The **Axonometric projections** ipelet

`https://github.com/Tomas38/ipe-projections`

Tomáš Drobil

2024-08-24 v1.0

Contents

List of Figures

About

This ipelet makes drawing 3D schemes using proper axonometric projection very easy. The goal of the ipelet is to make possible creating such schemes in Ipe software in an intuitive way without need for other (more complicated) solutions like TikZ.

1 Using the ipelet

With this ipelet, you can transform objects (lines, rectangles, circles and more) into their axonometric projection. You can transform objects into xy, xz or yz plane, or any other plane defined by rotation one of these three planes, see Fig. 1. It might be useful to use shortcuts when doing the projections. How the functions work should be clear from Fig. 2.

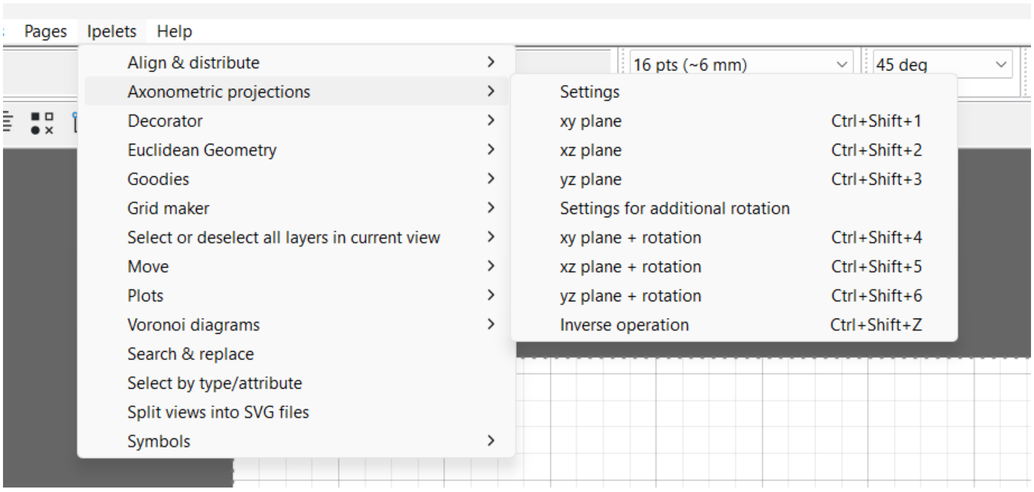


Figure 1: How to access Axonometric projections ipelet fuctions and settings.

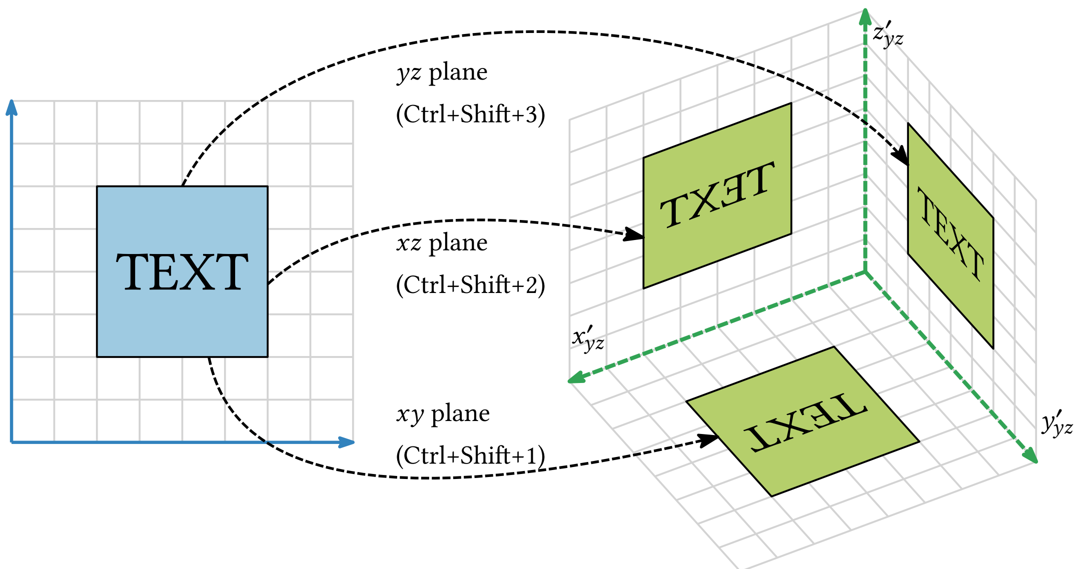


Figure 2: Basic demonstration of the main three projection functions ($\phi = -60^\circ$, $\theta = 40^\circ$, $\phi = 0^\circ$).

The projection without extra rotation is defined by three angles ϕ, θ and ψ representing rotation of the coordinate system along the z, y and x axis, respectively, see Fig. 3. Note that the effect of the last rotation is the same as performing a projection with $\psi = 0^\circ$ and then rotating the result in the drawing plane (i.e. using the Ctrl+R shortcut by the Goodies ipelet). The projection and math involved is described more deeply in the section below.

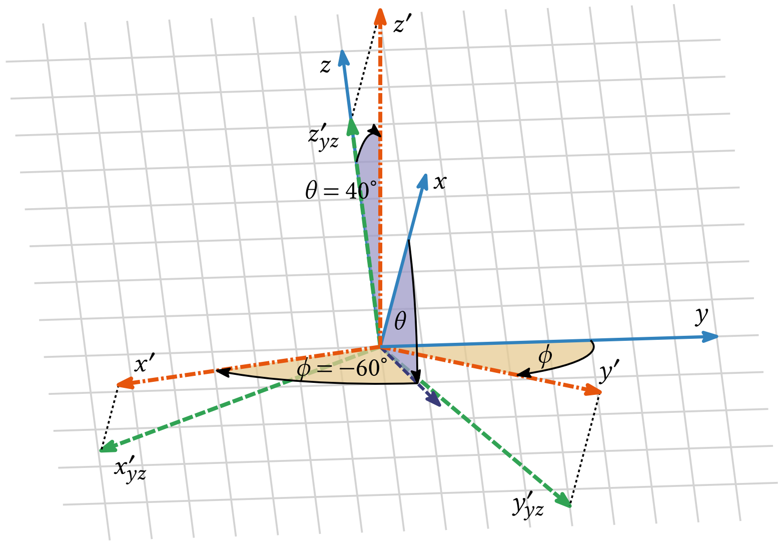


Figure 3: Relation between the coordinate system of the drawing plane (blue) and the coordinate system in which the projected 3D geometry exists (red).

Angles ϕ, θ and ψ can be set using a settings dialog. Values typed here are stored so there is no need for typing angle values before every projection (as this was an issue in the original version of this ipelet). However, whenever the ipe is closed, the values are lost. The dialog is also able to fill the values for a common special case of the axonometric projection – isometric projection.

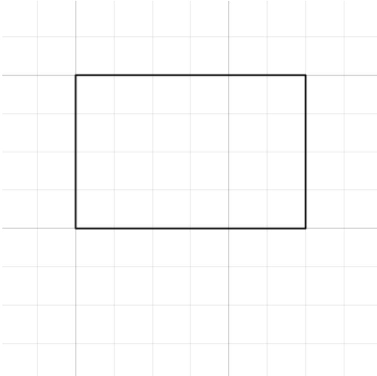
As mentioned above, the ipelet is able to perform a projection in all possible planes. These are described as planes rotated/tilted to xy, yz or xz planes, the "extra rotations" along x, y and z axii are described using angles α, β and γ respectively. These parameters are set in another settings dialog, the order from top to bottom determines the order of rotations and matters as matrix multiplication is not commutative (see the section below).

Performing an inverse operation is possible using the "Inverse operation" option. If the only transformation operation done to an object was using this ipelet, you can also revert the projection simply by removing matrix parameter if you edit the object as XML (right click, Edit as XML).

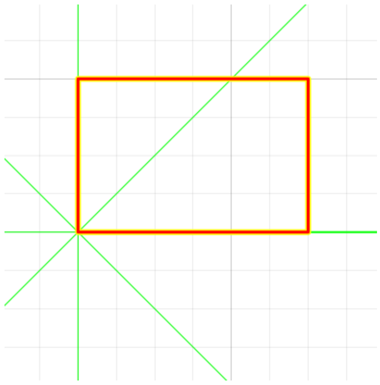
A useful feature might be using the projection on a (math) text, this is possible simply if the text is set as transformable.

2 A simple demo

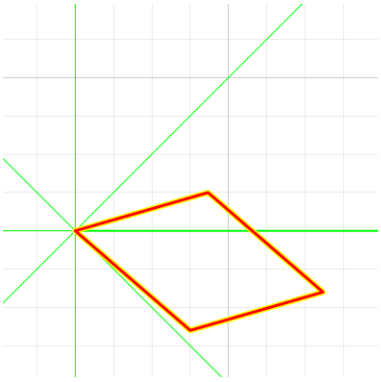
We will demonstrate usage of this ipelet on a simple cuboid with one vertex lying at the origin $(0,0,0)$. First, let's draw the bottom face lying in the xy plane (Fig. 4a). Next, we set the origin point (using F1 shortcut, Fig. 4b), select the rectangle and do the projection onto the xy plane (Fig. 4c) by either pressing Ctrl+Shift+1 or clicking "xy plane" in the menu. Similarly, the side and the back face is created, see Figs. 4d–4g. In the last step, the opposite faces are created using already projected faces by copying and translating these copies in the correct place using snapping on vertexes (Fig. 4h).



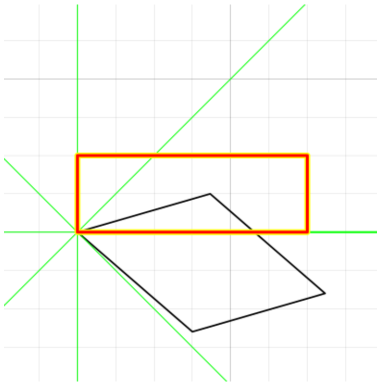
(a) Drawing the bottom face.



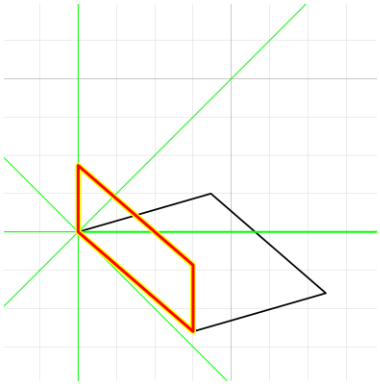
(b) Setting the origin (pressing F1) and selecting the rectangle.



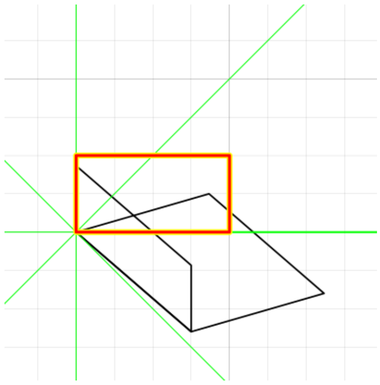
(c) Projection onto the xy plane.



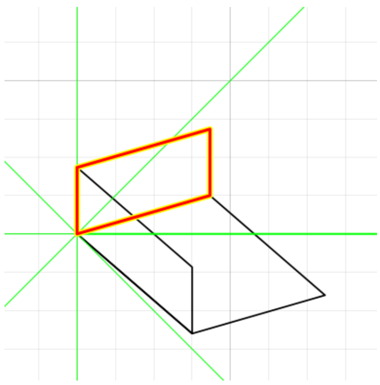
(d) Drawing the side face.



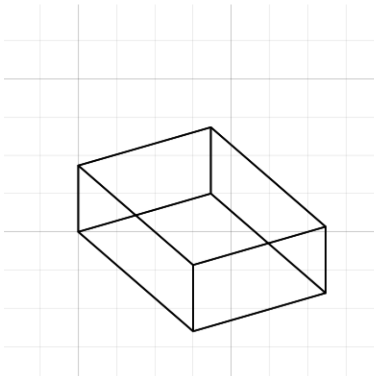
(e) Projection onto the xz plane.



(f) Drawing the last face.



(g) Projection onto the yz plane.



(h) Opposite faces are made using the copy and translate functions.

Figure 4: A simple demo – drawing a cuboid using the default axonometric projection with angles $\phi = 30^\circ, \theta = 30^\circ, \psi = 0^\circ$.

3 Mathematical background

An axonometric projection is a linear projection $\mathbb{R}^3 \to \mathbb{R}^2$, we would like to describe this projection. The $\mathbb{R}^3$ is described using orthonormal basis

$$\vec{\imath} = \begin{pmatrix}1\\0\\0\end{pmatrix}, \quad \vec{\jmath} = \begin{pmatrix}0\\1\\0\end{pmatrix}, \quad \vec{k} = \begin{pmatrix}0\\0\\1\end{pmatrix}. \tag{1}$$

Let's choose the yz plane as the $\mathbb{R}^2$ (we would like to have the z axis pointing upwards and the x axis pointing out of the drawing plane). Now the projection of an object projected in defined $\mathbb{R}^2$ would be simply the projection in the yz plane, i.e.

$$\begin{pmatrix}y'\\z'\end{pmatrix} = \begin{pmatrix}0 & 1 & 0\\0 & 0 & 1\end{pmatrix} \cdot \begin{pmatrix}x\\y\\z\end{pmatrix} = \begin{pmatrix}y\\z\end{pmatrix}. \tag{2}$$

If we would like to create an actual axonometric projection (and not that simple degenerate case), we have to "look at the coordinate system with the object from a different direction". This can be simply achieved by rotating the coordinate system including the object before the $\mathbb{R}^3 \to \mathbb{R}^2$ projection, which can be done using rotation matrices

$$R_x(\omega) = \begin{pmatrix}1 & 0 & 0\\0 & \cos(\omega) & -\sin(\omega)\\0 & \sin(\omega) & \cos(\omega)\end{pmatrix}, \tag{3}$$

$$R_y(\omega) = \begin{pmatrix}\cos(\omega) & 0 & \sin(\omega)\\0 & 1 & 0\\-\sin(\omega) & 0 & \cos(\omega)\end{pmatrix}, \tag{4}$$

$$R_z(\omega) = \begin{pmatrix}\cos(\omega) & -\sin(\omega) & 0\\\sin(\omega) & \cos(\omega) & 0\\0 & 0 & 1\end{pmatrix}, \tag{5}$$

which rotate along x, y and z axis, respectively. In order to keep the direction of the z axis, we will rotate first along the z axis by angle ϕ and than along the y axis by angle θ, see Fig. 3 and Fig. 5. If we don't mind keeping the direction of the z axis, we can additionally rotate along the x axis by angle ψ.

If we would like to to make a projection of planes tilted to xy, xz or yz plane, we can do so by rotating the coordinate system before performing rotations described in the previous paragraph. We define angles α, β and γ describing rotation along the x, y and z axes respectively. When doing extra rotations, user can define three matrices

$$R_1 = \begin{cases}R_x(\alpha)\\R_y(\beta)\\R_z(\gamma)\end{cases}, \quad R_2 = \begin{cases}R_x(\alpha)\\R_y(\beta)\\R_z(\gamma)\end{cases}, \quad R_3 = \begin{cases}R_x(\alpha)\\R_y(\beta)\\R_z(\gamma)\end{cases}. \tag{6}$$

Then, we can define a matrix describing the whole transformation using all the rotations mentioned

$$M = \begin{cases}R_x(\psi) \cdot R_y(\theta) \cdot R_z(\phi) & \text{for default projection}\\R_x(\psi) \cdot R_y(\theta) \cdot R_z(\phi) \cdot R_3 \cdot R_2 \cdot R_1 & \text{for additional rotation}\end{cases}. \tag{7}$$

The transformation using the matrix M on any given point can be written as

$$\begin{pmatrix}x'\\y'\\z'\end{pmatrix} = M \cdot \begin{pmatrix}x\\y\\z\end{pmatrix}. \tag{8}$$

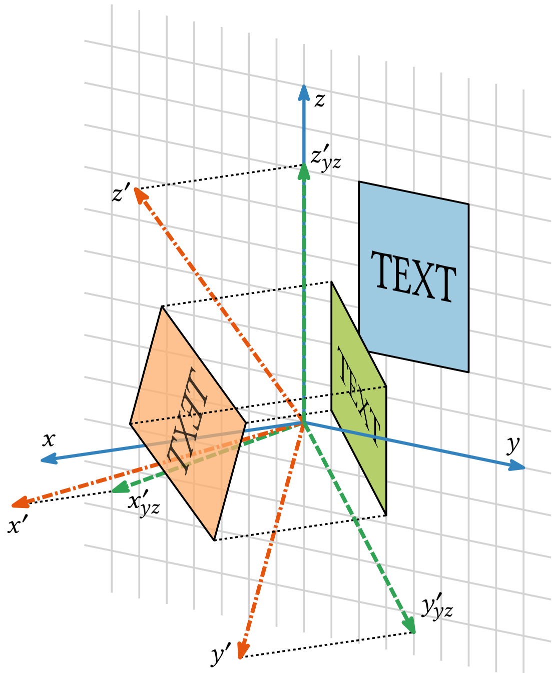


Figure 5: The original coordinate system (blue) is first rotated in the 3D space (red) and this new system is then projected onto the yz plane – drawing plane (green). In this case, yz plane projection is demonstrated.

Let's label the matrix elements

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} = \begin{pmatrix} \vec{\imath}_M & \vec{\jmath}_M & \vec{k}_M \end{pmatrix}, \tag{9}$$

$\vec{\imath}_M$, $\vec{\jmath}_M$ and $\vec{k}_M$ are basis vectors of the transformed coordinate system (they are transformed basis vectors in different words). The $\mathbb{R}^3 \to \mathbb{R}^2$ linear projection onto the yz plane can be thus written as

$$\begin{pmatrix} 0 \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \tag{10}$$

or simply

$$\begin{pmatrix} y' \\ z' \end{pmatrix} = \begin{pmatrix} m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \tag{11}$$

we will label vectors in the matrix as

$$\begin{pmatrix} m_{21} \\ m_{31} \end{pmatrix} = \vec{\imath}_{M,yz}, \quad \begin{pmatrix} m_{22} \\ m_{32} \end{pmatrix} = \vec{\jmath}_{M,yz}, \quad \begin{pmatrix} m_{23} \\ m_{33} \end{pmatrix} = \vec{k}_{M,yz}, \tag{12}$$

$\vec{\imath}_{M,yz}$, $\vec{\jmath}_{M,yz}$ and $\vec{k}_{M,yz}$ are projections of the transformed basis vectors. Using these vectors, the actual $\mathbb{R}^2 \to \mathbb{R}^2$ transformation in Ipe can be done. Ipe uses for transformations matrix defined as

$$M_{\mathrm{Ipe}} = \begin{pmatrix} a & c & s \\ b & d & t \end{pmatrix}, \tag{13}$$

so coordinate values of an object are transformed as

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & c & s \\ b & d & t \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} ax + cy + s \\ bx + dy + t \end{pmatrix} \tag{14}$$

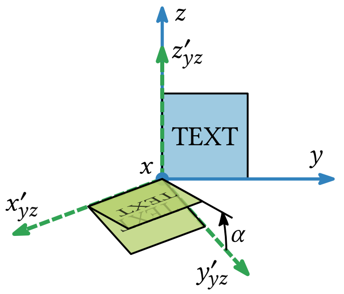
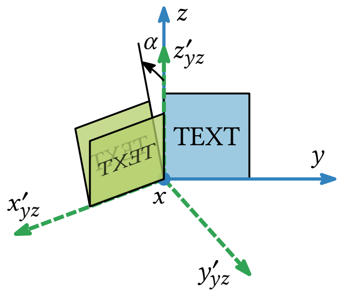
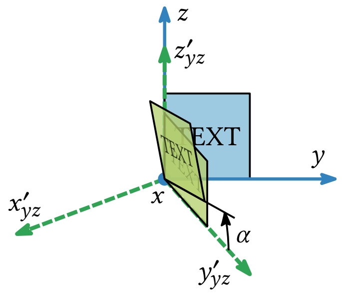
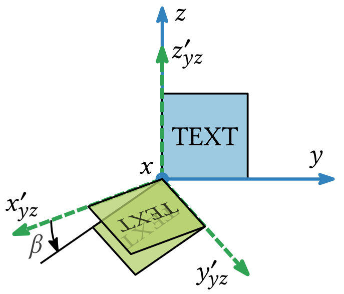
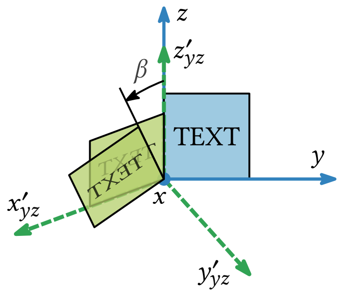
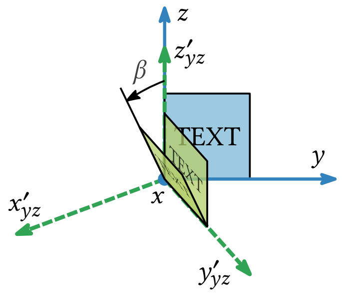
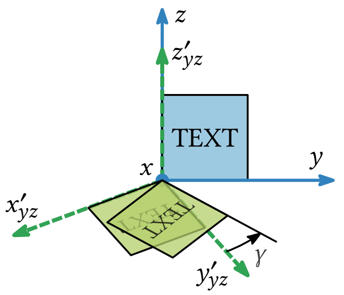
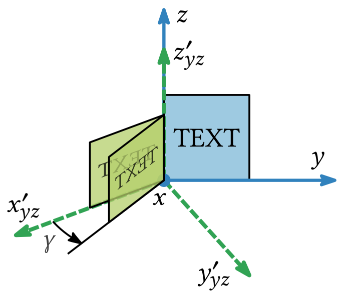
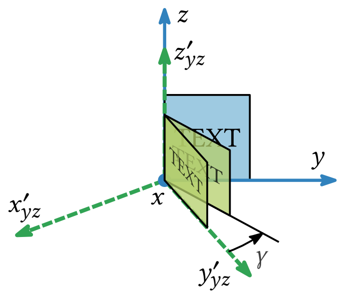
(now x, y, x', y' refers to the 2D coordinates of the Ipe drawing plane).

The ipe transformation matrix is then finally obtained as

$$M_{\text{Ipe,axonometric}} = \begin{cases} \begin{pmatrix} \vec{\imath}_{M,yz} & \vec{\jmath}_{M,yz} & \vec{0} \end{pmatrix} = \begin{pmatrix} m_{21} & m_{22} & 0 \\ m_{31} & m_{32} & 0 \end{pmatrix} & \text{for } xy \text{ plane,} \\ \begin{pmatrix} \vec{\imath}_{M,yz} & \vec{k}_{M,yz} & \vec{0} \end{pmatrix} = \begin{pmatrix} m_{21} & m_{23} & 0 \\ m_{31} & m_{33} & 0 \end{pmatrix} & \text{for } xz \text{ plane,} \\ \begin{pmatrix} \vec{\jmath}_{M,yz} & \vec{k}_{M,yz} & \vec{0} \end{pmatrix} = \begin{pmatrix} m_{22} & m_{23} & 0 \\ m_{32} & m_{33} & 0 \end{pmatrix} & \text{for } yz \text{ plane.} \end{cases} \tag{15}$$

Appendix A: Extra rotations

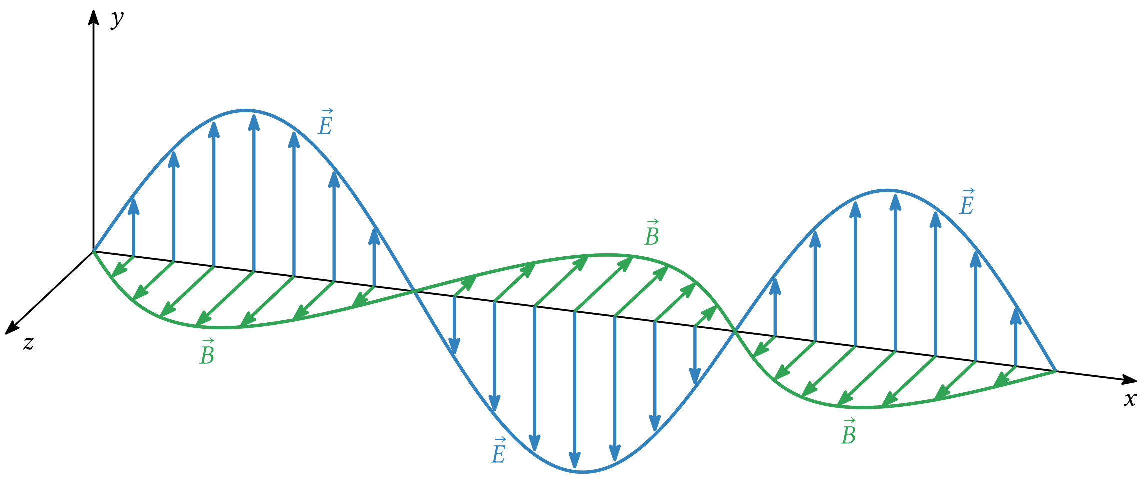
The effect of extra rotation is demonstrated below for extra rotation around one axis at a time.

xy plane	xz plane	yz plane
		
		
		

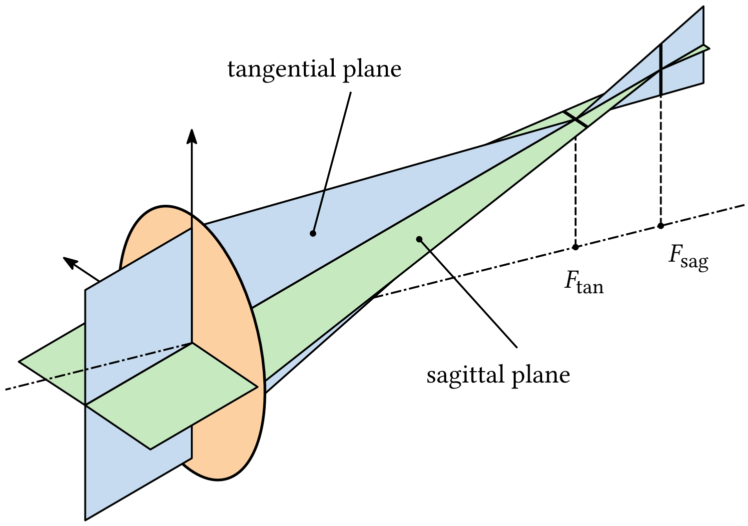
Appendix B: Examples of use

A few examples are presented below to illustrate some possible use of this ipelet.

3.1 Electromagnetic wave



3.2 Optical abberation



3.3 Cylindrical coordinates