

6 – Solides de l'espace

On a déjà vu comment tracer des courbes avec `Pstricks`, ou construire des petites figures de géométrie. On va encore utiliser `Pstricks` pour dessiner des solides dans l'espace.

Et comme on utilise `Pstricks`, on ajoute `\usepackage{pstricks-add}` dans le préambule.

1 Cube

Voici les instructions qui permettent de réaliser un cube :

```
\psset{xunit=0.8cm, yunit=0.8cm}
\def\xmin{0} \def\xmax{10} \def\ymin{0} \def\ymax{10}
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
\pspolygon[showpoints=true](1,7)(7,7)(9,9)(3,9)
\psline[showpoints=true](1,7)(1,1)(7,1)(9,3)(9,9)
\psline(7,1)(7,7)
\psline[linestyle=dashed](1,1)(3,3)(3,9)
\psline[linestyle=dashed, showpoints=true](3,3)(9,3)
\uput[d1](1,1){$A$} \uput[dr](7,1){$B$}
\uput[r](9,3){$C$} \uput[ul](3,3){$D$}
\uput[ul](1,7){$E$} \uput[100](7,7){$F$}
\uput[ur](9,9){$G$} \uput[ul](3,9){$H$}
\end{pspicture}
```

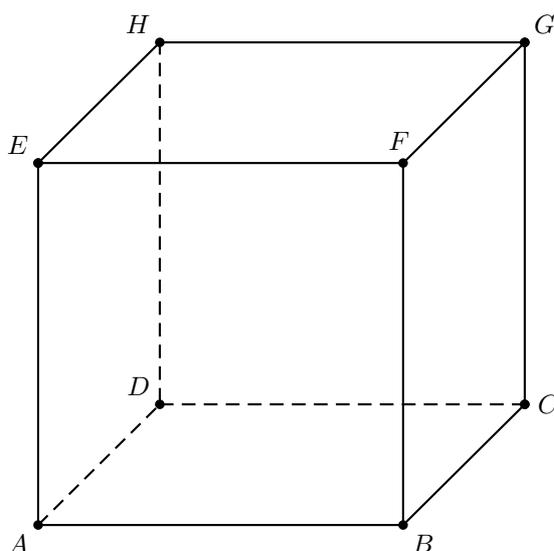
L'instruction `\pspolygon`, comme son nom l'indique, trace un polygone fermé.

L'option `[showpoints=true]` dessine les points aux sommets du polygone.

Par défaut, les points sont représentés par des petits disques dont on peut modifier la taille ; si on veut changer l'aspect des points, on peut entrer dans `\psset` l'instruction `dotstyle=+` pour avoir une petite croix, ou `dotstyle=x` pour avoir une autre croix comme symbole du point.

L'option `[showpoints=true]` peut être rajoutée si on trace un segment avec `\psline`, ou une courbe avec `\pscurve`. Pour ne pas tracer les points, il suffit de ne pas entrer cette option.

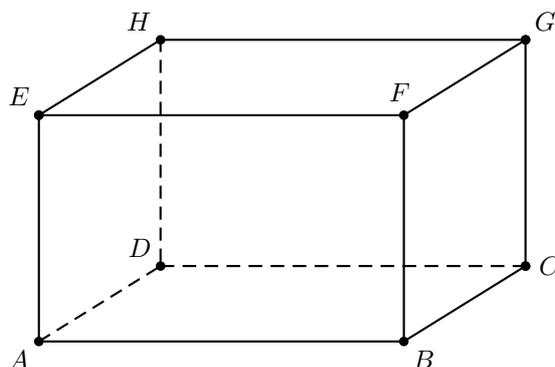
Cela donne un cube assez convenable :



2 Parallélépipède

Il faut changer peu de choses au cube pour en faire un parallélépipède rectangle : on réduit l'unité en x ou en y et le tour est joué.

En entrant `\psset{xunit=0.8cm, yunit=0.5cm}` comme première ligne d'instruction à la place de celle qui s'y trouve, on obtient un magnifique parallélépipède rectangle :



Et si on veut un parallélépipède non rectangle, c'est possible? Bien sûr!

On va utiliser l'instruction `\pstilt` qui a besoin de deux paramètres : le premier représente l'angle par rapport à l'horizontale (donc si on met 90 , il ne se passe rien), et en second paramètre ce que l'on veut « tilter ».

En tapant `\pstilt{60}{LaTeX}` on obtient *LaTeX* (angle de 60°).

En tapant `\pstilt{90}{LaTeX}` on obtient LaTeX (aucun changement!).

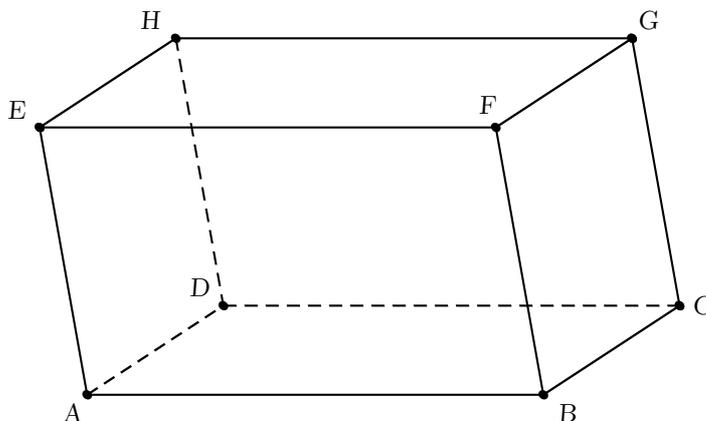
En tapant `\pstilt{120}{LaTeX}` on obtient \sXeX (angle de 120°).

L'instruction `\pstilt` fait partie de l'extension `pst-3d`, qu'il est inutile de charger car elle est incluse dans l'extension `pstricks-add` (alors qu'elle ne l'était pas dans l'extension `pstricks`).

En entrant :

```
\psset{xunit=1cm, yunit=0.6cm}
\def\xmin{0} \def\xmax{10}
\def\ymin{0} \def\ymax{10}
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
\pstilt{100}
{
    % début de ce que l'on veut incliner
\pspolygon[showpoints=true](1,7)(7,7)(9,9)(3,9)
\psline[showpoints=true](1,7)(1,1)(7,1)(9,3)(9,9)
\psline(7,1)(7,7)
\psline[linestyle=dashed](1,1)(3,3)(3,9)
\psline[linestyle=dashed, showpoints=true](3,3)(9,3)
\uput[dl](1,1){$A$} \uput[dr](7,1){$B$}
\uput[r](9,3){$C$} \uput[ul](3,3){$D$}
\uput[ul](1,7){$E$} \uput[100](7,7){$F$}
\uput[ur](9,9){$G$} \uput[ul](3,9){$H$}
}
% fin du \pstilt
\end{pspicture}
```

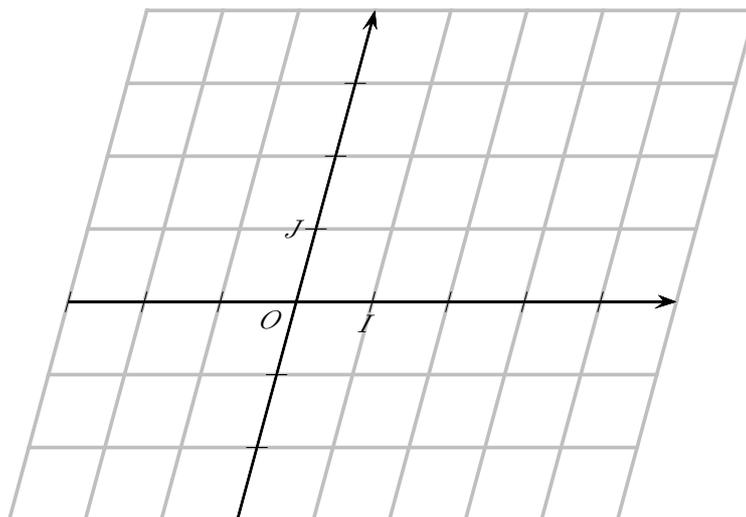
on obtient :



Attention : tout ce qui est entre les deuxièmes accolades du `\pstilt` sera incliné, y compris les noms des points ; il ne faut donc pas donner un angle trop important, sinon l'ensemble devient assez disgracieux !

3 Repère oblique

Autre application du `\pstilt` : construire un repère oblique.

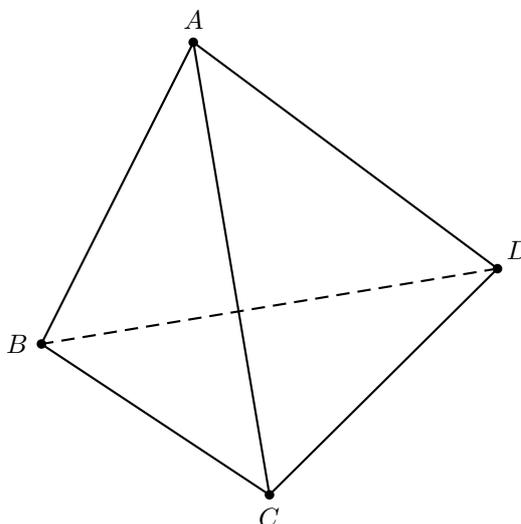


qui s'obtient en entrant :

```
\psset{xunit=1cm,yunit=1cm}
\def\xmin{-3} \def\xmax{5} \def\ymin{-3} \def\ymax{4}
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
\pstilt{75}{
\psgrid[gridlabels=0,gridwidth=0.5mm,subgriddiv=0,gridcolor=lightgray]
\psaxes[labels=none,arrowsize=3pt 3]{->}(0,0)(\xmin,\ymin)(\xmax,\ymax)
\uput[d](1,0){$I$} \uput[l](0,1){$J$}
\uput[-135](0,0){$O$}
}
\end{pspicture}
```

4 Tétraèdre

Enfin, comme il n'y a pas que le cube dans l'espace, voici un tétraèdre :



et son code :

```
\psset{xunit=1cm, yunit=1cm}
\def\xmin{0} \def\xmax{8} \def\ymin{0} \def\ymax{8}
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
\psline(3,7)(1,3)(4,1)(3,7)(7,4)(4,1)
\psline[linestyle=dashed](7,4)(1,3)
\point{3}{7}{u}{A$}
\point{7}{4}{ur}{D$}
\point{1}{3}{l}{B$}
\point{4}{1}{d}{C$}
\end{pspicture}
```

À cette occasion, on peut voir une commande personnelle appelée `\point`, qui place un point et écrit son nom en même temps ; elle nécessite quatre paramètres : l'abscisse du point, son ordonnée, la position du nom et le nom lui-même. On entre les paramètres entre des accolades.

```
\newcommand*\point}[4]
{
\psdots(#1,#2)
\uput[#3](#1,#2){#4}
}
% permet de placer un point et de marquer son nom en même temps
% paramètres : abscisse, ordonnée, emplacement et nom
```

On peut comprendre aisément le fonctionnement de cette commande qui ne fait qu'appliquer les deux commandes `\psdots` et `\uput`.

Elle est à copier dans le préambule du document.

J'ai l'habitude de commenter mes commandes, c'est-à-dire d'expliquer ce qu'elles font ; je vous conseille de faire la même chose car on oublie vite, surtout si on ne les utilise pas tous les jours !