ptolemaicastronomy — Diagrams of sphere models for variably strict conditionals (Lewis counterfactuals)*

Richard Zach[†]

Released 2018/04/08

1 Introduction

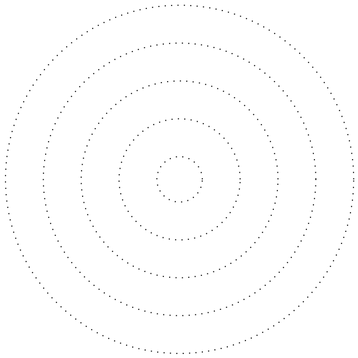
Lewis[1] introduced a sphere semantics for counterfactual conditionals. He jokingly referred to the diagrams depicting such sphere models as "Ptolemaic astronomy," hence the name of this package. It has nothing to do with Ptolemy or with astronomy, sorry.

The macros provided in this package aid in the construction of sphere model diagrams in the style of Lewis. The macros all make use of TikZ.

Source code can be found at `https://github.com/rzach/ptolemaic-astronomy`

2 Usage

`\spheresystem` To draw a sphere system with $\langle n \rangle$ layers, say `\spheresystem{`$\langle n \rangle$`}`:



```
\begin{tikzpicture}
  \spheresystem{5}
\end{tikzpicture}
```

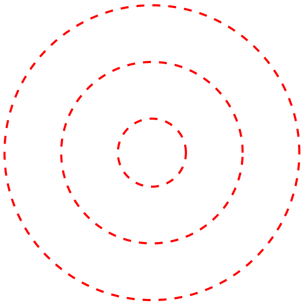
---

[*] This file describes version v1.00, last revised 2018/04/08.

[†] richardzach.org, E-mail: rzach@ucalgary.ca

[1] David K. Lewis, *Counterfactuals* (Blackwell 1973)

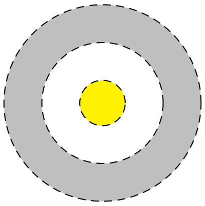
The width of each layer is determined by the TikZ parameter `layerwidth` and defaults to .5 TikZ units (so 0.5 cm by default). The radius of the center sphere is *not* `layerwidth`, but `layerwidth` × (1 − `innerfactor`). `innerfactor` defaults to 0.4. Spheres are drawn in `dotted` style by default. You can change this by passing an option to `\spheresystem`, e.g., to get red, dashed, thick lines and wider layers:



```
\begin{tikzpicture}
  \spheresystem[dashed, red,
     thick, layerwidth=.75]{3}
\end{tikzpicture}
```

`\spherelayer`
`\spherefill`

These macros shade the ⟨*n*⟩-th layer of the sphere model, or the entire ⟨*n*⟩-th sphere. The fill defaults to `lightgray` and can be changed with `[`⟨*options*⟩`]`. Note that the fill extends to the center of the layer boundary line, so you should fill first and then draw the spheres. For instance:



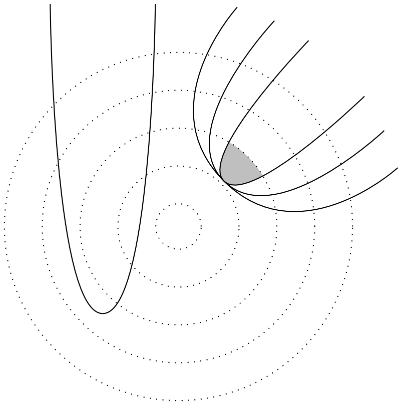
```
\begin{tikzpicture}
  \spherelayer{3}
  \spherefill[yellow]{1}
  \spheresystem[densely dashed]{3}
\end{tikzpicture}
```

`\proposition`
`\propositionintersect`

A proposition is a set of worlds which (usually) intersects with a sphere system. A common way of drawing them is as a parabola, and often we want to highlight the intersection of the proposition with the closest sphere with which it intersects. `\proposition{`⟨*direction*⟩`}{`⟨*n*⟩`}{`⟨*width*⟩`}{`⟨*length*⟩`}` will draw such a parabola. ⟨*direction*⟩ is the angle (0 is due east and 90 is due north) from which you want the proposition to reach into the sphere system. ⟨*n*⟩ is the innermost layer you want it to intersect. ⟨*width*⟩ and ⟨*length*⟩ describe the triangle with apex ⟨*width*⟩ degrees and sides of length ⟨*length*⟩. Use `\propositionintersect` to also highlight the intersection with the ⟨*n*⟩-th sphere. E.g., here are propositions that intersects the 3rd layer at 45 degrees, with a width of 20, 40, and 60 degrees, and the intersection of the first one with the innermost sphere it intersects.

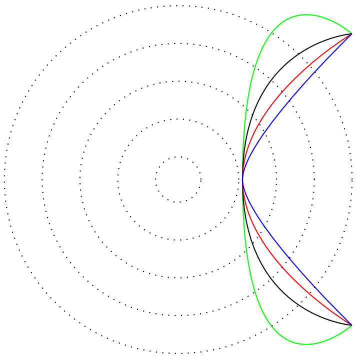
With the `shift` option you can also position propositions outside the center, e.g., a proposition extending from the north through the west side of the sphere system would use, say, `shift={(-1,-1)}`.

```
\begin{tikzpicture}
  \propositionintersect{45}{3}{20}{3}
  \proposition{45}{3}{40}{3}
  \proposition{45}{3}{60}{3}
  \proposition[shift={(-1,-1)}]{90}{1}{20}{4}
  \spheresystem{5}
\end{tikzpicture}
```



The degree of “pointedness” of propositions is determined by the `tension` parameter, which defaults to 1.7. Larger values make the proposition more bulgy, smaller values more pointy.

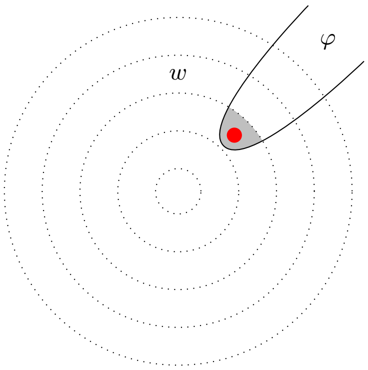
```
\begin{tikzpicture}
  \proposition[green,
     proposition/.style={tension=3}]{0}{3}{80}{3}
  \proposition{0}{3}{80}{3}
  \proposition[red,
     proposition/.style={tension=1}]{0}{3}{80}{3}
  \proposition[blue,
     proposition/.style={tension=.5}]{0}{3}{80}{3}
  \spheresystem{5}
\end{tikzpicture}
```



`\spherepos`

`\spherepos{`⟨*direction*⟩`}{`⟨*n*⟩`}{`⟨*code*⟩`}` moves to a position in the center of layer ⟨*n*⟩ in ⟨*direction*⟩ and then executes TikZ `path` code ⟨*code*⟩. It's useful to put labels or other things into the sphere system.

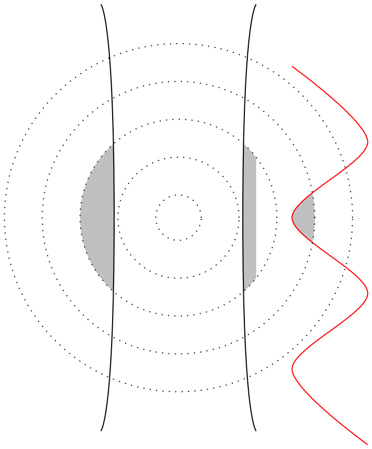
```
\begin{tikzpicture}
  \propositionintersect{45}{3}{20}{3}
  \spheresystem{5}
  \spherepos[fill,red]{45}{3}{circle[radius=.1]}
  \spherepos{90}{4}{node {$w$}}
  \spherepos{45}{6.5}{node {$\varphi$}}
\end{tikzpicture}
```



`\sphereintersect`

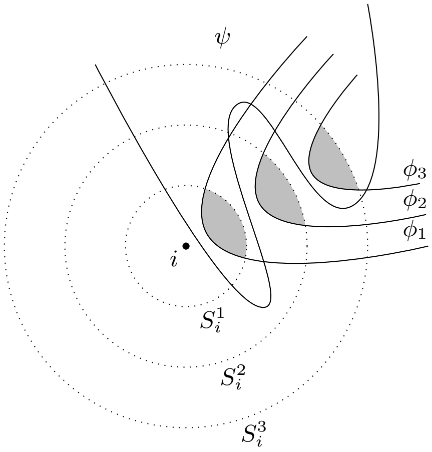
`\propositionintersect` uses `\sphereintersect[`⟨*options*⟩`]{`⟨*n*⟩`}{`⟨*code*⟩`}` to fill the area between the parabola and the outside edge of the ⟨*n*⟩-th sphere. (More precisely: what happens is that the area between the parabola and the line between its two endpoints is set as the clipping path, and then TikZ only shows the part of the shaded sphere within that clipping path.) That macro can also be used to intersect the respective layer with other paths, and in cases where the convex closure of the proposition does not include enough area. In that case, the clipping region has to be extended, and the path drawn separately. The example below shows what happens when a very wide parabola does not completely intersect with a sphere (on the right), how to use the trick to get the fill right (on the left), as well as how to intersect a more complex path with a sphere.

```
\begin{tikzpicture}
  \propositionintersect{0}{3}{140}{3}
  \sphereintersect{3}{\propositionplot{180}{3}{140}{3} -- (-2,-2)}
  \proposition{180}{3}{140}{3}
  \sphereintersect{4}{plot[smooth] coordinates
      {(1.5,2) (2.5,1) (1.5,0) (2.5,-1) (1.5,-2) (2.5,-3)}}
  \draw[red] plot[smooth] coordinates
      {(1.5,2) (2.5,1) (1.5,0) (2.5,-1) (1.5,-2) (2.5,-3)} ;
  \spheresystem{5}
\end{tikzpicture}
```



Finally, a complex example: the Sobel sequence diagram, Figure 2 from Lewis, p. 11:

```
\begin{tikzpicture}[scale=.8]\small
  % wider layers, pointier propositions
  \tikzset{layerwidth=1,innerfactor=0,
           proposition/.style={smooth,tension=1}}
  % fill the areas between three props and their innermost spheres
  \sphereintersect{3}{\propositionplot{30}{3.3}{30}{4}}
  \sphereintersect{2}{\propositionplot{30}{2.3}{45}{4}}
  \sphereintersect{1}{\propositionplot{30}{1.3}{60}{4}}
  % draw the sphere system
  \spheresystem{3}
  % draw the propositions
  \draw \propositionplot{30}{3.3}{30}{4};
  \draw \propositionplot{30}{2.3}{45}{4};
  \draw \propositionplot{30}{1.3}{60}{4};
  % draw \psi (coordinates figured out by trial and error)
  \draw plot[smooth,tension=1.2]
    coordinates {(-1.5,3) (1.2,-1) (.8,2.3) (2.8,.7) (3,4)};
  % draw and label the center world, spheres, and propositions
  \filldraw circle[radius=.05];
  \node at (-.2,-.2) {$i$};
  \spherepos{-70}{1.8}{node {$S^1_i$}}
  \spherepos{-70}{2.8}{node {$S^2_i$}}
  \spherepos{-70}{3.8}{node {$S^3_i$}}
  \spherepos{4}{4.3}{node {$\phi_1$}
                     node at +(0,.5) {$\phi_2$}
                     node at +(0,1) {$\phi_3$}}
  \spherepos{80}{4}{node {$\psi$}}
\end{tikzpicture}
```



3 Implementation

```
\RequirePackage{tikz}
```

```

\tikzset{
  sphere/.style = {dotted},
  sphere intersection/.style = {fill=lightgray},
  sphere layer/.style = {fill=lightgray},
  proposition/.style={smooth,tension=1.7},
}
```

`layerwidth`
`innerfactor`
TikZ parameters used to compute the sphere radii and can be set using TikZ's options mechanism or using `\tikzset`.

```
\pgfkeyssetvalue{/tikz/layerwidth}{.5}
\pgfkeyssetvalue{/tikz/innerfactor}{.4}
```

`\sphereplot` `\sphereplot{`⟨*n*⟩`}` gives the plot codes for the ⟨*n*⟩-th sphere

```
\newcommand{\sphereplot}[1]{
  circle
    [radius=(#1)*\pgfkeysvalueof{/tikz/layerwidth}-
      \pgfkeysvalueof{/tikz/layerwidth}*\pgfkeysvalueof{/tikz/innerfactor}]
}
```

`\spheresystem` `\spheresystem[`⟨*options*⟩`]{`⟨*n*⟩`}` draws a sphere system centered at the origin with ⟨*n*⟩ number of layers

```
\newcommand{\spheresystem}[2][]{
  \foreach \i in {1,...,#2}{
    \draw[sphere,#1] \sphereplot{\i} ;
  }
}
```

`\spherelayer` `\spherelayer[`⟨*options*⟩`]{`⟨*n*⟩`}` shades the ⟨*n*⟩-th layer

```
\newcommand{\spherelayer}[2][]{
  \begin{scope}[even odd rule]
    \fill[#1,sphere layer]
    \sphereplot{#2-1} \sphereplot{#2} ;
  \end{scope}
}
```

`\spherefill` `\spherefill[`⟨*options*⟩`]{`⟨*n*⟩`}` fills the ⟨*n*⟩-th sphere

```
\newcommand{\spherefill}[2][]{
    \fill[sphere intersection,#1]
    \sphereplot{#2} ;
}
```

`\sphereintersect` `\sphereintersect[`⟨*options*⟩`]{`⟨*n*⟩`}{`⟨*path*⟩`}` shades the area between ⟨*path*⟩ and the the ⟨*n*⟩-th sphere layer. Options only apply to the sphere layer.

```
\newcommand{\sphereintersect}[3][]{
  \begin{scope}[even odd rule]
    \path[clip] #3;
    \spherefill[#1]{#2}
  \end{scope}
}
```

`\propositionplot` `\propositionplot[`⟨*options*⟩`]{`⟨*direction*⟩`}{`⟨*n*⟩`}{`⟨*width*⟩`}{`⟨*length*⟩`}` produces the `plot` code for a proposition intersecting the ⟨*n*⟩-th layer in angle ⟨*direction*⟩ away from the center of the sphere system, with endpoints ⟨*length*⟩ away from the center at an angle of ⟨*direction*⟩ ± ⟨*width*⟩/2.

```
\newcommand{\propositionplot}[4]{
  plot [proposition]
  coordinates {+(#1+#3/2:#4)
    +(#1:#2*\pgfkeysvalueof{/tikz/layerwidth}-
    \pgfkeysvalueof{/tikz/layerwidth}*.9
    -\pgfkeysvalueof{/tikz/layerwidth}*\pgfkeysvalueof{/tikz/innerfactor})
    +(#1-#3/2:#4)}
}
```

`\proposition` `\proposition[`⟨*options*⟩`]{`⟨*direction*⟩`}{`⟨*n*⟩`}{`⟨*width*⟩`}{`⟨*length*⟩`}` actually draws the proposition. Note that ⟨*options*⟩ applies to `\draw`, not to `\plot`.

```
\newcommand{\proposition}[5][]{
  \draw[proposition,#1] \propositionplot {#2}{#3}{#4}{#5} ;
  }
```

`\propositionintersect` `\spherepropositionintersect` does the same as `\sphereproposition` but also shades the area of intersection with the ⟨*n*⟩-th sphere.

```
\newcommand{\propositionintersect}[5][]{
  \begin{scope}
  \path[clip] \propositionplot{#2}{#3}{#4}{#5};
  \spherefill[#1]{#3}
  \end{scope}
  \draw[proposition,#1] \propositionplot{#2}{#3}{#4}{#5};
}
```

`\spherepos` `\spherepos[`⟨*options*⟩`]{`⟨*direction*⟩`}{`⟨*n*⟩`}{`⟨*code*⟩`}` shifts the scope to a position in the center of the ⟨*n*⟩-th layer in direction angle from the center—and then puts a ⟨*code*⟩ `path` there.

```
\newcommand{\spherepos}[4][]{
  \begin{scope}[shift=(#2:#3*\pgfkeysvalueof{/tikz/layerwidth}-
      \pgfkeysvalueof{/tikz/layerwidth}/2-
        \pgfkeysvalueof{/tikz/layerwidth}*\pgfkeysvalueof{/tikz/innerfactor})]
    \path[#1] #4 ;
  \end{scope}
}
```

4 Change History

v1.00
  General: First public release 1

5 Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.