. The number of degrees past $0^\circ\mathcal{V}$, 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.

```
500 \def\horo@d@sdms{%
501   \begingroup
502     \def\horo@tmp{Z}\ifx\horo@r@offdigit\horo@tmp
503       \advance\dimen@\horo@r@offset\relax
504     \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.

```
505   \count0=\dimen@\relax
506   \divide\count0by1966080\relax
507   \count1=\count0\relax
508   \advance\count1by1\relax
509   \edef\horo@d@sign{\the\count1}%
510   \horo@fterdef\horo@d@sign
```

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

```
511   \multiply\count0by30\relax%
512   \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`.

```
513   \def\horo@tmp{D}\ifx\horo@r@offdigit\horo@tmp
514     \advance\dimen@\horo@r@offset\relax
515   \fi
516   \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`.

```
517   \def\horo@tmp{M}\ifx\horo@r@offdigit\horo@tmp
518     \advance\dimen@\horo@r@offset\relax
519   \fi
520   \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`.

```
521   \def\horo@tmp{S}\ifx\horo@r@offdigit\horo@tmp
522     \advance\dimen@\horo@r@offset\relax
523   \fi
524   \horo@d@dms@@\horo@d@sec{59}%
525 \endgroup
526 }
```

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

```
527 \def\horo@d@dms{%
528   \begingroup
529     \def\horo@tmp{Z}\ifx\horo@r@offdigit\horo@tmp
```



```

530     \advance\dimen@ \horo@r@offset\relax
531     \fi
532     \def\horo@tmp{D}\ifx\horo@r@offdigit\horo@tmp
533     \advance\dimen@ \horo@r@offset\relax
534     \fi
535     \horo@d@dms@@\horo@d@deg{100000}%
536     \def\horo@tmp{M}\ifx\horo@r@offdigit\horo@tmp
537     \advance\dimen@ \horo@r@offset\relax
538     \fi
539     \horo@d@dms@@\horo@d@min{59}%
540     \def\horo@tmp{S}\ifx\horo@r@offdigit\horo@tmp
541     \advance\dimen@ \horo@r@offset\relax
542     \fi
543     \horo@d@dms@@\horo@d@sec{59}%
544     \endgroup
545 }

```

`\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.

```

546 \def\horodsmstext#1{%
547     \begingroup
548     \ifhororoundauto\hororoundtoseckkeepmin\fi
549     \dimen@=#1\p@
550     \horo@d@sdms
551     \horo@d@deg\horodegrees\Zodiac{\horo@d@sign}%
552     \horo@d@min\horominutes\horo@d@sec\horoseconds
553     \endgroup
554 }

```

`\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.

```

555 \def\horotimetext#1{%
556     \begingroup
557     \ifhororoundauto\hororoundtruncate\fi
558     \dimen@=#1\p@
559     \horo@d@sdms
560     \horo@twodig\horo@d@min\horo@twodig\horo@d@sec
561     \horo@d@deg:\horo@d@min:\horo@d@sec
562     \endgroup
563 }

```

`\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.

```

564 \def\horo@nsew#1#2{%
565     \ifdim\dimen@<\z@\relax

```

```

566     \def\horo@calc@b{#2}\multiply\dimen@ by-1\relax
567     \else
568     \def\horo@calc@b{#1}%
569     \fi
570 }

```

`\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.

```

571 \def\horolatlon#1#2#3{%
572   \begingroup
573   \ifhororoundauto\hororoundtruncate\fi
574   \dimen@=#1\p@\horo@nsew{#2}{#3}%
575   \horo@d@dms
576   \horo@d@deg\horodegrees%
577   \horo@d@min\horominutes\horo@d@sec\horoseconds~\horo@calc@b
578   \endgroup
579 }

```

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

```

580 \def\hororxtext#1{%
581   \begingroup
582   \dimen@\csname horo#1Vel\endcsname\p@
583   \ifdim\dimen@<\z@\relax
584     \horoRetrogradeSymbol
585   \fi
586   \endgroup
587 }

```

§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.

```

588 \def\horocalparms#1#2#3#4#5#6{%
589   \edef\horocalcyear{#1}\edef\horocalcmonth{#2}\edef\horocalcday{#3}%
590   \horo@d@dms N#4Q\horocalctime
591   \horo@d@dms#5Q\horocalclon
592   \horo@d@dms#6Q\horocalclat
593 }

```

5. ASTROLOG CALCULATION BACK END

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

```

594 \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`.

```

595 \def\horoSun@strologname{Sun }\def\horo@stSunObj{Sun}
596 \def\horoMoon@strologname{Moo }\def\horo@stMooObj{Moon}
597 \def\horoMercury@strologname{Mer }\def\horo@stMerObj{Mercury}
598 \def\horoVenus@strologname{Ven }\def\horo@stVenObj{Venus}
599 \def\horoMars@strologname{Mar }\def\horo@stMarObj{Mars}
600 \def\horoJupiter@strologname{Jup }\def\horo@stJupObj{Jupiter}
601 \def\horoSaturn@strologname{Sat }\def\horo@stSatObj{Saturn}
602 \def\horoUranus@strologname{Ura }\def\horo@stUraObj{Uranus}
603 \def\horoNeptune@strologname{Nep }\def\horo@stNepObj{Neptune}
604 \def\horoPluto@strologname{Plu }\def\horo@stPluObj{Pluto}
605 \def\horoChiron@strologname{Chi }\def\horo@stChiObj{Chiron}
606 \def\horoCeres@strologname{Cer }\def\horo@stCerObj{Ceres}
607 \def\horoPallas@strologname{Pal }\def\horo@stPalObj{Pallas}
608 \def\horoJuno@strologname{Jun }\def\horo@stJunObj{Juno}
609 \def\horoVesta@strologname{Ves }\def\horo@stVesObj{Vesta}
610 \def\horoNorthNode@strologname{Nod }\def\horo@stNodObj{NorthNode}
611 \def\horoSouthNode@strologname{S.N }
612 \expandafter\def\csname horo@stS.NObj\endcsname{SouthNode}
613 \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.

```

614 \def\horo@stAscObj{CuspI}
615 \expandafter\def\csname horo@st2ndObj\endcsname{CuspII}
616 \expandafter\def\csname horo@st3rdObj\endcsname{CuspIII}
617 \def\horo@stNadObj{CuspIV}
618 \expandafter\def\csname horo@st5thObj\endcsname{CuspV}
619 \expandafter\def\csname horo@st6thObj\endcsname{CuspVI}
620 \def\horo@stDesObj{CuspVII}
621 \expandafter\def\csname horo@st8thObj\endcsname{CuspVIII}
622 \expandafter\def\csname horo@st9thObj\endcsname{CuspIX}
623 \def\horo@stMidObj{CuspX}
624 \expandafter\def\csname horo@st11tObj\endcsname{CuspXI}
625 \expandafter\def\csname horo@st12tObj\endcsname{CuspXII}
626 \def\horo@stAriSign{0}
627 \def\horo@stTauSign{30}
628 \def\horo@stGemSign{60}
629 \def\horo@stCanSign{90}
630 \def\horo@stLeoSign{120}
631 \def\horo@stVirSign{150}
632 \def\horo@stLibSign{180}
633 \def\horo@stScoSign{210}
634 \def\horo@stSagSign{240}
635 \def\horo@stCapSign{270}
636 \def\horo@stAquSign{300}
637 \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.

```

638 \def\horoPlacidusHouses{\def\horo@housenumber{0}}
639 \def\horoKochHouses{\def\horo@housenumber{1}}
640 \def\horoEqualHouses{\def\horo@housenumber{2}}
641 \def\horoCampanusHouses{\def\horo@housenumber{3}}

```

```

642 \def\horoMeridianHouses{\def\horo@houzenumber{4}}
643 \def\horoRegiomontanusHouses{\def\horo@houzenumber{5}}
644 \def\horoPorphyryHouses{\def\horo@houzenumber{6}}
645 \def\horoMorinusHouses{\def\horo@houzenumber{7}}
646 \def\horoPolichPageHouses{\def\horo@houzenumber{8}}
647 \def\horoAlcabitusHouses{\def\horo@houzenumber{9}}
648 \def\horoEqualMCHouses{\def\horo@houzenumber{10}}
649 \def\horoNeoPorphyryHouses{\def\horo@houzenumber{11}}
650 \def\horoWholeHouses{\def\horo@houzenumber{12}}
651 \def\horoVedicHouses{\def\horo@houzenumber{13}}

```

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

```

652 \newread\horo@tmpfile
653 \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.

```

654 \def\horoastrologopt{}

```

`\horoobjects` Default list of objects to calculate.

```

655 \def\horoobjects{Sun,Moon,Mercury,Venus,Mars,Jupiter,Saturn,%
656                 Uranus,Neptune,Pluto,NorthNode,Lilith,%
657                 Chiron,Ceres,Pallas,Juno,Vesta}

```

Placidus houses are the default:

```

658 \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.

```

659 \def\horocalculate{%
660   \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>astrologname` macro to the `\aftergroup` queue.

```

661   \begingroup
662   \aftergroup\def\aftergroup\horo@calc@a\aftergroup{%
663     \horoforeach{\horoobjects}{\horo@calc@c}%
664     \aftergroup}%
665   \endgroup

```

So at this point `\horocalca` 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 `\horocalcd`.

```
666 \dimen@\horocalctime\p@\horod@dms@\horotwodig\horod@min
667 \edef\horocalcd{\horod@deg:\horod@min\space GMT\space}%
```

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

```
668 \dimen@=\horocalclon\p@\horon@sew EW%
669 \horod@dms@
670 \horotwodig\horod@min
671 \edef\horocalce{\horod@deg:\horod@min\horocalcb\space}%
```

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

```
672 \dimen@=\horocalclat\p@\horon@sew NS%
673 \horod@dms@
674 \horotwodig\horod@min
675 \edef\horocalcf{\horod@deg:\horod@min\horocalcb\space}%
```

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

```
676 \immediate\write18{%
677   astrolog\space
678   -o0 \horotmpfname\space
679   -c \horohousenumber\space
680   -qa \horocalcmonth\space
681   \horocalcday\space
682   \horocalcyear\space
683   \horocalcd\horocalce\horocalcf
684   -R0 \horocalca
685   \horoastrologopt
686 }%
```

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

```
687 \openin\horotmpfile=\horotmpfname\relax
688 \aftergroup\horocalculatedfalse
689 \horocalc@parse
690 \closein\horotmpfile
```

Then the group ends, all the `\horofterdefs` happen, and that's the end of `\horocalculate`.

```
691 \endgroup
692 }
```

`\horocalcc` 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.

```
693 \def\horocalcc#1{%
694   \expandafter\aftergroup\csname horo#1@strologname\endcsname
695 }
```

`\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`;⁷ 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.

```

696 \def\horo@calc@parse{%
697   \read\horo@tmpfile to \horo@calc@parse@a
698   \ifeof\horo@tmpfile\else
699     \expandafter\horo@calc@parse@b\horo@calc@parse@a\space x\space Q%
700     \horo@calc@parse
701   \fi
702 }

```

`\horo@YF` Helper for `\horo@calc@parse@b`: simply expands to `-YF`.

```

703 \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.

```

704 \def\horo@calc@parse@b#1 #2Q{%
705   \def\horo@calc@parse@b@{#1}%
706   \ifx\horo@calc@parse@b@\horo@YF\horo@calc@parse@c#2Q\fi
707 }

```

`\horo@calc@parse@c` Second stage of parsing a line of Astrolog output. We use T_EX delimited argument parsing to find the fields on the line. They are a bit complicated but correspond 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`.

```

708 \def\horo@calc@parse@c#1: #2 #3 #4, #5 #6, #7 #8Q{%

```

First convert from degrees, sign, and decimal minutes, to decimal degrees:

```

709   \dimen@=#4\p@
710   \divide\dimen@ by 60\relax
711   \advance\dimen@ by #2\p@
712   \advance\dimen@ by \csname horo@st#3Sign\endcsname\p@

```

Save the result in Pos:

```

713   \expandafter\edef\csname
714     horo\csname horo@st#10bj\endcsname Pos\endcsname{%
715     \TG@rem@pt\dimen@}%
716   \expandafter\horo@fterdef\csname
717     horo\csname horo@st#10bj\endcsname Pos\endcsname

```

Save it in DPos as well:

```

718   \expandafter\edef\csname
719     horo\csname horo@st#10bj\endcsname DPos\endcsname{%
720     \TG@rem@pt\dimen@}%
721   \expandafter\horo@fterdef\csname
722     horo\csname horo@st#10bj\endcsname DPos\endcsname

```

⁷-YFing in designated areas only.

Save the velocity, which is already nicely formatted in #7, to Vel:

```
723 \expandafter\def\csname
724   horo\csname horo@st#10bj\endcsname Vel\endcsname{#7}%
725 \expandafter\horo@fterdef\csname
726   horo\csname horo@st#10bj\endcsname Vel\endcsname
```

Now we have at least a little bit of valid data, so things are probably cool.

```
727 \aftergroup\horocalculatedtrue
728 }
```

End of the `\ifhoro@strolog` conditional:

```
729 \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
```

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

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

```
731 \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.

```
732 \def\horoSunSEOpt{-p0}
733 \def\horoMoonSEOpt{-p1}
734 \def\horoMercurySEOpt{-p2}
735 \def\horoVenusSEOpt{-p3}
736 \def\horoMarsSEOpt{-p4}
737 \def\horoJupiterSEOpt{-p5}
738 \def\horoSaturnSEOpt{-p6}
739 \def\horoUranusSEOpt{-p7}
740 \def\horoNeptuneSEOpt{-p8}
741 \def\horoPlutoSEOpt{-p9}
742 \def\horoChironSEOpt{-pD}
743 \def\horoPholusSEOpt{-pE}
744 \def\horoCeresSEOpt{-pF}
745 \def\horoPallasSEOpt{-pG}
746 \def\horoJunoSEOpt{-pH}
747 \def\horoVestaSEOpt{-pI}
748 \def\horoMeanNodeSEOpt{-pm}
749 \def\horoTrueNodeSEOpt{-pt}
750 \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.

```
751 \def\horoAscendantSEOpt{-p0}
752 \def\horoMCSEOpt{-p0}
753 \def\horoARMCSEOpt{-p0}
754 \def\horoVertexSEOpt{-p0}
```

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

```
755 \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.

```
756 \def\horoAlcabitusHouses{\def\horo@houseletter{b}}
757 \def\horoAxialHouses{\def\horo@houseletter{x}}
758 \def\horoAzimuthalHouses{\def\horo@houseletter{h}}
759 \def\horoCampanusHouses{\def\horo@houseletter{c}}
760 \def\horoEqualHouses{\def\horo@houseletter{a}}
761 \def\horoKochHouses{\def\horo@houseletter{k}}
762 \def\horoKrusinskiHouses{\def\horo@houseletter{U}}
763 \def\horoMorinusHouses{\def\horo@houseletter{m}}
764 \def\horoPlacidusHouses{\def\horo@houseletter{p}}
765 \def\horoPolichPageHouses{\def\horo@houseletter{t}}
766 \def\horoPorphyryHouses{\def\horo@houseletter{o}}
767 \def\horoRegiomontanusHouses{\def\horo@houseletter{r}}
768 \def\horoVehlowHouses{\def\horo@houseletter{v}}
```

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

```
769 \newread\horo@tmpfile
770 \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.

```
771 \def\horo@readdata{%
772   \ifeof\horo@tmpfile
773     \def\horo@data{0.0}%
774     \horocalculatedfalse
775   \else
776     \begingroup
777     \let\do\@makeother\dospecials
778     \read\horo@tmpfile to \horo@data
779     \trimspaces\horo@data
```



```

780     \horo@fterdef\horo@data
781     \endgroup
782     \fi
783 }

```

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

```

784 \def\horo@readpos#1{%
785   \horo@readdata
786   \expandafter\let\csname horo#1Pos\endcsname\horo@data
787   \expandafter\let\csname horo#1DPos\endcsname\horo@data
788 }

```

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

```

789 \def\horo@readposvel#1{%
790   \horo@readpos{#1}%
791   \horo@readdata
792   \expandafter\let\csname horo#1Vel\endcsname\horo@data
793 }

```

`\horoobjects` Default object list for Swiss Ephemeris.

```

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

```

Placidus houses are default:

```

797 \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.

```

798 \def\horo@calcobj@#1#2{%

```

Start by opening a prophylactic group:

```

799   \begingroup

```

Convert the time of day to hours, minutes, and seconds:

```

800   \dimen@\horocalctime\p@\horo@d@dms@
801   \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.

```

802   \immediate\write18{%
803     swetest -f#2 -head\space
804             -b\horocalcday.\horocalcmonth.\horocalcyear\space
805             -ut\horo@d@deg:\horo@d@min:\horo@d@sec\space
806             \csname horo#1SEOpt\endcsname\space
807             \horosweopt\space
808     \ifhoro@egrep| egrep '[ 0-9.-]+' \space\fi
809     >\ifhoro@ppendtmp >\fi\space
810     \horo@tmpfname
811   }%

```

But after closing the prophylactic group...

```
812 \endgroup
```

... we set the flag true so that subsequent calls to `swetest` will append instead of overwriting.

```
813 \horo@ppendtmptrue
```

```
814 }
```

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

```
815 \def\horo@calobj#1{%
```

```
816 \horo@calobj@{#1}{l}%
```

```
817 \horo@calobj@{#1}{s}%
```

```
818 }
```

`\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@calobj` to get position and speed for all the regular objects. The internal function `\horo@calobj@` 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.

```
819 \def\horocalculate{%
```

```
820 \edef\horoCuspsSEOpt{-house\horocalclon,\horocalclat,%
```

```
821 \horo@houseletter\space-p}%
```

```
822 \horo@ppendtmpfalse
```

```
823 \horocalculatedtrue
```

```
824 \horoforeach{\horoobjects}{\horo@calobj}%
```

```
825 \horo@calobj@{Cusps}{l}%
```

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.

```
826 \openin\horo@tmpfile=\horo@tmpfname\relax
```

```
827 \horoforeach{\horoobjects}{\horo@readposvel}%
```

```
828 \horoforeach{\horoCusps,Ascendant,MC,ARMC,Vertex}{\horo@readpos}%
```

```
829 \closein\horo@tmpfile\relax
```

```
830 }
```

End the `\ifhoro@swetest` conditional:

```
831 \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.

```
832 \newif\ifhorodrawcusps\horodrawcuspstrue
```

`\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.

```
833 \newif\ifhoroboldangles\horoboldanglestrue
```

`\ifhoroanglearrows` Setting for whether to draw angular house cusps as arrows instead of regular lines.

```
834 \newif\ifhoroanglearrows\horoanglearrowstrue
```

Now we can enter a conditional on wheel charts not being disabled by package option.

```
835 \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.

```
836 \newlength{\horounitlength}
```

```
837 \setlength{\horounitlength}{0.00952\textwidth}
```

`\horoanglecuspswidth` Line width to use in drawing angular cusps when `\horoboldangles` is turned on.

```
838 \newlength{\horoanglecuspswidth}
```

```
839 \setlength{\horoanglecuspswidth}{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.

```
840 \newenvironment{horoscope}{%
```

```
841 \setlength{\unitlength}{\horounitlength}%
```

```
842 \begin{picture}(100,100)(-50,-50)%
```

```
843 }{%
```

```
844 \end{picture}%
```

```
845 }
```

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`

```
846 \newcommand{\horoCnote}[1]{%
```

```
847 \put(0,0){\makebox(0,0){\parbox{75\unitlength}{\centering #1}}}%
```

```
848 }
```

`\horoULnote`

```
849 \newcommand{\horoULnote}[1]{%
```

```
850 \put(-50,50){\makebox(40,0)[t]{\parbox[t]{40\unitlength}{\raggedright #1}}}%
```

```
851 }
```

`\horoURnote`

```
852 \newcommand{\horoURnote}[1]{%
853   \put(10,50){\makebox(40,0)[t]{\parbox[t]{40\unitlength}{\raggedleft #1}}}%
854 }
```

`\horoLLnote`

```
855 \newcommand{\horoLLnote}[1]{%
856   \put(-50,-50){\makebox(40,0)[b]{\parbox[t]{40\unitlength}{\raggedright #1}}}%
857 }
```

`\horoLRnote`

```
858 \newcommand{\horoLRnote}[1]{%
859   \put(10,-50){\makebox(40,0)[b]{\parbox[t]{40\unitlength}{\raggedleft #1}}}%
860 }
```

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

```
861 \dimendef\horo@dim@x=1\relax
862 \dimendef\horo@dim@y=2\relax
863 \dimendef\horo@dim@xa=3\relax
864 \dimendef\horo@dim@ya=4\relax
865 \dimendef\horo@dim@xb=5\relax
866 \dimendef\horo@dim@yb=6\relax
867 \dimendef\horo@dim@xc=7\relax
868 \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`.

```
869 \def\horo@putpolar#1#2#3{%
870   \begingroup
871     \edef\p@one{#1}\edef\p@two{#2}%
872     \horo@polarconvert\p@one\p@two\horo@dim@x\horo@dim@y
873     \put(\TG@rem@pt\horo@dim@x,\TG@rem@pt\horo@dim@y){\makebox(0,0){#3}}%
874   \endgroup
875 }
```

`\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.

```

876 \def\horoputradial#1#2#3{%
877   \begingroup
878     \edef\p@one{#1}\edef\p@two{#2}\edef\p@three{#3}%
879     \horo@polarconvert\p@one\p@three\horo@dim@xa\horo@dim@ya
880     \horo@polarconvert\p@two\p@three\horo@dim@xb\horo@dim@yb
881     \advance\horo@dim@xb by\horo@dim@xa
882     \advance\horo@dim@yb by\horo@dim@ya
883     \drawline(\TG@rem@pt\horo@dim@xa,\TG@rem@pt\horo@dim@ya)%
884             (\TG@rem@pt\horo@dim@xb,\TG@rem@pt\horo@dim@yb)%
885   \endgroup
886 }

```

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

```

887 \def\horoputline#1#2#3#4{%
888   \begingroup
889     \edef\p@one{#1}\edef\p@two{#2}\edef\p@three{#3}\edef\p@four{#4}%
890     \horo@polarconvert\p@one\p@two\horo@dim@xa\horo@dim@ya
891     \horo@polarconvert\p@three\p@four\horo@dim@xb\horo@dim@yb
892     \drawline(\TG@rem@pt\horo@dim@xa,\TG@rem@pt\horo@dim@ya)%
893             (\TG@rem@pt\horo@dim@xb,\TG@rem@pt\horo@dim@yb)%
894   \endgroup
895 }

```

§7.5. Arrowheads.—Although \LaTeX can draw arrows (which it calls “vectors”), the implementation has annoying limitations that make it unsuitable for use in horoscopes’ 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.

```

896 \def\horoputarrowhead#1#2#3{%
897   \begingroup
898     \edef\horo@pa@r{#1}\edef\horo@pa@th{#2}%
899     \edef\horo@pa@sc@le{#3}%

```

Convert tip coordinates to Cartesian.

```

900   \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.

```

901   \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.

```

902   \horo@dim@xb=\horo@dim@x\relax\horo@dim@yb=\horo@dim@y\relax
903   \advance\horo@dim@xb by-3\horo@dim@xa\relax
904   \advance\horo@dim@yb by-3\horo@dim@ya\relax

```

```

905 \advance\horodim@xb by-\horodim@ya\relax
906 \advance\horodim@yb by\horodim@xa\relax
907 \horodim@xc=\horodim@x\relax\horodim@yc=\horodim@y\relax
908 \advance\horodim@xc by-3\horodim@xa\relax
909 \advance\horodim@yc by-3\horodim@ya\relax
910 \advance\horodim@xc by\horodim@ya\relax
911 \advance\horodim@yc by-\horodim@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.

```

912 \multiply\horodim@xa by-2\relax\multiply\horodim@ya by-2\relax
913 \advance\horodim@xa by\horodim@x\relax
914 \advance\horodim@ya by\horodim@y\relax

```

Plot the arrowhead, using the documented-as-dangerous `\blacken` command.

```

915 \blacken\path%
916 (\TG@rem@pt\horodim@x,\TG@rem@pt\horodim@y)%
917 (\TG@rem@pt\horodim@xb,\TG@rem@pt\horodim@yb)%
918 (\TG@rem@pt\horodim@xa,\TG@rem@pt\horodim@ya)%
919 (\TG@rem@pt\horodim@xc,\TG@rem@pt\horodim@yc)%
920 (\TG@rem@pt\horodim@x,\TG@rem@pt\horodim@y)%
921 \endgroup
922 }

```

§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 L^AT_EX splines with two endpoints specified in polar coordinates and a single control point defined by 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.

```

923 \def\horoputcurve#1#2#3#4{%
924 \begingroup
925 \edef\horo@pc@r@one{#1}\edef\horo@pc@th@one{#2}%
926 \edef\horo@pc@rtw@{#3}\edef\horo@pc@thtw@{#4}%

```

Both θ coordinates are subjected to `\horochartrotate`.

```
927 \dimen@=\horopc@th@ne\p@\relax\horochartrotate
928 \edef\horopc@th@ne{\TG@rem@pt\dimen@}%
929 \dimen@=\horopc@t@tw@\p@\relax\horochartrotate
930 \edef\horopc@t@tw@\{\TG@rem@pt\dimen@}%
```

Convert first endpoint to Cartesian.

```
931 \horopolarconvert\horopc@r@ne\horopc@th@ne\horodim@xa\horodim@ya
```

Convert second endpoint to Cartesian.

```
932 \horopolarconvert\horopc@r@tw@\horopc@t@tw@\horodim@xb\horodim@yb
```

Compute the polar-coordinate midpoint. Radius is simple average, but angle must go through `\horomidpoint` to handle cases like warapping around 360° .

```
933 \dimen@=\horopc@r@ne\p@\relax
934 \advance\dimen@ by\horopc@r@tw@\p@\relax
935 \divide\dimen@ by2\relax
936 \edef\horopc@r@mid{\TG@rem@pt\dimen@}%
937 \horomidpoint\horopc@th@ne\horopc@t@tw@
938 \edef\horopc@th@mid{\TG@rem@pt\dimen@}%
```

Convert midpoint to Cartesian.

```
939 \horopolarconvert\horopc@r@mid\horopc@th@mid\horodim@x\horodim@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 `eepic`'s spline curves are based on the Chaikin technique of cutting corners off a polyline until it looks like a smooth curve. 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.

```
940 \multiply\horodim@x by -4\relax
941 \advance\horodim@x by \horodim@xa\relax
942 \advance\horodim@x by \horodim@xb\relax
943 \divide\horodim@x by -2\relax
944 \multiply\horodim@y by -4\relax
945 \advance\horodim@y by \horodim@ya\relax
946 \advance\horodim@y by \horodim@yb\relax
947 \divide\horodim@y by -2\relax
```

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

```
948 \spline(\TG@rem@pt\horodim@xa,\TG@rem@pt\horodim@ya)%
949 (\TG@rem@pt\horodim@x,\TG@rem@pt\horodim@y)%
950 (\TG@rem@pt\horodim@xb,\TG@rem@pt\horodim@yb)%
951 \endgroup
952 }
```

`\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`.

```
953 \def\horoconncurve#1#2#3{%
954   \horoputcurve{#1}{\csname horo#3Pos\endcsname}%
955             {#2}{\csname horo#3DPos\endcsname}%
956 }
```

§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.

```
957 \def\horo@getobjdpos#1{%
958   \dimen@\csname horo#1DPos\endcsname\p@
959   \horo@chartrotate
960 }
```

`\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.

```
961 \def\horo@getobjsdms#1{%
962   \expandafter\dimen@\csname horo#1Pos\endcsname\p@
963   \horo@d@sdms
964   \horo@getobjdpos{#1}%
965 }
```

§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.

```
966 \def\horo@pr#1#2#3{%
967   \horo@getobjdpos{#3}%
968   \horoputradial{#1}{#2}{\TG@rem@pt\dimen@}%
969 }
```

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

```
970 \def\horoputradials#1#2#3{%
971   \horoforeach{#1}{\horo@pr{#2}{#3}}%
972 }
```


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

```

973 \def\horo@pt{%
974   \ifdim\dimen@<360\p@
975     {\horo@chartrotate
976       \horoputradial{\horo@pta}{\horo@ptb}{\TG@rem@pt\dimen@}}%
977     \advance\dimen@ by \horo@ptc\p@
978     \expandafter\horo@pt
979   \fi
980 }

```

`\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.

```

981 \def\horoputticks#1#2#3{%
982   \begingroup
983     \edef\horo@pta{#1}\edef\horo@ptb{#2}\edef\horo@ptc{#3}%
984     \dimen@=\z@
985     \horo@pt
986   \endgroup
987 }

```

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

```

988 \def\horo@pa#1#2#3{%
989   \horo@getobjdpos{#3}%
990   \horoputarrowhead{#1}{\TG@rem@pt\dimen@}{#2}%
991 }

```

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

```

992 \def\horoputarrows#1#2#3{%
993   \horoforeach{#1}{\horo@pa{#2}{#3}}%
994 }

```

`\horo@psk` Helper for `\horoputsignkey`. This is another tail-recursive loop with `\count0` as the loop counter, going from 0 to 11.⁸ 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.

```

995 \def\horo@psk{
996   \ifnum\count0<12\relax
997     \dimen@=30pt\relax
998     \dimen@\count0\dimen@\relax
999     \advance\dimen@ by 15pt\relax
1000    \horo@chartrotate
1001    \advance\count0 by 1%
1002    \horo@putpolar{\horo@radius}{\TG@rem@pt\dimen@}%
1003                {\Zodiac{\the\count0}}%
1004    \expandafter\horo@psk
1005  \fi
1006 }

```

⁸Well, it's one louder, isn't it?

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

```
1007 \def\horoputsignkey#1{%
1008   \begingroup
1009     \def\horo@radius{#1}%
1010     \count0=0\relax
1011     \horo@psk
1012   \endgroup
1013 }
```

`\horoputcusplabel` 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).

```
1014 \def\horoputcusplabel#1#2#3{%
1015   \begingroup
1016     \horo@getobjsdms{#3}%
```

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

```
1017   \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.

```
1018   \def\horo@pcl@s{W}%
1019   \ifdim\dimen@<135\p@\def\horo@pcl@s{T}\fi
1020   \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.

```
1021   \begingroup
1022     \advance\dimen@ by -#2\p@\horo@fixdimen@
1023     \horo@putpolar{#1}{\TG@rem@pt\dimen@}{%
1024       \expandafter\if\horo@pcl@s T%
1025       \horo@d@min\horominutes%
1026     \else
1027       \horo@d@deg\horodegrees%
1028     \fi
1029   }%
1030   \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.

```
1031   \advance\dimen@ by #2\p@\horo@fixdimen@
```

```

1032 \horo@putpolar{#1}{\TG@rem@pt\dimen@}{%
1033 \expandafter\if\horo@pcl@s T%
1034 \horo@d@deg\horodegrees%
1035 \else
1036 \horo@d@min\horominutes%
1037 \fi
1038 }%
1039 \endgroup
1040 }

```

§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.

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

```

1041 \def\horo@putobjsymbol#1#2{%
1042 \begingroup
1043 \horo@getobjdpos{#2}%
1044 \horo@putpolar{#1}{\TG@rem@pt\dimen@}{\csname #2Symbol\endcsname}%
1045 \endgroup
1046 }

```

`\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° mark⁹ as possible.

```

1047 \def\horo@pol@i#1#2#3{%
1048 \begingroup
1049 \dimen1=#1\p@\advance\dimen1by#2\p@
1050 \horo@getobjsdms{#3}%
1051 \def\horo@pol@s{H}%
1052 \ifdim\dimen@<45\p@\def\horo@pol@s{R}\fi
1053 \ifdim\dimen@>225\p@\def\horo@pol@s{R}\fi
1054 }

```

`\horo@putobjdeglabel` 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.

```

1055 \def\horo@putobjdeglabel#1#2#3{%
1056 \horo@pol@i{#1}{#2}{#3}%
1057 \expandafter\if\horo@pol@s H%
1058 \horo@putpolar{#1}{\TG@rem@pt\dimen@}{\Zodiac{\horo@d@sign}}%
1059 \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen@}%
1060 {\horo@d@deg\horodegrees}%
1061 \else
1062 \horo@putpolar{#1}{\TG@rem@pt\dimen@}{\horo@d@deg\horodegrees}%

```

⁹Great A’Tuin’s left rear flipper.

```

1063     \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen@}%
1064         {\Zodiac{\horo@d@sign}}}%
1065     \fi
1066 \endgroup
1067 }

```

`\horo@putobjminlabel` Much the same as `\horo@putobjdegreelabel` but makes a three-chunk label that reads degrees, sign, minutes.

```

1068 \def\horo@putobjminlabel#1#2#3{%
1069   \horo@pol@i{#1}{#2}{#3}%
1070   \expandafter\if\horo@pol@s H%
1071     \horo@putpolar{#1}{\TG@rem@pt\dimen@}%
1072         {\horo@d@min\horominutes}%
1073     \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen@}%
1074         {\Zodiac{\horo@d@sign}}}%
1075     \advance\dimen1by#2\p@
1076     \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen@}%
1077         {\horo@d@deg\horodegrees}%
1078   \else
1079     \horo@putpolar{#1}{\TG@rem@pt\dimen@}%
1080         {\horo@d@deg\horodegrees}%
1081     \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen@}%
1082         {\Zodiac{\horo@d@sign}}}%
1083     \advance\dimen1by#2\p@
1084     \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen@}%
1085         {\horo@d@min\horominutes}%
1086   \fi
1087 \endgroup
1088 }

```

`\horo@putobjseclabel` Typesets a four-chunk label reading degrees, sign, minutes, seconds.

```

1089 \def\horo@putobjseclabel#1#2#3{%
1090   \horo@pol@i{#1}{#2}{#3}%
1091   \expandafter\if\horo@pol@s H%
1092     \horo@putpolar{#1}{\TG@rem@pt\dimen@}%
1093         {\horo@d@sec\horoseconds}%
1094     \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen@}%
1095         {\horo@d@min\horominutes}%
1096     \advance\dimen1by#2\p@
1097     \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen@}%
1098         {\Zodiac{\horo@d@sign}}}%
1099     \advance\dimen1by#2\p@
1100     \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen@}%
1101         {\horo@d@deg\horodegrees}%
1102   \else
1103     \horo@putpolar{#1}{\TG@rem@pt\dimen@}%
1104         {\horo@d@deg\horodegrees}%
1105     \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen@}%
1106         {\Zodiac{\horo@d@sign}}}%
1107     \advance\dimen1by#2\p@
1108     \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen@}%
1109         {\horo@d@min\horominutes}%
1110     \advance\dimen1by#2\p@
1111     \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen@}%

```

```

1112             {\horo@d@sec\horoseconds}%
1113     \fi
1114 \endgroup
1115 }

```

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

```

1116 \def\horoputrxlabel#1#2{%
1117   \begingroup
1118     \dimen@ \csname horo#2Vel\endcsname \p@
1119     \ifdim\dimen@ < \z@ \relax
1120       \horo@getobjdpos{#2}%
1121       \horo@putpolar{#1}{\TG@rem@pt\dimen@}{\horoRetrogradeSymbol}%
1122     \fi
1123 \endgroup
1124 }

```

`\horoputsmartlabel` 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@psl` with either `\horo@lblone` or `\horo@lbltwo` and a Q (which is still exotic, remember) to aid in parsing.

```

1125 \def\horoputsmartlabel#1{%
1126   \begingroup
1127     \horo@getobjsdms{#1}%
1128     \def\horo@psl@o{#1}%
1129     \def\horo@pol@s{H}%
1130     \ifdim\dimen@ < 45 \p@ \def\horo@pol@s{R} \fi
1131     \ifdim\dimen@ > 225 \p@ \def\horo@pol@s{R} \fi
1132     \dimen1 = \horo@outerrad \p@ \relax
1133     \expandafter \if \horo@pol@s H%
1134       \expandafter \horo@psl \horo@lblone Q%
1135     \else
1136       \expandafter \horo@psl \horo@lbltwo Q%
1137     \fi

```

```
1138 \endgroup
1139 }
```

`\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`.

```
1140 \def\horo@psl#1{%
1141   \def\horo@psl@a{#1}%
1142   \ifx\horo@psl@a\horo@cue\relax
1143   \else
1144     \horo@putpolar{\TG@rem@pt\dimen1}{\TG@rem@pt\dimen@}%
1145     {\horo@psl@#1}%
1146     \advance\dimen1by-\horo@delta\p@\relax
1147     \expandafter\horo@psl
1148   \fi
1149 }
```

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

```
1150 \def\horo@psl@#1{%
1151   \if#1d\relax\horo@d@deg\horodegrees\fi
1152   \if#1m\relax\horo@d@min\horominutes\fi
1153   \if#1s\relax\horo@d@sec\horoseconds\fi
1154   \if#1z\relax\Zodiac{\horo@d@sign}\fi
1155   \if#1y\relax\csname \horo@psl@o Symbol\endcsname\fi
1156   \if#1r\relax
1157     \begingroup
1158       \dimen@\csname \horo@psl@o Vel\endcsname\p@
1159       \ifdim\dimen@<\z@\relax\horoRetrogradeSymbol\fi
1160     \endgroup
1161   \fi
1162 }
```

`\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.

```
1163 \def\horoscanlabels#1{%
1164   \horo@scanlabels@ #1/#1/xQ%
1165 }
```

`\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 `\horo@lbl1len`, and may be used to automatically adjust radius step and text size.

```

1166 \def\horo@scanlabels@#1/#2/#3Q{%
1167   \def\horo@lblone{#1}%
1168   \def\horo@lbltwo{#2}%
1169   \begingroup
1170     \count0=0\relax
1171     \horo@scanlabels@@#1Q%
1172     \edef\horo@lble{the\count0}%
1173     \horo@fterdef\horo@lble
1174   \endgroup
1175 }

```

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

```

1176 \def\horo@scanlabels@@#1{%
1177   \def\horo@sls@@a{#1}%
1178   \ifx\horo@sls@@a\horo@cue\relax
1179   \else
1180     \advance\count0by1\relax
1181     \expandafter\horo@scanlabels@@
1182   \fi
1183 }

```

`\horosetsmartradii` Sets up the radii for `\horoputsmarlabel` 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 $\#2 - n \cdot \#3$ 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.

```

1184 \def\horosetsmartradii#1#2#3{
1185   \def\horo@outerrad{#1}%
1186   \begingroup
1187     \dimen@=#3\p@\relax
1188     \multiply\dimen@ by -\horo@lble\relax
1189     \advance\dimen@ by #2\p@\relax
1190     \edef\horo@delta{\TG@rem@pt\dimen@}%
1191     \horo@fterdef\horo@delta
1192   \endgroup
1193 }

```

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.

```
1194 \fi
1195 %% MAKING Q MUNDANE HERE!
1196 \catcode'\Q=11
1197 \ifhoro@wheels
```

§8.1. Configuration settings.—

`\horoaspectobjectsa` List of objects that can appear on one side of an aspect relation. By default, all objects.

```
1198 \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.

```
1199 \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.

```
1200 \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.

```
1201 \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.

```
1202 \def\horoConjunctionAngle{0}
1203 \def\horoOppositionAngle{180}
1204 \def\horoTrineAngle{120}
1205 \def\horoSquareAngle{90}
1206 \def\horoQuintileAngle{72}
1207 \def\horoBiquintileAngle{144}
1208 \def\horoSextileAngle{60}
1209 \def\horoQuincunxAngle{150}
1210 \def\horoSemisextileAngle{30}
1211 \def\horoSemisquareAngle{45}
1212 \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.

```
1213 \def\horoConjunctionOrb{6}
1214 \def\horoOppositionOrb{6}
1215 \def\horoTrineOrb{5}
1216 \def\horoSquareOrb{5}
```



```

1217 \def\horoQuintileOrb{2}
1218 \def\horoBiquintileOrb{2}
1219 \def\horoSextileOrb{4}
1220 \def\horoQuincunxOrb{3}
1221 \def\horoSemisextileOrb{3}
1222 \def\horoSemisquareOrb{2}
1223 \def\horoSesquiquadrateOrb{2}

```

§8.3. Drawing the aspect web.—

`\horoaspect` 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.

```

1224 % radius theta1 theta2 symbol
1225 \def\horoaspect#1#2#3#4{%
1226   \begingroup
1227   \edef\p@one{#1}\edef\p@two{#2}\edef\p@three{#3}%
      Apply chart rotation to both  $\theta$  coordinates:
1228   \dimen@=\p@two\p@relax\horo@chartrotate\edef\p@two{\TG@rem@pt\dimen@}%
1229   \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 as-
      pect:
1230   \horo@polarconvert\p@one\p@two\horo@dim@xa\horo@dim@ya
1231   \horo@polarconvert\p@one\p@three\horo@dim@xb\horo@dim@yb
1232   \drawline(\TG@rem@pt\horo@dim@xa,\TG@rem@pt\horo@dim@ya)%
1233           (\TG@rem@pt\horo@dim@xb,\TG@rem@pt\horo@dim@yb)%
      Compute the midpoint by doing an average on the Cartesian coordinates:
1234   \advance\horo@dim@xa by\horo@dim@xb
1235   \advance\horo@dim@ya by\horo@dim@yb
1236   \divide\horo@dim@xa by2\divide\horo@dim@ya by2%
      Put the symbol at the midpoint and end.
1237   \put(\TG@rem@pt\horo@dim@xa,\TG@rem@pt\horo@dim@ya){\makebox(0,0){#4}}%
1238   \endgroup
1239 }

```

`\horoaa` 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`.

```

1240 \def\horoaa#1#2#3#4#5#6{%
1241   \dimen@\csname horo#5Pos\endcsname\p@
1242   \advance\dimen@ by -\csname horo#6Pos\endcsname\p@
1243   \horo@fixdimen@diff
1244   \ifdim\dimen@<\z@\relax\multiply\dimen@ by -1\relax\fi
1245   \advance\dimen@ by -#1\p@
1246   \horo@fixdimen@diff
1247   \ifdim\dimen@<\z@\relax\multiply\dimen@ by -1\relax\fi
1248   \ifdim\dimen@<#2\p@\relax
1249     \horoaspect{#3}{\csname horo#5Pos\endcsname}%
1250     {\csname horo#6Pos\endcsname}{#4}%

```

```
1251 \fi
1252 }
```

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

```
1253 \def\horoautoaspect#1#2#3#4#5#6{%
1254   \begingroup
1255   \horo@dblforeach{#1}{#2}{\horo@aa{#3}{#4}{#5}{#6}}%
1256   \endgroup
1257 }
```

`\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.

```
1258 \def\horo@aa#1#2{%
1259   \horoautoaspect{\horoaspectobjectsa}{\horoaspectobjectsb}%
1260   {\csname horo#2Angle\endcsname}{\csname horo#2Orb\endcsname}%
1261   {#1}{\csname horo#2Symbol\endcsname}%
1262 }
```

`\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.

```
1263 \def\horoautoaspects#1{%
1264   \horoforeach{\horoaspects}{\horo@aa{#1}}%
1265 }
```

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.

```
1266 \newif\ifhorointhouselabels\horointhouselabelsfalse
```

Allocate a L^AT_EX counter for which house we’re in.

```
1267 \newcounter{horo@house}
```

`\horo@house@label` 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.

```
1268 \newcommand{\horo@house@label}{%
1269   \Roman{horo@house}}%
1270 }
```

`\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.

```
1271 \def\horo@houses{CuspIM,CuspIIM,CuspIIIM,CuspIVM,CuspVM,CuspVIM,%
1272             CuspVIIIM,CuspVIIIIM,CuspIXM,CuspXM,CuspXIM,CuspXIIM}
```

`\horo@findcuspmid` 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).

```
1273 \def\horo@findcuspmid#1#2{%
1274   \horo@midpoint{\csname horoCusp#1DPos\endcsname}%
1275                 {\csname horoCusp#2DPos\endcsname}%
1276   \expandafter\edef\csname horoCusp#1MPos\endcsname{\TG@rem@pt\dimen@}%
1277 }
```

`\horo@findcuspmins` 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.

```
1278 \def\horo@findcuspmins{%
1279   \horo@findcuspmid{I}{II}\horo@findcuspmid{II}{III}%
1280   \horo@findcuspmid{III}{IV}\horo@findcuspmid{IV}{V}%
1281   \horo@findcuspmid{V}{VI}\horo@findcuspmid{VI}{VII}%
1282   \horo@findcuspmid{VII}{VIII}\horo@findcuspmid{VIII}{IX}%
1283   \horo@findcuspmid{IX}{X}\horo@findcuspmid{X}{XI}%
1284   \horo@findcuspmid{XI}{XII}\horo@findcuspmid{XII}{I}%
1285 }
```

`\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.

```
1286 \def\horo@pihl#1#2{%
1287   \begingroup
1288     \horo@getobjdpos{#2}%
1289     \horo@putpolar{#1}{\TG@rem@pt\dimen@}{\horohouselabel}%
1290     \stepcounter{horohouse}%
1291   \endgroup
1292 }
```

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

```
1293 \def\horoputinthouselabels#1{%
1294   \setcounter{horohouse}{1}%
1295   \horoforeach{\horo@houses}{\horo@pihl{#1}}%
1296 }
```

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.—

<code>\horosignificantadj</code>	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. <code>1297 \def\horosignificantadj{0.1}</code>
<code>\horocuspadjusttrigger</code>	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. <code>1298 \def\horocuspadjusttrigger{65}</code>
<code>\horoadjcycles</code>	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. <code>1299 \def\horoadjcycles{30}</code>
<code>\horoposattobj</code>	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.

```
1300 \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.

```
1301 \def\horoposattcusp{7}
```

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

```
1302 \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.¹⁰ 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.

```
1303 \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.

```
1304 \def\horoomindist{6.0}
```

`\horoo mindist` Object/cusp minimum distance.

```
1305 \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.

```
1306 \fi
```

```
1307 \newif\ifhoro@djusted\relax
```

```
1308 \newif\ifhoro@djustcusps\relax
```

```
1309 \ifhoro@wheels
```

```
1310 \countdef\horo@i=1\relax
```

¹⁰Don'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.

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

`\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@`.

```
1311 \def\horo@findpdiff#1#2{%
1312   \dimen@=\csname horo#1Pos\endcsname\p@
1313   \multiply\dimen@ by -1\relax
1314   \advance\dimen@ by\csname horo#2Pos\endcsname\p@
1315   \horo@fixdimen@diff
1316 }
```

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

```
1317 \def\horo@pplycorrection#1{%
1318   \advance\dimen@ by \csname horo#1DPos\endcsname\p@
1319   \horo@fixdimen@
1320   \expandafter\edef\csname horo#1DPos\endcsname{\TG@rem@pt\dimen@}%
1321 }
```

`\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.

```
1322 \def\horo@djo@#1#2#3{%
1323   \horo@findpdiff{#3}{#2}%
1324   \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.

```
1325   \ifdim\horo@dim@x<45\p@\relax
1326     \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.

```
1327     \horo@findpdiff{#3D}{#2D}%
1328     \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.

```
1329     \advance\dimen@ by -#1\p@
1330     \ifdim\dimen@>\z@
1331       \dimen@=\z@
1332     \else
```

```

1333     \ifdim\dimen@<-\horomaxrepulsion\p@\relax
1334     \dimen@=-\horomaxrepulsion\p@%
1335     \fi
1336     \divide\dimen@ by\hororepulsion\relax
1337     \fi

```

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

```

1338     \ifdim\horo@dim@x>\z@\multiply\dimen@ by -1\relax\fi
    Apply the correction and end.
1339     \horo@plycorrection{#2}%
1340     \fi
1341     \fi
1342 }

```

`\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@`.

```

1343 \def\horo@djoo#1#2{%
1344   \edef\horo@tmpa{#1}\edef\horo@tmpb{#2}%
1345   \ifx\horo@tmpa\horo@tmpb\else
1346     \horo@djo@{\horoomindist}{#1}{#2}%
1347   \fi
1348 }

```

`\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.

```

1349 \def\horo@djoo@same#1#2{%
1350   \horo@findpdiff{#2D}{#2}%
1351   \divide\dimen@ by#1\relax
1352   \horo@plycorrection{#2}%
1353 }

```

`\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`.

```

1354 \def\horo@djcheckmovement#1{%
1355   \horo@findpdiff{#1D}{#1S}%
1356   \ifdim\dimen@<0\p@\relax\multiply\dimen@ by -1\relax\fi
1357   \ifdim\dimen@>\horosignificantadj\p@\relax\horo@djustedtrue\fi
1358 }

```

§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.

```

1359 \def\horo@djobcycle{%
1360   \message{\the\horo@i}%
1361   \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.

```

1362   \horocopyvar{\horoobjects}{DPos}{SPos}%

```

Adjust objects against objects.

```

1363   \horo@dblforeach{\horoobjects}{\horoobjects}{\horo@djo}%

```

Adjust objects against cusps. This is one-sided: objects move to accomodate cusps, but cusps do not move to accomodate 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.

```

1364   \horo@dblforeach{\horoobjects}{\horocusps}{\horo@djo@{\horocmindist}}%

```

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

```

1365   \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.

```

1366   \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.

```

1367   \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.

```

1368   \ifnum\horoadjcycles>\horo@i\relax
1369     \advance\horo@i by1\relax
1370     \expandafter\horo@djobcycle
1371   \fi
1372 }

```

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

```

1373 \def\horo@djcuspcycle{%
1374   \message{\the\horo@i}%
1375   \horo@djustedfalse
1376   \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@djo because it has the added handling for not adjusting anything against itself.

```

1377   \horo@dblforeach{\horocusps}{\horoobjects}{\horo@djo@{\horocmindist}}%
1378   \horo@dblforeach{\horocusps}{\horocusps}{\horo@djo}%
1379   \horo@dblforeach{\horoobjects}{\horoobjects}{\horo@djo}%
1380   \horo@dblforeach{\horoobjects}{\horocusps}{\horo@djo@{\horocmindist}}%

```


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.

```
1381 \horoforeach{\horoobjects}{\horo@djoo@same{\horoposattobj}}%
1382 \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.

```
1383 \horoforeach{\horoobjects,\horocusps}{\horo@djcheckmovement}%
1384 \ifhoro@djusted\else\horo@i=\horoadjcycles\relax\fi
1385 \ifnum\horoadjcycles>\horo@i\relax
1386   \advance\horo@i by1\relax
1387   \expandafter\horo@djcuspcycle
1388 \fi
1389 }
```

§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.

```
1390 \def\horo@djsavedpos#1{%
1391   \expandafter\horo@fterdef\csname horo#1DPos\endcsname
1392 }
```

`\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.

```
1393 \def\horo@djcheckcusps#1#2#3{%
1394   \edef\horo@tmpa{#2}\edef\horo@tmpb{#3}%
1395   \ifx\horo@tmpa\horo@tmpb
1396   \else
1397     \horo@findpdiff{#2D}{#3D}%
1398     \ifdim\dimen@<\z@ \multiply\dimen@ by-1\relax\fi
1399     \ifdim\dimen@<45\p@ \relax
1400       \multiply\dimen@ by 100\relax
1401       \divide\dimen@ by \horocuspadjusttrigger\relax
1402       \ifdim\dimen@<#1\p@ \relax\horo@djustcuspstrue\fi
1403   \fi
1404 \fi
1405 }
```

§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.

```

1406 \def\horoadjust{%
1407   \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.

```

1408   \ifhorointhouselabels
1409     \horo@findcuspmids
1410     \horocopyvar{\horocusps}{MPos}{MDPos}%
1411     \let\horo@savobjlist\horoobjects
1412     \edef\horoobjects{\horoobjects,\horo@houses}%
1413   \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.

```

1414   \begingroup
1415   \message{(%}
1416   \horo@i=1\relax

```

Do the actual adjustment cycles for objects only.

```

1417   \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.

```

1418   \horo@djustcuspsfalse
1419   \horo@dblforeach{\horoobjects}{\horoobjects}%
1420     {\horo@djcheckcusps{\horoomindist}}%
1421   \horo@dblforeach{\horoobjects}{\horocusps}%
1422     {\horo@djcheckcusps{\horoomindist}}%

```

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.

```

1423   \ifhoro@djustcusps
1424     \message{C}%
1425     \horo@i=1\relax
1426     \horo@djcuspcycle
1427   \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.

```

1428     \message{)}%
1429     \horoforeach{\horocusps,\horoobjects}{\horo@djsavedpos}%
1430     \endgroup
    Finally, restore the \horoobjects list if we tampered with it earlier.
1431     \ifhoro@inhouselabels\let\horoobjects\horo@savedobjlist\fi
1432     \fi
1433 }

    At this point we end the \ifhoro@wheels conditional.
1434 \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.

```
1435 \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.

```

1436 \def\horo@isclockwise#1#2{%
1437   TT\fi
1438   \begingroup
1439   \horo@findpdiff{#1}{#2}%
1440   \edef\horo@data{\the\dimen@}%
1441   \horo@fterdef\horo@data
1442   \endgroup
1443   \ifdim\horo@data>\z@\relax
1444 }

```

`\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.

```

1445 \def\horo@anghouse#1{%
1446   TT\fi
1447   \begingroup
1448   \def\horo@angh@{F}%
1449   \if\horo@isclockwise{CuspI}{#1}\if\horo@isclockwise{#1}{CuspII}%
1450   \def\horo@angh@{T}%
1451   \fi\fi
1452   \if\horo@isclockwise{CuspIV}{#1}\if\horo@isclockwise{#1}{CuspV}%
1453   \def\horo@angh@{T}%
1454   \fi\fi
1455   \if\horo@isclockwise{CuspVII}{#1}\if\horo@isclockwise{#1}{CuspVIII}%
1456   \def\horo@angh@{T}%

```

```

1457 \fi\fi
1458 \if\horo@isclockwise{CuspX}{#1}\if\horo@isclockwise{#1}{CuspXI}%
1459 \def\horo@angh@{T}%
1460 \fi\fi
1461 \horo@fterdef\horo@angh@
1462 \endgroup
1463 \if\horo@angh@ T\relax
1464 }

```

`\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`.

```

1465 \def\horo@succhouse#1{%
1466 TT\fi
1467 \beginingroup
1468 \def\horo@succh@{F}%
1469 \if\horo@isclockwise{CuspII}{#1}\if\horo@isclockwise{#1}{CuspIII}%
1470 \def\horo@succh@{T}%
1471 \fi\fi
1472 \if\horo@isclockwise{CuspV}{#1}\if\horo@isclockwise{#1}{CuspVI}%
1473 \def\horo@succh@{T}%
1474 \fi\fi
1475 \if\horo@isclockwise{CuspVIII}{#1}\if\horo@isclockwise{#1}{CuspIX}%
1476 \def\horo@succh@{T}%
1477 \fi\fi
1478 \if\horo@isclockwise{CuspXI}{#1}\if\horo@isclockwise{#1}{CuspXII}%
1479 \def\horo@succh@{T}%
1480 \fi\fi
1481 \horo@fterdef\horo@succh@
1482 \endgroup
1483 \if\horo@succh@ T\relax
1484 }

```

§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.

```

1485 \def\horo@montrealcurve#1#2#3{%
1486 \horo@midpoint{\csname horoCusp#2DPos\endcsname}%
1487 \csname horoCusp#3DPos\endcsname}%
1488 \horoputcurve{45}{\csname horoCusp#1Pos\endcsname}{15}{\TG@rem@pt\dimen@}%
1489 }

```

`\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.

```

1490 \def\horo@putmontrealobj#1{%
1491 \if\horo@succhouse{#1}%
1492 \horoputobjsymbol{42}{#1}%
1493 \horoputobjdeglabell{34}{4}{#1}%

```

```

1494   \horoputrxlabel{30}{#1}%
1495   \else
1496   \horoputobjsymbol{29}{#1}%
1497   \horoputobjdeglabel{21}{4}{#1}%
1498   \horoputrxlabel{17}{#1}%
1499   \fi
1500 }

```

§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.

```

1501 \def\horo@putqcobj#1{%
1502   \if\horo@anghouse{#1}%
1503     \horoputobjsymbol{28}{#1}%
1504     \horoputobjdeglabel{20}{4}{#1}%
1505     \horoputrxlabel{16}{#1}%
1506   \else
1507     \horoputobjsymbol{42}{#1}%
1508     \horoputobjdeglabel{34}{4}{#1}%
1509     \horoputrxlabel{30}{#1}%
1510   \fi
1511 }

```

`\horo@qccline` 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.

```

1512 \def\horo@qccline#1#2#3#4{%
1513   \begingroup
1514     \dimen@=\csname horoCusp#2\endcsname\p@\horo@chartrotate
1515     \dimen1=\dimen@\relax
1516     \dimen@=\csname horoCusp#4\endcsname\p@\horo@chartrotate
1517     \horoputline{#1}{\TG@rem@pt\dimen1}%
1518                   {#3}{\TG@rem@pt\dimen@}%
1519   \endgroup
1520 }

```

§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:

```

1521 \def\horo@dialstart#1#2{%
1522   \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.

```

1523 \begingroup
1524   \count0=#2\relax
1525   \edef\horodtone{\the\count0}%
1526   \multiply\count0 by 5\relax
1527   \edef\horodtfive{\the\count0}%
1528   \multiply\count0 by 2\relax
1529   \edef\horodtten{\the\count0}%
1530   \horofterdef\horodtone
1531   \horofterdef\horodtfive
1532   \horofterdef\horodtten
1533 \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.

```
1534 \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.

```

1535 \ifhoroaspectweb
1536   \horoautoaspects{#1}%
1537 \else
1538   \drawline(-1,0)(1,0)%
1539   \drawline(0,-1)(0,1)%
1540 \fi
1541 }

```

`\horo@dialwheel` Draw one wheel of a potentially multi-wheel dial chart. The argument #1 is the `\horooomindist` 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° .

```

1542 \def\horo@dialwheel#1#2#3#4#5#6#7{%
1543   \def\horooomindist{#1}\def\horooomindist{-90.0}%
1544   \put(0,0){\circle{#2}}%
1545   \def\hororightcoord{270}%

```

The rest is only meaningful if there is calculated data. We plot a set of radials to show objects' Pos values.

```

1546   \ifhorocalculated
1547     \horoputradials{\horoobjects}{#3}{#7}%

```

Do `\horoadjust` to come up with reasonable DPos values.

```
1548   \horoadjust
```

Then plot the connecting curves and the object symbols and end (closing off the conditional).

```

1549 \horoforeach{\horobjects}{\horconncurve{#4}{#5}}%
1550 \horoforeach{\horobjects}{\horputobjsymbol{#6}}%
1551 \fi
1552 }

```

§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.

```
1553 \def\horotextsize{0}
```

`\horochosetextsize` This makes some attempt to choose a reasonable text size based on `\horoblben`, 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 `\horoblben`.

```

1554 \def\horochosetextsize{%
1555 \count@=\horotextsize\relax
1556 \multiply\count@ by-1\relax
1557 \advance\count@ by\horoblben\relax

```

Based on the result, we choose a text size from `\Large` down to `\tiny`.

```

1558 \ifcase\the\count@\or\Large\or\large\or\normalsize\or\small\or
1559 \scriptsize\or\scriptsize\or\tiny\fi
1560 }

```

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.

```
1561 \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.

```
1562 \horoscanlabels{#1}%
1563 \horosetsmartradii{38}{4.666}{0.333}%
```

Set the default rounding mode, which is round to minutes keep degrees, if `\hororoundauto` is active.

```
1564 \ifhororoundauto\hororoundtominkeepdeg\fi
```

Draw the bounding circles (given by their diameter, which are twice their radii).

```
1565 \put(0,0){\circle{100}}%
1566 \put(0,0){\circle{90}}%
1567 \put(0,0){\circle{86}}%
1568 \put(0,0){\circle{42}}%
```

Draw the angle scale with its 1° and 5° ticks, and the sign key.

```
1569 \horoputticks{45}{5}{30}%
1570 \horoputticks{43}{1}{1}%
1571 \horoputticks{43}{2}{5}%
1572 \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.

```
1573 \ifhorocalculated
1574 \horoputradials{\horobjects}{41}{2}%
1575 \horoputradials{\horobjects}{21}{2}%
1576 \ifhorodrawcusps
1577 \horoputradials{\horocusps}{21}{1}%
```



```

1578     \horoputradials{\horocusps}{42}{1}%
1579     \fi
1580     \ifhoroboldangles
1581     \begingroup
1582     \allinethickness{\horoanglecuspswidth}%
1583     \horoputradials{\horoangularcusps}{21}{1}%
1584     \ifhoroanglearrows
1585     \horoputradials{\horoangularcusps}{42}{0.5}%
1586     \else
1587     \horoputradials{\horoangularcusps}{42}{1}%
1588     \fi
1589     \endgroup
1590     \fi
1591     \ifhoroanglearrows
1592     \horoputradials{\horoangularcusps}{43}{0.7}%
1593     \fi
1594     \fi

```

Compute the adjusted DPos values.

```

1595     \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.

```

1596     \ifhorocalculated
1597     \horoforeach{\horoobjects}{\horoconncurve{41}{40}}%
1598     \horoforeach{\horoobjects}{\horoconncurve{23}{24}}%
1599     \ifhorodrawcusps
1600     \horoforeach{\horocusps}{\horoconncurve{42}{42}}%
1601     \horoforeach{\horocusps}{\horoconncurve{22}{22}}%
1602     \horoputradials{\horocusps}{22}{20}%
1603     \fi
1604     \ifhoroboldangles
1605     \allinethickness{\horoanglecuspswidth}%
1606     \horoforeach{\horoangularcusps}{\horoconncurve{42}{42}}%
1607     \horoforeach{\horoangularcusps}{\horoconncurve{22}{22}}%
1608     \horoputradials{\horoangularcusps}{22}{20}%
1609     \thinlines
1610     \fi

```

Finally, the smart labels for objects, the aspect web, and the internal house labels.

```

1611     {\horochoosetextsize
1612     \horoforeach{\horoobjects}{\horoputsmartlabel}}%
1613     \ifhoroaspectweb\horoautoaspects{21}\fi
1614     \ifhoroinhouselabels\horoputinhouselabels{32}\fi
1615     \fi
1616 }

```

§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.

```
1617 \newcommand{\horowheelIqaluit}[1][1]{%
1618   \horo@dialstart{42}{#1}%
1619   \horoputticks{42}{1}{\horo@dtone}%
1620   \horoputticks{42}{2}{\horo@dtfive}%
1621   \horoputticks{42}{3}{\horo@dtten}%
1622   \horo@dialwheel{5.0}{84}{42}{45}{46}{47.5}{3}%
1623 }
```

`\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.

```
1624 \newcommand{\horowheelIgloolik}[3][1]{%
1625   \horo@dialstart{34}{#1}%
1626   #3\horocalcharmonic{#1}%
1627   \horoputticks{42}{1}{\horo@dtone}%
1628   \horoputticks{42}{2}{\horo@dtfive}%
1629   \horoputticks{42}{3}{\horo@dtten}%
1630   \horo@dialwheel{5.0}{84}{42}{45}{46}{47.5}{3}%
1631   #2\horocalcharmonic{#1}%
1632   \horoputticks{34}{1}{\horo@dtone}%
1633   \horoputticks{34}{2}{\horo@dtfive}%
1634   \horoputticks{34}{3}{\horo@dtten}%
1635   \horo@dialwheel{6.0}{68}{34}{37}{38}{39.5}{3}%
1636 }
```

`\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.

```
1637 \newcommand{\horowheelResolute}[4][1]{%
1638   \horo@dialstart{29}{#1}%
1639   #4\horocalcharmonic{#1}%
1640   \horoputticks{43}{1}{\horo@dtone}%
1641   \horoputticks{43}{2}{\horo@dtfive}%
1642   \horo@dialwheel{5.0}{86}{43}{45}{46}{47.5}{2}%
1643   #3\horocalcharmonic{#1}%
1644   \horoputticks{36}{1}{\horo@dtone}%
1645   \horoputticks{36}{2}{\horo@dtfive}%
1646   \horo@dialwheel{6.0}{72}{36}{38}{39}{40.5}{2}%
1647   #2\horocalcharmonic{#1}%
1648   \horoputticks{29}{1}{\horo@dtone}%
1649   \horoputticks{29}{2}{\horo@dtfive}%
1650   \horo@dialwheel{7.0}{58}{29}{31}{32}{33.5}{2}%
1651 }
```

`\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.

```

1652 \newcommand{\horowheelRankin}[5][1]{%
1653   \horo@dialstart{22}{#1}%
1654   #5\horocalcharmonic{#1}%
1655   \horoputticks{43}{1}{\horo@dtone}%
1656   \horoputticks{43}{2}{\horo@dtfive}%
1657   \horo@dialwheel{5.0}{86}{43}{45}{46}{47.5}{2}%
1658   #4\horocalcharmonic{#1}%
1659   \horoputticks{36}{1}{\horo@dtone}%
1660   \horoputticks{36}{2}{\horo@dtfive}%
1661   \horo@dialwheel{6.0}{72}{36}{38}{39}{40.5}{2}%
1662   #3\horocalcharmonic{#1}%
1663   \horoputticks{29}{1}{\horo@dtone}%
1664   \horoputticks{29}{2}{\horo@dtfive}%
1665   \horo@dialwheel{7.0}{58}{29}{31}{32}{33.5}{2}%
1666   #2\horocalcharmonic{#1}%
1667   \horoputticks{22}{1}{\horo@dtone}%
1668   \horoputticks{22}{2}{\horo@dtfive}%
1669   \horo@dialwheel{10.0}{44}{22}{24}{25}{26.5}{2}%
1670 }}

```

§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 accomodate 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.

```

1671 \newcommand{\horowheelMontreal}{%
1672   \def\horoomindist{10.0}\def\horoocmindist{6.0}%
      Set the default rounding mode, which is round to degrees keep sign, if
      \horo-roundauto is active.
1673   \ifhoro-roundauto\horo-roundto deg keepsign\fi
      Draw circles at radii 15 and 45.
1674   \put(0,0){\circle{90}}%
1675   \put(0,0){\circle{30}}%
      Put circular-reading labels at radius 47.5 for the four angles.
1676   \ifhoro-calculated
1677     \ifhoro-drawcusps
1678       \horo-foreach{\horo-angularcusps}{\horoputcusplabel{47.5}{5}}%
1679     \fi
1680   \fi

```

Do adjustment to find non-conflicting locations for everything else.

```
1681 \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 *cusps* 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.

```
1682 \ifhorocalculated
1683   \ifhorodrawcusps
1684     \horo@montrealcurve{I}{II}{III}%
1685     \horo@montrealcurve{IV}{II}{III}%
1686     \horo@montrealcurve{IV}{V}{VI}%
1687     \horo@montrealcurve{VII}{V}{VI}%
1688     \horo@montrealcurve{VII}{VIII}{IX}%
1689     \horo@montrealcurve{X}{VIII}{IX}%
1690     \horo@montrealcurve{X}{XI}{XII}%
1691     \horo@montrealcurve{I}{XI}{XII}%
```

Draw cusps separating cadent from angular houses. The end result looks like a sort of four-petaled flower, with two houses inside each petal and one more in each of the spaces between.

```
1692   \horoforeach{\horoangularcusps}{\horoconncurve{45}{15}}%
1693   \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.

```
1694   \horoforeach{\horoobjects}{\horo@putmontrealobj}%
1695   \fi
1696 }}
```

`\horowheelQuebecCity` As with the Montreal template, this one requires a fair bit of angular space between object labels.

```
1697 \newcommand{\horowheelQuebecCity}{%
1698   \def\horoocmindist{10.0}\def\horoocmindist{6.0}%
```

Set the default rounding mode, which is round to degrees keep sign, if `\horo-roundauto` is active.

```
1699   \ifhoro-roundauto\horo-roundto-deg-keep-sign\fi
```

Draw two circles for the outside; the cusp labels will go between them.

```
1700   \put(0,0){\circle{100}}%
1701   \put(0,0){\circle{90}}%
```

Do adjustment to find nice spacing for everything.

```
1702   \horoadjust
```

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.

```
1703 \ifhorocalculated
1704 \horo@findcuspmids
1705 \let\hororightcoord=\horoCuspVIIMPos
```

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).

```
1706 \ifhorodrawcusps
1707 \horoputradials{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.

```
1708 \horo@qcline{45}{IMPos}{20}{IIIDPos}%
1709 \horo@qcline{45}{IVMPos}{20}{IIIDPos}%
1710 \horo@qcline{45}{IVMPos}{20}{VIDPos}%
1711 \horo@qcline{45}{VIIMPos}{20}{VIDPos}%
1712 \horo@qcline{45}{VIIMPos}{20}{IXDPos}%
1713 \horo@qcline{45}{XMPos}{20}{IXDPos}%
1714 \horo@qcline{45}{XMPos}{20}{XIIDPos}%
1715 \horo@qcline{45}{IMPos}{20}{XIIDPos}%
```

Draw the square itself by connecting cusps of cadent houses.

```
1716 \horo@qcline{20}{IIIDPos}{20}{VIDPos}%
1717 \horo@qcline{20}{VIDPos}{20}{IXDPos}%
1718 \horo@qcline{20}{IXDPos}{20}{XIIDPos}%
1719 \horo@qcline{20}{XIIDPos}{20}{IIIDPos}%
1720 \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.

```
1721 \horocopyvar{\horocusps}{MPos}{DPos}%
1722 \ifhorodrawcusps
1723 \horoforeach{\horocusps}{\horoputcusplabel{47.5}{5}}%
1724 \fi
```

Finally, add the labels for the objects.

```
1725 \horoforeach{\horoobjects}{\horo@putqcobj}%
1726 \fi
1727}}
```

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.

```
1728 \fi
1729 \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;">General: egrep filtering option 8</p> <p style="padding-left: 40px;">Added smart rounding features 37</p> <p style="padding-left: 40px;">Added support for Krusinski houses with Swiss Ephemeris 79</p> <p style="padding-left: 40px;">Documentation on equal-house systems 16</p> <p style="padding-left: 40px;">Documentation on new shift</p>	<p>macros 19</p> <p>Fake SEOpt macros for always- calculated angles 79</p> <p>Label example to use rounding 27</p> <p><code>\horo@aa</code>: Fixed only-checking-one- direction bug 96</p> <p><code>\horomakeequalcusps</code>: New macro 65</p> <p><code>\horomakesigncusps</code>: New macro 65</p> <p><code>\horoshiftcusps</code>: New macro 65</p> <p><code>\horoshiftobjects</code>: New macro 64</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	\ChironSymbol .. 123, 240	H	
\@makeother	777	\hexstar	187
\@trimb	278, 281	\horo@aa	<u>1240</u> , 1255
\@trimc	281, 282	\horo@aaas ...	<u>1258</u> , 1264
A		\horo@angh@ ..	1448,
\afterassignment ..	282		1450, 1453, 1456,
\aftergroup ...	277,		1459, 1461, 1463
	284, 285, 428,	\horo@anghouse	<u>1445</u> , 1502
	431, 432, 444,	\horo@calc@a ...	662, 684
	448, 449, 662,	\horo@calc@b ...	566,
	664, 688, 694, 727		568, 577, 671, 675
\allinethickness ..		\horo@calc@c ...	663, <u>693</u>
 1582, 1605	\horo@calc@d ...	667, 683
\Aquarius	47, 71	\horo@calc@e ...	671, 683
\aquarius	171	\horo@calc@f ...	675, 683
\AquariusSymbol ...		\horo@calc@parse	689, <u>696</u>
	97, 111, 171, 202, 228	\horo@calc@parse@a	
\Aries	37, 61	 697, 699
\aries	161	\horo@calc@parse@b	
\AriesSymbol 699, <u>704</u>
	94, 101, 161, 192, 218	\horo@calc@parse@b@	
\ASC	250, 254	 705, 706
\AscendantSymbol	143, 254	\horo@calc@parse@c	
\astrosun	173	 706, <u>708</u>
B		\horo@calch ...	336, <u>338</u>
\begin	842	\horo@calcobj ..	<u>815</u> , 824
\bigtriangleup ...	185	\horo@calcobj@ ...	
\blacken	915		.. <u>798</u> , 816, 817, 825
C		\horo@chartrotate .	
\Cancer	40, 64	 <u>390</u> ,
\cancer	164		927, 929, 959,
\CancerSymbol		\DSC	252
	95, 104, 164, 195, 221	E	
\Capricorn	46, 70	\end	844
\CapricornSymbol ..		\ensuremath .	90–92, 185
	97, 110, 170, 201, 227	environments:	
\capricornus	170	horoscope	<u>840</u>
\catcode	273, 1196	G	
\centering	847	\Gemini	39, 63
\Ceres	241	\gemini	163
\CeresSymbol ...	124, 241	\GeminiSymbol	
\Chiron	240		94, 103, 163, 194, 220

\horo@d@dms ... 527 , 575	916, 920, 939–	\horo@djsavedpos ..
\horo@d@dms@ ... 494 ,	943, 949, 1324– 1390 , 1429
666, 669, 673, 800	1326, 1328, 1338	\horo@djustcuspsfalse
\horo@d@dms@@	\horo@dim@xa 1418
..... 485 , 496–	.. 863, 879, 881,	\horo@djustcuspstrue
498, 516, 520,	883, 890, 892, 1402
524, 535, 539, 543	901, 903, 906,	\horo@djustedfalse
\horo@d@min ... 497,	908, 911–913, 1361, 1375
520, 539, 552,	918, 931, 941,	\horo@djustedtrue 1357
560, 561, 577,	948, 1230, 1232,	\horo@dtfive
666, 667, 670,	1234, 1236, 1237	1527, 1531, 1620,
671, 674, 675,	\horo@dim@xb ... 865,	1628, 1633, 1641,
801, 805, 1025,	880, 881, 884,	1645, 1649, 1656,
1036, 1072, 1085,	891, 893, 902,	1660, 1664, 1668
1095, 1109, 1152	903, 905, 917,	\horo@dtone
\horo@d@sdms	932, 942, 950,	1525, 1530, 1619,
.. 500 , 550, 559, 963	1231, 1233, 1234	1627, 1632, 1640,
\horo@d@sec 498, 524,	\horo@dim@xc ... 867,	1644, 1648, 1655,
543, 552, 560,	907, 908, 910, 919	1659, 1663, 1667
561, 577, 801,	\horo@dim@y	\horo@dtten
805, 1093, 1112, 1153	.. 862, 872, 873, 1529, 1532,
\horo@d@sign 509, 510,	900, 902, 907,	1621, 1629, 1634
551, 1017, 1058,	914, 916, 920,	\horo@eathead .. 287 , 289
1064, 1074, 1082,	939, 944–947, 949	\horo@egrepfalse ... 17
1098, 1106, 1154	\horo@dim@ya	\horo@egreptrue 18
\horo@data . 460, 461,	.. 864, 879, 882,	\horo@fe
467, 773, 778–	883, 890, 892,	297 , 308
780, 786, 787,	901, 904, 905,	\horo@fe@a 306, 308
792, 1440, 1441, 1443	909, 910, 912,	\horo@fe@b 301, 307
\horo@dblforeach ..	914, 918, 931,	\horo@fe@c 298, 299
.... 326 , 1255,	945, 948, 1230,	\horo@findcuspmid .
1363, 1364, 1377–	1232, 1235–1237	... 1273 , 1279–1284
1380, 1419, 1421	\horo@dim@yb ... 866,	\horo@findcuspmids
\horo@ddms . 459 , 590–592	880, 882, 884,	... 1278 , 1409, 1704
\horo@delta	891, 893, 902,	\horo@findpdiff 1311 ,
... 1146, 1190, 1191	904, 906, 917,	1323, 1327, 1350,
\horo@dfe@a ... 318 , 328	932, 946, 950,	1355, 1397, 1439
\horo@dfe@b ... 310 , 322	1231, 1233, 1235	\horo@fixdimen@ ...
\horo@dfe@ction 314, 327	\horo@dim@yc ... 868, 341, 354,
\horo@dfe@x 314, 319, 320	907, 909, 911, 919	388, 392, 394 ,
\horo@dfe@xlist 327, 328	\horo@djcheckcusps	408, 1022, 1031, 1319
\horo@dfe@y 311, 312, 314	... 1393 , 1420, 1422	\horo@fixdimen@diff
\horo@dfe@ylist 322, 327	\horo@djcheckmovement 385,
\horo@dialstart 1354 , 1366, 1383	407 , 1243, 1246, 1315
.... 1521 , 1618,	\horo@djcuspccycle .	\horo@ft@a 284, 288
1625, 1638, 1653 1373 , 1426	\horo@fterdef
\horo@dialwheel ...	\horo@djo@ 1322 , 1346,	.. 283 , 417, 429,
.... 1542 , 1622,	1364, 1377, 1380	436, 437, 446,
1630, 1635, 1642,	\horo@djobccycle ..	453, 490, 510,
1646, 1650, 1657, 1359 , 1417	716, 721, 725,
1661, 1665, 1669	\horo@djoo ... 1343 ,	780, 1173, 1191,
\horo@dim@x ... 861,	1363, 1378, 1379	1391, 1441, 1461,
872, 873, 900,	\horo@djoo@same 1349 ,	1481, 1530–1532
902, 907, 913,	1365, 1381, 1382	\horo@getobjdpos ..
		.. 957 , 964, 967,

989, 1043, 1120, 1288	900, 901, 931,	\horo@scanlabels@@
\horo@getobjsdms ..	932, 939, 1230, 1231 1171, <u>1176</u>
<u>961</u> , 1016, 1050, 1127	\horo@ppendtmpfalse	\horo@shift 346, 349, <u>351</u>
\horo@houseletter 730, 822	\horo@sls@a . 1177, 1178
..... 756–768, 821	\horo@ppendtmptrue 813	\horo@stAquSign ... 636
\horo@housenumber .	\horo@pplycorrection	\horo@starfontfalse . 7
..... 638–651, 679	... <u>1317</u> , 1339, 1352	\horo@starfonttrue
\horo@houses	\horo@pr 11, 27
... <u>1271</u> , 1295, 1412	\horo@psk <u>995</u> , 1011	\horo@stAriSign ... 626
\horo@i	\horo@psl 1134, 1136, <u>1140</u>	\horo@stAscObj 614
1360, 1367–1369,	\horo@psl@ .. 1145, <u>1150</u>	\horo@stCanSign ... 629
1374, 1384–	\horo@psl@a . 1141, 1142	\horo@stCapSign ... 635
1386, 1416, 1425	\horo@psl@o	\horo@stCerObj 606
\horo@isclockwise 1128, 1155, 1158	\horo@stChiObj 605
<u>1436</u> , 1449, 1452,	\horo@pt	\horo@stDesObj 620
1455, 1458, 1469,	\horo@pta	\horo@stGemSign ... 628
1472, 1475, 1478	\horo@ptb	\horo@stJunObj 608
\horo@lblone . 1134, 1167	\horo@ptc	\horo@stJupObj 600
\horo@lbltwo . 1136, 1168	\horo@putmontrealobj	\horo@stLeoSign ... 630
\horo@marvosymfalse . 6 <u>1490</u> , 1694	\horo@stLibSign ... 632
\horo@marvosymtrue . 10	\horo@putpolar	\horo@stLilObj 613
\horo@midpoint	<u>869</u> , 1002, 1017,	\horo@stMarObj 599
<u>382</u> , 937, 1274, 1486	1023, 1032, 1044,	\horo@stMerObj 597
\horo@montrealcurve	1058, 1059, 1062,	\horo@stMidObj 623
... <u>1485</u> , 1684–1691	1063, 1071, 1073,	\horo@stMooObj 596
\horo@nsew	1076, 1079, 1081,	\horo@stNadObj 617
.. <u>564</u> , 574, 668, 672	1084, 1092, 1094,	\horo@stNepObj 603
\horo@outerrad 1132, 1185	1097, 1100, 1103,	\horo@stNodObj 610
\horo@pa	1105, 1108, 1111,	\horo@stPalObj 607
\horo@pa@r 898, 900	1121, 1144, 1289	\horo@stPisSign ... 637
\horo@pa@sc@le . 899, 901	\horo@putqcbj <u>1501</u> , 1725	\horo@stPluObj 604
\horo@pa@th 898, 900, 901	\horo@qcline	\horo@strologfalse
\horo@pc@r@ne <u>1512</u> , 1708–1719 12, 14, 16
..... 925, 931, 933	\horo@r@offdig <u>471</u>	\horo@strologtrue .. 15
\horo@pc@r@mid .. 936, 939	\horo@r@offdigit ..	\horo@stSagSign ... 634
\horo@pc@rtw@ 473–483, 502,	\horo@stSatObj 601
..... 926, 932, 934	513, 517, 521,	\horo@stScoSign ... 633
\horo@pc@th@ne . 925,	529, 532, 536, 540	\horo@stSunObj 595
927, 928, 931, 937	\horo@r@offset . <u>470</u> ,	\horo@stTauSign ... 627
\horo@pc@th@mid . 938, 939	474–483, 503,	\horo@stUraObj 602
\horo@pc@th@tw@ . 926,	514, 518, 522,	\horo@stVenObj 598
929, 930, 932, 937	530, 533, 537, 541	\horo@stVesObj 609
\horo@pcl@s .. 1018–	\horo@radius . 1002, 1009	\horo@stVirSign ... 631
1020, 1024, 1033	\horo@readdata	\horo@succh@ .. 1468,
\horo@pihl .. <u>1286</u> , 1295 <u>771</u> , 785, 791	1470, 1473, 1476,
\horo@pol@i .. <u>1047</u> ,	\horo@readpos	1479, 1481, 1483
1056, 1069, 1090 <u>784</u> , 790, 828	\horo@succhouse ...
\horo@pol@s	\horo@readposvel <u>789</u> , 827 <u>1465</u> , 1491
.... 1051–1053,	\horo@savecount@ ..	\horo@svc@a ... 447, <u>452</u>
1057, 1070, 1091,	.. 375, 380, 395, 405	\horo@svc@b ... 448, <u>455</u>
1129–1131, 1133	\horo@savobjlist	\horo@svo@a ... 430, <u>435</u>
\horo@polarconvert 1411, 1431	\horo@svo@b ... 431, <u>439</u>
.. <u>291</u> , 872, 879,	\horo@scanlabels@ .	\horo@swetestfalse
880, 890, 891, 1164, <u>1166</u> 14, 15

<code>\horo@swetesttrue</code>	13, 16	<code>\horoautoaspects</code>	..	<code>\horocuspjusttrigger</code>		
<code>\horo@templatesfalse</code>		...	1263 , 1536, 1613	1298 , 1401	
.....	21, 22	<code>\horoAxialHouses</code>	..	757	<code>\horoCuspIIIPos</code>	... 361
<code>\horo@templatestrue</code>	20	<code>\horoAzimuthalHouses</code>	758	<code>\horoCuspIIPos</code>	360
<code>\horo@textsymfalse</code>	.. 4	<code>\horoBiquintileAngle</code>		359	
<code>\horo@textsymtrue</code>	... 8	1207	<code>\horoCuspIVPos</code>	362
<code>\horo@tmp</code> 502,	<code>\horoBiquintileOrb</code>	1218	<code>\horoCuspIXPos</code>	367
513, 517, 521,		<code>\horoBiquintileSymbol</code>		<code>\horocusps</code>	332 ,
529, 532, 536, 540		152, 263	349, 446, 447,		
<code>\horo@tmpa</code>	... 1344,	<code>\horoboldanglestrue</code>	833	456, 828, 1364,		
1345, 1394, 1395		<code>\horocalcday</code>	589, 681, 804	1376–1378, 1380,		
<code>\horo@tmpb</code>	... 1344,	<code>\horocalcharmonic</code>		1382, 1383, 1410,		
1345, 1394, 1395		335 , 1522,	1421, 1429, 1577,		
<code>\horo@tmpfile</code>	1626, 1631, 1639,		1578, 1600–		
.. 652, 687, 690,		1643, 1647, 1654,		1602, 1721, 1723		
697, 698, 769,		1658, 1662, 1666		<code>\horoCuspsSEOpt</code>	... 820	
772, 778, 826, 829		<code>\horocalclat</code>	592, 672, 820	<code>\horoCuspVIIIPos</code>	.. 366	
<code>\horo@tmpfname</code>	<code>\horocalclon</code>	591, 668, 820	<code>\horoCuspVIIMPos</code>	. 1705	
..... 653, 678,		<code>\horocalcmonth</code>	<code>\horoCuspVIIPos</code>	331, 365	
687, 770, 810, 826		589, 680, 804	<code>\horoCuspVIPos</code> 364	
<code>\horo@twodig</code>	458 , 560,	<code>\horocalparms</code>	588 , 1729	<code>\horoCuspVPos</code> 363	
666, 670, 674, 801		<code>\horocalptime</code>	<code>\horoCuspXIIPos</code>	... 370	
<code>\horo@wasysymfalse</code>	.. 5	590, 666, 800	<code>\horoCuspXIIPos</code> 369	
<code>\horo@wasysymtrue</code>	... 9	<code>\horocalculate</code>	. 659 , 819	<code>\horoCuspXPos</code> 368	
<code>\horo@wheelsfalse</code>	.. 21	<code>\horocalculatedfalse</code>		<code>\horodegrees</code>	
<code>\horo@wheelstrue</code>	... 19	330, 688, 774	... 90, 551, 576,		
<code>\horo@YF</code> 703 , 706	<code>\horocalculatedtrue</code>		1027, 1034, 1060,		
<code>\horoadjcycles</code>	727, 823	1062, 1077, 1080,		
.... 1299 , 1367,		<code>\horocalcyyear</code>	1101, 1104, 1151		
1368, 1384, 1385		589, 682, 804	<code>\horo drawcuspstrue</code>	832	
<code>\horoadjust</code>	<code>\horoCampanusHouses</code>		<code>\horo dsms text</code> 546	
.... 1406 , 1548,		641, 759	<code>\horoEqualHouses</code>	640, 760	
1595, 1681, 1702		<code>\horoCeres@strologname</code>		<code>\horoEqualMCHouses</code>	648	
<code>\horoAlcabitusHouses</code>		606	<code>\horo foreach</code>	... 305 ,	
..... 647, 756		<code>\horoCeresSEOpt</code>	... 744	336, 346, 349,		
<code>\horoanglearrowstrue</code>	834	<code>\horoChiron@strologname</code>		421, 430, 447,		
<code>\horoanglecuspwidth</code>		605	663, 824, 827,		
... 838 , 1582, 1605		<code>\horoChironSEOpt</code>	.. 742	828, 971, 993,		
<code>\horoangularcusps</code>		<code>\horochoosetextsize</code>		1264, 1295, 1365,		
....	334 , 1583, 1585,	1554 , 1611	1366, 1381–1383,		
1587, 1592, 1606–		<code>\horoCnote</code>	1429, 1549, 1550,		
1608, 1678, 1692		846	1597, 1598, 1600,		
<code>\horoARMCSEOpt</code> 753	<code>\horoConjunctionAngle</code>		1601, 1606, 1607,		
<code>\horoAscendantSEOpt</code>	751	1202	1612, 1678, 1692,		
<code>\horoaspectobjects a</code>		<code>\horoConjunctionOrb</code>	1213	1694, 1723, 1725		
.....	1198 , 1259	<code>\horoConjunctionSymbol</code>		<code>\horo house label</code>	...	
<code>\horoaspectobjects b</code>		147, 183, 258	1268 , 1289	
.....	1199 , 1259	<code>\horoconncurve</code>	<code>\horo in house labels false</code>	
<code>\horoaspects</code>	. 1200 , 1264	953 , 1549, 1597,	1266	
<code>\horoaspectwebtrue</code>	1201	1598, 1600, 1601,		<code>\horoJuno@strologname</code>		
<code>\horoastrologopt</code>	654 , 685	1606, 1607, 1692		608	
<code>\horoautoaspect</code>	...	<code>\horocopyvar</code>	... 420 ,	<code>\horoJunoSEOpt</code> 746	
.....	1253 , 1259	440, 456, 1362,		<code>\horoJupiter@strologname</code>	
		1376, 1410, 1721		600	

\horoJupiterSEOpt . 737	\horoMoon@strologname	\horoputarrowhead .
\horoKochHouses 639, 761 596 896, 990
\horoKrusinskiHouses 762	\horoMoonSEOpt 733	\horoputarrows 992, 1592
\horolatlonstext ... 571	\horoMorinusHouses	\horoputaspect 1224, 1249
\horoblilen .. 1172, 645, 763	\horoputcurve
1173, 1188, 1557	\horoNeoPorphyryHouses 923, 954, 1488
\horoLilith@strologname 649	\horoputcusplabel .
..... 613	\horoNeptune@strologname	... 1014, 1678, 1723
\horoLilithSEOpt .. 750 603	\horoputinhouselabels
\horoLLnote 855	\horoNeptuneSEOpt . 740 1293, 1614
\horoLLnote 858	\horoNorthNode@strologname	\horoputline .. 887, 1517
\horom@rvAquarius 47, 202 610	\horoputobjdeglabel
\horom@rvAries .. 37, 192	\horoobjects 336, 346, 1055, 1493,
\horom@rvCancer . 40, 195	429, 430, 440,	1497, 1504, 1508
\horom@rvCapricorn	655, 663, 794,	\horoputobjminlabel 1068
..... 46, 201	824, 827, 1198,	\horoputobjseclabel 1089
\horom@rvGemini . 39, 194	1199, 1362-1366,	\horoputobjsymbol .
\horom@rvJupiter 54, 209	1376, 1377, 1379-	1041, 1492, 1496,
\horom@rvLeo 41, 196	1381, 1383, 1411,	1503, 1507, 1550
\horom@rvLibra .. 43, 198	1412, 1419, 1421,	\horoputradial
\horom@rvMars ... 53, 208	1429, 1431, 1547, 876, 968, 976
\horom@rvMercury 51, 206	1549, 1550, 1574,	\horoputradials ...
\horom@rvMoon ... 50, 205	1575, 1597, 1598,	970, 1547, 1574,
\horom@rvNeptune 57, 212	1612, 1694, 1725	1575, 1577, 1578,
\horom@rvPisces . 48, 203	\horoocmindist	1583, 1585, 1587,
\horom@rvPluto .. 58, 213 1305, 1364,	1602, 1608, 1707
\horom@rvSagittarius	1377, 1380, 1422,	\horoputrxlabel ...
..... 45, 200	1543, 1672, 1698 1116, 1494,
\horom@rvSaturn . 55, 210	\horooomindist	1498, 1505, 1509
\horom@rvScorpio 44, 199	1304, 1346, 1420,	\horoputsignkey ...
\horom@rvSun 49, 204	1543, 1672, 1698 1007, 1572
\horom@rvTaurus . 38, 193	\horoOppositionAngle	\horoputsmartlabel
\horom@rvUranus . 56, 211 1203 1125, 1612
\horom@rvVenus .. 52, 207	\horoOppositionOrb 1214	\horoputticks
\horom@rvVirgo .. 42, 197	\horoOppositionSymbol 981, 1569-
\horomakeequalcusps 148, 184, 259	1571, 1619-
..... 358, 379	\horoPallas@strologname	1621, 1627-
\horomakesigncusps 373 607	1629, 1632-
\horoMars@strologname	\horoPallasSEOpt .. 745	1634, 1640, 1641,
..... 599	\horoPholusSEOpt .. 743	1644, 1645, 1648,
\horoMarsSEOpt 736	\horoPlacidusHouses	1649, 1655, 1656,
\horomarvosym 191	.. 638, 658, 764, 797	1659, 1660, 1663,
\horomaxrepulsion .	\horoPluto@strologname	1664, 1667, 1668
... 1303, 1333, 1334 604	\horoQuincunxAngle 1209
\horoMCSEOpt 752	\horoPlutoSEOpt ... 741	\horoQuincunxOrb . 1220
\horoMeanNodeSEOpt 748	\horoPolichPageHouses	\horoQuincunxSymbol
\horoMercury@strologname 646, 765 154, 265
..... 597	\horoPorphyryHouses	\horoQuintileAngle 1206
\horoMercurySEOpt . 734 644, 766	\horoQuintileOrb . 1217
\horoMeridianHouses 642	\horoposattcusp ...	\horoQuintileSymbol
\horominutes 91, 1301, 1382 151, 262
552, 577, 1025,	\horoposattobj	\horoRegiomontanusHouses
1036, 1072, 1085,	... 1300, 1365, 1381 643, 767
1095, 1109, 1152		\hororepulsion 1302, 1336

<code>\hororetrogradeSymbol</code>	<code>\horosesequiquadrateOrb</code>	<code>\horotrinesymbol ..</code>
..... 146, 1223 149, 185, 260
257, 584, 1121, 1159	<code>\horosesequiquadrateSymbol</code>	<code>\horotrueNodeSEOpt</code> 749
<code>\hororightcoord</code> 157, 268	<code>\horoulnote</code> 849
..... 331 , 391,	<code>\horosetsmartradii</code>	<code>\horounitlength</code> 836 , 841
416, 417, 445, 1184 , 1563	<code>\horouranus@strologname</code>
446, 1534, 1545, 1705	<code>\horosextileAngle</code> 1208 602
<code>\hororotatechart</code> .. 411	<code>\horosextileOrb</code> .. 1219	<code>\horouranusSEOpt</code> .. 739
<code>\hororoundautotruer</code> 484	<code>\horosextileSymbol</code>	<code>\horournote</code> 852
<code>\hororoundclampfalse</code> 472 153, 187, 264	<code>\horovedicHouses</code> .. 651
<code>\hororoundstrictruncate</code>	<code>\horoshiftcusps</code> 348 , 371	<code>\horovehlowHouses</code> . 768
..... 473	<code>\horoshiftobjects</code> . 345	<code>\horovenus@strologname</code>
<code>\hororoundtodeg</code> ... 477	<code>\horosignificantadj</code> 598
<code>\hororoundtodegkeepsign</code> 1297 , 1357	<code>\horovenusSEOpt</code> ... 735
... 483, 1673, 1699	<code>\horosouthNode@strologname</code>	<code>\horovertexSEOpt</code> .. 754
<code>\hororoundtomin</code> ... 476 611	<code>\horovesta@strologname</code>
<code>\hororoundtominkeepdeg</code>	<code>\horosquareAngle</code> . 1205 609
..... 482, 1564	<code>\horosquareOrb</code> ... 1216	<code>\horovestaSEOpt</code> ... 747
<code>\hororoundtominkeepsign</code>	<code>\horosquareSymbol</code> .	<code>\horow@sySquare</code> . 33, 186
..... 481 150, 186, 261	<code>\horowasysym</code> 160
<code>\hororoundtosec</code> ... 475	<code>\horost@rAquarius</code> 71, 228	<code>\horowheelIglolik</code> 1624
<code>\hororoundtoseckkeepdeg</code>	<code>\horost@rAries</code> .. 61, 218	<code>\horowheelIqaluit</code> 1617
..... 479	<code>\horost@rCancer</code> . 64, 221	<code>\horowheelMontreal</code> 1671
<code>\hororoundtoseckkeepmin</code>	<code>\horost@rCapricorn</code>	<code>\horowheelQuebecCity</code>
..... 480, 548 70, 227 1697
<code>\hororoundtoseckkeepsign</code>	<code>\horost@rGemini</code> . 63, 220	<code>\horowheelRankin</code> . 1652
..... 478	<code>\horost@rJupiter</code> 78, 235	<code>\horowheelResolute</code> 1637
<code>\hororoundtruncate</code>	<code>\horost@rLeo</code> 65, 222	<code>\horowheelVancouver</code> 1561
..... 474, 557, 573	<code>\horost@rLibra</code> .. 67, 224	<code>\horowholeHouses</code> .. 650
<code>\hororxtext</code> 580	<code>\horost@rMars</code> ... 77, 234	
<code>\horosaturn@strologname</code>	<code>\horost@rMercury</code> 75, 232	
..... 601	<code>\horost@rMoon</code> ... 74, 231	
<code>\horosaturnSEOpt</code> .. 738	<code>\horost@rNeptune</code> 81, 238	
<code>\horosavecusps</code> 442	<code>\horost@rPisces</code> . 72, 229	
<code>\horosaveobjects</code> .. 426	<code>\horost@rPluto</code> .. 82, 239	
<code>\horoscanlabels</code> ...	<code>\horost@rSagittarius</code>	
..... 1163 , 1562 69, 226	
horoscope (environ-	<code>\horost@rSaturn</code> . 79, 236	
ment) 840	<code>\horost@rScorpio</code> 68, 225	
<code>\horoseconds</code> . 92, 552,	<code>\horost@rSquare</code> . 83, 261	
577, 1093, 1112, 1153	<code>\horost@rSun</code> 73, 230	
<code>\horosemisextileAngle</code>	<code>\horost@rTaurus</code> . 62, 219	
..... 1210	<code>\horost@rUranus</code> . 80, 237	
<code>\horosemisextileOrb</code> 1221	<code>\horost@rVenus</code> .. 76, 233	
<code>\horosemisextileSymbol</code>	<code>\horost@rVirgo</code> .. 66, 223	
..... 155, 266	<code>\horostarfont</code> 217	
<code>\horosemisquareAngle</code>	<code>\horosun@strologname</code> 595	
..... 1211	<code>\horosunSEOpt</code> 732	
<code>\horosemisquareOrb</code> 1222	<code>\horosweopt</code> ... 755 , 807	
<code>\horosemisquareSymbol</code>	<code>\horotextsize</code> 1553 , 1555	
..... 156, 267	<code>\horotextsym</code> ... 100 , 159	
<code>\horosesequiquadrateAngle</code>	<code>\horotimetext</code> 555	
..... 1212	<code>\horotrinesymbol</code> .. 1204	
	<code>\horotrinesymbol</code> 1215	

I

<code>\IC</code> 251
<code>\ifcase</code> 93, 1558
<code>\ifeof</code> 698, 772
<code>\ifhoro@djustcusps</code>
..... 1308, 1423
<code>\ifhoro@djusted</code> ...
... 1307, 1367, 1384
<code>\ifhoro@egrep</code> ... 17, 808
<code>\ifhoro@marvosym</code> ..
..... 6, 26, 35, 191
<code>\ifhoro@ppendtmp</code> 730 , 809
<code>\ifhoro@starfont</code> ..
..... 7, 60, 217
<code>\ifhoro@strolog</code> . 12, 594
<code>\ifhoro@swetest</code> . 13, 731
<code>\ifhoro@templates</code> .
..... 20, 1435
<code>\ifhoro@textsym</code> .. 4, 24
<code>\ifhoro@wasysym</code> ...
..... 5, 25, 31, 160
<code>\ifhoro@wheels</code> .. 19,
85, 835, 1197, 1309

<code>\ifhoroanglearrows</code>	...	834 , 1584, 1591
<code>\ifhoroaspectweb</code>
<code>\ifhoroboldangles</code>
<code>\ifhorocalculated</code>
<code>\ifhorodrawcusps</code>		832 , 1576, 1599, 1677, 1683, 1706, 1722
<code>\ifhorointhouselabels</code>	1266 , 1408, 1431, 1614
<code>\ifhoroundauto</code>
<code>\ifhoroundclamp</code>	472 , 488
<code>\ifnum</code>	397, 458, 488, 996, 1368, 1385
<code>\ifx</code>	299, 312, 320, 502,	513, 517, 521, 529, 532, 536, 540, 706, 1142, 1178, 1345, 1395
<code>\immediate</code>	676, 802
J		
<code>\jobname</code>	653, 770
<code>\Juno</code>	243
<code>\JunoSymbol</code>	...	126, 243
<code>\Jupiter</code>	54, 78
<code>\jupiter</code>	178
<code>\JupiterSymbol</code> 118, 178, 209, 235
L		
<code>\Large</code>	1558
<code>\large</code>	1558
<code>\Leo</code>	41, 65
<code>\leo</code>	165
<code>\LeoSymbol</code>	95, 105, 165, 196, 222
<code>\Libra</code>	43, 67
<code>\libra</code>	167
<code>\LibraSymbol</code>	96, 107, 167, 198, 224
<code>\Lilith</code>	249
<code>\LilithSymbol</code>	..	130, 249
<code>\long</code>	281, 282
M		
<code>\makebox</code>	847, 850, 853,	856, 859, 873, 1237
<code>\Mars</code>	53, 77
<code>\mars</code>	177
<code>\MarsSymbol</code> 117, 177, 208, 234
<code>\mathsf</code>	262, 263
<code>\MC</code>	253, 255
<code>\MCSymbol</code>	144, 255
<code>\MeanNodeSymbol</code>	..	98, 247
<code>\Mercury</code>	51, 75
<code>\mercury</code>	175
<code>\MercurySymbol</code> 115, 175, 206, 232
<code>\message</code>	..	1360, 1374, 1415, 1424, 1428
<code>\Moon</code>	50, 74
<code>\MoonSymbol</code> 114, 174, 205, 231
N		
<code>\NeedsTeXFormat</code>	1
<code>\Neptune</code>	57, 81
<code>\neptune</code>	181
<code>\NeptuneSymbol</code> 121, 181, 212, 238
<code>\newcounter</code>	1267
<code>\newif</code>	... 4–7, 12, 13,	17, 19, 20, 330, 472, 484, 730, 832–834, 1201, 1266, 1307, 1308
<code>\newlength</code>	836, 838
<code>\newread</code>	652, 769
<code>\nin@ty</code>	292
<code>\noexpand</code>	278
<code>\normalsize</code>	1558
<code>\NorthNode</code>	..	245, 247, 248
<code>\NorthNodeSymbol</code> 98, 99, 128, 245
O		
<code>\openin</code>	687, 826
<code>\Opposition</code>	259
<code>\opposition</code>	184
<code>\or</code>	... 94–97, 1558, 1559	
P		
<code>\p@four</code>	889, 891
<code>\p@one</code>	871, 872, 878, 879, 889, 890, 1227, 1230, 1231
<code>\p@three</code>	878–880, 889,	891, 1227, 1229, 1231
<code>\p@two</code>	871, 872, 878, 880, 889, 890, 1227, 1228, 1230
<code>\Pallas</code>	242
<code>\PallasSymbol</code>	..	125, 242
<code>\parbox</code>	847, 850, 853, 856, 859
<code>\path</code>	915
<code>\Pisces</code>	48, 72
<code>\pisces</code>	172
<code>\PiscesSymbol</code>	97, 112, 172, 203, 229
<code>\Pluto</code>	58, 82
<code>\pluto</code>	182
<code>\PlutoSymbol</code> 122, 182, 213, 239
<code>\ProcessOptions</code>	23
<code>\ProvidesPackage</code>	2
<code>\put</code>	847, 850, 853, 856, 859, 873, 1237, 1544, 1565–1568, 1674, 1675, 1700, 1701
Q		
<code>\Q</code>	273, 1196
<code>\Quincunx</code>	265
R		
<code>\raggedleft</code>	...	853, 859
<code>\raggedright</code>	...	850, 856
<code>\read</code>	697, 778
<code>\RequirePackage</code> 32, 36, 60, 86–88
<code>\Retrograde</code>	257
<code>\rightmoon</code>	174
<code>\Roman</code>	1269
S		
<code>\Sagittarius</code>	45, 69
<code>\sagittarius</code>	169
<code>\SagittariusSymbol</code>	96, 109, 169, 200, 226
<code>\Saturn</code>	55, 79
<code>\saturn</code>	179
<code>\SaturnSymbol</code> 119, 179, 210, 236
<code>\Scorpio</code>	44, 68
<code>\scorpio</code>	168
<code>\ScorpioSymbol</code>	96, 108, 168, 199, 225
<code>\scriptsize</code>	1559
<code>\Semisextile</code>	266
<code>\Semisquare</code>	267

<code>\Sesquiquadrate</code> . . .	268				
<code>\setcounter</code>	1294				
<code>\setlength</code> . 837, 839, 841					
<code>\Sextile</code>	264				
<code>\small</code>	1558				
<code>\SouthNode</code>	246				
<code>\SouthNodeSymbol</code> 129, 246					
<code>\space</code> 667, 671,					
675, 677–682,					
699, 803–809, 821					
<code>\spline</code>	948				
<code>\Square</code>	33, 83				
<code>\stepcounter</code>	1290				
<code>\string</code>	289				
<code>\Sun</code>	49, 73				
<code>\SunSymbol</code>					
.. 113, 173, 204, 230					
T					
<code>\Taurus</code>	38, 62				
<code>\taurus</code>	162				
<code>\TaurusSymbol</code>					
94, 102, 162, 193, 219					
<code>\textwidth</code>	837				
<code>\TG@sin</code>	292, 294				
<code>\TG@rem@pt</code>	342,				
343, 355, 356,					
416, 467, 715,					
		720, 873, 883,			
		884, 892, 893,			
		916–920, 928,			
		930, 936, 938,			
		948–950, 968,			
		976, 990, 1002,			
		1017, 1023, 1032,			
		1044, 1058, 1059,			
		1062, 1063, 1071,			
		1073, 1076, 1079,			
		1081, 1084, 1092,			
		1094, 1097, 1100,			
		1103, 1105, 1108,			
		1111, 1121, 1144,			
		1190, 1228, 1229,			
		1232, 1233, 1237,			
		1276, 1289, 1320,			
		1488, 1517, 1518			
		<code>\thinlines</code>			
		1609			
		<code>\tiny</code>			
		1559			
		<code>\toks</code>			
		277, 279			
		<code>\trimspaces</code> <u>275</u> , 468, 779			
		<code>\Trine</code>			
		260			
		<code>\TrueNodeSymbol</code> . 99, 248			
U					
		<code>\unitlength</code> 841, 847,			
		850, 853, 856, 859			
		<code>\uppercase</code>			
		461			
		<code>\Uranus</code>			
		56, 80			
		<code>\uranus</code>			
		180			
		<code>\UranusSymbol</code>			
		.. 120, 180, 211, 237			
V					
		<code>\Venus</code>			
		52, 76			
		<code>\venus</code>			
		176			
		<code>\VenusSymbol</code>			
		.. 116, 176, 207, 233			
		<code>\Vertex</code>			
		256			
		<code>\VertexSymbol</code> .. 145, 256			
		<code>\Vesta</code>			
		244			
		<code>\VestaSymbol</code> . . . 127, 244			
		<code>\vfuzz</code>			
		282			
		<code>\Virgo</code>			
		42, 66			
		<code>\virgo</code>			
		166			
		<code>\VirgoSymbol</code>			
		95, 106, 166, 197, 223			
W					
		<code>\write</code>			
		676, 802			
Z					
		<code>\Zodiac</code>			
		<u>93</u> , 551,			
		1003, 1017, 1058,			
		1064, 1074, 1082,			
		1098, 1106, 1154			