

# Package: sphereplot (via r-universe)

October 24, 2024

**Type** Package

**Title** Spherical Plotting

**Version** 2.0.8

**Date** 2023-08-30

**Author** Aaron Robotham

**Maintainer** Aaron Robotham <aaron.robatham@uwa.edu.au>

**Description** Various functions for creating spherical coordinate system plots via extensions to rgl.

**License** GPL-3

**Depends** rgl, checkmate

**Repository** <https://asgr.r-universe.dev>

**RemoteUrl** <https://github.com/asgr/sphereplot>

**RemoteRef** HEAD

**RemoteSha** 0e579de8f61a0e0540cb893a48f9e7ad69956ea0

## Contents

sphereplot-package	2
car2sph	2
pointsphere	3
rgl.sphcirc	4
rgl.sphglobe	5
rgl.sphgrid	6
rgl.sphline	8
rgl.sphMW	9
rgl.sphpoints	10
rgl.sphsun	11
rgl.sphtext	12
sph2car	13

<b>Index</b>	<b>15</b>
--------------	-----------

---

sphereplot-package      *Spherical plotting routines*

---

### Description

Various low level and high level routines for generate spherical plots. Includes celestial sphere style 3D grid and routines for overlaying additional points and text. Requires rgl.

### Details

Package: sphereplot  
Type: Package  
Version: 2.0.8  
Date: 2023-08-30  
License: GPL-3

Standard usage is to run rgl.sphgrid to make the 3D coordinate grid, then add points using rgl.sphpoints.

### Author(s)

Aaron Robotham

Maintainer: Aaron Robotham <aaron.robotham@uwa.edu.au>

---

car2sph      *Transforms 3D cartesian coordinates to spherical coordinates*

---

### Description

Transforms 3D cartesian coordinates to spherical coordinates. The user can choose to return the spherical coordinates in degrees or radians.

### Usage

```
car2sph(x, y, z, deg = TRUE)
```

### Arguments

x	x values, can also contain a matrix of x, y and z (in that order).
y	y values.
z	z values
deg	Should degrees be returned (default) or radians.

**Details**

This is a low level function that is used for plot transformations.

**Value**

A data.frame is returned containing the columns long (longitude), lat (latitude) and radius.

**Author(s)**

Aaron Robotham

**See Also**

[rgl.sphgrid](#), [rgl.sphcirc](#), [rgl.sphpoints](#), [rgl.sphtext](#), [rgl.sphsun](#), [rgl.sphMW](#), [pointsphere](#), [sph2car](#)

**Examples**

```
print(car2sph(x=1,y=1,z=0,deg=TRUE))
```

---

pointsphere	<i>Random sphere pointing</i>
-------------	-------------------------------

---

**Description**

Randomly generates data points within a sphere that are uniformly distributed.

**Usage**

```
pointsphere(N = 100, longlim = c(0, 360), latlim = c(-90, 90), rlim = c(0, 1))
```

**Arguments**

N	Number of random points.
longlim	Limits of longitude in degrees.
latlim	Limits of latitude in degrees.
rlim	Limits of radius.

**Details**

This function randomly generates data points within a sphere that are uniformly distributed. 3D pointing is based in efficient inversion of random uniform distributions, rather than a Monte-Carlo approach.

**Value**

Returns a data.frame containing the longitude, latitude and radius of the random points generated.

**Author(s)**

Aaron Robotham

**See Also**

[rgl.sphgrid](#), [rgl.sphcirc](#), [rgl.sphpoints](#), [rgl.sphtext](#), [rgl.sphsun](#), [rgl.sphMW](#), [sph2car](#), [car2sph](#)

**Examples**

```
rgl.sphgrid()
rgl.sphpoints(pointsphere(100,c(0,90),c(0,45),c(0.25,0.8)),deg=TRUE)
```

---

rgl.sphcirc

*Great circle generator*

---

**Description**

Function to generate a new great circle with arbitrary inclinations and radius.

**Usage**

```
rgl.sphcirc(CrossEq = 0, PeakDec = 0, radius = 1, deg = TRUE, col = "black", ...)
```

**Arguments**

CrossEq	The right ascension where the new circle crosses the equator. This should be the crossing before the Declination values of the new great circle become positive.
PeakDec	The peak declination the new great circle will reach.
radius	The radius of the new great circle.
deg	Specifies if input is in degrees (default) or radians.
col	The colour of the new great circle line.
...	Other arguments carried to lines3d.

**Details**

See [rgl.sphsun](#) and [rgl.sphMW](#) for examples of rgl.sphcirc in use.

**Value**

No value, used for plotting side effect.

**Author(s)**

Aaron Robotham

**See Also**

[rgl.sphgrid](#), [rgl.sphpoints](#), [rgl.sphtext](#), [rgl.sphsun](#), [rgl.sphMW](#), [pointsphere](#), [sph2car](#), [car2sph](#)

**Examples**

```
rgl.sphgrid()
rgl.sphcirc(radius=0.5,col='blue')
```

---

```
rgl.sphglobe
```

*Create a World, Night Sky or CMB Globe*

---

**Description**

Plots a background map of either the World (our Earth), night sky, or the Planck CMB map. This is nice for certain visualisations of the astronomical locations, and also plotting the world.

**Usage**

```
rgl.sphglobe(type = "sky1", radius = 1, res = 100, alpha = 0.99999, grid = TRUE,
  lit = FALSE, texture = NULL, longshift = 0, longtype = 'D', ...)
```

**Arguments**

type	Character scalar; used if 'texture' is NULL. One of: sky1 (low res) / sky2 (high res) / world1 (Light) / world2 (Dark) / world3 (simple) / cmb_gal (Planck in Galactic) / cmb_eq (Planck in Equatorial).
radius	Numeric scalar; radius to make the globe. If set to 'auto' it will be 1 for sky1 / sky2; 6371 for world1 / world2 (Earth radius in km); 13.8 for CMB (age of the Universe in Gyrs).
res	Numeric scalar; resolution of the globe (default should usually be fine).
alpha	Numeric scalar; transparency of the globe. Setting to 1 slows down the rendering for some reason.
grid	Logical; show coordinate grid?
lit	Logical; illuminate the globe?
texture	Character scalar; path to a texture to wrap onto the globe. This needs to be a rectangular projection to display correctly.
longshift	Shift to make to the texture in degrees along the longitude dimension. Many images are centred at either 0 or 180, where the default of 0 assumes 0 degrees longitude is at the centre of the texture. Only relevant if user provides input to 'texture'.
longtype	Specifies if longitudes should be labelled in hours (H- default) or degrees (D). See <a href="#">rgl.sphgrid</a> .
...	Extra arguments for <a href="#">rgl.sphgrid</a> .

### Details

Run for the side effect of generating a pretty globe to plot on top of.

The sky and World images are in the equatorial system, but I can only find Planck maps using a Galactic projection, so this is not in equatorial! But for giving the general impression of the CMB background, this might not matter.

All images were found online under open licenses. In particular the sky images were provided by NASA.

### Value

Run for the side effect of generating a pretty globe to plot on top of.

### Author(s)

Aaron Robotham

### See Also

[rgl.sphgrid](#), [rgl.sphline](#)

### Examples

```
## Not run:
rgl.sphglobe('world1')
rgl.sphglobe('world2')
rgl.sphglobe('world3')
rgl.sphglobe('sky1')
rgl.sphglobe('sky2')
rgl.sphglobe('cmb')

## End(Not run)
```

---

rgl.sphgrid

*Create a spherical plotting grid*

---

### Description

Creates a spherical plotting grid, within which further points can be added. Useful for astronomical plotting in particular, where this becomes the celestial sphere.

### Usage

```
rgl.sphgrid(radius = 1, axis.val = TRUE, col.long = 'red', col.lat = 'blue',
  col.rad = 'darkgreen', col.axes = 'grey50', col.grid = 'grey', deggap = 15,
  longtype = "H", add = FALSE, radaxis = TRUE, radlab = 'Radius', col.radlab = 'black')
```

**Arguments**

<code>radius</code>	The radial extent of the spherical grid.
<code>axis.val</code>	Logical; show numeric values on the spherical grid.
<code>deggap</code>	The attempted separation between spherical grid lines in degrees.
<code>col.long</code>	The colour for longitude axis value labels.
<code>col.lat</code>	The colour for latitude axis value labels.
<code>col.rad</code>	The colour for radius axis value labels.
<code>col.axes</code>	The colour for axes (i.e. grid line next to labels).
<code>col.grid</code>	The colour for grid lines.
<code>longtype</code>	Specifies if longitudes should be labelled in hours (H- default) or degrees (D).
<code>add</code>	Should the grid be added to the current plot, or if FALSE a new rgl device is launched.
<code>radaxis</code>	Logical determining whether the sphere radius vector is drawn and labeled (default is TRUE). Pretty labeling is used to choose the location of ticks and labels.
<code>radlab</code>	If 'radaxis' is TRUE then the 'radlab' parameter determines the name of the label.
<code>col.radlab</code>	Colour of the radius title.

**Details**

This function should be called first, and can generally be used with only the declaration of radius to good effect.

**Value**

No value, used for plotting side effect.

**Author(s)**

Aaron Robotham

**Examples**

```
rgl.sphgrid()
```

---

 rgl.sphline

*Draw Great and Minor Circle Line Segments*


---

### Description

Given points on a sphere, these functions will draw the connecting great and minor circle minimum distance line connecting them.

### Usage

```

rgl.sphline(long1, lat1, long2, lat2, radius = 1, deg = TRUE, col = "black", res = 1000,
  ...)
rgl.sphlines(long, lat, ...)
rgl.seglong(long1, long2, lat, radius=1, deg=TRUE, col='black', res=1000, ...)
rgl.seglat(long, lat1, lat2, radius=1, deg=TRUE, col='black', res=1000, ...)
rgl.segbox(long1, long2, lat1, lat2, radius=1, deg=TRUE, col='black', res=1000, ...)

```

### Arguments

long1	Numeric scalar; longitude of first location.
lat1	Numeric scalar; latitude of first location.
long2	Numeric scalar; longitude of second location.
lat2	Numeric scalar; latitude of second location.
long	Numeric vector; longitudes on plotting path.
lat	Numeric vector; latitudes on plotting path.
radius	Numeric scalar; radius of segment to draw.
deg	Logical; specifies if input is in degrees (default) or radians.
col	Specifies line colour.
res	Numeric scalar; resolution of line to draw (how many elements in the total great circle). Only make this large if plotting points very close together, otherwise the default is generally high enough resolution.
...	Other arguments to pass to <a href="#">lines3d</a> .

### Details

The main difference between `rgl.sphline` and `rgl.sphlines` is that the former takes scalar inputs but gives you the great outputs listed below, whilst `rgl.sphlines` can take vector inputs and plot complicated paths, but provides no return from the function.



**Value**

rgl.sphline and rgl.sphlines are called for the side effect of plotting on the current sphere.

rgl.seglong draws longitude lines of constant latitude (technically minor circle arcs, except at the equator).

rgl.seglat draws latitude lines of constant longitude (great circle arcs).

rgl.segbox draws boxes with the specified longitude and latitude limits.

rgl.sphline also returns various diagnostic outputs:

great_circle	Cartesian coordinates of the full great circle.
segment	Cartesian coordinates of the great circle segment.
CrossEq	Location where great circle crosses the equator.
PeakDec	Peak declination / latitude of the great circle.
AngSep	Angular separation of the two points in degrees.
CrossProd	Three element cross product vector for the two input positions.

**Author(s)**

Aaron Robotham

**See Also**

[rgl.sphglobe](#)

**Examples**

```
rgl.sphglobe('world1')
rgl.sphline(20,30,40,60)
rgl.sphpoints(c(20,40), c(30,60), size=10)
rgl.sphline(340,-20,80,30)
rgl.sphpoints(c(340,80), c(-20,30), size=10)
rgl.sphline(30,40,80,-60)
rgl.sphpoints(c(30,80), c(40,-60), size=10)
```

---

 rgl.sphMW

---

*Plot Galactic plane + Galactic centre*


---

**Description**

This function overplots the Galactic plane on the default Equatorial coordinates, and optionally will add the Galactic centre.

**Usage**

```
rgl.sphMW(radius = 1, col = "purple", type = "s", MWcenrad = 0.02, addMWplane = TRUE, ...)
```

**Arguments**

radius	The radius at which to draw the Galactic plane and Galactic centre.
col	The colour of the Galactic plane line and the Galactic centre.
type	Rgl plot type for the Galactic centre, default is to draw it as a 3D sphere, i.e. type 's'.
MWcenrad	The radius of the Galactic centre if plotted as a sphere.
addMWplane	Should the Galactic plane be drawn.
...	Extra arguments for lines3d.

**Value**

No value, used for plotting side effect.

**Author(s)**

Aaron Robotham

**See Also**

[rgl.sphgrid](#), [rgl.sphcirc](#), [rgl.sphpoints](#), [rgl.sphtext](#), [rgl.sphsun](#), [pointsphere](#), [sph2car](#), [car2sph](#)

**Examples**

```
rgl.sphgrid()
rgl.sphMW()
```

---

`rgl.sphpoints`      *Add points to spherical plots*

---

**Description**

This function allows the native plotting of spherical coordinates (in degrees or radians) and is expected to be used in conjunction with `rgl.sphgrid`, which produces the spherical grid.

**Usage**

```
rgl.sphpoints(long, lat, radius=1, deg = TRUE, col = 'black', ...)
```

**Arguments**

long	Numeric vector; longitude values, can also contain a matrix of long, lat and radius (in that order).
lat	Numeric vector; latitude values.
radius	Numeric vector; radius values.
deg	Logical; specifies if input is in degrees (default) or radians.
col	Specifies point colour.
...	Other arguments carried to <code>points3d</code> .

**Details**

This function uses sph2car in conjunction with points3d to plot points on a spherical coordinate system.

**Value**

No value, used for plotting side effect.

**Author(s)**

Aaron Robotham

**See Also**

[rgl.sphgrid](#), [rgl.sphcirc](#), [rgl.sphpoints](#), [rgl.sphtext](#), [rgl.sphsun](#), [rgl.sphMW](#), [pointsphere](#), [sph2car](#), [car2sph](#)

**Examples**

```
rgl.sphgrid()
rgl.sphpoints(40,50,0.5,deg=TRUE,col='red',cex=2)
```

---

```
rgl.sphsun
```

```
Plot ecliptic + Sun
```

---

**Description**

This function overplots the ecliptic plane on the default Equatorial coordinates, and optionally will add the Sun either for a desired date, or for today.

**Usage**

```
rgl.sphsun(Ydate = c(3, 21), radius = 1, col = "yellow", type = "s", sunrad = 0.02,
addeclip = TRUE, addsun = TRUE, ...)
```

**Arguments**

Ydate	The date for the location of the Sun on the spherical grid. Vector in c(M,D) format. If set to 'get' then the function will return the Sun's location for today.
radius	The radius at which to draw the ecliptic plane and Sun.
col	The colour of the ecliptic line and for the Sun.
type	RGL plot type for the Sun, default is to draw it as a 3D sphere, i.e. type 's'.
sunrad	The radius of the Sun if plotted as a sphere.
addeclip	Should the ecliptic plane be drawn.
addsun	Should the location of the Sun be plotted.
...	Extra arguments for lines3d.

**Value**

No value, used for plotting side effect.

**Author(s)**

Aaron Robotham

**See Also**

[rgl.sphgrid](#), [rgl.sphcirc](#), [rgl.sphpoints](#), [rgl.sphtext](#), [rgl.sphMW](#), [pointsphere](#), [sph2car](#), [car2sph](#)

**Examples**

```
rgl.sphgrid()
rgl.sphsun()

rgl.sphgrid()
rgl.sphsun('get', radius=2, col='red')

open3d()
```

---

rgl.sphtext

*Add text to spherical plot*

---

**Description**

Adds generic text to a spherical coordinate plot.

**Usage**

```
rgl.sphtext(long, lat, radius, text, deg = TRUE, col = 'black', ...)
```

**Arguments**

long	longitude values, can also contain a matrix of long, lat and radius (in that order).
lat	latitude values.
radius	radius values.
text	text values to be plotted.
deg	Specifies if input is in degrees (default) or radians.
col	Specifies text colour.
...	Other arguments carried to points3d.

**Details**

This function uses sph2car in conjunction with text3d to plot text on a spherical coordinate system.

**Value**

No value, used for plotting side effect.

**Author(s)**

Aaron Robotham

**See Also**

[rgl.sphgrid](#), [rgl.sphcirc](#), [rgl.sphpoints](#), [rgl.sphsun](#), [rgl.sphMW](#), [pointsphere](#), [sph2car](#), [car2sph](#)

**Examples**

```
rgl.sphgrid()  
rgl.sphtext(40,50,0.5, 'HI!', deg=TRUE, col='red', cex=2)
```

---

sph2car

*Transforms 3D spherical coordinates to cartesian coordinates*

---

**Description**

Transforms 3D spherical coordinates to cartesian coordinates. The user can choose to input the spherical coordinates in degrees or radians.

**Usage**

```
sph2car(long, lat, radius = 1, deg = TRUE)
```

**Arguments**

long	Longitude values, can also contain a matrix of long, lat and radius (in that order).
lat	Latitude values.
radius	Radius values.
deg	Specifies if input is in degrees (default) or radians.

**Details**

This is a low level function that is used for plot transformations.

**Value**

A data.frame is returned containing the columns x, y and z.

**Author(s)**

Aaron Robotham

**See Also**

[rgl.sphgrid](#),[rgl.sphcirc](#),[rgl.sphpoints](#),[rgl.sphtext](#),[rgl.sphsun](#),[rgl.sphMW](#),[pointsphere](#),[car2sph](#)

**Examples**

```
print(sph2car(45,0,sqrt(2),deg=TRUE))
```

# Index

- \* **circle**
    - rgl.sphcirc, 4
  - \* **galactic**
    - rgl.sphMW, 9
  - \* **package**
    - sphereplot-package, 2
  - \* **plot3d**
    - rgl.sphpoints, 10
    - rgl.sphtext, 12
  - \* **plot**
    - rgl.sphgrid, 6
  - \* **points3d**
    - rgl.sphpoints, 10
  - \* **points**
    - rgl.sphpoints, 10
  - \* **random**
    - pointsphere, 3
  - \* **sun**
    - rgl.sphsun, 11
  - \* **text3d**
    - rgl.sphtext, 12
  - \* **text**
    - rgl.sphtext, 12
  - \* **transform**
    - car2sph, 2
    - sph2car, 13
- car2sph, 2, 4, 5, 10–14
- lines3d, 8
- pointsphere, 3, 3, 5, 10–14
- rgl.segbox (rgl.sphline), 8  
rgl.seglat (rgl.sphline), 8  
rgl.seglong (rgl.sphline), 8  
rgl.sphcirc, 3, 4, 4, 10–14  
rgl.sphglobe, 5, 9  
rgl.sphgrid, 3–6, 6, 10–14  
rgl.sphline, 6, 8  
rgl.sphlines (rgl.sphline), 8  
rgl.sphMW, 3–5, 9, 11–14  
rgl.sphpoints, 3–5, 10, 10, 11–14  
rgl.sphsun, 3–5, 10, 11, 11, 13, 14  
rgl.sphtext, 3–5, 10–12, 12, 14  
sph2car, 3–5, 10–13, 13  
sphereplot (sphereplot-package), 2  
sphereplot-package, 2