The **blochsphere** package

Matthew Wardrop
`mister.wardrop@gmail.com`

v1.1 from 2015/09/15

Contents

1 Introduction

The `blochsphere` package is used to draw pseudo-3D Blochsphere diagrams, such as that shown in figure 1. It supports various annotations, such as great and small circles, axes, rotation markings and state vectors. It can be used in a standalone fashion, or nested within a `tikzpicture` environment by setting the environment option `nested` to `true`.

2 Usage

Using `blochsphere` is as simple as creating a 'blochsphere' environment.

```
% \begin{blochsphere}[<options>]
% ...
% \end{blochsphere}
```

By default, this will create a blank Bloch sphere ready for annotation. The options can be any of those described in the next section.

`\drawBall` [⟨*options*⟩]
This macro draws the basic Bloch sphere, according to the options defined in the next section.

`\drawBallGrid` [⟨*options*⟩] {⟨*latdegsep*⟩} {⟨*longdegsep*⟩}
This macro draws latitude and longitude circles at an interval defined by `latdegsep` and `longdegsep` respectively.

`\setDrawingPlane` [⟨*options*⟩] {⟨*tiltangle*⟩} {⟨*rotationangle*⟩}
This macro defines a transformation matrix within TikZ to create the effect of 2D projections of 3D planes, named according to the `plane` option. This is used internally in all of the drawing commands.

`\setLatitudinalDrawingPlane` [⟨*options*⟩] {⟨*latitude*⟩}
This is a special case of `\setDrawingPlane` that constructs the transform required to draw a 2D image in a latitudinal plane.

`\setLongitudinalDrawingPlane` [⟨*options*⟩] {⟨*longitude*⟩}
This is a special case of `\setDrawingPlane` that constructs the transform required to draw a 2D image in a longitudinal plane.

`\drawCircle` [⟨*options*⟩] {⟨*tiltangle*⟩} {⟨*rotationangle*⟩}
Draws a circle on the surface of a Bloch sphere around the axis which is tilted by `tiltangle` from the z-axis and with a rotation of `rotationangle` about the z-axis.

`\drawGreatCircle` [⟨*options*⟩] {⟨*tiltangle*⟩} {⟨*rotationangle*⟩}
An alias of `\drawCircle`.

`\drawSmallCircle` [⟨*options*⟩] {⟨*tiltangle*⟩} {⟨*rotationangle*⟩} {⟨*offsetLatitude*⟩}
Draws a circle on the surface of a Bloch sphere around the axis which is tilted by `tiltangle` from the z-axis and with a rotation of `rotationangle` about the z-axis, with an offset along the axis such that it reaches an effective latitude of `offsetLatitude`.

`\drawLatitudeCircle` [⟨*options*⟩] {⟨*latitude*⟩}
Draws a circle on the Bloch sphere corresponding to the given latitude.

`\drawLongitudeCircle`	[⟨*options*⟩] {⟨*longitude*⟩} Draws a great circle on the Bloch sphere corresponding to the given longitude.
`\drawRotationLeft`	[⟨*options*⟩] {⟨*tiltangle*⟩} {⟨*rotationangle*⟩} {⟨*axisoffset*⟩} {⟨*clearangle*⟩} Draw a left-handed rotation about the Bloch sphere, titled and rotated as for a circle with the same parameters, with a gap in the line where the arrow should go of `clearangle` degrees.
`\drawRotationRight`	[⟨*options*⟩] {⟨*tiltangle*⟩} {⟨*rotationangle*⟩} {⟨*axisoffset*⟩} {⟨*clearangle*⟩} As above, but a right-handed rotation annotation.
`\drawAxis`	[⟨*options*⟩] {⟨*tiltangle*⟩} {⟨*rotationangle*⟩} Draw a line along the axis tilted `tiltangle` degrees from the z-axis, rotated about the z-axis by `rotationangle` degrees.
`\labelPolar`	[⟨*options*⟩] {⟨*tikzlabel*⟩} {⟨*tiltangle*⟩} {⟨*rotationangle*⟩} Assign a tikz label to the point identified by an angle `tiltangle` from the z-axis and a rotation `rotationangle` about the z-axis.
`\labelLatLon`	[⟨*options*⟩] {⟨*tikzlabel*⟩} {⟨*latitude*⟩} {⟨*longitude*⟩} Assign a tikz label to the point identified the specified latitude and longitude.
`\drawStatePolar`	[⟨*options*⟩] {⟨*tikzlabel*⟩} {⟨*tiltangle*⟩} {⟨*rotationangle*⟩} Draw a vector from the origin to the point identified by an angle `tiltangle` from the z-axis and a rotation `rotationangle` about the z-axis, which will be labelled `tikzlabel`.
`\drawStateLatLon`	[⟨*options*⟩] {⟨*tikzlabel*⟩} {⟨*latitude*⟩} {⟨*longitude*⟩} Draw a vector from the origin to the point identified by the provided latitude and longitude, which will be labelled `tikzlabel`.

3 Package, Environment and Macro options

For convenience, all options at all levels share the same namespace. Therefore, all options passed to the package will be the defaults for all environment, which will in turn be the defaults for all subsequent macro calls.

`radius`	[default = 1.5cm] This option controls the radius of the qubit, and can be specified in any valid LaTeXunits.
`tilt`	[default = 15] This option controls the tilt (into the page) of the top of the bloch sphere (in degrees).
`rotation`	[default = -20] This option controls the (right-handed) rotation of the Blochsphere around the (potentially tilted) z-axis, also specified in degrees.
`color`	[default = white] This specifies the colour of the drawn ball.
`opacity`	[default = 0.7] This option controls the opacity of the ball, allowing lines drawn

behind the ball to be visible.

`style`	[default =] This option controls the current drawing style, and supports any valid tikz styling.
`scale`	[default = 1] This option allows one to temporarily change the scale of drawn circles/etc, allowing annotations to “hover” over the Bloch sphere.
`plane`	[default = current plane] The name which should label the current plane being defined, or the plane to be used.
`shift`	[default = (0,0,0)] The three dimensional offset by which the currently being drawn objects should be shifted.
`ball`	[default = 3d] Specifies how the ball should be drawn. Options are: `3d`, `circle`, `none`
`statecolor`	[default = black] The color with which the state vectors should be drawn.
`statewidth`	[default = 0.4pt] The width with which state vectors should be drawn.
`axisarrow`	[default =] The arrow to use at the end of a drawn axis.
`labelmark`	[default = false] Whether a black dot should be drawn at the point being labelled.
`nested`	[default = false] Whether the `blochsphere` environment is being nested in a `tikzpicture` environment. If it is nested, then this should be set to true.

4 Example

In this section we provide example code to generate the following diagram:

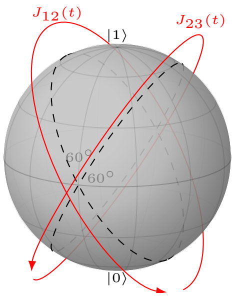

Figure 1: Example Bloch Sphere

```
\begin{blochsphere}[radius=1.5 cm,tilt=15,rotation=-20]
    \drawBallGrid[style={opacity=0.3}]{30}{30}

    \drawGreatCircle[style={dashed}]{-60}{0}{0}
    \drawGreatCircle[style={dashed}]{60}{0}{0}

    \drawRotationLeft[scale=1.3,style={red}]{-60}{0}{0}{15}
    \drawRotationRight[scale=1.3,style={red}]{60}{0}{0}{15}

    \node at (-0.8,1.9) {\textcolor{red}{\tiny £J_{12}(t)£}};
    \node at (1.1,1.8) {\textcolor{red}{\tiny £J_{23}(t)£}};

    \labelLatLon{up}{90}{0};
    \labelLatLon{down}{-90}{90};
    \node[above] at (up) {{\tiny £\left|1\right>£ }};
    \node[below] at (down) {{\tiny £\left|0\right>£}};

    \labelLatLon[labelmark=false]{d}{15}{90};
    \node at (d) {\color{gray}\fontsize{0.15cm}{1em}\selectfont £60^\circ£};

    \labelLatLon[labelmark=falsa]{d2}{5}{78};
    \node at (d2) {\color{gray}\fontsize{0.15cm}{1em}\selectfont £60^\circ£};
\end{blochsphere}
```

5 Implementation

5.1 Preamble

```
\NeedsTeXFormat{LaTeX2e}[1994/06/01]
\ProvidesPackage{blochsphere}[2015/08/17]
```

```
\usepackage{tikz,etoolbox,environ,ifthen,kvsetkeys,kvoptions}
\usetikzlibrary{decorations.pathreplacing, decorations.markings, calc, fadings}
```

5.2 Package/Environment/Macro options

Option: `radius`

```
\define@key{blochsphere}{radius}{%
\def\blochsphere@radius{#1}%
}
\providecommand{\blochsphere@radius}{1cm}
```

Option: `tilt`

```
\define@key{blochsphere}{tilt}{%
\def\blochsphere@tilt{#1}%
}
\providecommand{\blochsphere@tilt}{15}
```

Option: `rotation`

```
\define@key{blochsphere}{rotation}{%
\def\blochsphere@rotation{#1}%
}
\providecommand{\blochsphere@rotation}{-20}
```

Option: `color`

```
\define@key{blochsphere}{color}{%
\def\blochsphere@color{#1}%
}
\providecommand{\blochsphere@color}{white}
```

Option: `opacity`

```
\define@key{blochsphere}{opacity}{%
\def\blochsphere@opacity{#1}%
}
\providecommand{\blochsphere@opacity}{0.7}
```

Option: `style`

```
\define@key{blochsphere}{style}{%
\def\blochsphere@style{#1}%
}
\providecommand{\blochsphere@style}{}
```

Option: `scale`

```
\define@key{blochsphere}{scale}{%
\def\blochsphere@scale{#1}%
}
\providecommand{\blochsphere@scale}{1}
```

Option: `plane`

```
\define@key{blochsphere}{plane}{%
\def\blochsphere@plane{#1}%
}
\providecommand{\blochsphere@plane}{current plane}
```

Option: `shift`

```
\define@key{blochsphere}{shift}{%
\def\blochsphere@shift{{#1}}%
}
\providecommand{\blochsphere@shift}{{0,0,0}}
```

Option: `ball`

```
\define@key{blochsphere}{ball}{%
\def\blochsphere@ball{#1}%
}
\providecommand{\blochsphere@ball}{3d}
```

Option: `statecolor`

```
\define@key{blochsphere}{statecolor}{%
\def\blochsphere@statecolor{#1}%
}
\providecommand{\blochsphere@statecolor}{black}
```

Option: `statewidth`

```
\define@key{blochsphere}{statewidth}{%
\def\blochsphere@statewidth{#1}%
}
\providecommand{\blochsphere@statewidth}{0.4pt}
```

Option: `axisarrow`

```
\define@key{blochsphere}{axisarrow}{%
\def\blochsphere@axisarrow{#1}%
}
\providecommand{\blochsphere@axisarrow}{}
```

Option: `labelmark`

```
\define@key{blochsphere}{labelmark}{%
    \setboolean{blochsphere@labelmark}{#1}%
}
\newboolean{blochsphere@labelmark}
\setboolean{blochsphere@labelmark}{false}
```

Option: `nested`

```
\define@key{blochsphere}{nested}{%
\setboolean{blochsphere@nested}{#1}%
}
\newboolean{blochsphere@nested}
\setboolean{blochsphere@nested}{false}
```

```
\ProcessKeyvalOptions{blochsphere}\relax
```

5.3 `blochsphere` Environment

This is the central environment definition. Note that all macros defined below will only be defined within this environment.

blochsphere

```
\newenvironment{blochsphere}[1][]{
\begingroup
\makeatletter
\ignorespaces\setkeys{blochsphere}{#1}\unskip%
\pgfdeclarelayer{back}%
\pgfdeclarelayer{front}%
\pgfsetlayers{back,main,front}%
\pgfkeys{%
/tikz/on layer/.code={%
\pgfonlayer{##1}\begingroup%
\aftergroup\endpgfonlayer%
\aftergroup\endgroup%
},%
/tikz/node on layer/.code={%
\gdef\node@@on@layer{%
\setbox\tikz@tempbox=\hbox\bgroup\pgfonlayer{##1}\unhbox\tikz@tempbox\endpgfonlayer\egroup}%
\aftergroup\node@on@layer%
},%
/tikz/end node on layer/.code={%
\endpgfonlayer\endgroup\endgroup%
}%
}%
\def\node@on@layer{\aftergroup\node@@on@layer}%
```

\drawBall

```
\newcommand\drawBall[1][]{
\begingroup
\setkeys{blochsphere}{##1}
\expandafter\ifstrequal\expandafter{\blochsphere@ball}{3d}{
\fill[on layer=main,ball color=\blochsphere@color,opacity=\blochsphere@opacity] (0,0) circle (\
}{}
\expandafter\ifstrequal\expandafter{\blochsphere@ball}{circle}{
\draw[on layer=main,color=\blochsphere@color] (0,0) circle (\blochsphere@radius); % 3D lighting
}{}
\endgroup
}%
```

\drawBallGrid

```
\newcommand\drawBallGrid[3][]{
\begingroup
\setkeys{blochsphere}{##1}
\foreach \phi in {0,##2,...,89} {
\drawLatitudeCircle{\phi}
\drawLatitudeCircle{-\phi}
}
\foreach \theta in {0,##3,...,179} {
\drawLongitudeCircle{\theta}
}
\endgroup
}%
```

\setDrawingPlane

```
\newcommand\setDrawingPlane[3][]{%
\setkeys{blochsphere}{##1}
\pgfmathsetmacro\aphi{##2}
\pgfmathsetmacro\atheta{##3}
\pgfmathsetmacro\dot{cos(\aphi)}
\ifdim\dot pt<0.7 pt\relax%
\pgfmathsetmacro\norm{cos(asin(\dot))}
\pgfmathsetmacro\xx{-sin(\aphi)*sin(\blochsphere@rotation - \atheta)}
\pgfmathsetmacro\xy{sin(\aphi)*sin(\blochsphere@tilt)*cos(\blochsphere@rotation - \atheta)}
\pgfmathsetmacro\yx{-sin(2*\aphi - \blochsphere@rotation + \atheta)/4 - sin(2*\aphi + \blochsph
\pgfmathsetmacro\yy{(sin(\aphi)*cos(\blochsphere@tilt) - sin(\blochsphere@tilt)*sin(\blochsphere
\else
\pgfmathsetmacro\norm{sqrt(abs(sin(\blochsphere@rotation - \atheta)*cos(\aphi))^2 + abs(sin(\bl
\pgfmathsetmacro\xx{-sin(\blochsphere@rotation - \atheta)*cos(\aphi)}
\pgfmathsetmacro\xy{sin(\blochsphere@tilt)*cos(\aphi)*cos(\blochsphere@rotation - \atheta)}
\pgfmathsetmacro\yx{-cos(\aphi)^2*cos(\blochsphere@rotation - \atheta)}
\pgfmathsetmacro\yy{(sin(\aphi)*cos(\blochsphere@tilt) - sin(\blochsphere@tilt)*sin(\blochsphere
\fi
\computeOffset{shift}
  \begingroup\newcommand\tmp[1]{\endgroup\noexpand\relax%
\tikzset{\blochsphere@plane/.style={cm={\xx/\norm, \xy/\norm, \yx/\norm, \yy/\norm, \unexpanded
} \tmp{\shift}
}%
```

\setLatitudinalDrawingPlane

```
\newcommand\setLatitudinalDrawingPlane[2][]{%
\pgfmathsetmacro\yshift{sin(##2)*\the\blochsphere@radius}
\setDrawingPlane[##1,shift={{0,0,\yshift}}]{0}{0}
}%
```

\setLatitudinalDrawingPlane

```
\newcommand\setLongitudinalDrawingPlane[2][]{%
\setDrawingPlane[##1]{90}{##2}
}%
```

\drawCircle

```
\newcommand\drawCircle[3][]{
```

```
\begingroup
\setkeys{blochsphere}{##1}
\setDrawingPlane{##2}{##3}
\computeVisibility{##2}{##3}{agamma}{abeta}
\begingroup\edef\tmp{\endgroup%
\noexpand\draw[current plane,on layer=back,\unexpanded\expandafter{\blochsphere@style}] \unexpa
\noexpand\draw[current plane,on layer=front,\unexpanded\expandafter{\blochsphere@style}] \unexp
} \tmp
\endgroup
}%
```

\drawGreatCircle

```
\newcommand\drawGreatCircle[3][]{
\drawCircle[##1]{##2}{##3}
}
```

\drawSmallCircle

```
\newcommand\drawSmallCircle[4][]{
\begingroup
\pgfmathsetmacro\xshift{sin(##2)*cos(##3)*\blochsphere@radius*sin(##4)}
\pgfmathsetmacro\yshift{sin(##2)*sin(##3)*\blochsphere@radius*sin(##4)}
\pgfmathsetmacro\zshift{cos(##2)*\blochsphere@radius*sin(##4)}
\pgfmathsetmacro\radius{\blochsphere@radius*cos(##4)}
\drawCircle[##1,shift={{\xshift,\yshift,\zshift}},radius=\radius]{##2}{##3}
\endgroup
}%
```

\drawLatitudeCircle

```
\newcommand\drawLatitudeCircle[2][]{
\begingroup
\pgfmathsetmacro\yshift{sin(##2)*\blochsphere@radius}
\pgfmathsetmacro\radius{\blochsphere@radius*cos(##2)}
\drawCircle[##1,shift={{0,0,\yshift}},radius=\radius]{0}{0}
\endgroup
}%
```

\drawLongitudeCircle

```
\newcommand\drawLongitudeCircle[2][]{
\begingroup
\drawCircle[##1]{90}{##2+90}
\endgroup
}%
```

\drawRotationLeft

```
\newcommand\drawRotationLeft[5][]{
\begingroup
\setkeys{blochsphere}{##1}
\pgfmathsetmacro\xshift{sin(##2)*cos(##3)*##4}
\pgfmathsetmacro\yshift{sin(##2)*sin(##3)*##4}
\pgfmathsetmacro\zshift{cos(##2)*##4}
\setDrawingPlane[shift={\xshift,\yshift,\zshift}]{##2}{##3}
\computeVisibility{##2}{##3}{agamma}{abeta}
\begingroup\edef\tmp{\endgroup%
\noexpand\draw[current plane,on layer=back,\unexpanded\expandafter{\blochsphere@style}] \unexpa
\noexpand\draw[<-,current plane,on layer=front,\unexpanded\expandafter{\blochsphere@style}] \un
} \tmp
\endgroup
}%
```

\drawRotationRight

```
\newcommand\drawRotationRight[5][]{
\begingroup
\setkeys{blochsphere}{##1}
\pgfmathsetmacro\xshift{sin(##2)*cos(##3)*##4}
\pgfmathsetmacro\yshift{sin(##2)*sin(##3)*##4}
\pgfmathsetmacro\zshift{cos(##2)*##4}
\setDrawingPlane[shift={\xshift,\yshift,\zshift}]{##2}{##3}
\computeVisibility{##2}{##3}{agamma}{abeta}
\begingroup\edef\tmp{\endgroup%
\noexpand\draw[current plane,on layer=back,\unexpanded\expandafter{\blochsphere@style}] \unexpa
\noexpand\draw[->,current plane,on layer=front,\unexpanded\expandafter{\blochsphere@style}] \un
} \tmp
\endgroup
}%
```

\drawAxis

```
\newcommand\drawAxis[3][] {
\begingroup
\setkeys{blochsphere}{##1}
\pgfmathsetmacro\behind{ifthenelse(-sin(##2)*sin(\blochsphere@rotation - ##3)*cos(\blochsphere@
\pgfmathsetmacro\newphi{180-##2}
\pgfmathsetmacro\newtheta{##3+180}
\ifnum\behind=0\relax
\labelPolar[scale=1]{blochspheretmpp}{##2}{##3}
\labelPolar{blochspheretmps}{##2}{##3}
\labelPolar{blochspheretmpe}{\newphi}{\newtheta}
\else
\labelPolar[scale=1]{blochspheretmpp}{\newphi}{\newtheta}
\labelPolar{blochspheretmpe}{##2}{##3}
\labelPolar{blochspheretmps}{\newphi}{\newtheta}
\fi
```

```
\begingroup\edef\tmp{\endgroup%
\ifnum\behind=0\relax
\noexpand\draw[on layer=front,\unexpanded\expandafter{\blochsphere@style},\blochsphere@axisarro
\noexpand\draw[on layer=back,\unexpanded\expandafter{\blochsphere@style}] (blochspheretmpe) --
\else
\noexpand\draw[on layer=front,\unexpanded\expandafter{\blochsphere@style},\blochsphere@axisarro
\noexpand\draw[on layer=back,\unexpanded\expandafter{\blochsphere@style}] (blochspheretmps) --
\fi
} \tmp
\endgroup
}%
```

\labelPolar

```
\newcommand\labelPolar[4][]{%
\begingroup
\setkeys{blochsphere}{##1}
\setLongitudinalDrawingPlane{##4+90}
\pgfmathsetmacro\behind{ifthenelse(-sin(##3)*sin(\blochsphere@rotation - ##4)*cos(\blochsphere@
\ifthenelse{\boolean{blochsphere@labelmark}}
        {
            \ifnum\behind=1\relax
                \path[current plane,on layer=back] (90-##3:\blochsphere@radius*\blochsphere@sca
            \else
                \path[current plane,on layer=front] (90-##3:\blochsphere@radius*\blochsphere@sc
            \fi
        }
        {
            \ifnum\behind=1\relax
                \path[current plane,on layer=back] (90-##3:\blochsphere@radius*\blochsphere@sca
            \else
                \path[current plane,on layer=front] (90-##3:\blochsphere@radius*\blochsphere@sc
            \fi
        }
\endgroup
}%
```

\labelLatLon

```
\newcommand\labelLatLon[4][]{%
\begingroup
\pgfmathsetmacro\newphi{90-##3}
\labelPolar[##1]{##2}{\newphi}{##4}
\endgroup
}%
```

\drawStateLatLon

```
\newcommand\drawStateLatLon[4][]{
```

```
\begingroup
\pgfmathsetmacro\newphi{90-##3}
\drawStatePolar[##1]{##2}{\newphi}{##4}
\endgroup
}%
```

\drawStatePolar

```
\newcommand\drawStatePolar[4][]{
\begingroup
\setkeys{blochsphere}{##1}
\labelPolar{##2}{##3}{##4}
\pgfmathsetmacro\behind{ifthenelse(-sin(##3)*sin(\blochsphere@rotation - ##4)*cos(\blochsphere@
\ifnum\behind=1\relax
\tikzset{test/.style={
postaction={
decorate,
decoration={
markings,
mark=at position \pgfdecoratedpathlength-0.5pt with {\arrow[\blochsphere@statecolor,line width=
mark=between positions 0 and \pgfdecoratedpathlength-5pt step 0.5pt with {
\pgfmathsetmacro\myval{multiply(divide(
\pgfkeysvalueof{/pgf/decoration/mark info/distance from start}, \pgfdecoratedpathlength),100)};
\pgfsetfillcolor{\blochsphere@statecolor};
\pgfsetfillopacity{0.4-\myval/100*0.2}
\pgfpathcircle{\pgfpointorigin}{\blochsphere@statewidth};
\pgfusepath{fill};}
}}}}
\else
\tikzset{test/.style={
postaction={
decorate,
decoration={
markings,
mark=at position \pgfdecoratedpathlength-0.5pt with {\arrow[\blochsphere@statecolor,line width=
mark=between positions 0 and \pgfdecoratedpathlength-5pt step 0.5pt with {
\pgfmathsetmacro\myval{multiply(divide(
\pgfkeysvalueof{/pgf/decoration/mark info/distance from start}, \pgfdecoratedpathlength),100)};
\pgfsetfillcolor{\blochsphere@statecolor};
\pgfsetfillopacity{0.4+\myval/100*0.6}
\pgfpathcircle{\pgfpointorigin}{\blochsphere@statewidth};
\pgfusepath{fill};}
}}}}
\fi
\path [test,on layer=main] (0,0) -- (##2);
\endgroup
}%
```

\computeOffset

```
\newcommand\computeOffset[2][]{%
\setkeys{blochsphere}{##1}%
\pgfmathsetmacro{\x}{\blochsphere@shift[0]*cos(\blochsphere@rotation) + \blochsphere@shift[1]*s
\pgfmathsetmacro\y{\blochsphere@shift[0]*sin(\blochsphere@rotation)*sin(\blochsphere@tilt) - \b
\expandafter\def\csname ##2\endcsname{(\x pt,\y pt)}%
}%
```

\computeVisibility

```
\newcommand\computeVisibility[5][]{
\setkeys{blochsphere}{##1}
\pgfmathsetmacro\aphi{##2}
\pgfmathsetmacro\atheta{##3}
\pgfmathsetmacro\d{sqrt(\blochsphere@shift[0]^2+\blochsphere@shift[1]^2+\blochsphere@shift[2]^2
\pgfmathsetmacro\tatheta{\atheta+\blochsphere@rotation}
\pgfmathsetmacro\dot{cos(\aphi)}
\ifdim\dot pt<0.7 pt\relax%
\pgfmathsetmacro\domaintest{ifthenelse(and(\blochsphere@rotation - \atheta==0,or(\aphi==0,sin(\
\ifdim\domaintest pt=0 pt\relax
\pgfmathsetmacro\agamma{0}
\else
\pgfmathsetmacro\agamma{-90+atan(cos(\blochsphere@tilt)*cos(\blochsphere@rotation - \atheta)/(s
\fi
\else
\pgfmathsetmacro\domaintest{ifthenelse(and(Mod(\blochsphere@rotation-\atheta,180)==90,Mod(\bloc
\pgfmathsetmacro\domaintesttwo{ifthenelse(Mod(\blochsphere@rotation-\atheta,180)==90,0,1)}
\pgfmathsetmacro\domaintestthree{ifthenelse(Mod(\blochsphere@tilt,180)==90,0,1)}
\ifdim\domaintest pt=0 pt\relax
\pgfmathsetmacro\agamma{360-90*\blochsphere@tilt/abs(\blochsphere@tilt)}
\else
\ifdim\domaintesttwo pt=0 pt\relax
\pgfmathsetmacro\agamma{360-90*(sin(\aphi)*tan(\blochsphere@tilt)+cos(\aphi)*sin(\blochsphere@r
\else
\ifdim\domaintestthree pt=0 pt\relax
\pgfmathsetmacro\agamma{360-90*\blochsphere@tilt/abs(\blochsphere@tilt)*cos(\blochsphere@rotati
\else
\pgfmathsetmacro\agamma{360-atan( (sin(\aphi)*tan(\blochsphere@tilt)+cos(\aphi)*sin(\blochspher
\fi
\fi
\fi
\fi
\pgfmathsetmacro\aalpha{acos(-sin(\aphi)*sin(\blochsphere@rotation)*cos(\atheta)*cos(\blochsphe
\pgfmathsetmacro\aalphatest{atan(\d/\blochsphere@radius)}
\ifdim\aalpha pt<\aalphatest pt\relax
\pgfmathsetmacro\abeta{0}
\else
\pgfmathsetmacro\abeta{acos(\d*cot(\aalpha)/\blochsphere@radius)}
\fi
\pgfmathsetmacro\abeta{ifthenelse(\blochsphere@shift[2]<0,\abeta+2*(90-\abeta),\abeta)}
\expandafter\pgfmathsetmacro\csname ##4\endcsname{\agamma}
\expandafter\pgfmathsetmacro\csname ##5\endcsname{\abeta}
}%

\tikzset{%
>=latex, % option for nice arrows
inner sep=0pt,%
outer sep=2pt,%
mark coordinate/.style={inner sep=0pt,outer sep=0pt,minimum size=3pt,
fill=black,circle}%
}%
\pgfdeclareradialshading[tikz@ball]{ball}{\pgfqpoint{-10bp}{10bp}}{%
color(0bp)=(tikz@ball!30!white);
color(9bp)=(tikz@ball!75!white);
color(18bp)=(tikz@ball!90!black);
color(25bp)=(tikz@ball!70!black);
color(50bp)=(black)
}%
\ifthenelse{\boolean{blochsphere@nested}}{}{%
\begin{tikzpicture}
\drawBall
}%
}

{
\ifthenelse{\boolean{blochsphere@nested}}{}{
\end{tikzpicture}%
}%
\endgroup%
}
\endinput
```