

Three dials, and a few more: a practical introduction to accurate gnomonics

Denis Roegel*

22 August 2007

Abstract

Sundials provide an interesting application for the METAPOST graphical programming language, at the intersection of geometry, astronomy and time. This article considers three classical sundial drawings (normal hours, horizontal coordinates, and Babylonian/Italian hours), shows how to reproduce them accurately, and extends the study to related dials, such as those showing the declination, temporary hours, or sidereal time. Analemmatic sundials, related to common sundials, are also covered.

Contents

1	Introduction	2
2	Programming prerequisites	3
3	Common vertical and horizontal dials	7
3.1	The local sphere and its gnomonic projection	7
3.2	Construction with descriptive geometry	10
3.3	Horizontal dials	19
4	Azimuth and elevation dials	20
4.1	Intersection of a cone with a vertical plane	20
4.2	Construction of the dial	26

*LORIA & Nancy Université, roegel@loria.fr.

5	Babylonian and Italian hour dials	30
5.1	Principles	30
5.2	Construction of the dial	32
6	Temporary hours sundials	48
6.1	Sunrise angles	51
6.2	Horizontal projection of temporary hour lines	51
6.3	Vertical projection of temporary hour lines	52
7	Declination sundials	54
8	Sunrise and sunset	56
9	Sidereal time sundials	57
9.1	Drawing a vertical sidereal sundial	58
9.2	Reading a sidereal sundial	62
10	Analemmatic sundials	64
11	Conclusion	67

1 Introduction

Sundials are a fascinating topic where astronomy, geometry and time meet naturally. A great number of treatises have been written on sundials since the middle ages, the most important being perhaps Clavius’ book, published in 1581 [3].

Dials were often associated with clocks, and the original dials that prompted this study were computed around 1572 by the mathematicians Conrad Dasypodius and David Wolkenstein as part of the then new Strasbourg astronomical clock (1571–1574). They were painted by Tobias Stimmer in 1572 and finally engraved in 1669 [8, p. 123–125, plates 44–46]. In 1912, Paul Werkmeister published a first thorough analysis of these dials, and he provided reproductions of the dials drawn according to the principles of descriptive geometry [14]. More recently, these dials were described by Rohr [10] and Staub [13].

In the following sections, we show how Werkmeister’s drawings can be *accurately* reproduced with METAPOST. Our drawings do not only provide a new illustration of the use of METAPOST in descriptive geometry (following a previous study on gears [9]), but they also open the way to more elaborate scientific applications, including in 3-D.

Although the initial focus of our work was the construction of three dials using descriptive geometry, we have actually departed from that goal, in two ways. First, although we aimed at reproducing Werkmeister's drawings, we did not strictly resort to the rules of descriptive geometry. There is indeed a difference between reproducing a descriptive geometry drawing, and reproducing it *with descriptive geometry*. Instead, we have actually used all techniques at our hand, including analytical geometry, to produce the results we sought.

The second departure from the initial goal concerns the scope of our study. Sundials are a vast topic, and although our focus was on three dials, we thought it was important to cover a number of dials that are related to the three basic ones. We have however decided to restrict the study to vertical and horizontal dials, and to problems that are purely geometric. In that frame, drawing a sundial is little more than finding the intersection of two surfaces, one of which is rotating. Further investigations would consider non planar dials (with the general intersection problem being exactly the same), special dials such as bifilar sundials, or dials showing uniform angles, such as those related to the mean time. The latter problems are more physical in nature, since they are (partly) related to the uneven motion of the Earth around the Sun. This explains why analemmas are not considered in this study, although, oddly enough, analemmatic dials are briefly covered.

For the three basic drawings, an approach similar to the one used in a previous study of descriptive geometry will be used [9]. We will detail the construction of the basic drawings, labelling the various points, but we will not show how the lines are drawn, except in cases where this is not obvious.

2 Programming prerequisites

The next sections deal with different types of sundials, and the drawings are often made with 3-D constructions that are projected. The main METAPOST features, as well as those used for 3-D purposes, are briefly described here, and the other interesting features are described when they occur, but the reader is still assumed to have some basic knowledge of METAPOST programming, such as can be found in John Hobby's manual [6] or in *The L^AT_EX Graphics Companion* [5].

Most of the METAPOST code is rather self-explanatory, in particular the manipulation of coordinates. An expression such as `z5` (or `z[5]` which is equivalent), for instance, represents a point in the plane, or a complex number (which we denote z_5), and it can receive a value or be involved in a computation. Moreover, `z5` is actually equal to the pair `(x5,y5)`, so that

defining one amounts to defining the other, and vice-versa. Of course, such a pair may at times be only incompletely defined.

A variable gets a value either through an assignment ($:=$), or through an equation ($=$). Equations are very convenient, because they make it possible to express geometrical relations very naturally. The unknowns may then appear on either side of the equation. Assignments are usually used when we want to give a new value to a variable that already has one.

A very convenient construction is `whatever`. Although not strictly necessary, its use is so practical that it is hard not to use it. It is usually used as follows: `whatever[A,B]` represents an undefined point between A and B, like `.5[A,B]` represents the middle between A and B, and more generally `a[A,B]` is an abbreviation for $A+a*(B-A)$. Now, if A, B, C and D are four points such that AB is not parallel to CD, then, provided the intersection does not result in an overflow, it can be obtained by the mere equation `whatever[A,B]=whatever[C,D]`. And in this equation, we stress that the two occurrences of `whatever` do represent *different* and *unnamed* variables. `whatever` is used in this study in several places.

Variables have types, and a number of basic types are available. Those of most interest to us are the types `numeric` (a numerical value, float or integer), `pair` (a pair of numerics) and `path` (a curve).

Curves in the plane are actually Bézier curves, and they are excellent approximations to the curves of interest to us, provided we use the right control points. METAPOST can build a Bézier curve going through a list of points, and although this curve is usually not unique, METAPOST chooses the control points in a way that tries to be best. It is of course possible to have METAPOST construct a Bézier curve with given control points, but this is a feature seldom used. So, if we want to build a path `p` going through `z3`, `z7` and `z1`, we merely write `p=z3..z7..z1`.

Arrays of basic types are often used, as sundials are inherently repetitive. The same graphical constructions appear with a number of variants, for instant different declinations, different hour angles, etc.

One noteworthy feature is METAPOST's ability to build a path as it unrolls a loop. This may seem strange at first, but it is very convenient. So, instead of writing `p=z1..z2..z3..z4..z5`, for instance, we can write `p=z1` for `i=2 upto 5: ..z[i] endfor;`, which is exactly the same, but more flexible. We use this construction several times, but, when doing so, we often need to do some caretaking from one value of the loop to the next, and this caretaking would disrupt the path construction. So, in order to circumvent this problem, we use a special macro `hide` which executes its parameters, but has no side-effects in a loop constructing a path. An example showing the construction of a path, and at the same time the recording of the sums

of its x -components, is:

```
p=z1 for i=2 upto 5: hide(sx:=sx+x[i];) ..z[i] endfor;
```

Paths are of course used to find intersections, and operations such as rotations or scalings are applied to points or paths.

Eventually, components of a construction are drawn with `draw`. This macro can be used with paths, or mere points, and the pen used for drawing can be changed. Lines can also be dashed and colored. However, in this study we do not go into the details of drawing, and concentrate primarily on the constructions. Once the points and paths are defined, drawing is usually straightforward.

For the 3-D approach, we manipulate vectors which are numerical triplets. To that effect, we use a synonym of the `color` type, and we define `Xp`, `Yp` and `Zp` to access the components of a vector:

```
let vector=color;
let Xp=redpart;
let Yp=greenpart;
let Zp=bluepart;
```

A few basic vector constants are defined to make some macros friendlier:

```
vector vo,vi,vj,vk;
vo=(0,0,0);
vi=(1,0,0);
vj=(0,1,0);
vk=(0,0,1);
```

The coordinate system we use is shown in figure 1: for a vertical sundial, \vec{v}_i is perpendicular to the wall, \vec{v}_j is parallel to it, and \vec{v}_k is vertical, so that $(\vec{v}_i, \vec{v}_j, \vec{v}_k)$ is a right-handed orthonormal system. The origin of the coordinates is the tip of the gnomon, except when stated otherwise.

Once vectors are available, we can define a number of useful operations such as the dot product and the vector product:

```
def dotprod(expr Va,Vb)=
  (Xp(Va)*Xp(Vb)+Yp(Va)*Yp(Vb)+Zp(Va)*Zp(Vb))
enddef;

def vecprod(expr Va,Vb)=
  (Yp(Va)*Zp(Vb)-Zp(Va)*Yp(Vb),
   Zp(Va)*Xp(Vb)-Xp(Va)*Zp(Vb),
   Xp(Va)*Yp(Vb)-Yp(Va)*Xp(Vb))
enddef;
```

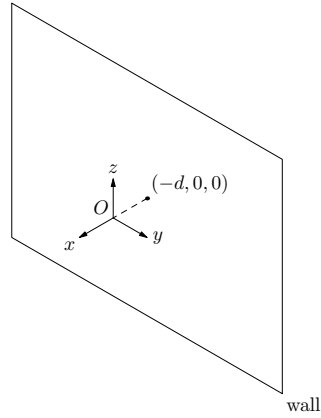


Figure 1: Reference directions. d is the distance between the tip O of the gnomon and the wall.

In order to compute $\|\vec{v}\|$, we do not use $\sqrt{\vec{v} \cdot \vec{v}}$, because that could lead to overflows. Instead, we first compute an intermediate vector where the absolute value of the greatest component is 1. So, we introduce `maxv` which returns the greatest of the absolute values of the components, and this is used in the macros `norm` (which returns the norm), and `normed` (which returns a normed vector):

```
def maxv(expr V)=
  max(abs(Xp(V)),abs(Yp(V)),abs(Zp(V)))
enddef;

def norm(expr V)=
  (maxv(V)*sqrt(dotprod(V/maxv(V),V/maxv(V))))
enddef;

def normed(expr V)=
  (V/norm(V))
enddef;
```

Finally, it is also useful to have a means to rotate a given vector \vec{V}_a by a certain angle a around another vector \vec{V}_b , and this is achieved with the following macro:

```
vardef rotatearound(expr Va,Vb,a)=
  save v;
  vector v[];
  v0=normed(Vb);
  v1=dotprod(Va,v0)*v0; % Va's part parallel to Vb
```

```

v2=Va-v1; % Va's part orthogonal to Vb
v3=vecprod(v0,v2);
v4=v2*cosd(a)+v3*sind(a)+v1;
v4
enddef;

```

3 Common vertical and horizontal dials

The most common type of dial shows the “hour angle” of the Sun (or merely the “solar time”) on a vertical plane. Such a dial is depicted at the top of figure 5. The hour angle is the angle between a plane going through the Sun and the polar axis on one side, and the meridian plane on the other side, counted positively towards the West (hence increasing with time) and measured in hours, such that 15 degrees equal one hour. Since the hour angle is proportional to time (ignoring small changes such as the slowing down of the Earth’s rotation), hours are well defined: the time between noon and 1 p.m. is the same as between 1 p.m and 2 p.m., and so on. We will see later that this is not true for all dials, or, in other words, there are different notions of hours where the length of an hour is not constant during the day or during the year, or when changing locations. In this section, however, there are exactly 24 hours of the same duration in one day. Such hours are also called “equinoctial hours,” because they can be obtained by taking a twelfth of the length of the day or the night at the equinoxes.

3.1 The local sphere and its gnomonic projection

Figure 2 shows a sundial on a vertical wall, with a style parallel to the axis of the world. The “style” is the mathematical line which is the source of the shadow, and we will sometimes also use “gnomon” for the same meaning. The hour lines are shown from 5 a.m. (true solar time) to 4 p.m.

Throughout this study, the style is assumed to have no thickness. A more elaborate computation should of course take into account the physical characteristics of the style, as well as the fact that the Sun is not a point, and that solar rays are deviated by refraction. We ignore all these factors here.

Figure 4 shows the same dial, but makes the wall apparent as well as a sphere on which the motion of the Sun is visible.

The sphere is a representation of a local equatorial sphere on which a number of lines are shown: the horizontal plane H (parallel to the horizon), the equatorial plane Eq (parallel to the equator), the meridian plane \mathcal{M} , and the two tropical planes, positionned like the Cancer (☉) and Capricorn (☿)

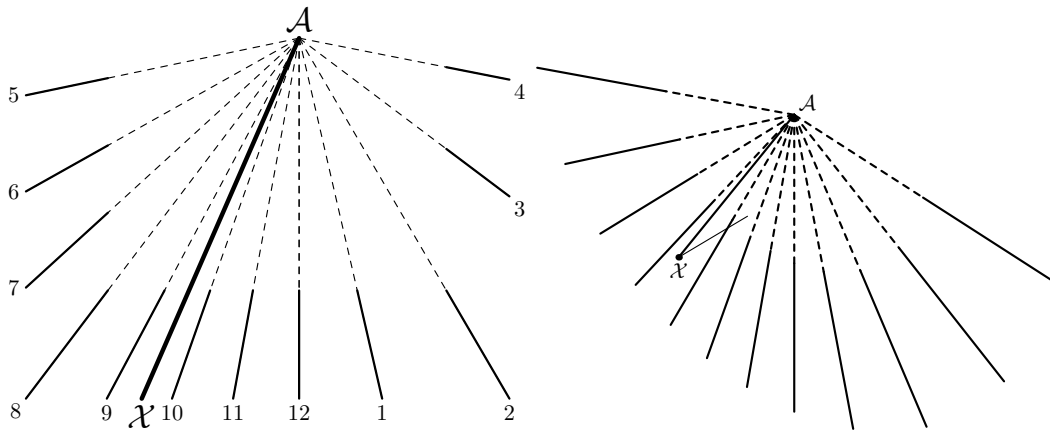


Figure 2: A vertical dial, with a gnomon \mathcal{AX} parallel to the world axis, and with hour lines from 5 a.m. till 4 p.m. The left figure shows the dial from the front, and the right figure shows it with a perspective. (The lengths of the lines are not the same in both drawings.) The other line leaving \mathcal{X} in the right-hand drawing is an horizontal fixation of the style.

circles, at angles of $\pm 23^\circ 26'$ with the equator. The angle between the horizon and the North pole is exactly the latitude φ (figure 3).

The Sun rises East and sets West, but always remains between the two tropics. We can assume that on a given day the angular distance between the Sun and the equator, namely the declination, is constant. As a consequence, the Sun seems to travel along a parallel to the equator.

If the meridian plane is rotated by multiples of 15° , it cuts the sphere at lines which are the hour lines, shown in figure 4. When the Sun goes through the meridian, it is noon solar time. The hour arc right East of the meridian is reached at 11 a.m., and the arc West of the meridian is reached at 1 p.m., and so on. These arcs are therefore like the meridians on a terrestrial globe.

This sphere also explains the principles of equatorial sundials. These dials are usually made of a wide ring along the equator, but only on half of its circumference, in the direction opposite to the Sun. The axis of the Earth is materialized in the form of an arbor, so that the shadow of that arbor reaches the wide ring. The plane containing the Sun and the arbor corresponds to the hour line and the ring is marked accordingly with parts of great circles every 15° , exactly the way it is done in figure 4.

Figure 4 also shows how the 10 a.m. line on the wall can be constructed. A line is drawn from the 10 a.m. Summer point (on the Cancer tropic) through the tip of the style, and hits the wall at a point which is then connected to the

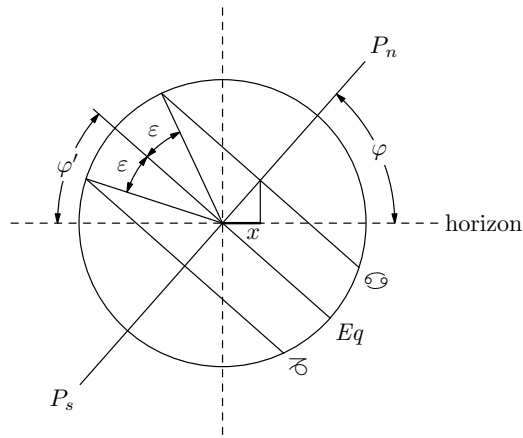


Figure 3: Position of the polar axis and the tropics. x is the distance between the projection of the center of the Cancer tropic and the center of the equator.

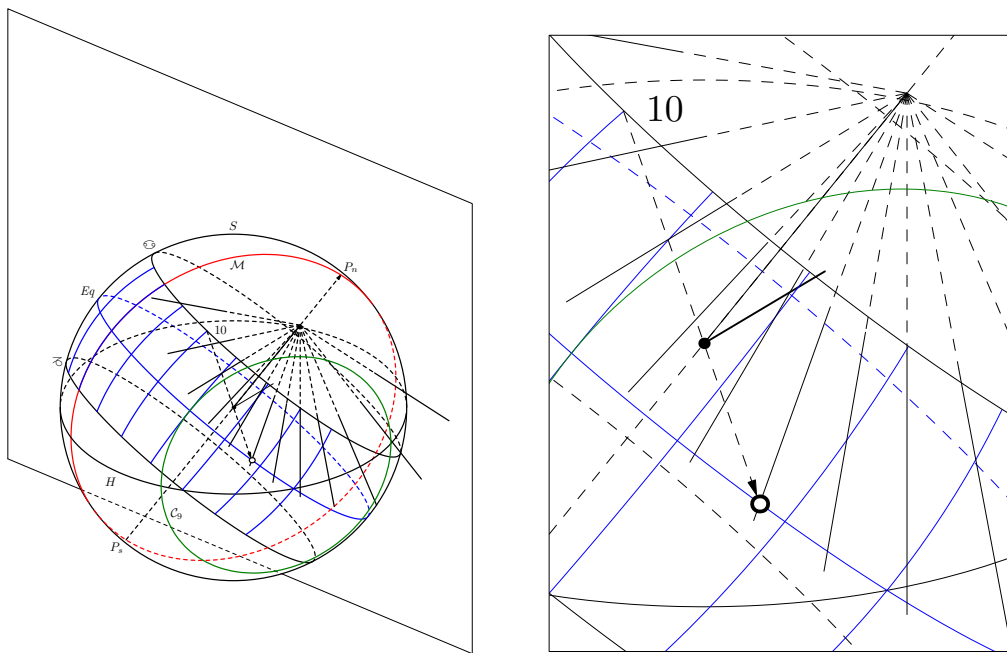


Figure 4: The same vertical sundial, with a local sphere centered on the end of the style. The smaller circle C_9 parallel to the meridian shows how the equatorial points can be obtained from evenly-spaced points on the horizon. The righthand picture is an excerpt of the left one showing the projection of the 10 a.m. Cancer point. The projection is indicated by a small circle.

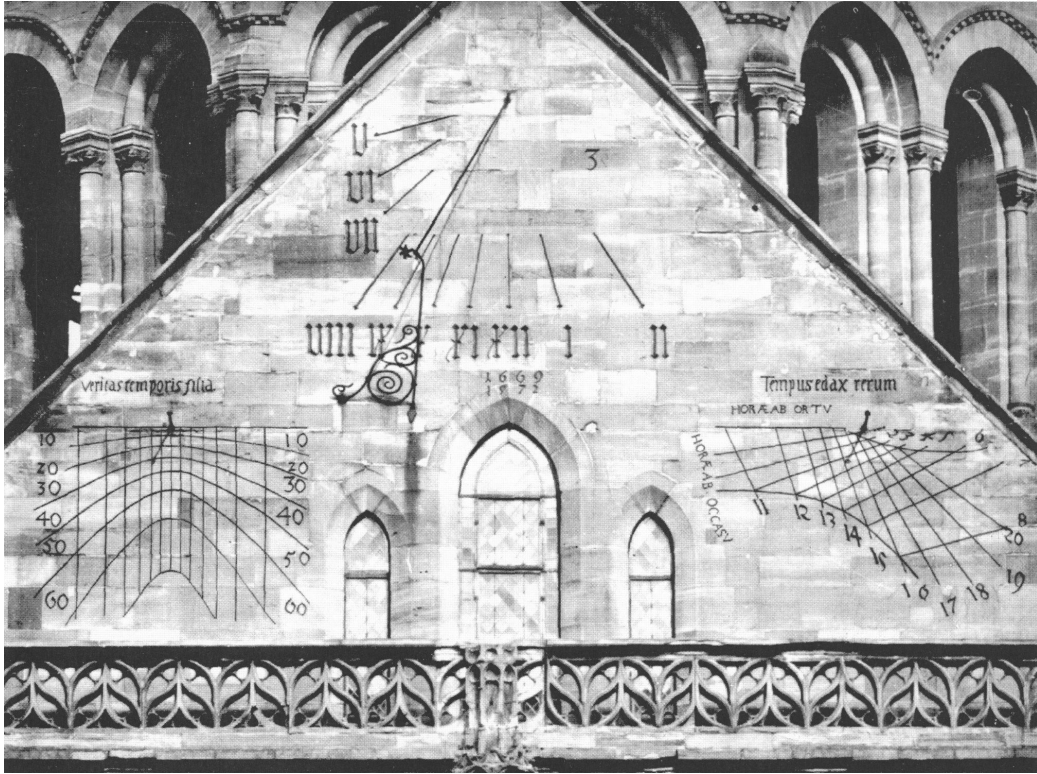


Figure 5: The three sundials at the South gable of the Strasbourg cathedral [2].

intersection of the polar axis with the wall. Since the hour lines of the sphere — the meridians — are generated by the rotation of the meridian plane, and since the style is parallel to the polar axis, it follows that all points on a same meridian project on a same line going through the intersection between the style and the wall.

3.2 Construction with descriptive geometry

As an example, we consider a graphical construction of a dial drawn for the Strasbourg cathedral. This dial is located on the South wall, but it is not exactly oriented towards the South. In sundial parlance, the dial is referred as a declining one.

The original dial (figure 5, top) has a peculiarity, in that the mark for 3 p.m. has no line, and is not engraved in the stone like the other marks, although it was initially the case, according to ancient etchings. Rohr gave a

very simple interpretation of that oddity, namely that the dial was certainly engraved in the Summer of 1669 and an error was made [10, p. 111–112]. Indeed, since the Sun cannot reach the 3 p.m. mark during that period of the year, because of the wall declination, only the marks until 2 p.m. must have been engraved. The last one, for 3 p.m., was probably added, with a different style, and no line, in 1872 after it was noticed that there could be a shadow at 3 p.m. In principle, the 4 p.m. mark should also appear, because at the Winter solstice the Sun sets slightly after 4 p.m., and South of the wall.

Such a dial can be constructed with descriptive geometry methods and this was done by Werkmeister [14] (figure 6). Here, we merely reproduce the result of the descriptive geometry approach, that is, every line of the original construction is reproduced, but sometimes we may resort to analytical geometry to find certain coordinates. This, of course, could be avoided, but we decided to present the construction as a mix of different techniques. As an exercise, the reader may rewrite our code in a purely descriptive geometry way. This could be done by defining macros for the various ways rulers and compasses are used in the pure construction.

Figure 7 shows the result of our construction. The upper part of the drawing is the vertical projection, whereas the lower part is the horizontal projection. The horizontal line dividing the two parts is the “ground line.”

The upper part therefore shows the wall, and \mathcal{A} is the foot of the gnomon. The gnomon is parallel to the polar axis and appears as the inclined segment $b'\mathcal{A}$.

On the horizontal projection, the gnomon is the segment Ba and the ground line is the projection of the wall. (Ba) is parallel to the meridian line and α is the angle between the wall and East, the so called gnomonic declination. This angle and the latitude φ are defined as follows:

```
numeric alpha,phi;
alpha=29+40/60; % angle between the wall and the East
phi=48+35/60; % latitude
```

These are the values for the Strasbourg cathedral, and they could of course be changed. Our purpose, however, is to reproduce drawings that were made for these particular circumstances, and therefore we will only consider slight changes to these values. We will also see later that the more elaborate the drawings, the more sensitive they become to the parameters. Certain curves, for instance, may intersect only in certain cases, and handling properly all the different cases is a challenge, especially if one wants to automate the positioning of labels in drawings with ever-changing angles.

We now define a number of points to which lines will be attached. Some of the points are chosen at arbitrary locations, others correspond to certain

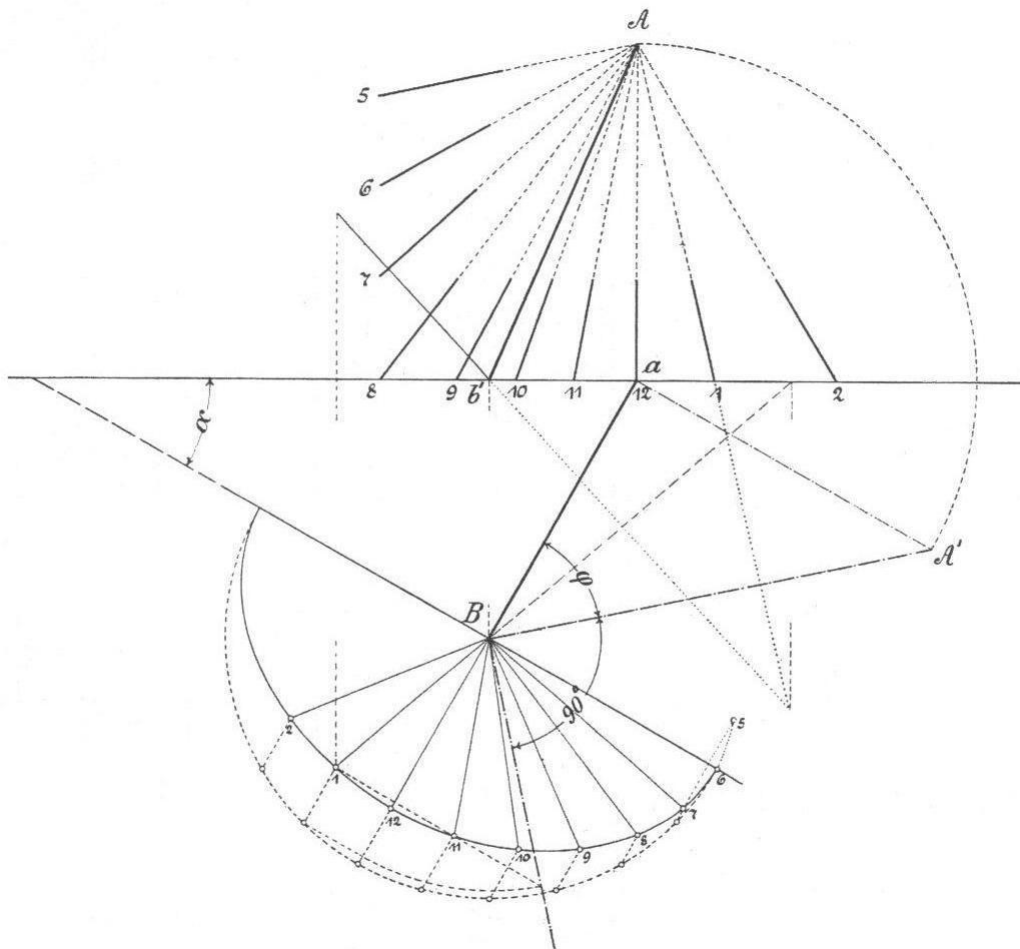


Figure 6: The original drawing by Werkmeister for the upper dial of the South gable [14]. The new drawing is shown in figure 7.

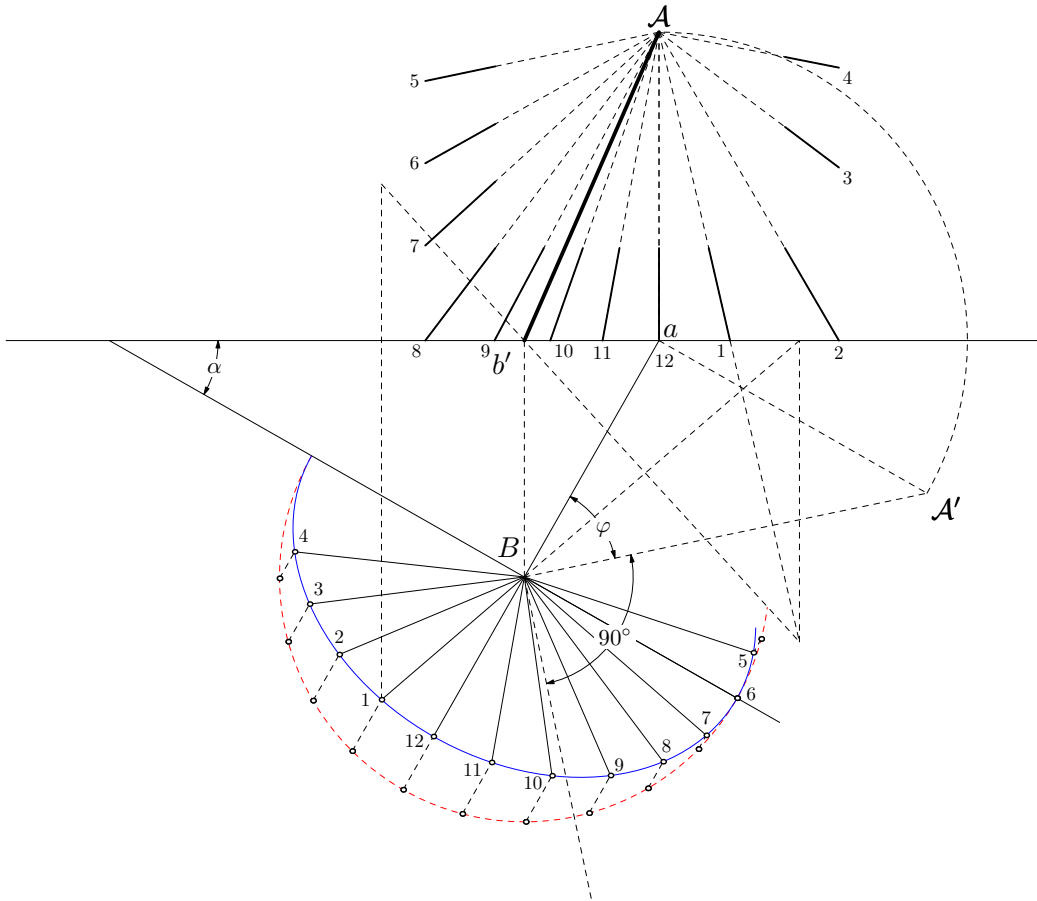


Figure 7: The construction of the solar time dial with descriptive geometry. See the original drawing in figure 6. Our construction adds the lines for 3 p.m. and 4 p.m. The latter is a possible case overseen in the original construction.

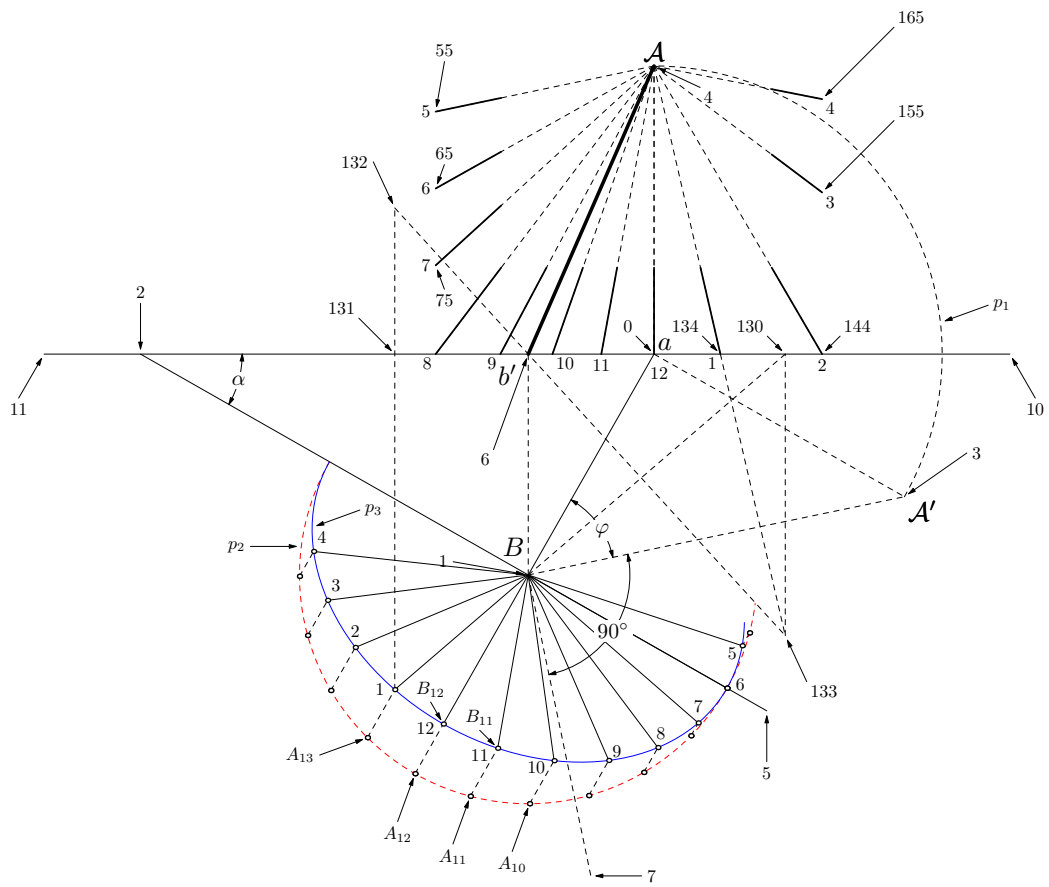


Figure 8: Details of the construction of the solar time dial.

geometrical properties. The whole drawing (upper and lower parts) is considered as one drawing in the plane, with 2-dimensional coordinates. We take the origin at the foot of the gnomon in the horizontal projection. This is the point labelled a .

```
z0=origin; % horizontal projection of gnomon foot
```

The following variables will be useful later. They show the definition of the arrays A , B and p . In METAPOST, arrays are not given a size, and the indexes can have any numeric value, including negative and non integer:

```
numeric sr,t;
pair A[],B[];
path p[];
```

Besides α and φ , there is a third important parameter, namely the length of the gnomon. Although it does not affect the general appearance of the drawing, it affects its dimensions. However, instead of the length of the gnomon, we actually use its horizontal projection l_h , and we express it as a multiple of some predefined unit u :

```
numeric lh;
lh=5u; % length of the horizontal projection of the gnomon
```

We can now set other points, and we refer to figure 8 which gives the details of all the points: z_1 is the tip of the gnomon, z_2 the intersection of the East-West line going through B with the wall, etc.:

```
z1-z0=lh*dir(-90-alpha); % z1=gnomon tip (B)
z2=z1+whatever*dir(180-alpha)=z0+whatever*left;
z10-z0=7u*right;z0-z11=12u*right; % ground line
```

Right now, the foot \mathcal{A} of the gnomon in the vertical projection has not yet been drawn. We know that \mathcal{A} is somewhere on the vertical line going through b , but we do not know where. The exact position of \mathcal{A} depends on α , φ and l_h . In fact, since the height of the North pole is equal to the latitude, we have $l_h = l \cos \varphi$, where l is the length of the gnomon. If we have the angle φ , the length l can easily be constructed as it is the length of segment $B\mathcal{A}'$:

```
z3=z1+whatever*((z0-z1) rotated -phi)
    =z0+whatever*((z0-z1) rotated -90); % z3=A'
```

At the same time, $a\mathcal{A}'$ is equal to the distance $a\mathcal{A}$ because the wall is vertical. We can now construct \mathcal{A} (point z_4):

```
z4-z0=arclength(z0--z3)*up; % z4=gnomon foot
```

The arc showing how \mathcal{A} is obtained from \mathcal{A}' can be drawn using a circle p_1 defined as follows, only the part $(0, t)$ of this circle being drawn later:

```
p1=fullcircle rotated angle(z3-z0)
      scaled 2arclength(z0--z4) shifted z0;
t=xpart(p1 intersectiontimes (z0--(z4+u*up))); % used for the arc A'-> A
```

Next, we consider the equator as it appears in figure 4, and we project it onto the horizontal plane. For that purpose, we must choose a radius s_r for the local sphere:

```
sr=4.5u; % the exact value is not relevant
```

z_5 is defined towards East:

```
z5-z1=1.2sr*unitvector(z1-z2); % z5 towards East
```

and slightly more than half of the projection of the local sphere is defined as p_2 :

```
p2=subpath(0,4.5) of fullcircle scaled 2sr rotated (180-alpha) shifted z1;
```

We define A_i ($i = 0, \dots, 23$) to be equidistant points on the circle with A_0 being towards North. We have therefore $A_i - z_1 = s_r e^{j(\frac{\pi}{2} - \alpha - i\frac{\pi}{12})}$, where j is the complex number such that $j^2 = -1$. In degrees, and using the fact that $z_1 - z_2$ is directed East (hence towards A_6), we can easily express the difference $A_i - z_1$ as follows for $i = 5$ until 16:

```
for i=5 upto 16:
  A[i]-z1=sr*unitvector(z1-z2) rotated (-(i-6)*15);
endfor;
```

The closest mathematical expression for the previous equation would be $A_i - z_1 = s_r e^{j \arg(z_1 - z_2)} e^{-j(i-6)\frac{\pi}{12}}$.

Now, the vertical projection b' of the tip of the gnomon is easily defined using B and a 's coordinates:

```
z6=(x1,y0);
```

Another dashed line is defined 90 degrees South of \mathcal{A}' :

```
z7-z1=(z3-z1) scaled .8 rotated -90;
```


In order to draw the horizontal projection of the equator, we use a macro `paralleltransform` defined as follows:

```
def paralleltransform(expr angleA,d,phi,angleB)=
  rotated angleA scaled cosd(d) yscaled cosd(90-phi) rotated angleB
enddef;
```

Such a transformation appears several times throughout this work, and it is convenient to abbreviate it. In the present case, the projected equator is an ellipse whose semi-major axis is s_r and whose semi-minor axis is $s_r \cos(90-\varphi)$, and this is exactly what `paralleltransform` does here, besides rotating the ellipse. The second parameter of the macro is not used here, and will only be used when projecting smaller circles such as tropics. And like p_2 , p_3 is actually defined as slightly more than half of the ellipse, and it is shifted to be centered in $B(z_1)$. And finally, the ellipse is actually rotated by $180^\circ - \alpha$ so that its origin lies West.

```
p3=subpath(0,4.5)
  of fullcircle scaled 2sr
    paralleltransform(0,0,phi,180-alpha)
  shifted z1;
```

The equator is also subdivided and the way this is done is that we consider the A_i divisions (the first rotation in `paralleltransform` produces A_i), and these divisions are scaled along the minor axis of the ellipse; yielding the B_i divisions. These divisions correspond to the hour angles of the Sun:

```
for i=5 upto 16:
  B[i]-z1=(sr*right) paralleltransform(-(i-6)*15,0,phi,angle(z1-z2));
endfor;
```

Like above, the previous expression actually corresponds to the following equation $B_i - z_1 = (\Re(s_r e^{-j(i-6)\frac{\pi}{12}}), \Im(s_r e^{-j(i-6)\frac{\pi}{12}}) \sin \varphi) e^{j \arg(z_1 - z_2)}$ and the conciseness of the METAPOST code is obvious.

Finally, each of these divisions is used to construct the projection on the vertical dial. This is valid because the whole circle going through the poles and through a certain hour mark B_i is projected on a straight line going through \mathcal{A} . We can therefore satisfy ourselves with the projection of the tip of the gnomon when the Sun is located in B_i .

Now, we loop i from 5 to 16, that is from 5 a.m. until 4 p.m. solar time:

```
for i=5 upto 16:
```

A number of points are defined. z_{10i} , for instance, is the horizontal projection of the shadow when the Sun is at B_i . Figure 8 shows the construction of z_{130} , for 1 p.m.

```
z[10i]=whatever[B[i],z1]=whatever[z6,z0];
```

Next, we compute how much B_i is above the horizontal plane going through B . It is immediate that this is $s_r \sin(90 - \varphi) \cos(15(i - 12))$, angles being expressed in degrees. Hence the vertical projection z_{10i+2} (for instance z_{132}) and the intermediate point z_{10i+1} (for instance z_{131}):

```
z[10i+1]=(xpart(B[i]),y0);
z[10i+2]-z[10i+1]=sind(90-phi)*cosd(15*(i-12))*sr*up;
```

Finding the points z_{10i+3} (for instance z_{133}) is immediate, as the line ($z_{10i+2}z_{10i+3}$) is the vertical projection of the solar ray. When $i \neq 16$, this point is the shadow of the tip of the gnomon at the equinoxes for the hour i (with the convention that $i = 13$ corresponds to 1 p.m., and so on), provided we consider z_{10i+3} as part of the vertical projection, and provided the Sun casts a shadow at that time. This, however, is not true for $i = 16$, that is, for 4 p.m. Nevertheless, the construction below will provide the right shadow line for $i = 16$, for a reason that will be explained shortly. First, the definition of z_{10i+3} :

```
z[10i+3]=whatever[z[10i+2],z6]=whatever[z[10i],z[10i]+down];
```

For 8 a.m. until 2 p.m. we stop the dial lines at the ground line, and the points z_{10i+4} (for instance z_{134}) are used for that purpose:

```
z[10i+4]=whatever[z4,z[10i+3]]=whatever[z6,z0];
```

The lines for 3 and 4 p.m., however, stop shorter:

```
if i>14: % 3 and 4 p.m.
    z[10i+5]=whatever[z[10*14+4],z[10*14+4]+up]=whatever[z4,z[10*i+4]];
fi;
```

Now, we close the loop:

```
endfor;
```

Finally, we define the ends for the 5, 6 and 7 a.m. lines, all ending at the vertical of z_{84} :

```

for i=5 upto 7:
  z[10i+5]=whatever[z[10*8+4],z[10*8+4]+up]=whatever[z4,z[10*i+4]];
endfor;

```

When $i = 16$, that is for 4 p.m., there is actually no shadow at the equinoxes, but what the previous computation produces is the imaginary shadow for the symmetric hour 4 a.m. This shadow is imaginary, because the Sun has not yet risen at 4 a.m. at the equinoxes, but if the Earth were transparent, there would be a shadow. This shadow is given by the line $z_4 z_{163}$, somewhat unintentionally, but still correctly. z_{164} , then, produces the shadow for 4 p.m., because the two shadow lines (for 4 a.m. near the equinoxes and for 4 p.m. near the Winter solstice) are exactly opposite lines.

We have now drawn all the shadow lines, based on the positions of the Sun at the equinoxes and of the shadow it casts on the tip of the gnomon. However, the shadow of the tip of the gnomon can be well beyond the wall, in which case only an initial part of the style serves as an index.

More generally, we can also use such a construction to check whether the Sun projects a shadow at a certain hour and time of the year. For instance, on figure 25 which shows the normal hour lines in an horizontal projection (not shown in figure 7), one can readily see that the Sun's shadow should be visible at 3 p.m. at least between September and March, but not at the Summer solstice. We can also observe that the shadow is possible at 4 p.m. (provided nothing obstructs the Sun, of course), but only near the Winter solstice. At the equinoxes, however, the Sun is already too much North to be able to cast a shadow on the wall. And the same figure 25 also shows that the 4 a.m. line is not reached, but could still be useful, since the Sun rises at the earliest slightly after 4 a.m.

The previous construction is of course somewhat tied to the values of α and φ , and different values would entail different constraints for the lines to be drawn. It is up to the reader to adapt the previous definitions, or to generalize them appropriately.

3.3 Horizontal dials

These dials are constructed in the same way as the vertical dials, but by inverting the two projection planes. In fact, it is easy to see that an horizontal dial for latitude φ corresponds to a non-declining vertical dial for a latitude $90 - \varphi$, but with symmetric hours marks. More precisely, if one wants to produce an horizontal dial, it suffices to draw a non-declining vertical dial for the colatitude, and replace the marks h by $24 - h$: 6 a.m. becomes 6 p.m., 7 a.m. becomes 5 p.m., etc., 11 a.m. becomes 1 p.m., the noon mark remains the same, 1 p.m. becomes 11 a.m., and so on.

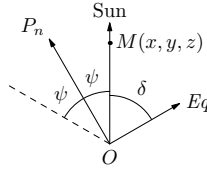


Figure 9: The cone of angle ψ .

Of course, both horizontal and vertical dials are only special cases of dials on more general surfaces. On planes the hour curves are lines, but on other surfaces they do appear differently, since the hour curves are obtained by the intersection of planes with the projection surfaces.

4 Azimuth and elevation dials

Azimuth and elevation dials are sundials locating the Sun with respect to its horizontal direction (Azimuth, 0° being South, 90° West, etc.) as well as vertical direction (elevation, or height, counted from 0 till 90°), or, in other words, with respect to its horizontal coordinates. An example of such a dial where the gnomon is horizontal is given at the bottom left corner of figure 5.

These dials contain two sets of lines, one for the azimuth values and one for the elevation values. The first lines are vertical lines which are obtained in a very straightforward way. The second lines are hyperbolas obtained as the intersections of cones with the surface of the wall. We will first consider these intersections, before poring over the code.

4.1 Intersection of a cone with a vertical plane

We consider the horizontal plane (Oxy), with (Ox) opposite the vertical plane and (Oy) parallel to it. The vertical projection plane is given by the equation $x = -d$ (see figure 1).

4.1.1 Constant declination shadow lines

Let us now consider the constant declination shadow lines. Because of the rotation of the Earth, the apparent direction of the Sun describes a cone centered on $O = (0, 0, 0)$, with its axis the polar axis (OP_n), and with a cone angle $\psi = 90 - \delta$, where δ is the declination of the Sun (figure 9).

We assume the polar axis given by a point A such that \overrightarrow{OA} is a unit vector directed towards the celestial North pole P_n . We set $\overrightarrow{OA} = (x_A, y_A, z_A)$ and

we have:

$$x_A = -\cos \varphi \cos \alpha \quad (1)$$

$$y_A = \cos \varphi \sin \alpha \quad (2)$$

$$z_A = \sin \varphi \quad (3)$$

(Section 9 shows how this vector can easily be computed using a combination of rotations.)

Now, $M(x, y, z)$ belongs to the intersection of the cone and the vertical plane if

$$\frac{\overrightarrow{OA} \cdot \overrightarrow{OM}}{\|\overrightarrow{OM}\| \cdot \|\overrightarrow{OA}\|} = \cos \psi \quad (4)$$

$$x = -d \quad (5)$$

It comes:

$$xx_A + yy_A + zz_A = \cos \psi \sqrt{x^2 + y^2 + z^2} \quad (6)$$

Squaring each side leads to:

$$y^2 y_A^2 + d^2 x_A^2 + z^2 z_A^2 = \cos^2 \psi (d^2 + y^2 + z^2) + 2yy_A x_A d + 2zz_A x_A d - 2yzy_A z_A \quad (7)$$

This is a second degree equation in z :

$$z^2(z_A^2 - \cos^2 \psi) + z(2yy_A z_A - 2z_A x_A d) + y^2 y_A + d^2 x_A^2 - \cos^2 \psi (d^2 + y^2) - 2yy_A x_A d = 0 \quad (8)$$

We set:

$$c_A = z_A^2 - \cos^2 \psi \quad (9)$$

$$c_B = 2z_A(yy_A - x_A d) \quad (10)$$

$$c_C = y^2 y_A^2 + d^2 x_A^2 - \cos^2 \psi (d^2 + y^2) - 2x_A y_A d y \quad (11)$$

and the equation is now

$$c_A z^2 + c_B z + c_C = 0 \quad (12)$$

z can now easily be expressed as a function of y , as shown later in the macro `def_declination_line` (section 4.2).

In the special case where $\delta = 0$, that is when $\psi = 90^\circ$, eq. (6) leads to $yy_A + zz_A = dx_A$ which is the equation of a line.

It can be rewritten:

$$z = \frac{dx_A - yy_A}{z_A} = -\frac{\sin \alpha}{\tan \varphi} y - d \frac{\cos \alpha}{\tan \varphi} \quad (13)$$

If $z = 0$, we have $y \sin \alpha = -d \cos \alpha$, hence $y = -\frac{d}{\tan \alpha}$. The intersection of that line with the horizon is therefore: $\left(-\frac{d}{\tan \alpha}, 0\right)$.

If $y = 0$, then $z = -d \frac{\cos \alpha}{\tan \varphi}$, and we have the point $\left(0, -d \frac{\cos \alpha}{\tan \varphi}\right)$

4.1.2 Constant height shadow lines

The formulas for constant height shadow lines are special cases of the previous formulas, because at the poles declination lines are height lines. So, taking $OA = (0, 0, 1)$, that is $x_1 = y_A = 0$ and $z_A = 1$, eq. (8) becomes:

$$z^2(1 - \cos^2 \psi) - \cos^2 \psi (d^2 + y^2) = 0 \quad (14)$$

that is

$$z^2 \sin^2 \psi = \cos^2 \psi (d^2 + y^2) \quad (15)$$

and

$$z = \pm \frac{\sqrt{d^2 + y^2}}{\tan \psi} = \pm \tan h \cdot \sqrt{d^2 + y^2} \quad (16)$$

where $h = 90 - \psi$.

When $h > 0$, we have actually simply

$$z = -\tan h \cdot \sqrt{d^2 + y^2} \quad (17)$$

the other solution being for negative values of the elevation.

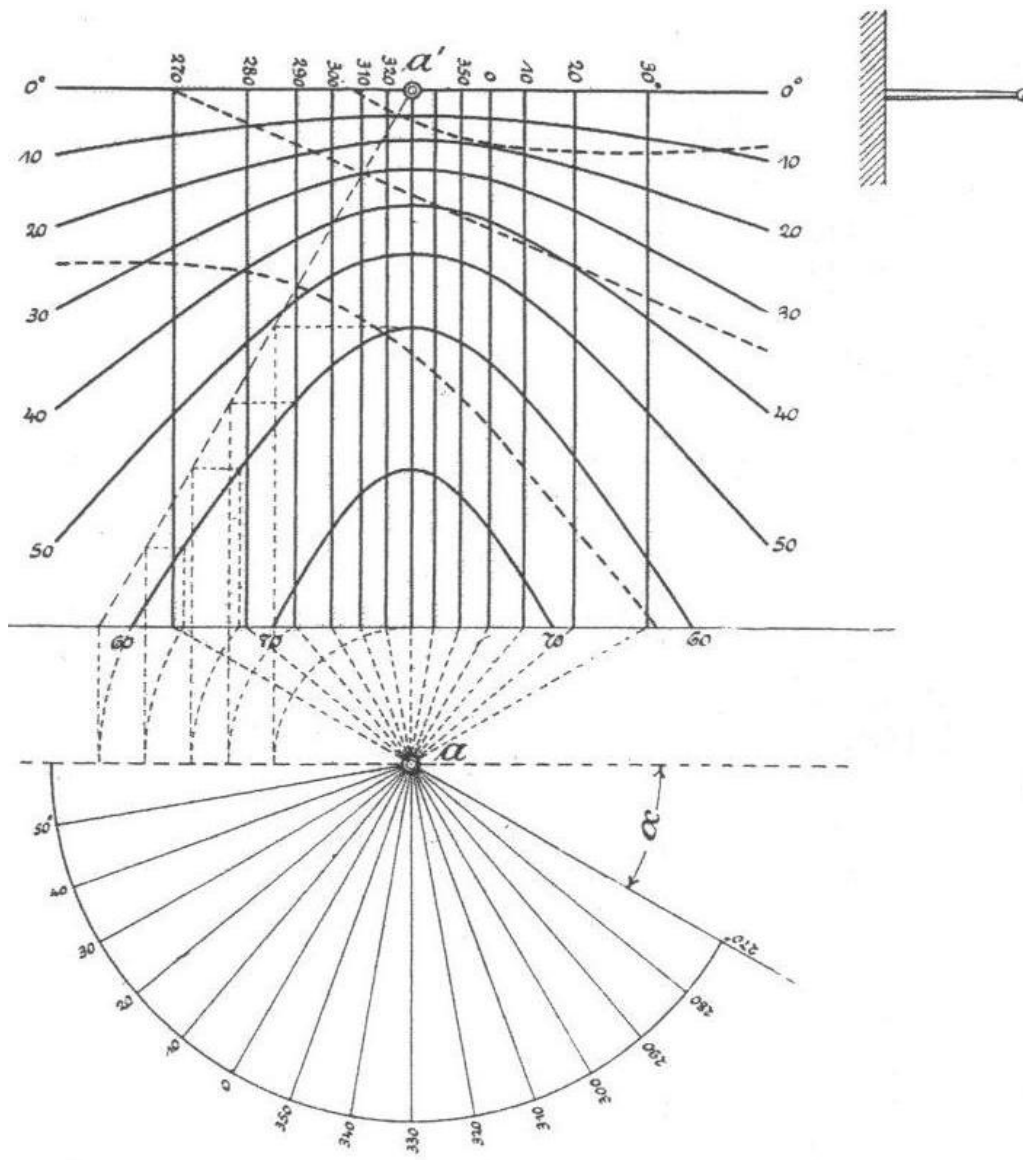


Figure 10: The original drawing by Werkmeister for the lower left dial of the South gable [14]. The new drawing is shown in figure 11.

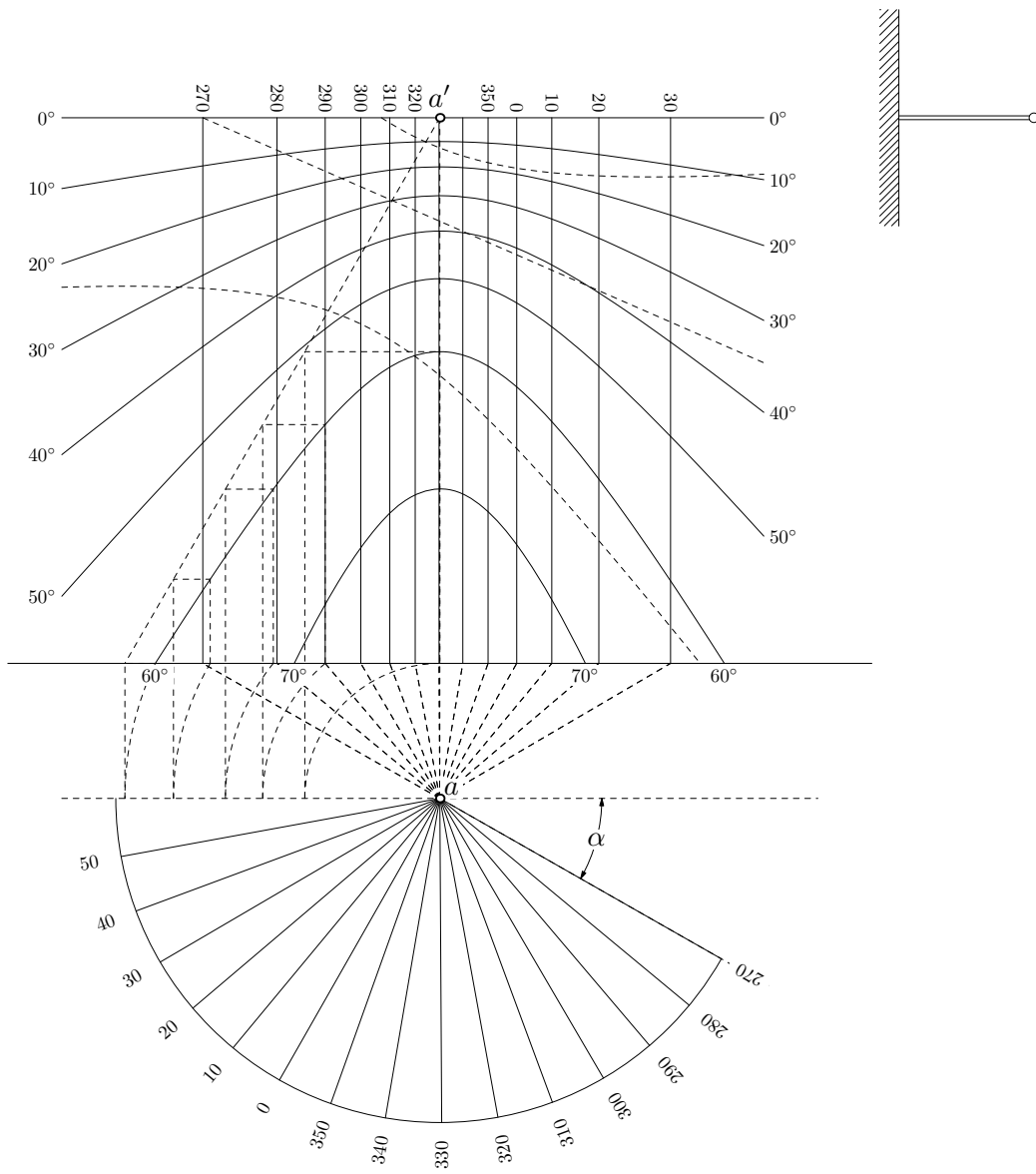


Figure 11: The construction of the elevation/azimuth dial with descriptive geometry. Note that the 330° position is not totally vertical, contrary to what is shown by Werkmeister (figure 10).

4.2 Construction of the dial

Figure 11 shows our reproduction of the original drawing by Werkmeister (figure 10). In order to make the construction easier to understand, we have included figure 12 showing the location of the various points.

Like in the first case, the construction starts with definitions of the values of the declining angle (or declination) α and the latitude φ :

```
numeric alpha,phi;  
alpha=29+40/60;  
phi=48+35/60;
```

We define several variables that will be useful later:

```
numeric lh,lv,sr,t,nsteps;  
numeric xA,yA,zA,psi,cA,cB,cC;  
pair A[],B[],C[],D[],E[];  
path p[];
```

The azimuth of the Sun is shown on a circle whose radius is s_r .

```
sr=6u; % circle radius in horizontal projection
```

This circle is centered at z_1 and the foot of the horizontal gnomon is at z_0 . The gnomon is at right angle with the wall and its length is l_h .

```
z0=origin; % horizontal projection of the foot of the gnomon  
lh=2.5u; % length of the gnomon's horizontal projection  
z0-z1=lh*up; % z1=tip of the gnomon in horizontal projection
```

In the vertical projection, the gnomon a' is set at a distance l_v above the ground line. This is of course an arbitrary distance.

```
lv=10.1u; % dial height  
z2-z0=lv*up; % z2=tip of the gnomon in vertical projection
```

Next, we define the ground line ($z_{11}z_{12}$) and a line parallel to it and going through the tip of the gnomon:

```
z0-z11=z12-z0=8u*right; % ground line  
z1-z13=z14-z1=7u*right; % parallel to the wall
```

Point z_3 marks the Eastern direction, declining by an angle α :

```
z3-z1=7u*dir(-alpha); % z3 towards East
```

The horizontal circle is p_0 , which is here rotated in order to have it start on the left side. (In METAPOST circles are paths, and have a beginning and an end.) t is a numerical value which will be used to show only the part of the circle until the East. Drawing this part is achieved with `draw subpath(0,t)` of p_0 .

```
p0=fullcircle rotated -180 scaled 2sr shifted z1;
t=xpart(p0 intersectiontimes (z1--(z1+2sr*dir(-alpha))));
```

Azimuth positions A_i for angles ranging 270° until 410° are drawn, every 10 degrees. These points enable the construction of the projections. When the Sun has a certain azimuth A , it is actually in a vertical plane, and this plane projects as a vertical line on the dial. B_i is the projection of A_i on the wall, through the tip of the gnomon, in its horizontal projection. C_i is the vertical projection of A_i , in the horizontal gnomon plane. For instance (see figure 12) A_{39} corresponds to an azimuth of 30° , it is projected as B_{39} and C_{39} .

```
for i=27 upto 41:
  A[i]-z1=sr*dir(-90-alpha-i*10);
  B[i]=whatever[z1,A[i]]=whatever[z0,z0+right];
  C[i]=(xpart(B[i]),y2);
endfor;
```

The horizon is delimited by $(z_{15}z_{16})$ on the vertical projection:

```
% horizontal line going through the gnomon foot
z2-z15=7u*right;z16-z2=6u*right;
```

Next, we construct seven hyperbolas, for the heights 10° , 20° , \dots , 70° . We already have the equation (17) for the hyperbola corresponding to height h . In our case, this equation becomes

$$z - y_2 = -\tan h \sqrt{l_h^2 + x^2} \quad (18)$$

or

$$z = y_2 - \tan h \sqrt{l_h^2 + x^2} \quad (19)$$

In order to draw such a hyperbola, we have x vary between two limits, in our case between x_{15} and x_{16} . We construct a smooth curve going through the hyperbola points of abscissa x_{15} , $x_{15} + \frac{1}{nsteps}(x_{16} - x_{15})$, $x_{15} + \frac{2}{nsteps}(x_{16} - x_{15})$, \dots , x_{16} . Each hyperbola is therefore made of $nsteps + 1$ points.

Now, $\sqrt{l_h^2 + x^2}$ is actually the distance between z_1 and the point on the ground line with abscissa x , that is, the point of coordinates (x, y_0) . In METAPOST, the distance between two points A and B can be obtained easily by computing the length of the path $A--B$ with `arclength`. So, given a point of abscissa $(j/nsteps)[x15,x16]$, the value of $\sqrt{l_h^2 + x^2}$ is `arclength(((j/nsteps)[x15,x16],y0)--z1)`. Combining all this and introducing the macro `hyppoint`, with the parameter X being $\sqrt{l_h^2 + x^2}$ and b being the elevation:

```
def hyppoint(expr x,X,b)=
  (x,y2-X*tan(b))
enddef;
```

we obtain the following code:

```
nsteps=20;
for i=1 upto 7:
  p[i]=hyppoint(x15,arclength((x15,y0)--z1),i*10)
  for j=1 upto nsteps:
    ..hyppoint((j/nsteps)[x15,x16],
               arclength(((j/nsteps)[x15,x16],y0)--z1),i*10)
  endfor;
endfor;
```

We cut the elevation curves for 60° and 70° so that they don't extend below the ground line. Interestingly, the following code works, and shows that `cutbefore` and `cutafter` do not choose the same intersection:

```
p6:=p6 cutbefore (z11--z12) cutafter (z11--z12);
p7:=p7 cutbefore (z11--z12) cutafter (z11--z12);
```

The figure 11 also shows how the elevation curves can be obtained in a geometrical way. The construction is shown for the 60° hyperbola. Consider for instance the hyperbola point at azimuth 290° . The distance between this point and the upper line $(z_{15}z_{16})$ is $\tan 60^\circ \times l$, where l is the distance between a and the ground line point with the selected azimuth. This length l is transferred onto the line $(z_{13}z_{14})$ with a circular arc. Then a vertical line is drawn and reaches a line starting at a' with a slope of 60° . This is of course also an asymptote to the 60° hyperbola. By construction, the point where this inclined line is reached is exactly at a distance $\tan 60^\circ \times l$ from the upper line, and the hyperbola point is then immediately obtained as the intersection of two lines, one horizontal from the last intersection, and one vertical for the azimuth. We do not detail all these construction drawings, but merely define the asymptote:

```

% z17 is defined on the ground line as a point through
% which the left asymptote of the 60 degrees hyperbola goes:
z17=whatever[z11,z12]=z2+whatever*dir(-90-30);

```

We now only have to draw the projection of the equator and the two tropics. First, we have already seen (eq. (13)) that the projection of the equator is a straight line. Two points are readily obtained: z_{18} on the horizon line, and z_{19} at the vertical of the gnomon foot. The line is extended to a third point, z_{20} , having abscissa x_{16} :

```

z18=(-lh/tan(alpha),y2);
z19=(0,y2-lh*cosd(alpha)/tan(phi));
z20=whatever[z18,z19];x20=x16;

```

Finally, the two tropics, the Cancer and Capricorn parallels, are obtained by tracing the two hyperbolas. (The name of the tropic is Capricorn, but the sign is Capricornus.) We use the macro `def_declination_line` that defines two arrays `dec_lineA` and `dec_lineB`, given a pole distance ψ . These two arrays correspond to the two sought hyperbolas, as these hyperbolas are the intersections of only one cone with the wall. The macro is therefore only called once. It may however be called more than once if other declinations are desired, for instance those for the entrances in Zodiac signs.

The parameters of this macro are the initial and final values of the abscissa, the unit vector VA directed towards the North pole (already seen in section 4.1.1), the distance l_h , the cone angle $\psi = 90 - 23^\circ 26'$ and `zorigin`, the point corresponding to the vertical projection of the tip of the gnomon. The rest of the code follows equations (9), (10) and (11), and then solves the 2nd degree equation. Special care is taken to avoid overflows. The hyperbolas are plotted on 31 points and stored in the arrays given as parameters.

```

vardef def_declination_line(expr ybegin,yend)
  (expr VA,lh,psi,zorigin)(text dec_lineA)(text dec_lineB)=
  save xA,yA,zA,Y,cA,cB,cC,Delta;
  numeric xA,yA,zA,Y,cA,cB,cC,Delta;
  xA=Xp(VA);
  yA=Yp(VA);
  zA=Zp(VA);
  for i=0 upto 30:
    Y:=(i/30)[ybegin,yend]; % will vary
    cA:=(zA*zA-cosd(psi)*cosd(psi))/100; % z^2 term
    cB:=2zA*(-lh*xA+yA*Y)/100;
    cC:=(Y/10)*(Y/10)*yA*yA+lh*lh*xA*xA/100
      -cosd(psi)*cosd(psi)*((Y/10)*(Y/10)+lh*lh/100)
      -2*yA*xA*lh*Y/100;
    Delta:=cB*cB-4*cA*cC;

```

```

    if Delta<0:Delta:=0;fi;
    dec_lineA[i]:= (Y,ypart(zorigin)+(-cB-sqrt(Delta))/(2cA));
    dec_lineB[i]:= (Y,ypart(zorigin)+(-cB+sqrt(Delta))/(2cA));
endfor;
enddef;

```

Calling the macro `def_declination_line` is then straightforward, and the two arrays are D (Cancer) and E (Capricorn). (Vector VA is assumed given, but section 9 shows that it can easily be computed.)

```

% VA is a unit vector for the polar axis
VA=(-cosd(phi)*cosd(alpha),cosd(phi)*sind(alpha),sind(phi));
psi=90-(23+26/60);
def_declination_line(x15,x16)(VA,lh,psi,z2)(D)(E);

```

Finally, the two projections are obtained as follows, each curve being appropriately cut:

```

% projection of Cancer tropic
p8=(D0 for i=1 upto 30:..D[i] endfor) cutafter (z11--z12);
% projection of Capricorn tropic
p9=(E0 for i=1 upto 30:..E[i] endfor) cutbefore (z15--z16);

```

5 Babylonian and Italian hour dials

5.1 Principles

Normal sundials show equinoctial hours starting at midnight. When the Sun culminates, these dials show 12 p.m.

A millenium ago, midnight, however, was not a clearly defined instant. There was no visible change at midnight, and there was no good reason to start counting at an instant that was so hard to define. Instead, there are two other instants which are much better candidates for serving as time origins, namely sunrise and sunset. These choices led to Babylonian (or Babylonian) and Italian (or Italic) hours: Babylonian hours are equinoctial hours counted from sunrise, whereas Italian hours are equinoctial hours counted from last sunset.

As a consequence, the Babylonian hour is 0 at sunrise and the Italian hour is 0 at sunset.

Sundials can be made to show these two types of hours. The dial depicted at the lower right of figure 5 is of that kind.

In order to have a good understanding of the differences between “normal” hours and Babylonian and Italian hours, it is useful to consider the

corresponding divisions or curves on the local sphere (figure 13). \mathcal{M} is the meridian plane, H the horizon, P_n and P_s are the poles and Eq is the equatorial plane. The hour angle lines are obtained by rotating the meridian plane around the polar axis by multiples of 15° . They are shown from 6 a.m. until 1 p.m. The figure shows immediately that the Sun rises earlier in Summer (Cancer tropic) than in Winter (Capricorn tropic). It also shows that when the Sun is on the equator, it rises at 6 a.m.

Babylonian hour lines $\mathcal{B}_1, \mathcal{B}_2$, etc. appear somewhat parallel to the horizon. These lines start on the Cancer tropic at points PB_0 (on the horizon), $PB_1, PB_2, \dots, PB_{23}$. Similarly they end on the Capricorn tropic at points PC_0 (on the horizon), $PC_1, PC_2, \dots, PC_{23}$. Obviously these curves are great circles and are obtained by rotating the horizon plane around the polar axis by increments of 15° .

Italian hour lines are constructed symmetrically, but starting at sunset. They are also great circles obtained by rotating the horizon plane around the polar axis. Figure 13 shows the Italian hour curve \mathcal{I}_{18} , corresponding to 18 hours after the last sunset.

These curves are then projected and their projections on a vertical surface are of course straight lines.

The construction of the lines for Babylonian and Italian hours is shown by various authors such as Rohr [11, p. 90], but we will confine ourselves to reproducing the drawing made by Werkmeister using descriptive geometry (figure 15). The horizontal and vertical projections of the local sphere are shown in figure 14. The horizontal gnomon is visible on the vertical wall on the left.

As it was the case for the previous figures, it should be clear that the two projections in Werkmeister's figures are of a different nature. The projection on the horizontal surface is a mere parallel projection along vertical lines, but the projection on the vertical surface is actually a central perspective, or a gnomonic projection. The lines joining a point in the scene to its projection always go through the tip of the gnomon.

The two projections are shown simultaneously in figure 16. The vertical projection (top) shows the two sets of lines, marked with numbers such as $\underline{3}$ for Babylonian hours, and $\overline{17}$ for Italian hours. These hours can be used to find the length of the day. Indeed, if the Babylonian hour is b and the Italian hour i , during the day the number of hours until the sunset is $24 - i$, and hence the day length is $24 - i + b$; during the night, if we could have the values of b and i , we would obtain the length of the night by computing $24 - b + i$. So, during the day, it would actually be more convenient to use the number of hours until the next sunset. Then adding this number to the Babylonian hour would give the length of the day. On figure 16, this

reasoning can be used to find that the two tropics correspond to about 8 and 16 hours of daylight. If we had added declination curves (see section 7), we would have been able to use them to find the length of the day as a function of the season.

The horizontal plane (bottom) shows the projection of the equator, as well as of the Cancer and Capricorn parallels. All Babylonian hour arcs which are above the horizon are shown in plain lines and are each identified by three points: $b_0a_0c_0$ for hour 0, $b_1a_1c_1$ for hour 1, etc., where a_i , b_i and c_i are the projections of A_i , B_i and C_i on the horizontal plane. The points a_i lie on the equator and a_6 is on the meridian. The starts and ends of the arcs are marked with small unfilled circles.

Only one Italian arc is shown, namely $b'_{18}a'_{18}c'_{18}$ for the Italian hour 18. All these arcs start and end at the filled circles, which in general do not coincide with the unfilled ones. However, the Italian arc 18 coincides with the Babylonian arc 6 exactly on the meridian and at the same time on the equator, because the Sun goes through the meridian at the equator (hence at the equinoxes) exactly six hours after sunrise, hence the coincidence with the Babylonian hour 6, but there are then also exactly six hours until sunset, or 18 hours since last sunset, hence the coincidence with Italian hour 18. More generally, Babylonian hour n coincides with Italian hour $n + 12$ at the equator.

On figure 16, the unfilled and filled circles are very close, but this is a coincidence. With a different latitude, these points are closer or further apart. The conditions for their coincidence are easy to compute (see section 5.2.1) and b_{10} coincides with b'_{18} (and all others at the same time) when the latitude is $\arctan\left(\frac{1}{2\tan\varepsilon}\right)$ which is about 49.08° . When this is true, the minimal and maximal lengths of the days are exactly 8 and 16 hours. The sundials drawn by Werkmeister are made for Strasbourg, located at $48^\circ 35'$ N, which is very close to this value, hence the coincidence. Unfortunately, this is misleading for the understanding of Werkmeister's drawing, since the casual reader is led to believe that an Italian hour line goes through b_{10} and c_{10} , which is not true in general, and not even exactly the case in Strasbourg.

5.2 Construction of the dial

Figure 16 shows our reproduction of the original drawing by Werkmeister (figure 15). In order to make the construction easier to understand, we have included figure 17 showing the location of the various points.

Like in the previous cases, the construction starts with definitions of the values of the declining angle (or declination) α and the latitude φ :

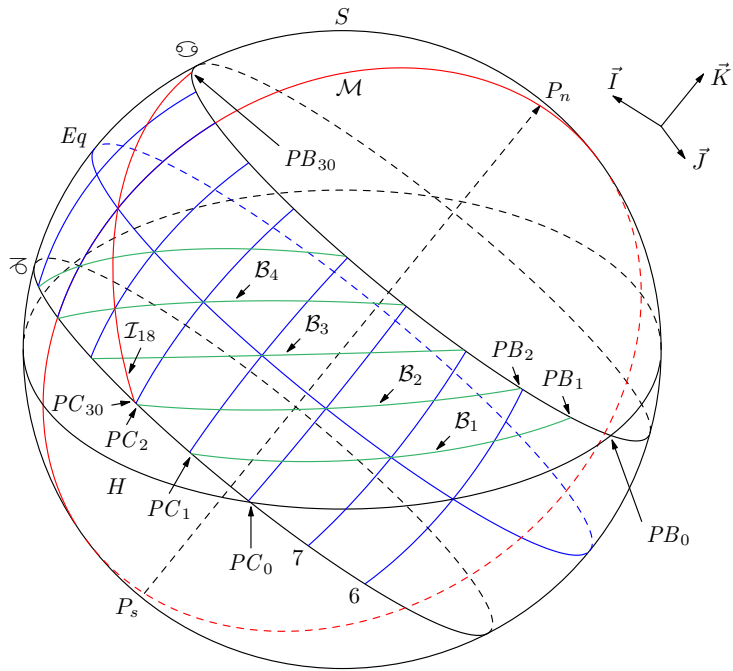


Figure 13: The local sphere with Babylonian and Italian hours for $\varphi = 48^\circ 35'$. Babylonian hours (counted from sunrise) correspond to the curves $\mathcal{B}_0, \mathcal{B}_1, \mathcal{B}_2$, etc. The corresponding tropic points in space are PB_i and PC_i . Italian hour correspond to the curves $\mathcal{I}_0, \mathcal{I}_1$, etc. The tropic points in space corresponding to \mathcal{I}_i are $PB_{24+(24-i)}$ and $PC_{24+(24-i)}$.

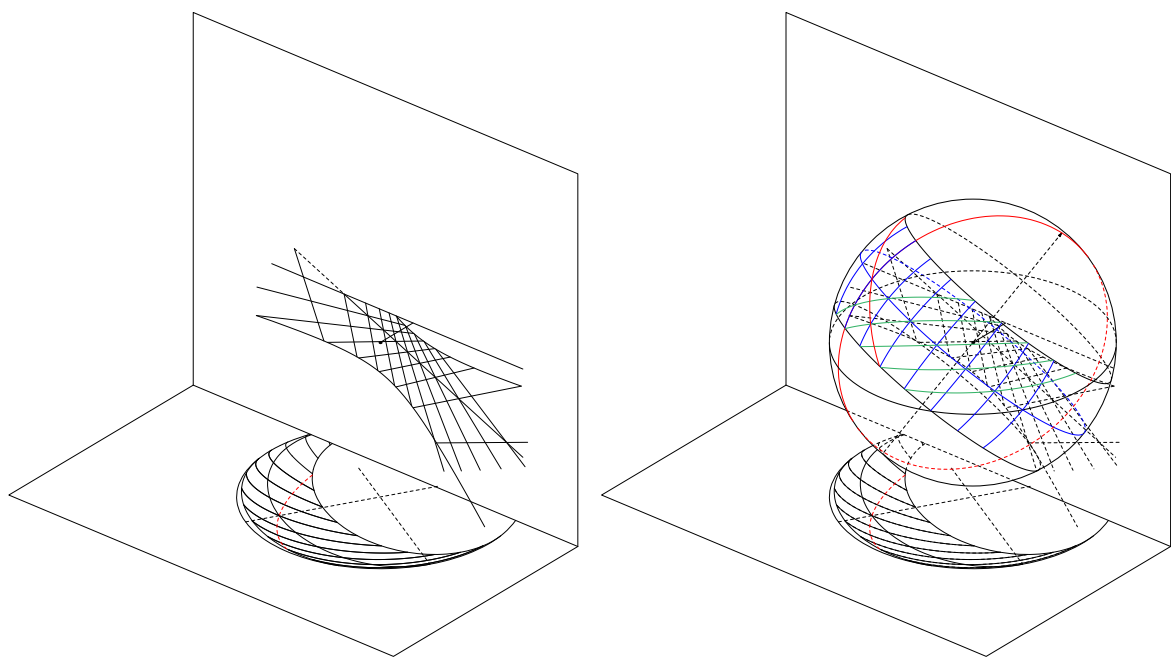


Figure 14: The projections of the local sphere on the two planes, with and without the sphere.

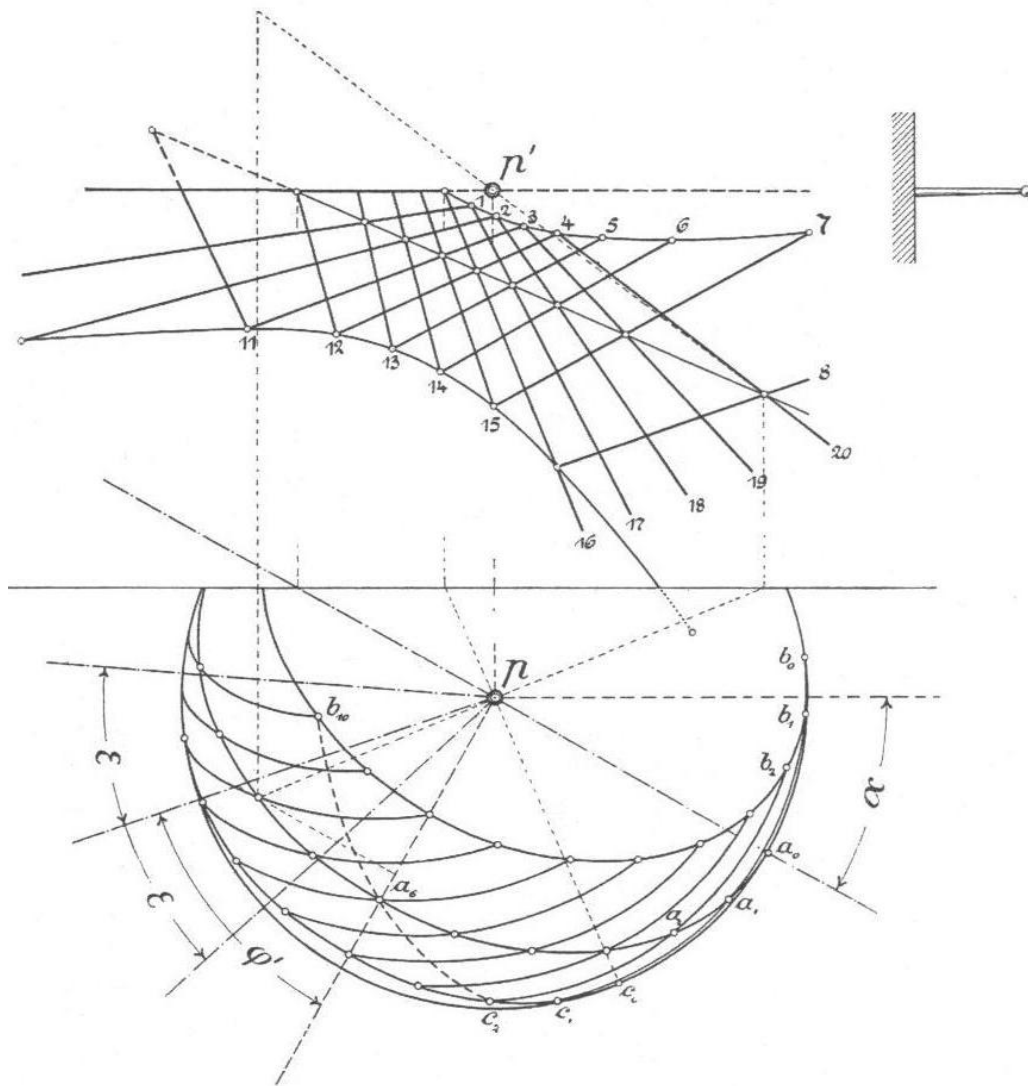


Figure 15: The original drawing by Werkmeister for the lower right dial of the South gable [14]. The new drawing is shown in figure 16.

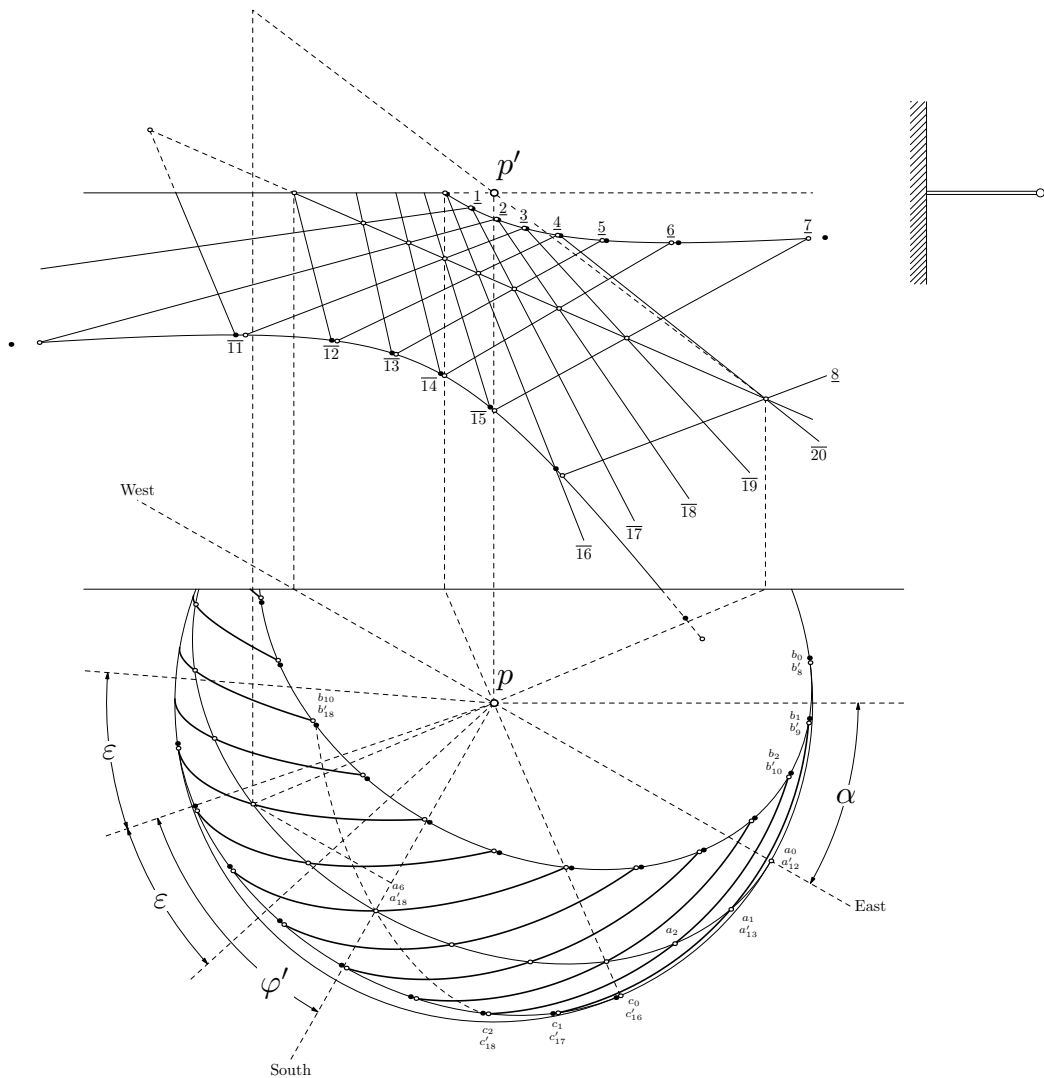


Figure 16: The simultaneous projections on the two planes for $\varphi = 48^\circ 35'$. See the original drawing in figure 15 (where filled and unfilled points are identified). The underlined numbers, such as 3, correspond to Babylonian hours, hours counted from sunrise. On the horizontal projection (bottom), they correspond to unprimed points (for instance $b_2 a_2 c_2$ for 2). The overlined numbers, such as $\overline{17}$, correspond to Italian hours, hours counted from last sunset. On the horizontal projection, they correspond to primed points (for instance $b'_{18} a'_{18} c'_{18}$ for $\overline{18}$).

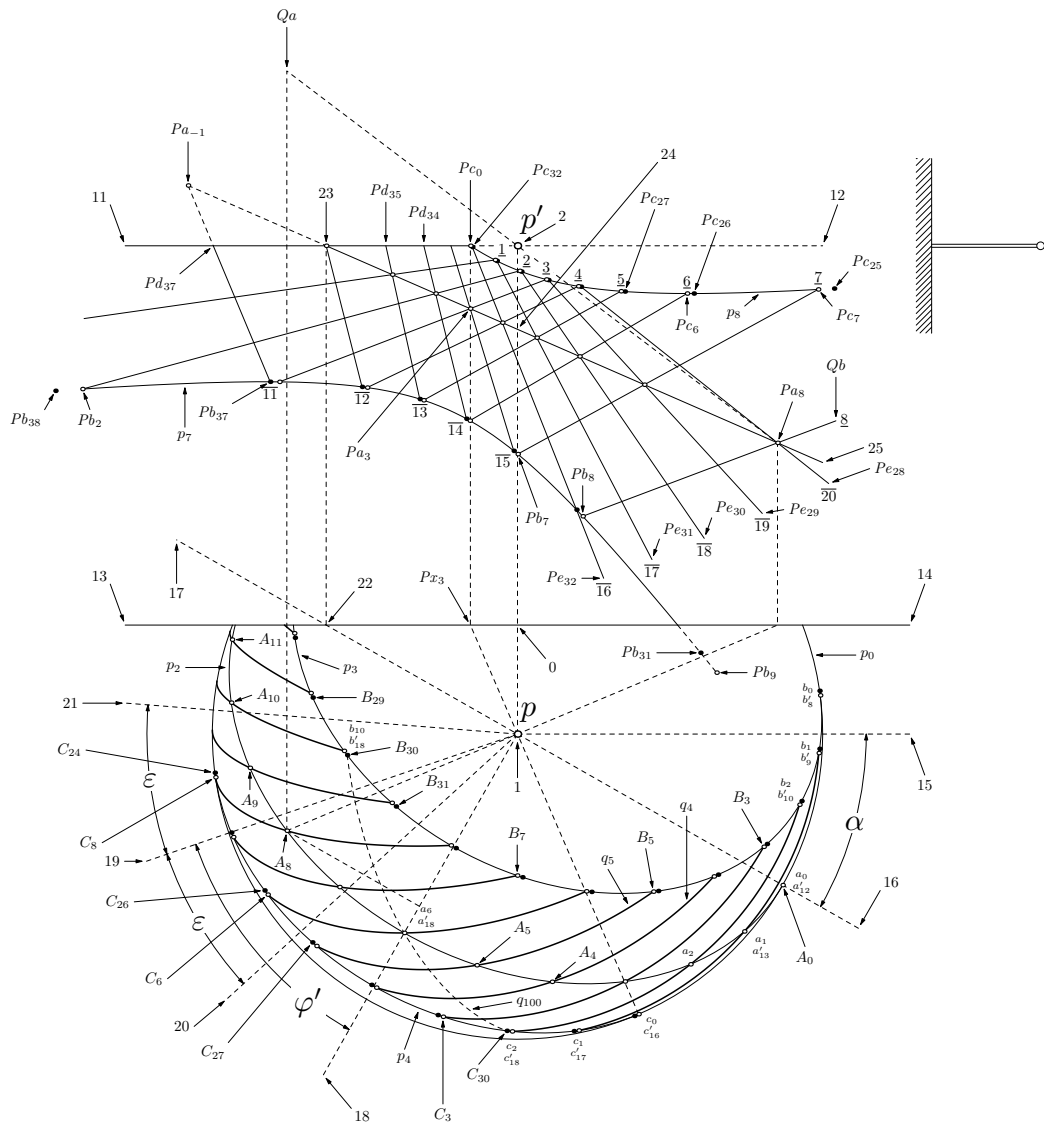


Figure 17: Details of the simultaneous projections on the two planes for $\varphi = 48^{\circ}35'$.

```

numeric alpha,phi;
alpha=29+40/60;
phi=48+35/60;

```

We then define a number of variables that will be useful later.

```

numeric phip,eps,lh,lv,sr;
path p[],q[];
pair H[],Px[],Pa[],Pb[],Pc[],Pd[],Pe[],A[],B[],C[],D[],Paa,Pab;
numeric nsteps;
numeric ca,sa,cancer_a,capricorn_a,ta,tb,tc,td;
numeric si;
pair Qa,Qb;
vector IA,JA,KA,MB,MC,I,J,K,M,PB[],PC[];

```

Next, we define the colatitude and the obliquity of the ecliptic:

```

phip=90-phi; % colatitude
eps=23+26/60; % obliquity of the ecliptic

```

Like for the first drawing, we consider the local sphere of radius s_r and this sphere is centered at z_1 in the horizontal projection. z_0 is located on the ground line, above z_1 and the length of the gnomon is l_h .

```

sr=7u; % radius of the local sphere
z0=origin; % horizontal projection of the basis of the gnomon
lh=2.5u; % gnomon length
z0-z1=lh*up; % z1=gnomon tip in h. proj.

```

The vertical projection z_2 of the gnomon is located at a distance l_v above z_0 . This is an arbitrary distance chosen to reproduce faithfully the original drawing.

```

lv=8.7u; % height of the dial
z2-z0=lv*up; % z2=gnomon tip in v. proj.

```

The horizon, or sunrise-sunset line, is $(z_{11}z_{12})$, and these points are defined as follows:

```

z2-z11=9u*right;z12-z2=7u*right;

```

Next we define points for the ground line, for a parallel to the wall, for the East-West line, and for the South:

```

z0-z13=z14-z0=9u*right; % ground line
z15-z1=9u*right; % z1--z15 is parallel to the wall
z1-z17=z16-z1=9u*dir(-alpha); % East-West line
z18-z1=9u*dir(-90-alpha); % South line

```

We now consider the projections of the local sphere and of its various circles. First, p_0 is the projection of the sphere itself and p_1 is that part of p_0 which is below the ground line. The rotation we perform on p_0 ensures that the origin of p_1 lies on the ground line, but we have to be careful that this does not interfere with the removal of the end of the circle, hence the slight adjustment to the angle, in order to avoid rounding surprises.

```
p0=fullcircle scaled 2sr shifted z1;
% p1 is the part of p0 below the ground line
ta=angle((p0 intersectionpoint (z13--z14))-z1);
p1=(p0 rotatedaround(z1,ta+.01)) cutbefore (z13--z14);
```

The projection of the equator is identical to the one made for the first drawing, except that we only draw half of the equator, and cut the part which is above the ground line:

```
p2=(subpath(0,4) of fullcircle scaled 2sr
      paralleltransform(0,0,phi,180-alpha) shifted z1)
  cutbefore (z13--z14);
```

The projections of the tropics are similar in shape and orientation to the projection of the equator, but the original circle has a different radius, and the center of the tropics do not project on z_1 . Since the tropics correspond to parallels at latitude ε , their radius is $s_r \cos \varepsilon$. The second parameter of the macro `paralleltransform` takes this scaling down into account. The center of the equator is shifted by $s_r \sin \varepsilon \cos \varphi$ in the direction $90 - \alpha$ (see figure 3) for the Cancer tropic, and in the direction $-90 - \alpha$ for the Capricorn tropic. Like in the first drawing, the two paths have their origins towards the West. Later, this will make it easier to trim the paths.

```
% The horizontal projections of the parallels are defined as follows:
% upper parallel (Cancer)
p3=fullcircle scaled 2sr paralleltransform(0,eps,phi,180-alpha)
      shifted (z1+dir(90-alpha)*sr*sind(eps)*cosd(phi));
% lower parallel (Capricorn)
p4=subpath(0,4)
  of fullcircle scaled 2sr
      paralleltransform(0,eps,phi,180-alpha)
      shifted (z1+dir(-90-alpha)*sr*sind(eps)*cosd(phi));
```

These two paths will later be adjusted and yield p_{13} and p_{14} .

We now define again equidistant points, like we did for the first drawing. These points are numbered differently, and the code is slightly different, but equivalent to the code given in the first place. A_i is a point on the projected equator and A_6 is South (whereas it was East in the first drawing) and A_0 is East.

```

for i=-1 upto 11:
  % A[i]=hour point on the projected equator
  A[i]-z1=(sr*right) rotated (-i*15) yscaled cosd(phis) rotated -alpha;
endfor;

```

The projection z_{23} of A_0 on the wall is easy to find, because it is on the horizon line:

```

z22=whatever[z16,z17]=z0+whatever*left;
z23=(x22,y2); % vertical projection of A0

```

Like we did in the previous drawing, we draw the projection of the equator by considering the point z_{24} which is at the vertical from the gnomon foot (this corresponds to z_{19} in figure 12 and the computation is identical) and a third point z_{25} at the vertical of z_{12} (this corresponds to z_{20} in figure 12).

```

z24=(0,y2-lh*cosd(alpha)/tan(phi));
% the vertical projection of the equator is prolonged:
z25=whatever[z23,z24];x25=x12;

```

Hour points on the equator are straightforward: the intersection of $(A_i z_1)$ with the ground line defines Px_i and the projection of A_i lies on the equator projection and at the vertical of Px_i . This projection is Pa_i . Figure 17 shows such a construction for $i = 3$.

```

% we draw the hour points from one hour before sunrise (-1)
% to eight hours after sunrise (8)
for i=-1 upto 8:
  % Px[i]=h. projection of A[i] on the wall
  Px[i]=whatever[A[i],z1]=whatever[z13,z14];
  % Pa[i]=v. projection of A[i] on the wall
  Pa[i]=whatever[Px[i],Px[i]+up]=whatever[z23,z25];
endfor;

```

In order to ease the computation related to the parallels, we perform the computations in space, using three vectors \vec{I} , \vec{J} and \vec{K} (figure 13). This coordinate system is expressed in a system where \vec{i} is towards the South, \vec{j} towards East and \vec{k} towards the zenith (see figure 1).

```

% K towards the celestial pole
K=(-cosd(phi)*cosd(alpha),cosd(phi)*sind(alpha),sind(phi));
% I towards South
I=(cosd(alpha)*sind(phi),-sind(alpha)*sind(phi),cosd(phi));
% and J as follows:
J=vecprod(K,I);

```


5.2.1 Intersections between the tropics and the horizon

Now, we need to find the intersections PB_0 (sunrise), PB_{24} (sunset), PC_0 (sunrise) and PC_{24} (sunset) of the tropics with the horizon. From these points, other hour points will be obtained by rotation, and all these points, when projected, will define the sundial lines for Babylonian and Italian hours.

The calculation is in fact rather straightforward. A parallel at latitude ε is given by the parametric equation

$$\overrightarrow{PM}(\theta) = \overrightarrow{PN} + s_r \cos \varepsilon \cos \theta \vec{I} + s_r \cos \varepsilon \sin \theta \vec{J} \quad (20)$$

where P is the center of the local sphere, N is the center of the parallel, and θ is the angle between \overrightarrow{NM} and \vec{I} . We have of course $\overrightarrow{PN} = s_r \sin \varepsilon \vec{K}$.

What is sought is the intersection of that parallel with the horizontal plane of P . Projecting this equation on the vertical axis, we have:

$$0 = s_r \sin \varepsilon \sin \varphi + s_r \cos \varepsilon \cos \theta \cos \varphi \quad (21)$$

hence

$$\cos \theta = -\tan \varepsilon \cdot \tan \varphi \quad (22)$$

From that value, we determine the value of $\sin \theta$, which is assumed positive. $\cos \theta$ and $\sin \theta$ define θ uniquely on the interval $[0, 360[$. We call `cancer_a` the angle corresponding to the (sunrise) intersection of the Cancer tropic with the horizon:

```
ca=-tan(eps)*tan(phi); % cos(a)
sa=sqrt(1-ca*ca);      % sin(a)
cancer_a=angle((ca,sa));
```

A similar calculation could be done for the Capricorn tropic, but the change of ε into $-\varepsilon$ actually leads to an angle $180 - \text{cancer_a}$, when measured from the South.

```
capricorn_a=180-cancer_a;
```

The two intersections that we have considered are of course related to the extreme lengths of the days. So, the longest day corresponds to a sunrise at position PB_0 and the length of the day is $\frac{\text{cancer_a}}{180} \times 24$ hours. Similarly, the shortest day corresponds to a sunrise at position PC_0 and the length of the day is $\frac{\text{capricorn_a}}{180} \times 24$ hours.

We can now return to the problem of identifying the filled and unfilled points in figure 16. It is clear that on the Cancer tropic, these points coincide

when the length of the day is an integer number of hours. This condition is always fulfilled on both tropics at the same time, or not fulfilled at all, for mere symmetry reasons. So, when the longest day has ℓ hours, the shortest has $24 - \ell$ hours, and when ℓ is an integer, so is $24 - \ell$.

In figure 16, b_{10} nearly coincides with b'_{18} . On the Cancer tropic, the former point is 10 hours after sunrise, whereas the latter corresponds to 6 hours before sunset. So, the coincidence will be exact when the length of the longest day is 16 hours. This occurs when

$$\frac{\text{cancer_a}}{180} \times 24 = 16 \quad (23)$$

hence when $\text{cancer_a} = 120^\circ$. But $\cos(\text{cancer_a}) = -\tan \varepsilon \tan \varphi$, and therefore $\tan \varepsilon \tan \varphi = 0.5$, hence $\varphi = \arctan\left(\frac{1}{2\tan \varepsilon}\right)$.

5.2.2 Projection of the Cancer tropic

We now define the points PB_i , for $i = 0$ until 12, as these are the only values of interest in our drawing. These points are obtained by starting at the sunrise, and decrementing the parameter angle by multiples of 15 degrees. In addition, we call the macro `project_hv` which merely projects a spatial point into two points in the plane. This macro is defined as follows. The projection on the horizontal plane is simple, since it is a parallel projection along a vertical axis. The projection on the vertical plane is slightly more complex, because we perform a central projection through the origin.

```
% project 3-D point Po on Ph (horizontally) and Pv (vertically)
vardef project_hv(expr Po)(text Ph)(text Pv)=
  save N;vector N;
  % the horizontal projection is:
  Ph:=z1+(Yp(Po),-Xp(Po));
  % we now project Po on the wall (coordinate x=-lh)
  % the result is N
  N=whatever[Po,(0,0,0)]=(-lh,whatever,whatever);
  Pv:=(whatever,whatever);
  % N is now transformed into a 2-D coordinate,
  % obtaining the Y-part of Pv:
  ypart(Pv)=y2+Zp(N);
  % and the following gives a value to the X-part of Pv:
  (xpart(Pv),y0)=whatever[Ph,z1];
enddef;
```

The code defining points PB_0 , B_0 , Pb_0 , PB_1 , B_1 , Pb_1 , etc., is:

```
% Babylonian hours:
for i=0 upto 12:
```

```

PB[i]=sr*sind(eps)*K+sr*cosd(eps)*cosd(-i*15+cancer_a)*I
      +sr*cosd(eps)*sind(-i*15+cancer_a)*J;
project_hv(PB[i])(B[i])(Pb[i]);
endfor;

```

Italian hour points are defined similarly, and PB_{24} is the point at sunset, PB_{25} the point one hour before sunset, PB_{26} the point two hours before sunset, etc. These points are projected on the horizontal and vertical planes like before.

```

for i=24 upto 24+16:
% i-24=number of hours until sunset
% 24-(i-24)=48-i=number of hours since sunset
PB[i]=sr*sind(eps)*K+sr*cosd(eps)*cosd((i-24)*15-cancer_a)*I
      +sr*cosd(eps)*sind((i-24)*15-cancer_a)*J;
project_hv(PB[i])(B[i])(Pb[i]);
endfor;

```

The projection of the Cancer tropic can be obtained with an excellent approximation by joining a number of its points with a smooth curve. A first solution merely connects points Pb_2 to Pb_9 , but a close examination will show that, in our setting, the curve slightly drifts from its ideal shape in certain places.

```

p5=Pb2 for i=3 upto 9:..Pb[i] endfor;

```

A better approximation is obtained by adding two points between each pair (Pb_i, Pb_{i+1}) , hence plotting the curve every 20 minutes, solar time. The following loop has si range over the values $2, 2 + \frac{1}{3}, 2 + \frac{2}{3}, 3, 3 + \frac{1}{3}, \dots, 8 + \frac{2}{3}, 9$, and it constructs a point M on the tropic on the fly, and projects it on Pab (which is ignored) and Paa (which is used to construct this new path p_7).

```

p7=Pb2
for i=0 upto 21: % 21=3*(9-2)
  hide(si:=(i/21)[2,9];
    M:=sr*sind(eps)*K+sr*cosd(eps)*cosd(-si*15+cancer_a)*I
      +sr*cosd(eps)*sind(-si*15+cancer_a)*J;
    project_hv(M)(Pab)(Paa);
  )
  ..Paa
endfor;

```

5.2.3 Projection of the Capricorn tropic

The same construction is done for the Capricorn parallel, the only change being the use of arrays PC , C and Pc , as well as a different equation for the parallel, stemming from the replacement of ε by $-\varepsilon$.

Hence, the following loop defines points PC_i and its projections C_i and Pc_i for the Babylonian hours. PC_0 corresponds to sunrise at the Winter solstice:

```
for i=0 upto 12:
  PC[i]=-sr*sind(eps)*K+sr*cosd(eps)*cosd(-i*15+capricorn_a)*I
          +sr*cosd(eps)*sind(-i*15+capricorn_a)*J;
  project_hv(PC[i])(C[i])(Pc[i]);
endfor;
```

Italian hours on the Capricorn parallel are defined as follows, PC_{24} corresponding to sunset at the Winter solstice:

```
for i=24 upto 24+13:
  % i-24=number of hours until sunset
  % 24-(i-24)=48-i=number of hours since sunset
  PC[i]=-sr*sind(eps)*K+sr*cosd(eps)*cosd((i-24)*15-capricorn_a)*I
          +sr*cosd(eps)*sind((i-24)*15-capricorn_a)*J;
  project_hv(PC[i])(C[i])(Pc[i]);
endfor;
```

Like above, a path p_6 is constructed joining the points Pc_0 to Pc_7 on the vertical projection:

```
p6=Pc0 for i=1 upto 7:..Pc[i] endfor;
```

But, like for the Cancer tropic, this path is not perfect, and a much better approximation is p_8 , obtained with additional points every 20 minutes, solar time:

```
p8=Pc0
for i=0 upto 21:
  hide(si:=(i/21)[0,7];
    M:=-sr*sind(eps)*K+sr*cosd(eps)*cosd(-si*15+capricorn_a)*I
        +sr*cosd(eps)*sind(-si*15+capricorn_a)*J;
    project_hv(M)(Pab)(Paa);
  )
  ..Paa
endfor;
```

5.2.4 Trimming the tropics

p_3 is the projection of the Cancer parallel, and we remove what is beyond B_0 , as well as what is above the ground line:

```
% upper parallel: B0=sunrise
tb=xpart(p3 intersectiontimes (z1--(B0+(B0-z1))));
p13=(subpath(0,tb) of p3) cutbefore (z13--z14); % upper parallel
```

p_4 is the projection of the Capricorn parallel, and we keep only the part between C_{24} and C_0 :

```
% lower parallel: C0=sunrise,C24=sunset
tc=xpart(p4 intersectiontimes (z1--(C0+(C0-z1))));
td=xpart(p4 intersectiontimes (z1--(C24+(C24-z1))));
p14=subpath(td,tc) of p4; % lower parallel
```

5.2.5 Babylonian arcs

We now define twelve arcs for the Babylonian hours 1 to 12. q_1, q_2, \dots, q_{12} are the horizontal projections of these arcs (see figure 17). q_1 is the projection of \mathcal{B}_1 , q_2 is the projection of \mathcal{B}_2 , etc.

In order to draw such an arc, we consider a parametric equation of the arc. For each value of $i = 1, \dots, 12$, we consider two points: PB_i on the Cancer tropic and PC_i on the Capricorn tropic. These two points correspond to Babylonian hour i . We build three vectors \vec{I}_A, \vec{J}_A and \vec{K}_A , such that $\vec{I}_A = \frac{PB_i}{\|PB_i\|}$, $\vec{K}_A = \frac{PB_i \wedge PC_i}{\|PB_i \wedge PC_i\|}$ and $\vec{J}_A = \vec{K}_A \wedge \vec{I}_A$.

Then, a parametric equation of the arc is

$$\overrightarrow{PM}(\theta) = s_r \cos \theta \vec{I}_A + s_r \sin \theta \vec{J}_A \quad (24)$$

where P is the center of the local sphere and M is a point of the arc with parameter θ .

Using this equation, it is easy to find where the arcs intersect with the horizontal plane, something which is needed to draw the arcs properly, plain when seen, and dashed when not seen.

So, projecting eq. (24) on a vertical axis, equating it to 0, we have:

$$\cos \theta I_z + \sin \theta J_z = 0 \quad (25)$$

where I_z and J_z are the z -components of \vec{I}_A and \vec{J}_A .

Hence

$$\tan \theta = -\frac{I_z}{J_z} \quad (26)$$

This yields two values for θ , corresponding to the two intersections with the horizontal plane. However, we will draw the arcs starting at the Cancer tropic and going to the Capricorn tropic, and so we actually need the value of θ corresponding to that direction. We use here an ad-hoc selection of the right value which is sufficient for our needs, but a better solution should be implemented for the general case. The path in the horizontal plane is then constructed by taking $nsteps$ points between the Cancer tropic position and the intersection with the horizontal plane: if θ_h is the angle corresponding to the horizontal plane, we consider the $nsteps + 1$ points

$$\begin{aligned}
& s_r \vec{I}_A, \\
& s_r \cos\left(\frac{1}{nsteps}\theta_h\right) \vec{I}_A + s_r \sin\left(\frac{1}{nsteps}\theta_h\right) \vec{J}_A, \\
& \dots \\
& s_r \cos\left(\frac{i}{nsteps}\theta_h\right) \vec{I}_A + s_r \sin\left(\frac{i}{nsteps}\theta_h\right) \vec{J}_A, \\
& \dots \\
& s_r \cos\left(\frac{nsteps-1}{nsteps}\theta_h\right) \vec{I}_A + s_r \sin\left(\frac{nsteps-1}{nsteps}\theta_h\right) \vec{J}_A, \\
& s_r \cos(\theta_h) \vec{I}_A + s_r \sin(\theta_h) \vec{J}_A
\end{aligned}$$

The horizontal projection of one of these points M is then merely $z_1 + (M_y, -M_x)$, given our reference directions (figure 1). This is all summarized in the code below, which ends by trimming the arcs either at the Capricorn tropic (for Babylonian hours 1 to 7), or at the ground line (for Babylonian hour 12). The other arcs, from 8 to 11 end at the horizon.

```

nsteps=50;
for i=1 upto 12:
  IA:=normed(PB[i]);
  KA:=vecprod(normed(PB[i]),normed(PC[i]));
  KA:=KA/norm(KA);
  JA:=vecprod(KA,IA);
  ta:=-Zp(IA)/Zp(JA);
  if i>6:
    ta:=angle((1,ta));
  else:
    ta:=180+angle((1,ta));
  fi
  M:=sr*IA;
  q[i]=(z1+(Yp(M),-Xp(M)))
  for j=1 upto nsteps:
    hide(M:=sr*cosd(ta*(j/nsteps))*IA+sr*sind(ta*(j/nsteps))*JA;)
    .. (z1+(Yp(M),-Xp(M)))
  endfor;
  if i<8:q[i]:=q[i] cutafter p4;fi;

```

```

    if i=12:q[i]:=q[i] cutafter (z13--z14);fi;
endfor;

```

5.2.6 Italian arcs

Drawing all Italian arcs would be similar, and we only draw one as an example. This is the arc going through PB_{30} and PC_{30} , hence it is the arc for the time six hours before sunset, or 18 hours after the previous sunset. It is therefore the arc \mathcal{I}_{18} . This arc goes through A_6 , since A_6 is six hours before sunset. The arc which is constructed, q_{100} and which extends to the horizon, will eventually be cut after p_4 , the Capricorn tropic.

```

IA:=PB30/sr;IA:=IA/norm(IA);
JA:=PC30/sr;
KA:=vecprod(IA,JA);KA:=KA/norm(KA);
JA:=vecprod(KA,IA);
M:=sr*IA;
nsteps:=10;
ta:=-Zp(IA)/Zp(JA);
ta:=180+angle((1,ta));
q100=(z1+(Yp(M),-Xp(M)))
  for j=1 upto nsteps:
    hide(M:=sr*cosd(ta*(j/nsteps))*IA+sr*sind(ta*(j/nsteps))*JA;)
    .. (z1+(Yp(M),-Xp(M)))
  endfor;

```

5.2.7 Drawing the vertical projection

Werkmeister's drawing gave as an example the construction of the vertical projection of A_8 . In our case, we merely define Qa as being at the vertical of A_8 and on the line connecting the gnomon projection z_2 and A_8 's projection Pa_8 :

```

Qa=whatever[A8,A8+up]=whatever[Pa8,z2];

```

The remaining definitions are only for various improvements to the drawing, such as the positioning of Babylonian hour 8 (position Qb), and pleasing extensions to the Italian lines 16 to 20:

```

Qb=1.3[Pb8,Pa8]; % position of numeral 8

% distant points on the italic hour lines
Pe32=10[Pc32,Pa4]; % 16 hours since sunset
Pe31=10[Pc31,Pa5]; % 17 hours since sunset
Pe30=10[Pc30,Pa6]; % 18 hours since sunset
Pe29=10[Pc29,Pa7]; % 19 hours since sunset

```

```

Pe28=10[Pc28,Pa8]; % 20 hours since sunset

% uniformization of distances
for i=28 upto 32:
  % Pe[i] is beyond the intersection of italic hour line (48-i)
  % with the Babylonian hour line for 8 hours since sunrise:
  Pe[i]:=((Pc[i]--Pe[i]) intersectionpoint
          ((Pb8+(Pb8-Qb))--Qb))+1.5u*unitvector(Pe[i]-Pc[i]);
endfor;

```

Finally, we define points Pd_{33} to Pd_{37} , which are the intersections of Italian hour lines 15, 14, 13, 12, and 11 with the sunrise-sunset line:

```

for i=33 upto 37:
  Pd[i]=whatever[Pa[36-i],Pb[i]]=whatever[z11,z12];
endfor;

```

6 Temporary hours sundials

Normal, Babylonian and Italian hours are all equinoctial hours, that is, hours equal to a twenty-fourth part of a day. The only difference between these time reckonings is the origin: midnight, sunrise or sunset.

Temporary hours (also called temporal, planetary, or simply unequal hours), instead, do not always have the same length. The day and the night are each divided in twelve equal hours, and the length of an hour varies with the season.

In addition to the normal hours, figure 18 shows the temporary hour lines tp_0 to tp_6 , tp_0 being the sunrise line and tp_6 being the meridian line. Each tropic, and each intermediate parallel, is divided in 12 parts between sunrise and sunset, and in 12 parts between sunset and sunrise. Obviously, tp_1 and H_i (the normal hour curve i) intersect on the equator (at the equinoxes), as this is when temporary hours are equal to normal hours.

Figure 19 shows the horizontal and vertical projections of the temporary hour lines. On the vertical projection (top), the temporary hour lines are not straight lines, albeit get very close (see [4] and [12, p. 295–304]). When drawing these curves, one can for instance join the ends by a straight line and observe that the correct curve does not coincide with the segment. A consequence of this observation is the fact that the curves tp_i are not parts of great circles.

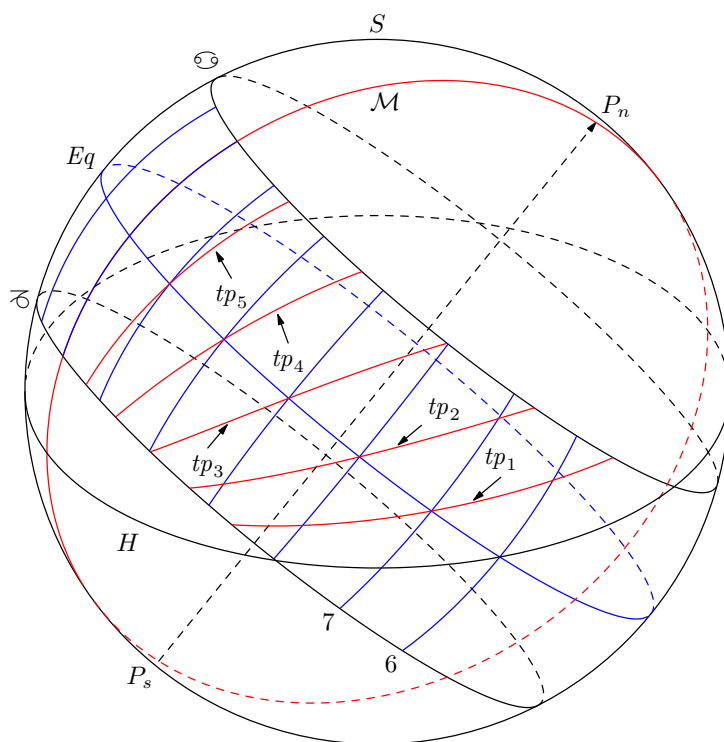


Figure 18: The local sphere with normal and temporary hours.

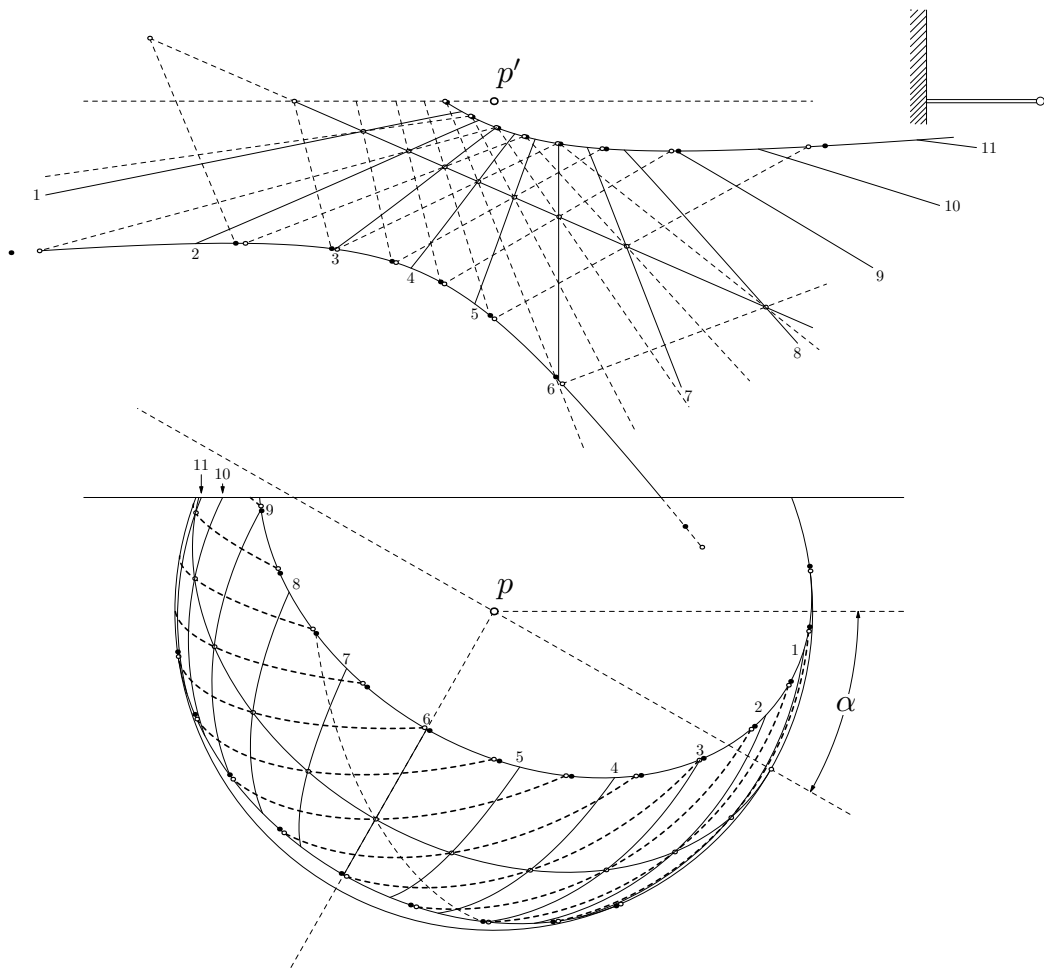


Figure 19: The simultaneous projections of the temporary hours on the two planes. p is the tip of the gnomon in the horizontal projection and p' in the vertical projection. The Babylonian and Italian hour lines are drawn with dashed lines.

6.1 Sunrise angles

In order to define and project the temporary hour lines, it is useful to have a macro returning the hour angle of sunrise and sunset for the various declinations of the Sun. We have already seen how to compute these values in the two extreme cases, for the tropics. The formulæ remain the same:

$$\cos \theta = -\tan \delta \tan \varphi \quad (27)$$

$$\sin \theta = \sqrt{1 - \cos^2 \theta} \quad (28)$$

This is summarized in macro `sunrise` which takes the declination δ and the latitude φ as parameters, and returns the sought angle:

```
vardef sunrise(expr dec,phi)=
  save sunrise_ca,sunrise_sa,sunrise_a;
  sunrise_ca:=-tan(dec)*tan(phi); % cos(a)
  sunrise_sa:=sqrt(1-sunrise_ca*sunrise_ca); % sin(a)
  sunrise_a:=angle((sunrise_ca,sunrise_sa));
  sunrise_a
enddef;
```

6.2 Horizontal projection of temporary hour lines

We first consider the horizontal projections and construct exactly eleven paths $tph_1, tph_2, \dots, tph_{10}$, and tph_{11} for the temporary hours 1 to 11 after sunrise. For each curve, we start at the Cancer tropic ($\delta = \varepsilon$) and we end at the Capricorn tropic ($\delta = -\varepsilon$), with $nsteps - 1$ intermediate points. For every declination δ , we compute the sunrise angle.

Now, when the sunrise angle is θ , the angle from sunrise to sunset corresponds to 2θ , and one temporary hour is equivalent to an angle $h = \frac{2\theta}{12}$. So, after having computed the temporary hour (starting at sunrise), we merely subtract it to the sunrise angle, and from this we compute the actual point on temporary hour curve i at declination δ . MB is a point on a local sphere of radius 1, on the parallel of declination δ and with the parametric angle corresponding to temporary hour i . The expression is the same as that used in equation (20), except that ε is to be replaced by δ and the radius is 1. Finally, when projecting the point on the horizontal plane, we apply the same transformations as those used in the previous section when projecting the Babylonian and Italian arcs, and we multiply the projections by s_r , as this factor was omitted in MB . The code also shows how the projected points are connected, and computations are done in between, but hidden by `hide`.

And the curves are of course cut above the ground line. This actually only concerns the cases $i = 10$ and $i = 11$, but we can remove the parts above z_{13} -- z_{14} in every case.

```
nsteps:=10;
path tph[];
for i=1 upto 11:
  % we construct and project nsteps points:
  tph[i]=
    hide(dec:=eps;
      sunrise_a:=sunrise(dec,phi);
      temp_hour_an:=2*sunrise_a*(i/12);
      MB:=sind(dec)*K+cosd(dec)*cosd(sunrise_a-temp_hour_an)*I
          +cosd(dec)*sind(sunrise_a-temp_hour_an)*J;)
    (z1+(Yp(sr*MB),-Xp(sr*MB)))
  for j=1 upto nsteps:
    hide(dec:=(j/nsteps)[eps,-eps]; % varies from eps to -eps
      sunrise_a:=sunrise(dec,phi);
      temp_hour_an:=2*sunrise_a*(i/12);
      MB:=sind(dec)*K+cosd(dec)*cosd(sunrise_a-temp_hour_an)*I
          +cosd(dec)*sind(sunrise_a-temp_hour_an)*J;)
    ..(z1+(Yp(sr*MB),-Xp(sr*MB)))
  endfor;
  draw tph[i] cutbefore (z13--z14);
endfor;
```

6.3 Vertical projection of temporary hour lines

To a great extent, drawing the vertical projections is similar to drawing the horizontal projections. The points in space are computed in the same way. The two changes are the variation of the initial declination, and the different projection.

In the horizontal projection, the curves were always drawn for declinations between ε and $-\varepsilon$. In the vertical projection, we cannot always do that, because the projected curves end up outside of the page. In order to control the drawing, we have introduced the macro `Eps` which has the initial value of the declination vary as a function of the temporary hour i :

```
def Eps(expr i,eps)=
  if i=1: .8eps
  elseif i=7: .6eps
  elseif i=8: .1eps
  elseif i=9: -.37eps
  elseif i=10: -.7eps
  elseif i=11: -.95eps
  else:eps
```

```

fi
enddef;

```

The result can be seen in figure 19 where temporary hour lines 1, and 7 to 11 extend in various ways but stop before reaching the projection of the Cancer tropic.

As far as the projection is concerned, we cannot merely project to the wall at right angle. The projection is gnomonic and goes through the tip of the gnomon. The point we project is $s_r \cdot MB$ and it is projected on the wall which is at distance l_h in the negative x -direction. We do therefore introduce a macro `projV` which takes a point V to project, and a position on the x -axis. It finds point Q on the projection plane:

```

vardef projV(expr V,xw)=
  save Q;
  vector Q;
  Q=whatever[V,vo]=(xw,whatever,whatever);
  Q
enddef;

```

Now, the whole construction is done as follows. The projected point for declination δ and hour curve i is named vx , and its coordinates are used to construct the projection in the upper part of the drawing:

```

vector vx;
path tpv[];
for i=1 upto 11:
  % we construct and project nsteps points:
  tpv[i]=
    hide(dec:=Eps(i,eps);
      sunrise_a:=sunrise(dec,phi);
      temp_hour_an:=2*sunrise_a*(i/12);
      MB:=sind(dec)*K+cosd(dec)*cosd(sunrise_a-temp_hour_an)*I
          +cosd(dec)*sind(sunrise_a-temp_hour_an)*J;
      vx:=projV(sr*MB,-lh);)

    (z2+(Yp(vx),Zp(vx)))
  for j=1 upto nsteps:
    hide(dec:=(j/nsteps)[Eps(i,eps),-eps];
      sunrise_a:=sunrise(dec,phi);
      temp_hour_an:=2*sunrise_a*(i/12);
      MB:=sind(dec)*K+cosd(dec)*cosd(sunrise_a-temp_hour_an)*I
          +cosd(dec)*sind(sunrise_a-temp_hour_an)*J;
      vx:=projV(sr*MB,-lh);)
    ..(z2+(Yp(vx),Zp(vx)))
  endfor;
  draw tpv[i];
endfor;

```

7 Declination sundials

Assuming the declination of the Sun is constant during a day, the shadow of the style describes a hyperbola. There is one hyperbola for each value of the declination. So far, we have only drawn the extreme hyperbolas, corresponding to the Cancer and Capricorn tropics (figures 11, 16, and 19). If we draw the curves corresponding to other declinations, they can be used to read the signs of the zodiac. In figure 20, we show the declination lines for a true longitude of the Sun of 0° (start of Aries and Libra), 30° (start of Taurus and Virgo), and so on. On the right, and from top to bottom, we show the sign entrances from the Winter solstice to the Summer solstice. On the left, and from bottom to top, we show the sign entrances from the Summer solstice to the Winter solstice.

In section 4.2, we saw how to draw the two extreme declination lines, using the macro `def_declination_line`. This macro is reused here, but we have to compute the right cone angles ψ .

The (true) longitude L of the Sun, that is the angle between γ and the Sun measured along the ecliptic, is related to its declination δ through

$$\sin \delta = \sin \varepsilon \sin L \quad (29)$$

where ε is the obliquity of the ecliptic.

From that, we obtain $\sin \delta$, and then δ , assuming $\delta > 0$. And the cone angle is such that $\psi = 90 - \delta$.

The declination lines are then computed for $L = 0^\circ, 30^\circ, 60^\circ$, and 90° , and, at the same time, for the opposite values. For $L = 0^\circ$, we obtain of course the projection of the equator, and for $L = \pm 90^\circ$ those of the tropics. Like in section 4.2, we define VA to be the unit vector directed towards the North pole.

Finally, the points for the declination line corresponding to longitude $30k$ ($k = 0, 1, 2, 3$) are stored in the array `dec_line[2k]`, and those corresponding to longitude $-30k$ ($k = 0, 1, 2, 3$) are stored in the array `dec_line[2k + 1]`. (The projection of the equator appears twice.) These points are used to build the paths `dec_path[2k]` and `dec_path[2k + 1]`, with the following meanings:

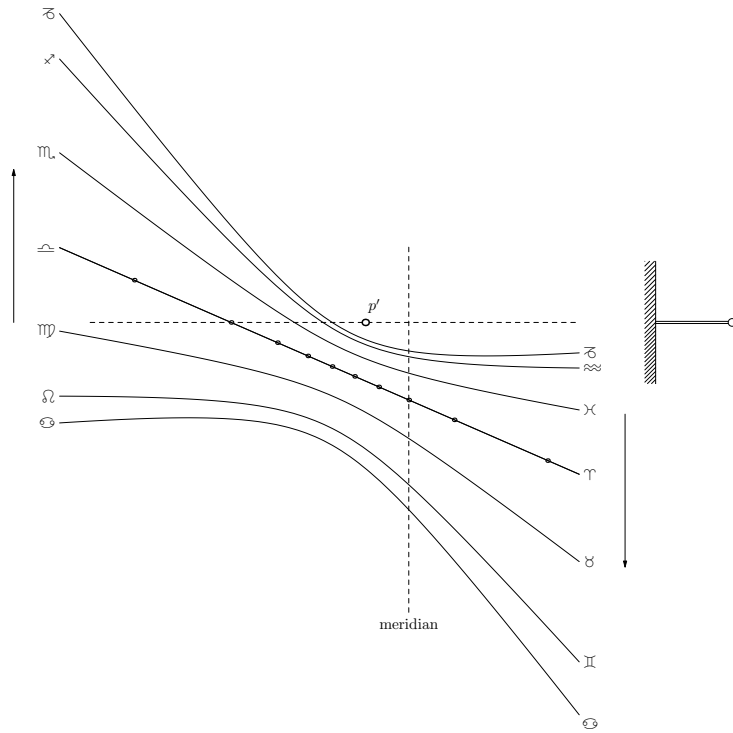


Figure 20: The declination lines for the sign entrances. The dots on the equator line are the hour points, like before. p' is the gnomon, and the vertical dashed line is the intersection of the meridian plane going through the tip of the gnomon with the wall. The arrows show in which order the signs appear: Aries (Υ) is followed by Taurus ($\var�$), followed by Gemini (II), Cancer (C), Leo (L), Virgo (M), Libra (L), Scorpio (M), Sagittarius (X), Capricornus (C), Aquarius (A) and, finally, Pisces (P).

path	longitude	start of
dec_path0	0° and 180°	Aries (♈) and Libra (♎)
dec_path2	30° and 150°	Taurus (♉) and Virgo (♍)
dec_path4	60° and 120°	Gemini (♊) and Leo (♌)
dec_path6	90°	Cancer (♋)
dec_path1	0° and 180°	Aries (♈) and Libra (♎)
dec_path3	210° and 330°	Scorpio (♏) and Pisces (♓)
dec_path5	240° and 300°	Sagittarius (♐) and Aquarius (♑)
dec_path7	270°	Capricornus (♐)

The code is now straightforward:

```

pair dec_line[] [];
path dec_path[];
numeric sdec,cdec;

VA=(-cosd(phi)*cosd(alpha),cosd(phi)*sind(alpha),sind(phi));
for k=0 upto 3:
  % we consider the longitudes 0, 30, 60 and 90 deg.
  sdec:=sind(eps)*sind(30k);
  cdec:=1+-sdec; % Pythagorean subtraction
  psi:=90-angle((cdec,sdec)); % cone angle
  def_declination_line(x11-u,x12)(VA,lh,psi,z2)
    (dec_line[2k])(dec_line[2k+1]);
  % projection of positive declinations
  dec_path[2k]=(dec_line[2k][0]
    for i=1 upto 30:..dec_line[2k][i] endfor);
  % projection of negative declinations
  dec_path[2k+1]=(dec_line[2k+1][0]
    for i=1 upto 30:..dec_line[2k+1][i] endfor);
  draw dec_path[2k];
  draw dec_path[2k+1];
endfor

```

8 Sunrise and sunset

A sundial can easily show certain sunrises and sunsets. However, depending on the orientation of the wall and the period of the year, sunrise can occur before the wall is lit and sunset can occur when the wall is no longer lit.

Notwithstanding, in the cases where the reading is possible, the times for sunrise and sunset can be obtained using the declination lines and the normal hours (figure 21). The intersection(s) of the current declination curve with the horizon line shows where the Sun is located at sunrise and/or sunset. It is then straightforward to check to which time this corresponds. At the

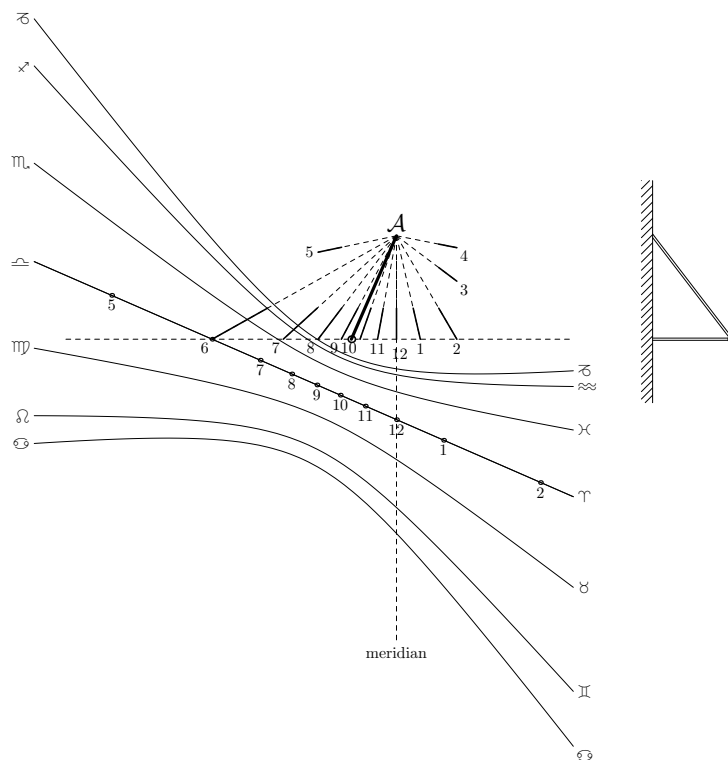


Figure 21: The declination curves with the normal vertical sundial to read sunrise and sunset. The two extreme curves are the solstitial curves and the straight line is the equinox line with the hour points. The vertical dashed line is the intersection of the meridian plane going through the tip of the gnomon with the wall.

start of Pisces (♓) or Scorpio (♏), for instance, the dial shows readily that sunrise occurs around 7 (solar time), at this location, of course.

9 Sidereal time sundials

A sundial can also show sidereal time. This was apparently first reported by Hugo Michnik in 1922 [7]. Although Michnik's work is very detailed, understanding sidereal sundials is straightforward and does not require any computation or complex drawing.

The principle is as follows: the sidereal time is the hour angle of the γ point, and hence, at two different times having the same sidereal time T_s , the positions of the stars with respect to the Earth are exactly the same. This

includes the position of the apparent path of the Sun, the ecliptic (figure 22). Depending on the time of the year, the Sun is somewhere on this path and the projection of the Sun is a point on the projection of the ecliptic at this moment. But at a given time, and since the ecliptic is a great circle of the local sphere, the projection of the ecliptic is a straight line, and hence to every value of the sidereal time corresponds a straight line—the projection of the ecliptic at this time (figure 23).

It can be seen from figure 23 that the solstitial lines are the envelopes of the sidereal hour lines. This is of course a consequence of the ecliptic being tangent to the two tropics (figure 22). Moreover, at noon, the solstitial curves are met at particular values of the sidereal time: $T_s = 6$ h at the Summer solstice, and $T_s = 18$ h at the Winter solstice, which is obvious from figure 22. At the time of the equinoxes, noon corresponds to $T_s = 0$ h (Spring) and $T_s = 12$ h (Fall).

We can of course have vertical or horizontal sidereal sundials, and in fact sidereal sundials on any other surface. Michnik gives for instance an horizontal dial [7, p. 447–448]. In this section, however, we only consider a vertical dial.

Once the sidereal time lines have been drawn (one for each value of the sidereal time), the shadow of the tip of the gnomon can be used to read sidereal time.

9.1 Drawing a vertical sidereal sundial

In order to draw a vertical sidereal hour line, we first find the unit vector towards the North pole of the ecliptic. This vector is a function of the sidereal time, or of the position of γ . But first, we define the celestial North pole. We have already used a vector for that purpose, namely VA , but without showing how it is obtained. Here, we pause to examine how this vector can be constructed. First, we let uPn be a unit vector towards the North:

```
vector uPn;
uPn=rotatearound(vj,vk,90-alpha);
% uPn=vj*sind(alpha)-vi*cosd(alpha); % equivalent
```

Then, Eqa is defined to be a unit vector towards East:

```
vector Eqa;
Eqa=vecprod(uPn,vk);
% Eqa=vj*cosd(alpha)+vi*sind(alpha); % equivalent
```

And uPn is rotated by φ around the Eastern vector, yielding a unit vector towards the celestial North:

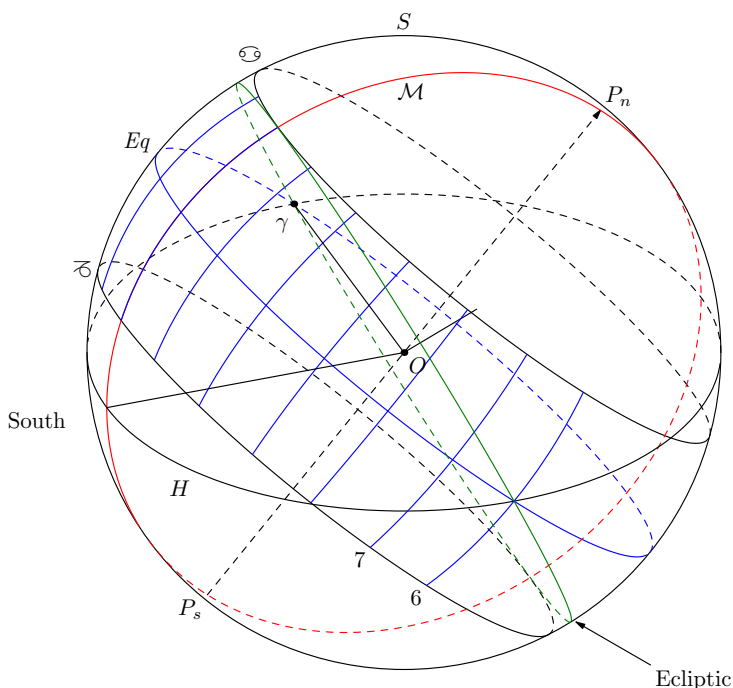


Figure 22: The local sphere with the ecliptic at $T_s = 6$ h.

```
uPn:=rotatearound(uPn,Eqa,phi);
```

Then, in order to draw sidereal hour line i , we first define the ecliptic North pole. And again, we do this in two stages. First, by rotating uPn by $-\varepsilon$ around Eqa , we actually obtain the ecliptic North pole when $T_s = 6$ h (see figure 22). The rest is elementary, since we merely have to rotate this vector around uPn by the angular difference between the sidereal time and 6. This angle difference is equal to $15 \times 6 - 15i = 90^\circ - 15i$.

```
vector ecliptic_pole;
for i=0 upto 23:
    ecliptic_pole:=rotatearound(uPn,Eqa,-eps);
    ecliptic_pole:=rotatearound(ecliptic_pole,uPn,90-15i);
    % drawing of sidereal line i
    ...
endfor;
```

Each line is drawn by a call to the macro `sidereal_line`, with a number of parameters. Since some of the parameters depend on i , we can either make 24 different calls, or define new macros computing the parameters from i . Let us assume, for simplicity, that we have 24 different calls, only one being

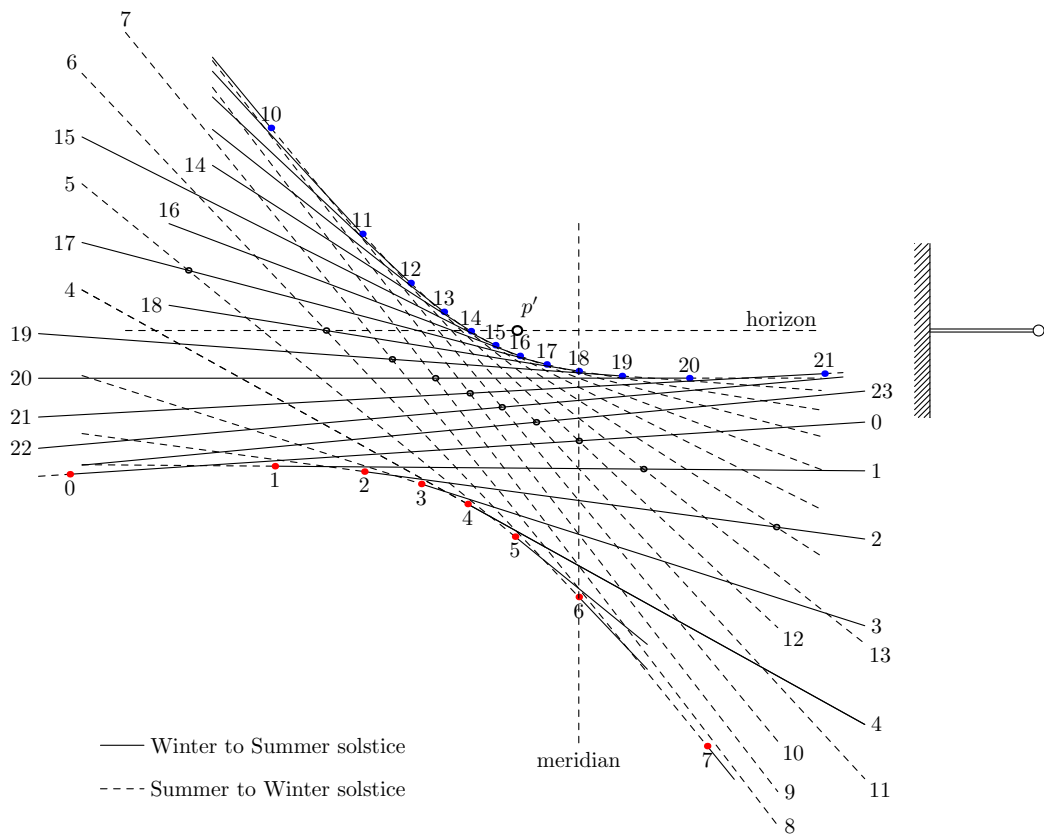


Figure 23: The sidereal time lines on a vertical dial. Plain lines are to be used from Winter solstice to Summer solstice, whereas dashed lines are to be used from Summer solstice to Winter solstice. The part over the horizon is drawn only for convenience and is of course never reached by the shadow. The vertical dashed line is the intersection of the meridian plane going through the tip of the gnomon with the wall. The lower hyperbola is met at the Summer solstice, whereas the upper one is met at the Winter solstice.

chose for a given value of i . The parameters shown here are simplified, but the three main parameters are the y -range for the lines (the y -axis is parallel to the wall, and directed towards the right, see figure 1), and the pole of the ecliptic:

```
if i=0:sidereal_line(-11u,8u,ecliptic_pole);fi;
...
if i=10:sidereal_line(-7u,6u,ecliptic_pole);fi;
...
if i=23:sidereal_line(-10u,8u,ecliptic_pole);fi;
```

In order to draw a line, we determine two of its points and connect them. The two points are those for the y -range, named here y_A and y_B . The two points, p_a and p_b are merely shifted by z_2 , as this is the origin of the coordinates with the macro `sidereal_line_point`:

```
vardef sidereal_line(expr ya,yb,pole)=
  save pa,pb;
  pair pa,pb;
  pa=z2+(ya,sidereal_line_point(ya,ecliptic_pole));
  pb=z2+(yb,sidereal_line_point(yb,ecliptic_pole));
  draw pa--pb;
enddef;
```

Finally, `sidereal_line_point` is the function corresponding to the projected ecliptic. It takes two parameters, the value on the y -axis, and the ecliptic pole. This function is easy to establish: assuming the pole vector components to be (P_x, P_y, P_z) , an equation of the ecliptic plane, taking the center of the local sphere as the origin, is

$$xP_x + yP_y + zP_z = 0 \quad (30)$$

The intersection with the wall is obtained by setting $x = -l_h$, hence:

$$z = \frac{l_h P_x - y P_y}{P_z} \quad (31)$$

which, when translated in METAPOST code, becomes:

```
def sidereal_line_point(expr y,pole)=
  ((lh*Xp(pole)-y*Yp(pole))/Zp(pole))
enddef;
```

9.2 Reading a sidereal sundial

9.2.1 The problem of multiple lines

The dials produced with the previous procedure appear very cluttered, because there are two sets of lines, and reading can become difficult. Indeed, except at the time of solstices, and given a position of the Sun on the local sphere, there are actually two possible ways to position the ecliptic, which means that the shadow of the Sun lies on two different lines. Only one must be kept.

There is also another problem, namely that near the solstices (upper and lower curves), reading the sidereal time is difficult because there is some uncertainty in the position of the shadow and many lines cross nearby.

The first problem can be avoided by distinguishing the lines in one way or another, or by drawing two different dials. In figure 23, for instance, the lines are split in two groups: one is drawn in plain lines and is valid from the Winter to the next Summer solstice, the other is drawn in dashed lines and is valid from the Summer to the next Winter solstice. The extremal curves are valid at the solstices.

9.2.2 Moving through the lines

In order to see how the sidereal time can be read, it is convenient to follow the Sun during its journey through the days, and then through the year. If we start on the first day of Spring, the Sun is located at the γ point and is of course in the equator plane. Hence, the path of its shadow is the equator line (\sphericalangle/Υ). On this day, of course, at noon, the Sun as well as γ are in the meridian plane, and therefore $T_s = 0$. The shadow is then located at the intersection of the equator line with the meridian line. Two sidereal lines go through this point: 0 and 12. The latter is to be discarded for now.

Now after noon, the shadow moves towards the right, and the lines 1, 2, etc. are crossed until the shadow does no longer reach the wall (in our particular case with $\alpha = 29^\circ 40'$). The next day, the case is almost the same, but the path followed by the shadow is slightly below the equator. It starts at sunrise, which is slightly before 6 a.m. (solar time), and therefore $T_s \approx 18$ and it ends at sunset, slightly after 6 p.m., hence $T_s \approx 6$. The sidereal time for noon increases by about 4 minutes every day, and after about two weeks noon occurs at $T_s = 1$, then $T_s = 2$, etc., until $T_s = 6$ at the Summer solstice. The shadow of the Sun at noon is now at its lowest point, reaching the Cancer curve. For these three months, the sidereal lines that are used are those that are mostly horizontal.

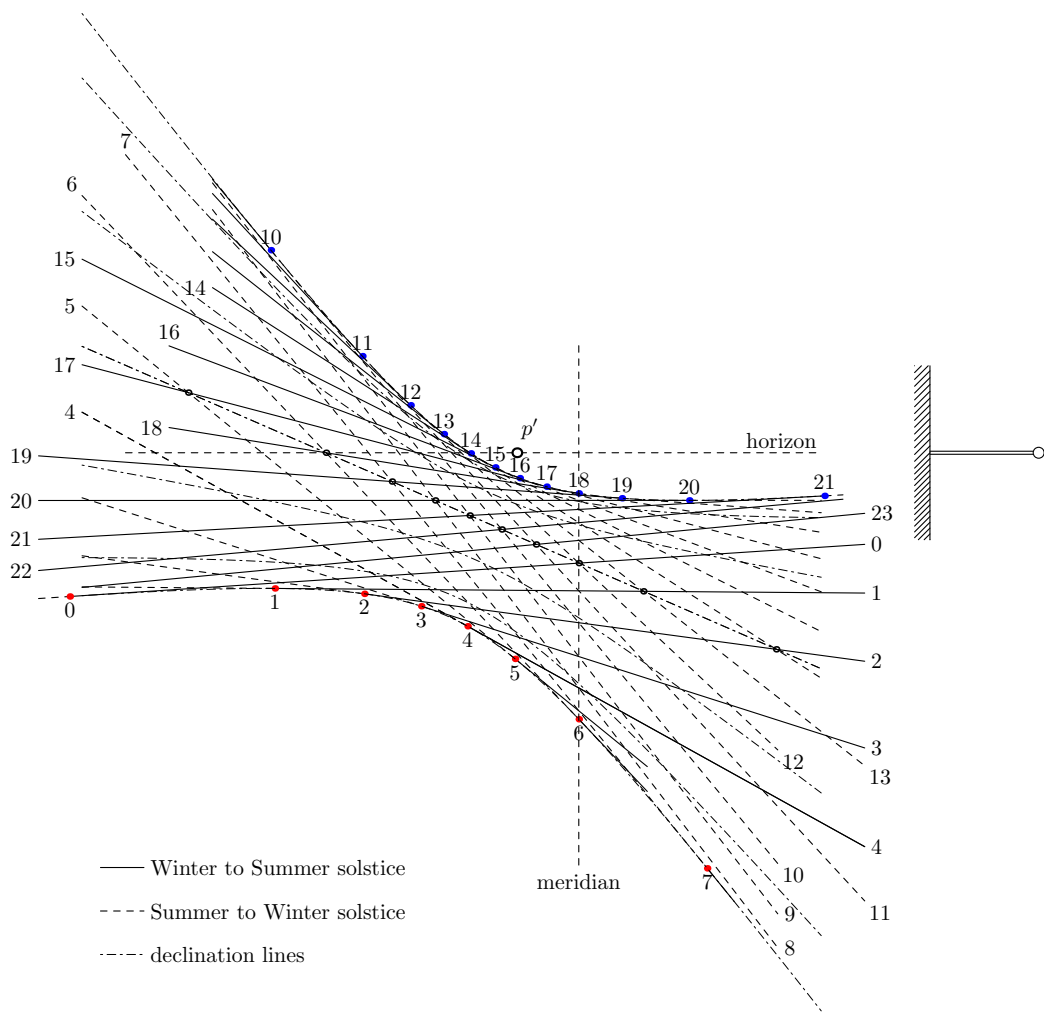


Figure 24: The sidereal time lines, together with the declination curves (thin dashed lines).

9.2.3 A practical approach

A similar reasoning can be applied to follow the shadow from the Summer solstice to the Winter solstice, and so on, but one needs to be very careful in this reasoning. In fact, it is a lot easier to find out which sidereal line to use by first computing mentally an approximation of the sidereal time. The sidereal time at noon is approximately the number of half months since the beginning of Spring. And the sidereal time n hours after/before noon is approximately n hours more/less than the sidereal time at noon. These approximations are amply sufficient to choose between the two sets of lines. Of course, we need to know the solar time, that might be provided by a conventional sundial.

An example will make this clear. Consider reading the sidereal time around 10 a.m. on June 30. On this day at noon, the sidereal time is approximately $T_s = 6.5$. At 10 a.m., it must therefore be approximately $T_s = 4.5$. This means that we need only consider the two lines 4 and 5, and the reading is made a lot simpler. This procedure can be applied even when there is an uncertainty of several hours on the solar time, except near the solstitial lines, where the reading is intrinsically difficult.

10 Analemmatic sundials

Analemmatic dials are another kind of sundials, of which there are actually numerous varieties [1]. Here, we consider only the commonest type, namely an horizontal dial with a vertical style.

In this kind of dial, the hours are marked along an ellipse, and the style must be positioned according to the period of the year, or, what is equivalent, according to declination of the Sun. Such dials are adequate for places such as squares, where a person can act as the style.

Although these dials seem strange and mysterious, they are actually very simple, and do not require any fancy mathematics, contrary to what is often seen, for instance in Rohr's book [11].

Figure 25 shows the construction of the elliptic dial. The dial is actually the horizontal projection of the equator circle on a local sphere. It is therefore the same curve as the equator curve seen in previous construction. On the figure, this dial is drawn with a thicker line. On the figure, we have also put the center of the sphere O where the tip of the gnomon was located in previous examples. We have also shown the position of the wall, for mere reference. The horizontal gnomon or the wall do not play any part here and they can now be discarded.

Figure 25 also shows the horizontal projections of the two tropics (\ominus and \oslash), as well as the normal hour lines corresponding to the hours of true solar time.

We now consider what happens at a certain time, for instance when the hour angle of the Sun is 22, that is when it is 10 a.m. solar time. At the equinoxes, the Sun is located in $S_{22,0}$ and the projection of O reaches the local sphere at the opposite point Γ_{22} . If we now assume that O is on the ground and that a vertical style is put at O , then the shadow of that style is the line $O\Gamma_{22}$. This reasoning is true for every position of the Sun, and therefore the analemmatic dial clearly works at the equinoxes: the shadow of a style put at O goes through a point of the dial symmetrical to the direction of the Sun. We can therefore merely translate the numbering of the hours, or swap “a.m.” for “p.m.”

Consider now what happens when it is 10 a.m. at the Summer solstice. The Sun is at $S_{22,\varepsilon}$. The direction of its rays are parallel to $S_{22,\varepsilon}O$. What is expected is that the rays reach the same point Γ_{22} . How can this be achieved? If the style is put at O , the shadow of the Sun will be more towards Γ_{21} which would not be correct. The answer is simple: draw a parallel to $(S_{22,\varepsilon}O)$ and going through Γ_{22} . This line cuts the meridian at G . It is clear that a vertical style put at G will cast a shadow which goes through Γ_{22} at the Summer solstice. If we know where the Sun is, we can now easily find how to position the style. It is however useful to be able to compute this position.

This is done as follows. In figure 25, we can actually consider that there are two parallel lines, $(S_{22,\varepsilon}O)$ and $(G'\Gamma_{22})$, and that these lines are in space, G being the horizontal projection of a point G' . Figure 26 shows the plane containing the polar axis and point Γ_{22} . Beware that this plane is not vertical, except in rare circumstances. The line marked “horizon” is therefore only the intersection of the figure plane with the horizon, not a depiction of the whole horizon. Point G' is on the polar axis, slightly beyond the center \ominus_c of the Cancer tropic, and its position only depends on r and ε , not on the time of day. We have $OG' = r \tan \varepsilon$. Now, since the angle between the polar axis and the horizon is exactly the latitude φ , it comes that $OG = OG' \cos \varphi = r \tan \varepsilon \cos \varphi$. This is the position of the vertical style at the Summer solstice.

Since we didn't assign any value to ε , and since we did not even use the fact that it is the obliquity of the ecliptic, we can actually conclude that the result is valid when ε is replaced by the declination δ of the Sun: the style should be put such that $OG = r \tan \delta \cos \varphi$. In this formula, δ is a function of the day, and r is the semi-major axis of the projected ellipse.

As it appears, the position of the style at the Summer solstice is very close—but not identical—to that of the center of the Cancer parallel. The distance between O and the projection of \ominus_c is $r \sin \varepsilon \cos \varphi$ and the distance

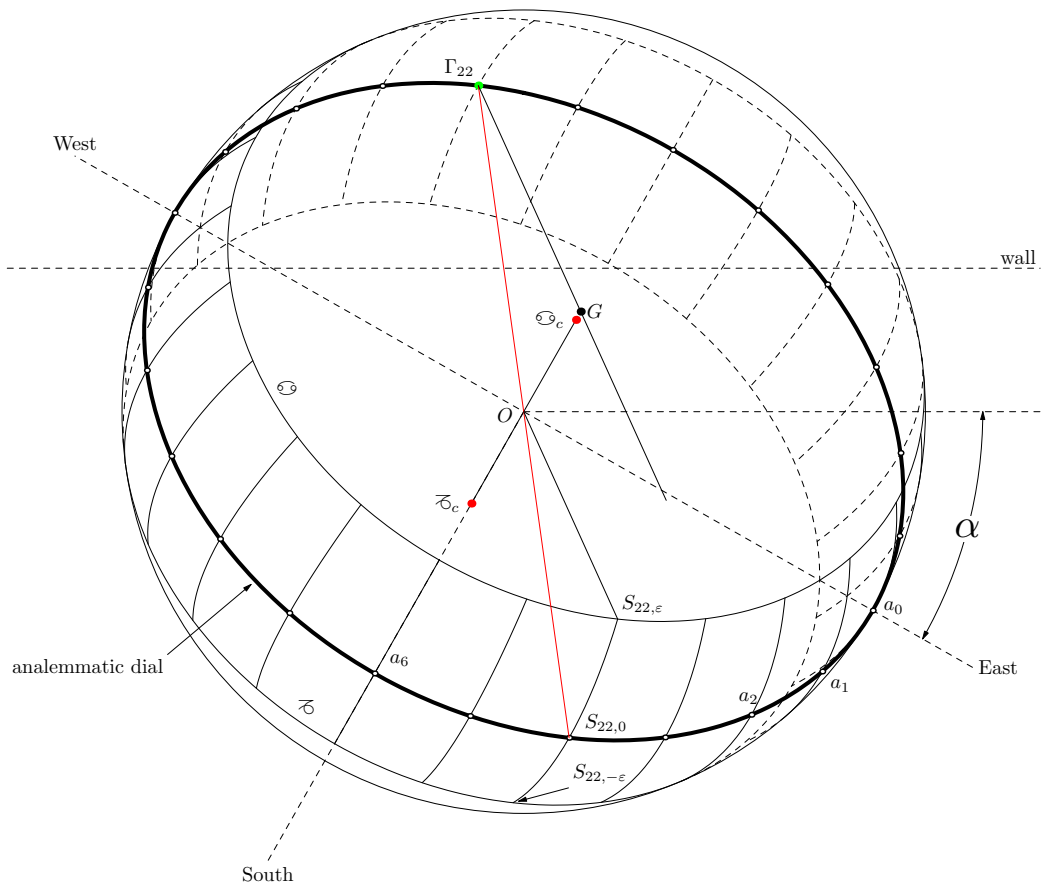


Figure 25: The construction of the analemmatic sundial.

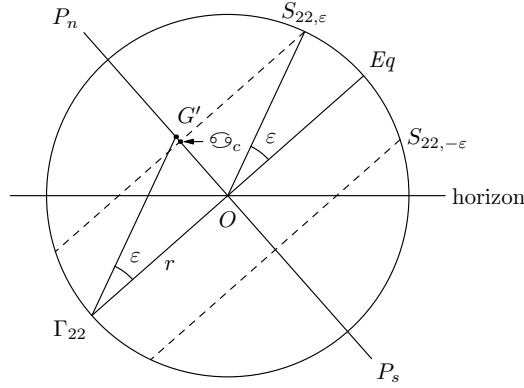


Figure 26: Position of the style in the analemmatic sundial.

between the two points is $r(\tan \varepsilon - \sin \varepsilon) \cos \varphi$ which of course tends to 0 when the declination tends to 0, but a lot faster than either term alone. This is easy to prove, using the developments of sin and tan:

$$\sin x = x - \frac{x^3}{3!} + o(x^3) \quad (32)$$

$$\tan x = x + \frac{x^3}{3} + o(x^3) \quad (33)$$

Hence:

$$r(\tan \varepsilon - \sin \varepsilon) \cos \varphi = r \left(\frac{\varepsilon^3}{2} + o(\varepsilon^3) \right) \cos \varphi \quad (34)$$

$$\approx r \frac{\varepsilon^3}{2} \cos \varphi \quad (35)$$

where ε is expressed in radians.

11 Conclusion

Our study has now come to an end, and, taking three particular dials as a starting point, we have covered a wide range of material related to sundials. Dials showing equinoctial hours (normal, Babylonian and Italian) have been considered, as well as dials showing temporary hours. More unusual dials such as elevation and azimuth dials, declination dials, sidereal time dials, and analemmatic dials have also been studied.

In each case, we have strived to produce accurate drawings and to show how they were obtained using a high-level drawing language such as METAPOST. It should be easy to adapt what we have done to other graphical languages, and even to generalize it.

Computations were kept at a minimum, and we have for instance refrained from giving equations for the curves on several dials, because we didn't need them. We didn't give any equation for the Babylonian lines, for instance, and it didn't prevent us from drawing them accurately.

And although this wasn't that much the focus of this study, we have also tried to produce parametric drawings, in such a way that the dials can be adjusted to a different latitude and orientation without difficulty. However, as it can be expected, producing sundials totally automatically is not that easy, because there are also aesthetic requirements. For instance, the labelling of figure 17 is only right for certain values of φ and α . Adapting the location of the labels, arrows, intermediate points, line lengths, etc. as functions of φ , of α , and also of the gnomon dimensions, may prove a real challenge that we haven't really tackled. Nevertheless, for "pure" sundials, without labels, and only simple lines, producing parametric drawings is rather easy, and hence experimenting different configurations is also possible.

References

- [1] Margarida Archinard. Les cadrans solaires analemmatiques. *Annals of Science*, 62(3):309–346, July 2005.
- [2] Henri Bach, Jean-Pierre Rieb, and Robert Wilhelm. *Les trois horloges astronomiques de la cathédrale de Strasbourg*. Strasbourg: Éditions Ronald Hirlé, 1992.
- [3] Christophorus Clavius. *Gnomonices libri octo, in quibus non solum horologiorum solarium, sed aliarum quoquam rerum, quæ ex gnomonis umbra cognosci possunt, descriptiones geometricè demonstrantur*. Roma: Franciscus Zanettus, 1581. Several digital versions are available on internet.
- [4] Thomas Stephens Davies. An Inquiry into the Geometrical Character of the Hour-Lines upon the Antique Sun-Dials. *Transactions of the Royal Society of Edinburgh*, 12:77–122, 1834.
- [5] Michel Goossens, Frank Mittelbach, Sebastian Rahtz, Denis Roegel, and Herbert Voß. *The L^AT_EX Graphics Companion, Second Edition*. Boston: Addison-Wesley, 2008. First printing, July 2007.
- [6] John Douglas Hobby. A user’s manual for METAPOST. Computing Science Technical Report 162, AT&T Bell Laboratories, 1992.
- [7] Hugo Michnik. Konstruktion einer siderischen Sonnenuhr. *Astronomische Nachrichten*, 216(5184):441–450, 1922.
- [8] Günther Oestmann. *Die astronomische Uhr des Straßburger Münsters: Funktion und Bedeutung eines Kosmos-Modells des 16. Jahrhunderts*. Berlin: Verlag für Geschichte der Naturwissenschaften und der Technik, 2000. Second edition. First edition in 1993.
- [9] Denis Roegel. A complex drawing in descriptive geometry. *TUGboat*, 28(2):218–228, 2007.
- [10] René Rodolphe Joseph Rohr. *Les cadrans solaires anciens d’Alsace*. Colmar: Éditions Alsatia, 1971.
- [11] René Rodolphe Joseph Rohr. *Sundials: History, Theory, and Practice*. New York: Dover Publications Inc., 1996.
- [12] Denis Savoie. *La gnomonique*. Paris: Les Belles Lettres, 2001.

- [13] Hervé Staub. *Les horloges silencieuses d'Alsace*. Strasbourg: Éditions Coprur, 1997.
- [14] Paul Werkmeister. Über die Zeitmesser des Strassburger Münsters insbesondere die Sonnenuhren am Giebel der Südseite. *Strassburger Münsterblatt*, 6:62–74, 1912.