Using 3D Graphics to Explain Why a Sundial Can Go Backwards in the Tropics

Sun Yifan

An academic exercise presented in partial fulfillment for the degree of Bachelor of Science with Honors in Mathematics

Supervisor: Associate Professor Helmer Aslaksen

Department of Mathematics

National University of Singapore

2010/2011
Acknowledgements

I would like to express my deep gratitude to my supervisor, A/P Aslaksen, for his patient guidance, enthusiastic encouragement and useful suggestions for this final year project. I am so grateful for his willingness to give time so generously. Despite his busy schedule, he still meets me every week to guide and support me throughout the year. I have benefited and learnt greatly from this project.

I would also like to thank my lovely family, my understanding boyfriend Jesse, and my supportive friends Huimin and Jingxian for their continued encouragement and help. Last but not least, a special thank goes to the spirits in Heaven, who give and support my life on this beautiful Earth. I have had a really special and meaningful year because of you all.
Summary

The objective of this project is to find a good 3D graphics tool to explain the Hezekiah Phenomenon, which refers to a story in the Bible about a sundial going backwards. We will explain why in the tropics this does in fact happen for part of the year.

To understand the Sun’s path and the shadow path of a vertical gnomon (the stick of a sundial) throughout the year at different latitudes, we will need some knowledge of basic astronomy, which we will discuss in this thesis.

During a trip to Mexico, my supervisor realized that the Sun changed direction in the sky. This was the topic of another thesis some years ago (S. Lee, X.J. Sim, and Y. Yang, 2006). However, that thesis used freehand drawing to illustrate the phenomenon. This project will try to illustrate it using Mathematica. In the earlier thesis, the phenomenon was referred to as Maya Phenomenon, but another student, Wu Chengyuan, alerted my supervisor to the story of Hezekiah in the Bible, so we now refer to it as the Hezekiah Phenomenon.

One of the objectives of this project is also to explore and evaluate Mathematica in 3D modeling and visualization in astronomy. Our interest lies in modeling the Sun’s path in the sky and the gnomon’s shadow path on the ground. Based on the criteria of Wu (2009), we found that Mathematica is inherently 3D, easy to use, efficient to code, has powerful 3D graphics, and provides an extensive astronomical dataset. It is a perfect tool for our purpose.

We will use the package AstronomicalData to model how the Sun and the shadow depend on the observer’s position and the date and time. The 3D graphics generated can be rotated and zoomed. We also built a control panel to change the observer’s position and date interactively.

In addition, we studied where, which part of the year, and what time of the day we can observe the Hezekiah Phenomenon, and when it will be most noticeable.

Lastly, we created animations of the Sun’s path and the shadow path.
# Table of Contents

ACKNOWLEDGEMENTS .................................................................................................................. 2
SUMMARY ....................................................................................................................................... 3
TABLE OF CONTENTS ...................................................................................................................... 4
1 BASICS OF ASTRONOMY ........................................................................................................... 5
2 MATHEMATICA .......................................................................................................................... 9
3D .................................................................................................................................................... 9
AUTOMATIC DASHING OF HIDDEN LINES .................................................................................. 10
COMPUTATION OF INTERSECTIONS .......................................................................................... 10
USABILITY ....................................................................................................................................... 11
EFFICIENCY OF CODE ............................................................................................................... 11
SPEED ........................................................................................................................................... 11
dRAWING OF CONES .................................................................................................................. 11
ROTATING THE OUTPUT ............................................................................................................. 12
3 SUN’S PATH AND SHADOW PATH OF A VERTICAL GNOMON ................................................. 14
4 THE HEZEKIAH PHENOMENON ................................................................................................. 17
5 FINDING THE BEST LOCATION AND TIME TO OBSERVE THE HEZEKIAH PHENOMENON ................................................................................................................................. 21
6 ANIMATION OF THE SUN’S PATH AND THE SHADOW PATH ................................................. 24
GLOSSARY ....................................................................................................................................... 26
APPENDIX 1 THE HEZEKIAH PHENOMENON ............................................................................ 28
APPENDIX 2 AZIMUTH AND ALTITUDE PLOTS .......................................................................... 40
BIBLIOGRAPHY ............................................................................................................................. 42
1 Basics of Astronomy

This sector is taken from the Chapter 1 and 2 in the thesis of S. Lee, X.J. Sim, and Y. Yang in 2006, and we have amended and added some parts.

1.1 Horizontal Coordinate System

To construct a coordinate system on a sphere we use a base circle and a base point on the circle. The first coordinate measures the angular distance from the base circle and the second coordinate measures the angular distance along the base circle.

For horizontal coordinate system we use the horizon NSEW as the base circle, and the North point as the base point. An observer is at point O and looking up at the celestial sphere, which refers to an imaginary sphere surrounding the earth. The point Z that marks the overhead position of the observer is called the zenith. We can define the position of a celestial body, C, by stating its azimuth and altitude. The azimuth is defined by projecting point C onto a point D on the horizon plane and measuring the clockwise angle from N to D. Its altitude is the angular distance measured vertically upwards from the horizon, marked by the arc DC. Since the celestial bodies move across the sky, the altitude and azimuth values depend on the observer’s location and time.
1.2 Equatorial Coordinate System

We will now introduce the equatorial coordinate system. We use the celestial equator, which refers to the projection of the Earth’s equator onto the celestial sphere, as the base circle in this system. Spring equinox, where the ecliptic crosses the celestial equator, is the base point. In this way, we can indicate the position of a celestial body C on the celestial sphere. The Earth rotates about its axis which points to the North celestial pole and it orbits the Sun in a plane called the ecliptic. Alternatively, the ecliptic may be defined as the apparent path of the Sun on the celestial sphere. Project the position of C onto the celestial equator to the point D. The angular distance between the spring equinox and point D is the right ascension of C. From D, the angular distance measured upwards along a meridian line from the celestial equator to the celestial body C is its declination.
1.3 Altitude vs. Latitude

The *latitude* of an observer standing at point O is equal to the altitude of the North Star.
During the summer solstice, the Earth’s axis is tilted towards the Sun while during the winter solstice, it tilts away from the Sun. Thus, an observer situated at the Northern hemisphere will receive more Sun during the summer solstice than the winter solstice. However, during the equinoxes, everywhere on the Earth will experience equal amount of day and night.
Mathematica

Mathematica is a commercial software program that widely used in technical computing, such as scientific, engineering and mathematical fields. It has powerful and extensive built-in features and data sets from many academic and industrial areas, including astronomy.

Mathematica is split into two parts, the kernel and the front end. The kernel interprets input expressions including Mathematica code and data, and returns result expressions. The front end provides a Graphical User Interface (GUI), which is used to create and edit documents in Notebook format containing program code and results including typeset mathematics, graphics, GUI components, tables, and sounds (Wolfram Mathematica 8 Documentation).

3D

Mathematica offers a number of powerful functions to create and manipulate 3D graphics using familiar physical metaphors. It provides not only real-time 3D manipulation, but also detailed programmatic control of features such as orientation, viewing geometry and lighting, all tightly integrated with Mathematica's symbolic language and real-time dynamic capabilities.

3D objects in Mathematica are represented and rendered by combining 3D graphics primitives, such as Point[{x,y,z}] (point with coordinates (x, y, z)), Line[{{x1,y1,z1},{x2,y2,z2},...}] (line through the points {x1, y1, z1}, {x2, y2, z2}, ...), Polygon[{{x1,y1,z1},{x2,y2,z2},...}] (filled polygon with the specified list of corners), and Arrow[{pt1,pt2}] (arrow pointing from pt1 to pt2), etc. Such 3D graphics primitives can be created simply by one line of code. For example, “Graphics3D [Sphere[{0, 0, 0}]]” will produce the 3D graphics below:

![Figure 2.1 Draw a Sphere in Mathematica](image)
To build complex 3D graphics, we can add more parameters in the code above, such as the viewing angle, size, color and opacity, and combine the sphere with other 3D graphics primitives. For example, the 3D graphics below, which represents a simple horizontal reference system, can be created using less than 10 lines of code combining several 3D graphics primitives (points, lines, circles, spheres):

![Figure 2.2 Combine 3D Graphics Primitives](image)

**Automatic Dashing of Hidden Lines**

As can be observed in Figure 2.2, Mathematica does not automatically dash the hidden lines, but marks the hidden lines in a lighter color. The difference is easily distinguishable.

**Computation of Intersections**

To compute the intersections of graphs, we can solve the equation system in Mathematica via function `Solve`. For example, if we want to get the intersection of a circle and a line, we can use `Solve[x^2 + y^2 == 1 && x == 0, {x, y}]` and system will return the result of 2 intersection points: `{y -> -1, x -> 0}, {y -> 1, x -> 0}`. Similarly we can compute the intersections of any graphs.

After we get the intersection points, we can draw the graphs and their intersections, as illustrated in Figure 2.3 below.
Usability

3D programming in Mathematica is very easy to use, because of the idea of the symbolic graphics language, i.e. building up 3D graphics from symbolic primitives, which can be manipulated using all standard Mathematica functions and seamlessly integrated with text, math, or tables.

Efficiency of Code

To draw Fig.2.3, only approximate 20 lines of code are needed. Compared to other software packages that we have used such as PGF/TikZ, the efficiency of code is one of the advantages of using Mathematica.

Speed

It takes less than 1 second to compile the code and output graphics in Figure 2.2.

Drawing of Cones

It is vital for us to be able to draw cones and conic sections for the purpose of the project. This is because the sunlight can be visualized as a spotlight with a conical beam and the
The gnomon’s shadow path can be viewed as conic sections made by the sunlight and the horizon. The changes of the gnomon’s shadow, therefore, can be studied through the conic sections by changing the position of the beam, which is due to changes in latitude (S. Lee, X.J. Sim, and Y. Yang, 2006).

Although cones are not 3D graphics primitives in Mathematica, it is fairly simple to draw a 3D cone compared to many other software packages. We can draw cones geometrically, by rotating a line around the cone’s axis. For example, the two lines of code below produce a double cone with the vertex at (0,0,0), the axis of (0, 0, 1), and the angular aperture of $\pi/6$.

\begin{verbatim}
RevolutionPlot3D[{r/Tan[ArcCos[1 - (Pi/6)/(2 Pi)] } , {r, 0, 1}, RevolutionAxis -> {0, 0, 1}],
RevolutionPlot3D[{r/Tan[ArcCos[1 - (Pi/6)/(2 Pi)] } , {r, 0, -1}, RevolutionAxis -> {0, 0, 1}],
\end{verbatim}

![Figure 2.4 Draw a Double Cone, 3D](image)

**Rotating the Output**

The 3D visualization and animation tools give users the capability to rotate and zoom the 3D graphics in real-time using a mouse. In addition, we can add GUI to the graphics so that we can use a control panel to change the input variables, such as the longitude/latitude of the observer, the length of the vertical gnomon, and the radius of the horizon circle.
In conclusion, Mathematica is intuitive and simple to use, provides powerful features and functions in 3D programming, and is perfect for our purpose of modeling in interactive astronomy. Combining with the result of Wu (2009), Table 2.1 summarizes the evaluation results among a variety of software packages.

<table>
<thead>
<tr>
<th>Inherently 3D?</th>
<th>Metapost</th>
<th>Asymptote</th>
<th>PSTricks</th>
<th>Sketch</th>
<th>PGF/Tikz</th>
<th>Tikz-3dplot</th>
<th>Mathematica</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Automatic dashing of hidden lines</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Calculate points of intersection</td>
<td>Yes (2D)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Draw curve of intersection between solid and plane</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Able to draw cones</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Output is rotatable</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
3 Sun’s Path and Shadow Path of a Vertical Gnomon

Figure 3.1 is from the previous honors thesis by S. Lee, X.J. Sim, and Y. Yang (2006). In the picture, the NSEW plane is the observer’s horizon. The picture shows the horizon plane, the Sun’s path, and the gnomon’s shadow path for an observer in the Arctic.

However, the picture is manually drawn and might not be very accurate. Moreover, it is not interactive and we are not able to manipulate the latitude/time of the observer to show the interaction of the Sun’s path and the gnomon’s shadow path dynamically. Therefore, if we want to have a similar illustration of an observer in the tropics, then we need to start over and draw a new picture from scratch.

Figure 3.1. Horizontal Reference System (S. Lee, X.J. Sim, and Y. Yang, 2006)

Figure 3.2.a is the interactive 3D model that we built in Mathematica. It exhibits the Sun’s path, the gnomon’s shadow path and the equinoctial shadow line for an observer at latitude 74.5 N on 11 August.

In the model, the Sun’s path is the perimeter of the circular base of the cone’s upper nappe. The observer’s latitude determines the angle between the cone’s axis and the Z-axis while the date of observation determines the declination of the Sun, and the angular aperture of the cone. Therefore, the Sun’s path depends on the shape of the cone, which depends on the observer’s latitude and date. In the model, we have:

angle between the cone’s axis and Z-axis: $u = 90^\circ - \text{Latitude}$;

angular aperture of the cone: $v= 90^\circ - |\text{the Sun’s Declination}|$;
In Mathematica, we use the built-in AstronomicalData package to compute the Sun’s declination.

```math
the Sun’s declination = AstronomicalData["Sun", {"Declination", ObservationDate}, ObserverLatitude]
```

We put a vertical gnomon at the center of horizon and we make the tip of the vertical gnomon the origin and the vertex of the two cones. The gnomon’s shadow is the conic section obtained as the intersection of the lower nappe of the cone and the horizon plane. Since on the one hand the Earth is supposed to be a small dot in the middle of the sphere, while at the same time we are supposed to be able to see the gnomon and the path of the shadow, we cheat by positioning the tip of the vertical gnomon at $(0, 0, 0)$. As a result, the base of the gnomon is located at $(0, 0, 0)$ minus the height of the gnomon, and the horizon plane becomes $z$ - height of the gnomon.

Therefore, given the observer’s position and date as the input variables, the model is able to give the interaction of the Sun’s path and the gnomon’s shadow path. The graphics can be dynamically rotated and zoomed to get a better view (Figure 3.2.b).

---

**Figure 3.2.a. Sun’s Path and Shadow Path, 74.5° N, 11 August (Mathematica)**
The control panel can be used to manipulate the input variables such as the height of the gnomon, the Sun’s declination, and the latitude of the observer and date. For example, without changing the code, we can get the Sun’s path and the gnomon’s shadow path for an observer standing at Latitude 35° N on 11 March by simply changing the relevant variables on the control panel (Figure 3.2.c). We also have the option of using the Sun’s declination instead of the date.
4 The Hezekiah Phenomenon

First, we would like to explain where the name Hezekiah Phenomenon comes from. In the Bible there is a story about God making the shadow of the sundial move backward as a sign for King Hezekiah.

The Bible gives two versions of the story of King Hezekiah and the sundial. First in 2 Kings, Chapter 20.

8 And Hezekiah said unto Isaiah, What [shall be] the sign that the LORD will heal me, and that I shall go up into the house of the LORD the third day? 9 And Isaiah said, This sign shalt thou have of the LORD, that the LORD will do the thing that he hath spoken: shall the shadow go forward ten degrees, or go back ten degrees? 10 And Hezekiah answered, It is a light thing for the shadow to go down ten degrees: nay, but let the shadow return backward ten degrees. 11 And Isaiah the prophet cried unto the LORD: and he brought the shadow ten degrees backward, by which it had gone down in the dial of Ahaz. (2 Kings 20: 8--11, King James Version)

The story is then repeated in Isaiah 38.

8 Behold, I will bring again the shadow of the degrees, which is gone down in the sun dial of Ahaz, ten degrees backward. So the sun returned ten degrees, by which degrees it was gone down. (Isaiah 38: 8, King James Version)

The Bible considers this a miracle, and few people seem to be aware that in the tropics, this is normal behavior for part of the year. We will refer to this as the Hezekiah Phenomenon, and one of the goals of this project is to help visualize and explain this.

Now the next step is to add more elements and data into the simple model that we built in Figure 3.2, in order to illustrate the Hezekiah Phenomenon.

Figure 4.1 is from the previous honors thesis by S. Lee, X.J. Sim, and Y. Yang (2006). It exhibits the changes of the Sun’s azimuth, and the turning points of the Hezekiah Phenomenon (T1 and T2). Again, the graphics is manually drawn, which might be inaccurate. Moreover, we are not able to manipulate the latitude/time of the observer to show the interaction of the Sun’s path and the gnomon’s shadow path dynamically.
Figure 4.1 The Hezekiah Phenomenon (S. Lee, X.J. Sim, and Y. Yang, 2006)

Figure 4.2 illustrates the model that we built in Mathematica. Given the position of the observer and date, it shows the Sun’s path, the gnomon’s shadow path, and the turning points of T1 and T2 dynamically. The control panel functions as a simple user interface to manipulate the input variables (the position of observer, the date, the height of the gnomon) to view the results.

Figure 4.2 The Hezekiah Phenomenon (Mathematica)
Besides, we can use Mathematica to model an inside view of the Hezekiah Phenomenon. Figure 4.3 illustrates the model that we built. The observer is standing on the ground and looking to the East. When the Sun rises to position T1, the Sun’s path becomes tangent to one of the vertical circles (represented by the green lines in the graph) of the Sun. A vertical circle is defined as a great circle joining the zenith to the nadir of an observer. Hence the azimuth had reached its easternmost position at T1, and T1 is the turning point of the Sun in the morning. This graph enables us to have an idea what the Sun changing direction in the morning looks like.

Similarly, Figure 4.4 also shows an inside view of the Hezekiah Phenomenon. This time the observer is looking at the point where the Sun’s path crosses the meridian. In this picture we see how the azimuth is changing in the morning, and it looks like the Sun will pass into the southern part of the Sky. But because of the steepness of the Sun’s path, it crosses the meridian in the North. That is why we get turning points at T1 and T2, and between these the azimuth is decreasing (with a jump discontinuity when it crosses the meridian).
Figure 4.4 The Hezekiah Phenomenon (Mathematica)

![Mathematica interface](image)

Figure 4.4 illustrates how the turning point T2 looks like when looking down upon at the shadow of the gnomon. The blue line, which connects the base of the gnomon and the point T2, is tangent to the shadow path. This again proves that T2 is the turning point.

Figure 4.5 The Hezekiah Phenomenon on the Shadow of the Gnomon (Mathematica)
5 Finding the Best Location and Time to Observe the Hezekiah Phenomenon

The Hezekiah Phenomenon is only observable within the tropics in the period between the two zenith passages that includes the summer solstice (S. Lee, X.J. Sim, and Y. Yang, 2006). We will refer to this as the Hezekiah Period. However, even in this period, not every position or every day within the range allows a clear observation of the phenomenon.

1) The graphs below illustrate the Sun’s azimuth change during the morning time at different latitudes on the summer solstice (23 Jun).

2) The graphs below illustrate the Sun’s azimuth change during the morning time (i.e. from sunrise to noon), on different dates at the same latitude (22° N).

3) The graphs below illustrate the Sun’s azimuth change during the morning time (i.e. from sunrise to noon), on different dates at the same latitude (13° N)
The graphs below illustrate the Sun’s azimuth change during the morning time (i.e. from sunrise to noon), on different dates at the same latitude (0°, Equator).

From the graphs above, we can conclude that the Hezekiah Phenomenon is more noticeable at latitudes in the middle of the Equator and the Tropic of Cancer, e.g. around 12°, and in the periods that are in the middle part between the summer solstice and one of the zenith passages.

Also, we can see that during the Hezekiah Period, the limit of the azimuth as time approaches the meridian passage is 0°. Outside of the Hezekiah Period, the limit is 180°.

Figure 5.1 shows an example for a good location and time to observe the Hezekiah Phenomenon, which is at 18° N on 30 Jun. It illustrates the Sun’s azimuth change from sunrise (around 5:20 AM) to noon (12 PM), and from noon to sunset (7:00 PM). In the morning, there is an increase of the azimuth from sunrise to 9:30AM. From 9:30AM to noon, the azimuth decreases. Therefore, 9:30AM is the turning point T1. From sunrise to T1, there are about 4 hours while from T1 to noon, there are about 2.5 hours; so it provides sufficient time for the observation. However, in some cases, if T1 happens soon after
sunrise or shortly before noon, we will not be able to observe the Hezekiah Phenomenon because it will be difficult to see the increase and decrease of the azimuth.

Figure 5.1 Plot of the Sun’s Azimuth at a Tropical Location (18° N) during Critical Periods (30 Jun)
6 Animation of the Sun’s Path and the Shadow Path

In the last part of the project, we added "time" into the control panel. Thus given a time (e.g. 8:30AM), a date as well as a latitude, the position of the Sun and the shadow of the sundial will be plotted on the graph. Also, the animation of the Sun’s path and the shadow path can be given by expanding the control panel and clicking PLAY in the “time” row. However, Mathematica takes a long time to calculate and run the program, and the speed of the animation is not ideal.

Figure 6.1 shows the animation of the Sun’s path and the shadow path. The red dot represents the position of the Sun at 9:00AM 11 Aug at 13° N, and the blue dot represents the position of the shadow of the gnomon.

![Figure 6.1 Animation of the Sun’s Path and the Shadow Path (Mathematica)](image)

Besides, by changing the view angle, we can look at the shadow point and shadow path as a 2D graph. Similarly, the animation of the shadow path can be shown on the graph.

Figure 6.2 shows the animation of the shadow path. The red dot represents the position of the shadow of the gnomon at 6:42AM 11 Aug at 35° N.
Figure 6.2 Animation of the Shadow Path (Mathematica)
Glossary

Altitude: angular distance measured positive upward from the horizon to the star along the star’s vertical circle. (DC in Figure 1.1)

Azimuth: angular distance measured from the north point to the base of the star’s vertical circle along the horizon in an easterly direction. (NOD in Figure 1.1)

Celestial equator: the projection of the Earth’s equator onto the celestial sphere. (Blue circle in Figure 1.2) This great circle will be 90° distant from the celestial poles at all points. Just as the horizon serves as the fundamental reference circle in the horizon system, so, too, is the celestial equator the fundamental reference circle in the equatorial system.

Celestial sphere: an imaginary sphere of arbitrary extent centered about an observer located at some position on the Earth where all the objects in sky can be thought of as lying upon the sphere. Projected, from their corresponding geographic equivalents, are the celestial equator and the celestial poles.

Equinoxes: the two points on the celestial sphere at which the Sun crosses the celestial equator. That is, the two intersection points of the two great circles, the ecliptic and the celestial equator. The spring equinox (Labeled in Figure 1.2) is one of the points of intersection where the Sun passes from the southern to the northern hemisphere. The fall equinox is the other point of intersection opposite to spring equinox, where the Sun passes from north to south. The equinox dates are March 21 and September 22 approximately.

Fall equinox: See Equinoxes.

Gnomon: a scientific instrument that can be used for finding the declination of the Sun through the year and to study the path of the Sun among other things. It is one of the first scientific instruments ever made, originating with the Chaldean astronomers of Babylon and from there brought to the Greek world. It is still very instructive, an excellent example of random and systematic errors of measurement, vividly demonstrating astronomical principles.

Horizon (of an observer): the horizontal plane through the eyes of the observer and is perpendicular to the observer.

Inclination: the angular distance of the plane of the orbit of the planet and the ecliptic, normally stated in degrees.

Latitude: the position north/south of the Equator measured from 0° to 90°.

Meridian: great circle on the celestial sphere that passes through the zenith and north point of horizon, through the celestial pole, up to the zenith, and through the south point on the
horizon and is perpendicular to the local horizon. Thus the azimuth at meridian in the North is 0°.

*Midnight Sun*: a phenomenon occurring in latitudes north of the Arctic Circle and south of the Antarctic Circle where the Sun is continuously visible for 24 hours.

*Nadir*: the point directly opposite the zenith.

*North celestial pole*: See *Celestial poles*.

*Opening angle of a right cone*: the angle made by a cross section through the vertex and center of the base of the cone.

*Right ascension*: angular distance measured from the spring equinox to the star’s hour circle along the celestial equator in an easterly direction (spring equinox to D in Figure 1.2) because this direction lies along the Equator and we measure the passage of time by rotation of the Earth, the right ascension coordinate is usually expressed in hours and minutes of time instead of angular measure. The celestial equator is divided into twenty-four hours instead of 360°; therefore, one hour of time measure is equivalent to 15° of angle measure.

*Solstices*: See *Summer solstice* and *Winter solstice*.

*Spring equinox*: See *Equinoxes*.

*Summer solstice*: point on the celestial sphere where the Sun reaches its greatest distance north of the celestial equator, about June 21.

*True anomaly*: the angle between the vectors Sun-Planet and Sun-Perihelion. *True longitude*: is the longitude at which an orbiting body could actually be found if its inclination were zero.

*Vertical circles*: great circles passing through the zenith and nadir, perpendicular to the horizon. ZS, ZD, and ZW each depict quarters of vertical circles in Figure 1.1.

*Winter solstice*: point on the celestial sphere where the Sun reaches its greatest distance south of the celestial equator, about December 21.

*Zenith*: Z, of the observer, O, the point directly overhead (opposite the direction of a plumb line) in Figure 1.1.

*Zenith passage*: The day where the Sun passes the zenith.
Appendix 1  the Hezekiah Phenomenon

(* Function to find the time of T1 (T2 is symmetric to T1 in the afternoon)
We cannot set TimeZone globally, so since Mathematica reads $TimeZone from our computer in Singapore at
longitude 104 and TimeZone 8, we set longitude manually to 120. *)

findT1T2Time[Latitude0_, month_, date_] :=
  (specificdate = DateList[{2010, month, date}];
   location = {Latitude0, 120};
   sunRiseTime = AstronomicalData["Sun", {"NextRiseTime", specificdate, location}];
   sunSetTime = AstronomicalData["Sun", {"NextSetTime", specificdate, location}];
   SunRiseTimeInMins = DateDifference[specificdate, sunRiseTime, "Minute"];
   SunSetTimeInMins = DateDifference[specificdate, sunSetTime, "Minute"];
   MidDayInMins = (SunSetTimeInMins[[1]] + SunRiseTimeInMins[[1]])/2;
   MorningData =
     Table[{DatePlus[specificdate, {i, "Minute"}], AstronomicalData["Sun", {"Azimuth", DatePlus[specificdate, {i, "Minute"}],
       location}]}, {i, SunRiseTimeInMins[[1]], MidDayInMins, 7}];
   dec1 = MorningData[[All, 1]];
   dec2 = MorningData[[All, 2]];
   max0 = Max[dec2];
   pos = Position[dec2, max0];
   t1 = Extract[dec1, pos][[1]]
  )

(* Function to find the dates for the two zenith passages. Code is not optimal. *)
findZenithPassageDate1 [Latitude0_] :=
  (dec = Table[{DateList[{2010, 1, i}], AstronomicalData["Sun", {"Declination", DateList[{2010, 1, i}]}]}, {i, 78, 173}];
   (*Define dec1 to be the first row of the dec table/date list*)
   dec1 = dec[[All, 1]];
   (*Define dec2 to be the second row of the dec table/Declination list*)
   dec2 = dec[[All, 2]];
   (*Define dec3 to be the difference between declination and latitude*)
   dec3 = Abs[dec[[All, 2]] - Latitude0];
   (*Find the min of errors*)
   min0 = Min[dec3];
   (*Find the position of the specific declination with min difference in the dec table*)
   pos = Position[dec3, min0];
   (*Find the date of the specific declination*)
   Extract[dec1, pos][[1]]
  )

findZenithPassageDate2 [Latitude0_] := (
dec = Table[{DateList[{2010, 1, i}], AstronomicalData["Sun", {"Declination", DateList[{2010, 1, i}]}]}, {i, 173, 263}]; (*Define dec1 to be the first row of the dec table/date list*)
dec1 = dec[[All, 1]]; (*Define dec2 to be the second row of the dec table/Declination list*)
dec2 = dec[[All, 2]]; (*Define dec3 to be the difference between declination and latitude*)
dec3 = Abs[dec[[All, 2]] - Latitude0]; (*Find the min of errors*)
min0 = Min[dec3]; (*Find the position of the specific declination with min difference in the dec table*)
pos = Position[dec3, min0]; (*Find the date of the specific declination*)
Extract[dec1, pos][[1]]

(* Draw the sphere. *)
itemSphere[LatitudeInput_, m_, d_, b_] := Show[
  Graphics3D[{GrayLevel[.3], PointSize[.02], (* find the 2 zenith passage dates using the functions findT1T2 which has been written at the beggining. *)
    ZenithPassageDate1 = findZenithPassageDate1[LatitudeInput];
    ZenithPassageDate2 = findZenithPassageDate2[LatitudeInput];
    (* Draw point T1 and T2 using the functions findT1T2 which has been written at the beginning. *)
    If [LatitudeInput <= 23.5 && DateDifference[DateList[{2010, m, d}], ZenithPassageDate2] >= 0 && DateDifference[DateList[{2010, m, d}], ZenithPassageDate1] <= 0,
      Point[t1Coordinate], Point[{0, 0, 0}]],
    If [LatitudeInput <= 23.5 && DateDifference[DateList[{2010, m, d}], ZenithPassageDate2] >= 0 && DateDifference[DateList[{2010, m, d}], ZenithPassageDate1] <= 0,
      Text[Style["T1", Medium, Bold, Red], t1Coordinate + {0, 0, 0.05}], Point[{0, 0, 0}]],
    If [LatitudeInput <= 23.5 && DateDifference[DateList[{2010, m, d}], ZenithPassageDate2] >= 0 && DateDifference[DateList[{2010, m, d}], ZenithPassageDate1] <= 0,
      Text[Style["T2", Medium, Bold, Red], t2Coordinate + {0, 0, 0.05}], Point[{0, 0, 0}]],
    Point[{0, 0, 0}],
    RGBColor[.5, 0, 0], PointSize[.02], LightBlue, Opacity[.5], Sphere[{0, 0, 0}, 1]]
  (* Mark S, N, W, E on the graph *)}
(* Draw the Sphere without T1 or T2. *)

itemSphere2[b_, x_, y_, z_, t_] :=
Show[
  Graphics3D[{GrayLevel[.3], PointSize[.02],
     If[(60 t - sunRiseTime) > 0 && (sunSetTime - 60 t) > 0,
       Point[{{x, y, z}, {b x/z, b y/z, b}}, VertexColors -> {Red, Blue}],
       Point[{{0, 0, 0}}],
       RGBColor[.5, 0, 0], PointSize[.02],
       LightBlue, Opacity[.5], Sphere[{0, 0, 0}, 1],
       Text[Style["S", Medium, Bold, Black], {0, 1.05, b}],
       Text[Style["N", Medium, Bold, Black], {0, -1.05, b}],
       Text[Style["W", Medium, Bold, Black], {1.05, 0, b}],
       Text[Style["E", Medium, Bold, Black], {-1.05, 0, b}]
     ]},
  (* Point the Sun given specific month, date, latitude, time*)
  Point[{0, 0, 0}], RGBColor[.5, 0, 0], PointSize[.02],
  LightBlue, Opacity[.5], Sphere[{0, 0, 0}, 1],
  Text[Style["S", Medium, Bold, Black], {0, 1.05, b}],
  Text[Style["N", Medium, Bold, Black], {0, -1.05, b}],
  Text[Style["W", Medium, Bold, Black], {1.05, 0, b}],
  Text[Style["E", Medium, Bold, Black], {-1.05, 0, b}]
]]

(* Draw the Celestial Equator. *)

itemCE :=
Show[
  ParametricPlot3D[{Sin[t], Cos[t], 0}.RotationMatrix[90 Degree - LatitudeRad, {-1, 0, 0}], {t, 0, 2 Pi},
    Exclusions -> Range[0, Pi], MaxRecursion -> 0, PlotStyle -> {Thin, Dashed, ColorData[1, 2]}
  ]
]

(* Draw the Sun Path: draw the first half, and then draw the second half which is symmetric with the first half. *)

itemSunPath[v_] :=
Show[
  sunpath = If[Cos[v] < Sin[90 Degree - LatitudeRad],
    ParametricPlot3D[{Sin[v] Sin[t], Sin[v] Cos[t], Cos[v]}.RotationMatrix[90 Degree - LatitudeRad, {-1, 0, 0}],
      {t, 0, Pi/2 + ArcSin[1/Tan[v] 1/Tan[90 Degree - LatitudeRad]]},
      MaxRecursion -> 0, PlotStyle -> {Thick, ColorData[1, 1]}],
    ParametricPlot3D[{Sin[v] Sin[t], Sin[v] Cos[t], Cos[v]}.RotationMatrix[90 Degree - LatitudeRad, {-1, 0, 0}],
      {t, 0, Pi}, MaxRecursion -> 0, PlotStyle -> {Thick, ColorData[1, 1]}
    ]],
  sunpath = If[Cos[v] < Sin[90 Degree - LatitudeRad],
    ParametricPlot3D[{Sin[v] Sin[t], Sin[v] Cos[t], Cos[v]}.RotationMatrix[90 Degree - LatitudeRad, {-1, 0, 0}],
      {t, 3 Pi/2 - ArcSin[1/Tan[v] 1/Tan[90 Degree - LatitudeRad]], 2 Pi},
      MaxRecursion -> 0, PlotStyle -> {Thick, ColorData[1, 1]}
    ]]
ParametricPlot3D[{Sin[v] Sin[t], Sin[v] Cos[t], Cos[v]}, RotationMatrix[90 Degree - LatitudeRad, {-1, 0, 0}], {t, Pi, 2 Pi}, MaxRecursion -> 0, PlotStyle -> {Thick, ColorData[1, 1]}]

(* Draw the cone for the Sun's path (Primary Cone). *)
itemPC[v_] :=
  Show[
    ParametricPlot3D[{z Cos[q] Sin[v], z Sin[q] Sin[v],
                     z Cos[v]}, RotationMatrix[Pi/2 - LatitudeRad, {-1, 0, 0}], {q, 0, 2 Pi}, {z, 0, 1},
                     Boxed -> False, Axes -> False,
                     (* Draw the sun path *)
                     Mesh -> {{0.0}},
                     MeshStyle -> AbsoluteThickness[4],
                     PlotStyle -> Opacity[0.4]]
  ]

(* Draw the Equinoctial Shadow Line. *)
itemESL[b_] :=
  Show[
    ParametricPlot3D[{z Cos[q], z Sin[q], 0}, RotationMatrix[Pi/2 - LatitudeRad, {-1, 0, 0}], {q, 0, 2 Pi}, {z, -1, 0},
                     Boxed -> False, Axes -> False,
                     (* Draw the shadow *)
                     MeshFunctions -> {Function[{x, y, z, q, zz}, (b - z)]},
                     MeshStyle -> AbsoluteThickness[4],
                     PlotStyle -> Opacity[0.4]]
  ]

(* Draw the gnomon with length d, which can be manipulated in the control panel. *)
itemGnomon[b_] :=
  Show[
    Table[Graphics3D[{Thickness[i], Line[{{0, 0, b}, {0, 0, 0}}]}], {i, {0.01}}]]
(* Draw the plane through the base of the gnomon. *)

```math
itemPlane[b_] :=
  Show[
    ParametricPlot3D[{p Sqrt[1 - b^2] Sin[c], p Sqrt[1 - b^2] Cos[c], b}, {p, 0, 1}, {c, 0, 2 Pi}, Boxed -> False, Axes -> False, Mesh -> None, PlotStyle -> Opacity[0.8]]
  ]
```

(* Draw the NS Line and WE Line. *)

```math
itemNSWELine[b_] :=
  Show[
    Table[Graphics3D[{Thickness[i], Line[{{0, -Sqrt[1 - b^2], b}, {0, Sqrt[1 - b^2], b}}]}], {i, {0.002}}],
    Table[Graphics3D[{Thickness[i], Line[{{-Sqrt[1 - b^2], 0, b}, {Sqrt[1 - b^2], 0, b}}]}], {i, {0.002}}]
  ]
```

(* Hezekiah Phenomenon with Automatic T1 and T2 *)

```math
Manipulate[
  Show[
    LatitudeRad = LatitudeInput/180*Pi;
    (* Use Month and Date to calculate the declination of the sun and then v = 90 - absolute value of declination. *)
    v = If [UseMonthDate == "True",
      2 Pi (90 - Abs[AstronomicalData["Sun", 
        "Declination", DateList[{2010, m, d}]}])/360, (0.5 Pi - v0)];
    T1T2Time =
      If [LatitudeInput <= 23.5, findT1T2Time[LatitudeInput, m, d], 0];
    altitude0 = AstronomicalData["Sun", 
      {"Altitude", T1T2Time, location}];
    azimuth0 = AstronomicalData["Sun", 
      {"Azimuth", T1T2Time, location}];
    
    (* Calculate coordinates of T1 and T2 using altitude0 and azimuth0. *)
    t1Coordinate = {Cos[Pi (270 - azimuth0)/180] Sin[Pi (90 - altitude0)/180],
                   Sin[Pi (270 - azimuth0)/180] Sin[Pi (90 - altitude0)/180],
                   Cos[Pi (90 - altitude0)/180]};
    t2Coordinate = {-Cos[Pi (270 - azimuth0)/180] Sin[Pi (90 - altitude0)/180],
                   -Sin[Pi (270 - azimuth0)/180] Sin[Pi (90 - altitude0)/180],
                   Cos[Pi (90 - altitude0)/180]};
    
    (* Draw the Sphere. *)
    itemSphere[LatitudeInput, m, d, b],
    (* Draw the Celestial Equator. *)
    itemCE,
    (* Draw the Sun Path: draw the first half, and then draw the second half which is symmetric with the first half *)
    itemSunPath[v],
    (* Draw the cone of the Sun's path (Primary Cone), and the shadow path (Secondary Cone) *)
    itemPC[v],
    itemSC[v, b],
  ]
```
(* Draw the Equinoctial Shadow Line *)
itemESL[b],

(* Draw the gnomon and the plane*)
itemGnomon[b],
itemPlane[b],
itemNSWELine[b],

(* Size, ViewPoint, and ViewAngle of the graph *)
ImageSize -> {480, 360},
Boxed -> False,
ViewAngle -> Pi/10,
ViewPoint -> {Pi, -Pi/3, 1}
],

(* Control Panel *)
"Horizontal Plane:",
{{b, -0.2, "Height"}, -0.4, 0, 0.05, ImageSize -> Tiny,
Appearance -> "Labeled"},
"The red dot circle is the Celestial Equator.",
Delimiter,
(* Date in the year*)
{UseMonthDate, {"True", "False"}},
{{v0, 23.5 Degree, "Declination"}, 0 Degree, 23.5 Degree, 1 Degree,
Appearance -> "Labeled"},
{{m, 6, "Month"}, 1, 12, 1, Appearance -> "Labeled"},
{{d, 30, "Date"}, 1, 31, 1, Appearance -> "Labeled"},
Delimiter,
(* Manipulate Latitude: This graph only shows the Sun's path and the shadow path in the North Hemisphere. *)
"(0 is at the equator, and 90 is at the North Pole.",
{{LatitudeInput, 18, "Latitude"}, 0, 90, 0.5, Appearance -> "Labeled"}]

(*----------------------------------Hezekiah Phenomenon: Zoom in at T1 ----------------------*)
Manipulate[
Show[
LatitudeRad = LatitudeInput/180*Pi;
(* Use Month and Date to calculate the declination of the sun and then \v = 90 - absolute value of declination. *)

v = If [UseMonthDate == "True", 2 Pi (90 - Abs [
    AstronomicalData["Sun", {"Declination", DateList[{2010, m, d}]}])/360, (0.5 Pi - v0)];
T1T2Time =
If \[\text{LatitudeInput} \leq 23.5\], find \(T1T2Time[\text{LatitudeInput}, m, d, 0]\);

\[
\text{altitude0} = \text{AstronomicalData["Sun", \{"Altitude", T1T2Time, location\}];}
\]

\[
\text{azimuth0} = \text{AstronomicalData["Sun", \{"Azimuth", T1T2Time, location\}];}
\]

(* Calculate coordinates of T1 and T2 using \text{altitude0} and \text{azimuth0}.*

\[
\text{t1Coordinate} = \{\cos[\pi (270 - \text{azimuth0})/180],
\sin[\pi (90 - \text{altitude0})/180],
\cos[\pi (90 - \text{altitude0})/180]\};
\]

\[
\text{t2Coordinate} = \{-\cos[\pi (270 - \text{azimuth0})/180],
\sin[\pi (90 - \text{altitude0})/180],
\cos[\pi (90 - \text{altitude0})/180]\};
\]

(* Draw the Sphere. *)

\itemSphere[\text{LatitudeInput}, m, d, b],

(* Draw the Sun Path: draw the first half, and then draw the second half which is symmetric with the first half *)

\itemSunPath[v],

(* Draw the cone of sun path (Primary Cone), and the shawdow path (Secondary Cone) *)

\itemPC[v],

(* Draw the gnomon and the plane*)

(*Draw a vertical circle, which can be tangent to T1.*)

\text{ParametricPlot3D[\{\cos[\theta] \sin[t], \sin[\theta] \sin[t], \cos[t]\}, \{t, 0, 2 \pi\}, \text{PlotStyle} \rightarrow \{\text{Thin, ColorData[1, 4]}}\],

\text{Table[ ParametricPlot3D[\{\cos[i] \sin[t], \sin[i] \sin[t], \cos[t]\}, \{t, 0, 2 \pi\}, \text{PlotStyle} \rightarrow \{\text{Thin, ColorData[1, 4]}\}, \{i, 0, 11 \pi/6, \pi/6 \} ]},

\itemGnomon[b],

\itemPlane[b],

\itemNSWELine[b],

(* Change the size, ViewPoint, ViewCenter and ViewAngle of the graph.*)

\text{Boxed} \rightarrow \text{False},

(* Set ViewCenter *)

\text{ViewCenter} \rightarrow \{\text{t1Coordinate[1]}, \text{t1Coordinate[2]}, 2 \text{t1Coordinate[3]}\},

\text{ViewAngle} \rightarrow \pi/2,

\text{ViewPoint} \rightarrow \{0, 0, 0\}

].

(* Control Panel *)

"Horizontal Plane:",

\{\{b, -0.005, "Height"\}, -0.4, 0, 0.005, \text{ImageSize} \rightarrow \text{Tiny}, \text{Appearance} \rightarrow \"Labeled\"},

"The red dot circle is the Celestial Equator.",

\text{Delimiter},

(* Date in the year*)

\{\text{UseMonthDate}, \{"True", "False"\} \},
Manipulate[
    Show[
        LatitudeRad = LatitudeInput/180*Pi;
        (* Use Month and Date to calculate the declination of the sun and then v = 90 - absolute value of declination.*)
        v = If [UseMonthDate == "True", 2 Pi (90 - Abs [AstronomicalData["Sun", {"Declination", DateList[{2010, m, d}]})]/360, (0.5 Pi - v0)];

        (* Calculate the coordinate where the Sun's path crosses the Meridian.*)
        crossCoordinate = {0, -Cos[v + LatitudeRad], Sin[v + LatitudeRad]};
        T1T2Time = If [LatitudeInput <= 23.5, findT1T2Time[LatitudeInput, m, d], 0];
        altitude0 = AstronomicalData["Sun", {"Altitude", T1T2Time, location}];
        azimuth0 = AstronomicalData["Sun", {"Azimuth", T1T2Time, location}];

        (* Calculate coordinates of T1 and T2 using altitude0 and azimuth0. *)
        t1Coordinate = {Cos[Pi (270 - azimuth0)/180] Sin[Pi (90 - altitude0)/180],
            Sin[Pi (270 - azimuth0)/180] Sin[Pi (90 - altitude0)/180],
            Cos[Pi (90 - altitude0)/180]};
        t2Coordinate = {-Cos[Pi (270 - azimuth0)/180] Sin[Pi (90 - altitude0)/180],
            Sin[Pi (270 - azimuth0)/180] Sin[Pi (90 - altitude0)/180],
            Cos[Pi (90 - altitude0)/180]};

        (* Draw the Sphere.*)
        itemSphere[LatitudeInput, m, d, b],
        (* Draw the Sun Path: draw the first half, and then draw the second half which is symmetric with the first half *)
        itemSunPath[v],
        (* Draw the cone of sun path (Primary Cone), and the shawdow path (Secondary Cone) *)
        itemPC[v],
        (*Draw 2 vertical circles, which can be tangent to T1 and T2*)
        ParametricPlot3D[{ Cos[theta1] Sin[t], Sin[theta1] Sin[t],
            Cos[t]}, {t, 0, 2 Pi}, PlotStyle -> {Thin,
azimuth0 = AstronomicalData["Sun", {"Azimuth", {2010, m, d, t}, location}];
altitude0 = AstronomicalData["Sun", {"Altitude", {2010, m, d, t}, location}];
x = Cos[Pi (270 - azimuth0)/180] Sin[Pi (90 - altitude0)/180];
y = Sin[Pi (270 - azimuth0)/180] Sin[Pi (90 - altitude0)/180];
z = Cos[Pi (90 - altitude0)/180];
sunRiseTime = AstronomicalData["Sun", {"NextRiseTime", {2010, m, d}, location}][[4]]*60 + AstronomicalData["Sun", {"NextRiseTime", {2010, m, d}, location}][[5]];
sunSetTime = AstronomicalData["Sun", {"NextSetTime", {2010, m, d}, location}][[4]]*60 + AstronomicalData["Sun", {"NextSetTime", {2010, m, d}, location}][[5]];

(* Draw the Sphere without T1 or T2. *)
itemSphere2[b, x, y, z, t],
(* Draw the Celestial Equator. *)
itemCE,
(* Draw the Sun Path: draw the first half, and then draw the second half which is symmetric with the first half *)
itemSunPath[v],
(* Draw the cone of sun path (Primary Cone) *)
itemPC[v],
(* Draw the Equinoctial Shadow Line *)
itemESL[b],
(* Find the intersection of the plane and the line *)
Table[Graphics3D[{{Thickness[i],
   Line[{{x, y, z}, {-x, -y, -z}}]}}, {i, {0.002}}],
If [{60 t - sunRiseTime} > 0 && (sunSetTime - 60 t) > 0,
Table[Graphics3D[{{Thickness[i],
   Line[{{0, 0, b}, {b x/z, b y/z, b}}]}}, {i, {0.005}}],
Table[Graphics3D[{{Thickness[i],
   Line[{{0, 0, 0}, {0, 0, 0}}]}}, {i, {0.001}}]]],
(* Draw the gnomon and the plane *)
itemGnomon[b],
itemPlane[b],
itemNSWELine[b],
PlotRange -> {{-1.1, 1.1}, {-1.1, 1.1}, {-1.1, 1.1}},
ImageSize -> {480, 360}, Boxed -> False,
SphericalRegion -> True,
(* Change the ViewPoint and ViewAngle of the graph. *)
ViewAngle -> Pi/10,
ViewPoint -> {Pi, -Pi/3, 1}].
"Horizontal Plane:",
{b, -0.25, "Height"}, -0.4, 0, 0.05, ImageSize -> Tiny, Appearance -> "Labeled"},
"The red dot circle is the Celestial Equator.",
Delimiter,

(* Date in the year*)
{UseMonthDate, {"True", "False"}},
{{v0, 23.5 Degree, "Declination"}, 0 Degree, 23.5 Degree, 1 Degree, Appearance -> "Labeled"},
{{m, 8, "Month"}, 1, 12, 1, Appearance -> "Labeled"},
{{d, 11, "Date"}, 1, 31, 1, Appearance -> "Labeled"},
{{t, 9, "Time"}, 0, 24, 0.1, Appearance -> "Labeled"},
Delimiter,
"(0 is at the equator, and 90 is at the North Pole.)",
{{LatitudeInput, 13, "Latitude"}, 0, 90, 0.5, Appearance -> "Labeled"}
]

(*------------------------------------------- Shadow Path Animation 2D version -------------------------------------------*)
Manipulate[
  Show[
    LatitudeRad = LatitudeInput/180*Pi;
    (* Use Month and Date to calculate the declination of the sun and then v = 90 - absolute value of declination. *)
    v = If [UseMonthDate == "True", 2 Pi (90 - Abs [AstronomicalData["Sun", {"Declination", DateList[{2010, m, d}]}])/360, (0.5 Pi - v0)];
    location = {LatitudeInput, 120};
    azimuth0 = AstronomicalData["Sun", {"Azimuth", {2010, m, d, t}, location}];
    altitude0 = AstronomicalData["Sun", {"Altitude", {2010, m, d, t}, location}];
    x = Cos[Pi (270 - azimuth0)/180] Sin[Pi (90 - altitude0)/180];
    y = Sin[Pi (270 - azimuth0)/180] Sin[Pi (90 - altitude0)/180];
    z = Cos[Pi (90 - altitude0)/180];
    sunRiseTime = AstronomicalData["Sun", {"NextRiseTime", {2010, m, d}, location}][[4]]*60 + AstronomicalData["Sun", {"NextRiseTime", {2010, m, d}, location}][[5]]; 
    sunSetTime = AstronomicalData["Sun", {"NextSetTime", {2010, m, d}, location}][[4]]*60 + AstronomicalData["Sun", {"NextSetTime", {2010, m, d}, location}][[5]]; 

    (* Draw the Sphere without T1 or T2. *)
    itemSphere2[b, x, y, z, t],
    (* Draw the shawdow path (Secondary Cone) *)
    itemSC[v, b],

    (* find the intersection of the plane and the line *)
]
Table[Graphics3D[{Thickness[i], Line[{{x, y, z}, {-x, -y, -z}}]}, {i, {0.002}}],
If ([60 t - sunRiseTime] > 0 && (sunSetTime - 60 t) > 0,
Table[Graphics3D[{Thickness[i], Line[{{0, 0, b}, {b x/z, b y/z, b}}]}, {i, {0.005}}],
Table[Graphics3D[{Thickness[i], Line[{{0, 0, 0}, {0, 0, 0}}]}, {i, {0.001}}]],
(* Draw the gnomon and the plane *)
itemGnomon[b],
itemPlane[b],
itemNSWELine[b],
PlotRange -> {{-100, 100}, {-100, 100}, {-100, 100}},
Boxed -> False,
ViewPoint -> {0, 0, -b}]

"Horizontal Plane:",
{b, -0.02, "Height"}, -0.1, 0, 0.01, ImageSize -> Tiny,
Appearance -> "Labeled",
Delimiter,
(* Date in the year*)
{UseMonthDate, {"True", "False"}},
{v0, 23.5 Degree, "Declination"}, 0 Degree, 23.5 Degree, 1 Degree, Appearance -> "Labeled"},
{m, 8, "Month"}, 1, 12, 1, Appearance -> "Labeled"},
{d, 11, "Date"}, 1, 31, 1, Appearance -> "Labeled"},
{t, 9, "Time"}, 0, 24, 0.1, Appearance -> "Labeled"},
Delimiter,
"(0 is at the equator, and 90 is at the North Pole.)",
{LatitudeInput, 35, "Latitude"}, 0, 90, 0.5, Appearance -> "Labeled"}]
Appendix 2  Azimuth and Altitude Plots

Manipulate[
  Show[
  specificdate = DateList[{2010, m, d}];
  location = {latitude, 120};
  v = 90 - Abs[
    AstronomicalData["Sun", {"Declination", DateList[{2010, m, d}]}]];
  sunRiseTime = If [latitude <= (v - 1),
    AstronomicalData["Sun", {"NextRiseTime", specificdate, location}], 0];
  sunSetTime = If [latitude <= (v - 1),
    AstronomicalData["Sun", {"NextSetTime", specificdate, location}], 1440];
  SunRiseTimeInMins = If [latitude <= (v - 1),
    DateDifference[specificdate, sunRiseTime, "Minute"], {0, Minute}];
  SunSetTimeInMins = If [latitude <= (v - 1),
    DateDifference[specificdate, sunSetTime, "Minute"], {1440, Minute}];
  MidDayInMins = (SunSetTimeInMins[[1]] + SunRiseTimeInMins[[1]])/2;
  data1 = Tooltip[
    Table[
      {DatePlus[specificdate, {i, "Minute"}],
        AstronomicalData["Sun", {"Azimuth", DatePlus[specificdate, {i, "Minute"}],
          location}]},
      {i, SunRiseTimeInMins[[1]], SunSetTimeInMins[[1]], 13}];
    DateListPlot[data1],
    "Azimuth Plot:",
    "{m, 6, "Month"}, 1, 12, 1, Appearance -> "Labeled"},
    "{d, 30, "Date"}, 1, 31, 1, Appearance -> "Labeled"},
    "{latitude, 18, "Latitude"}, 0, 90, 1, Appearance -> "Labeled"}];
data1 = Tooltip@Table[{DatePlus[specificdate, {i, "Minute"}],
   AstronomicalData["Sun", {"Altitude", DatePlus[specificdate, {i, "Minute"}],
      location}]], {i, SunRiseTimeInMins[[1]], SunSetTimeInMins[[1]], 13}];
DateListPlot[data1],
"Altitude Plot:",
{{m, 6, "Month"}, 1, 12, 1, Appearance -> "Labeled"},
{{d, 30, "Date"}, 1, 31, 1, Appearance -> "Labeled"},
{{latitude, 18, "Latitude"}, 0, 90, 1, Appearance -> "Labeled"}


Bibliography

