

# Polyhedra L<sup>A</sup>T<sub>E</sub>X package - v. 0.3

Zafeirakis zafeirakopoulos

November 6, 2023

## Abstract

This package provides macros for creating polyhedral objects in 2D and 3D. It requires tikz and tikz-3dplot. The macros provided can be used for drawing vertices, edges, rays, polygons and cones.

## 1 Introduction

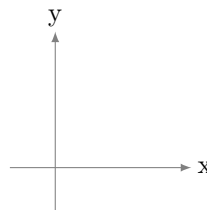
All the drawing commands work with keyword arguments and are the same in 2d and 3D.

## 2 Functionality

### 2.1 The *polyhedron* environment

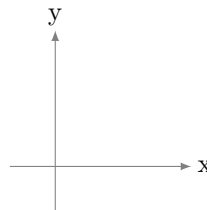
The *polyhedron* environment is essentially a wrapper for a tikzpicture environment. It accepts as extra arguments the dimension of the picture (either 2 or 3) and if the dimension is 3, then the arguments for the three angles defining the rotated coordinate system are given. For the 3D coordinate system, we use tikz-plot3d.

```
\begin{polyhedron}{dim=2}  
  \axes{x={1,3},y={1,3}}  
\end{polyhedron}
```



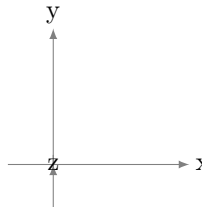
The dimension argument defaults to 2.

```
\begin{polyhedron}{}  
  \axes{x={1,3},y={1,3}}  
\end{polyhedron}
```



Let us define a 3D polyhedron.

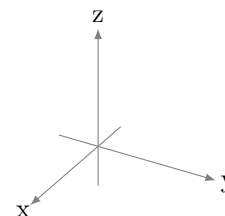
```
\begin{polyhedron}{dim=3}
  \axes{x={1,3},y={1,3},z={1,3}}
\end{polyhedron}
```



Since we did not give the angles for the main coordinate system, we get the default tikz  $xy$ -plane.

Now lets rotate the coordinate system by 60 degrees around the  $x$ -axis and 120 degrees around the  $z$ -axis.

```
\begin{polyhedron}{dim=3,phi=60,theta=120}
  \axes{x={1,3},y={1,3},z={1,3}}
\end{polyhedron}
```



## 2.2 Input

The following macros work within the `polyhedron` environment.

### 2.2.1 Point.

To let A be a point  $(x,y)$  or  $(x,y,z)$

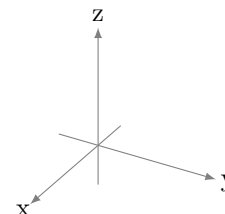
```
\point{(x,y,z)}{A}
```

After defining a point, we can use the handle (variable name) A, instead of writing down the coordinates in the macros that follow.

### 2.2.2 Axes.

We saw already some examples using the `axes` command.

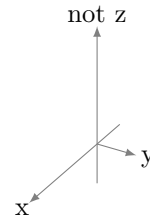
```
\begin{polyhedron}{dim=3,phi=60,theta=120}
  \axes{x={1,3},y={1,3},z={1,3}}
\end{polyhedron}
```



The command takes the following (self-explanatory) arguments. All arguments are optional, defaulting to the ones shown here.

```
x={0,1},y={0,1},z={},xlabel={x},ylabel={y},zlabel={z},labels={true}
```

```
\begin{polyhedron}{dim=3,phi=60,theta=120}
 \axes{x={1,3},z={1,3},zlabel={not z}}
\end{polyhedron}
```



## 2.3 Drawing commands

All drawing commands accept the following options:

**color** The color of the polyhedral object to draw.

**status** The status of the object. The possible values are:

- normal
- focus
- alert
- open

**opacity** The opacity of the object.

**thickness** The width of the lines

**pattern** The draw pattern

### 2.3.1 Vertex

The *vertex* command is used to draw a vertex at a point  $(x,y,z)$  or A. The standard options of color, status, opacity, thickness and pattern apply. The extra options for vertex are:

**point** sets the coordinates of the vertex and it can be either a tikz coordinate tuple or a point already defined.

**text** The text of the vertex label.

**textcolor** The color of the label's text.

**anchora** The label anchor relative to the vertex: north, south, east, west

**anchorb** The text's anchor relative to the label: above, below, right, left

```
\begin{polyhedron}{}
 \vertex{point={0,0},status=normal}
 \vertex{point={1,0},status=alert}
 \vertex{point={2,0},status=open}
 \vertex{point={3,0},color=red,pattern={north west lines}}
 \vertex{point={4,0},color=orange,text={A}}
\end{polyhedron}
```

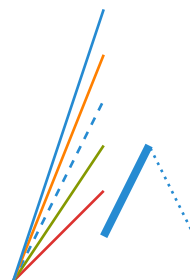


### 2.3.2 Edges and Rays

The command `edge` is used to draw an edge from point A to point B. The extra arguments are:

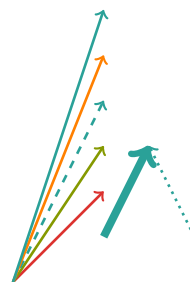
**points** sets the coordinates of the vertices of the edge. It is a list of (two) points in `{}`.

```
\begin{polyhedron}{}
\edge{points={(0,0),(2,2)},status=alert}
\edge{points={(0,0),(2,3)},status=focus}
\edge{points={(0,0),(2,4)},status=open}
\edge{points={(0,0),(2,5)},color=orange}
\edge{points={(0,0),(2,6)},status=normal}
\edge{points={(2,1),(3,3)},thickness=3}
\edge{points={(4,1),(3,3)},pattern={dotted}}
\end{polyhedron}
```



Similarly, the command `ray` draw a ray. The options are exactly the same as for `edge`. Note that the sequence in which the points are given matters.

```
\begin{polyhedron}{}
\ray{points={(0,0),(2,2)},status=alert}
\ray{points={(0,0),(2,3)},status=focus}
\ray{points={(0,0),(2,4)},status=open}
\ray{points={(0,0),(2,5)},color=orange}
\ray{points={(0,0),(2,6)},status=normal}
\ray{points={(2,1),(3,3)},thickness=3}
\ray{points={(4,1),(3,3)},pattern={dotted}}
\end{polyhedron}
```



### 2.3.3 Cone

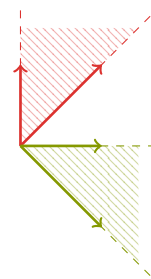
The command `cone` is used to draw a polyhedral cone. The extra arguments are:

**vertex** Sets the apex of the cone.

**generators** A list of points

**openness** A list of boolean values declaring the openness of a simplicial cone.

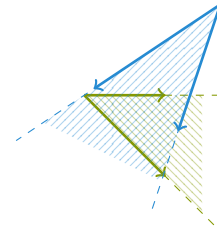
```
\begin{polyhedron}{}
\cone{vertex={(0,0)}, generators={(0,2),(2,2)},
status=alert}
\cone{vertex={(0,0)}, generators={(2,0),(2,-2)},
status=focus}
\end{polyhedron}
```



```

\begin{polyhedron}{}
\cone{vertex={(0,0)}, generators={(2,0),(2,-2)},
      status=focus}
\cone{vertex={(3,2)}, generators={(0,0),(2,-1)},
      pattern={north east lines}}
\end{polyhedron}

```



### 2.3.4 Polygon

The command *polyhedron* is used to draw a 2-dimensional polytope in 2D or 3D. Attention must be paid in the sequence of the points given. The command does not take the convex hull, but rather draws the polygonal line defined by the sequence of points.

The extra arguments are:

**points** The vertices of the cone.

**boundary** True or False, for drawing or not the boundary of the polygon.

```

\begin{polyhedron}{}
\polygon{points={(0,2),(2,2),(2,0)}, status=alert}
\polygon{points={(0,-2),(-2,-2),(-2,0)}, status=open}
\polygon{points={(0,1),(0,0),(1,0),(1,1)}, status=open,
          boundary=false, color=orange}
\end{polyhedron}

```



## 3 Known limitations and errors