

GÉOMÉTRIE DU PLAN

1 Du texte...

Exercice 1 (`\overrightarrow` / `\vv`).

1. A l'aide des menus de votre éditeur et/ou de l'aide de l'éditeur, donnez le code de : \overrightarrow{AB} .
2. Comparer le résultat avec l'instruction `\vv{AB}` du package `esvect`. Ce package offre quelques variantes pour les flèches de vecteurs. Elles sont plus épaisses, ce qui est utile pour les photocopies.

Exercice 2 (carré scalaire).

À l'aide des menus de votre éditeur, codez :

1.

$$\|\overrightarrow{AB}\|^2 = \overrightarrow{AB}^2 = |z_B - z_A|^2$$

2.

$$\left(\overrightarrow{AB}, \overrightarrow{AC}\right) \equiv \alpha \pmod{2\pi}$$

Exercice 3 (caractère fantôme).

Dans une classe où l'on découvre les vecteurs, les élèves doivent compléter :

$$\overrightarrow{AB} = \overrightarrow{\dots D} = \overrightarrow{\dots}$$

Quel problème pose le simple code suivant : `\vv{AB}=\vv{\ldots D}=\vv{\ldots\ldots}` ?

On cherchera à le régler en recherchant le fonctionnement de `\vphantom` par exemple dans la FAQ <http://www.faqs.org/faqs/fr/faq-latex-francaise/part2/>.

Exercice 4.

Définir un raccourci appelé `\oi j` qui a pour effet d'afficher : (O, \vec{i}, \vec{j})

Rappel de la syntaxe : `\newcommand{raccourci}{contenu à afficher}`

Exercice 5.

Définir une commande `\repere` telle que

- `\repere{0}{i, j}` affiche (O, \vec{i}, \vec{j})
- `\repere{0}{i, j, k}` affiche $(O, \vec{i}, \vec{j}, \vec{k})$.

On pourra utiliser la boucle `foreach` du package `pgffor`. Elle est documentée dans le `pgf manual`. On trouvera facilement le `pgf manual` sur le web en tapant « tex catalogue pgf » dans google.

Essayer cet exemple pour en comprendre la syntaxe :

`\foreach \x in {1,2,3} {\$x =\x$, }`



2 ...et des figures

2.1 Les outils disponibles

2.1.1 Les packages

Il existe pour les principaux packages pour le dessin des extensions orientées géométrie euclidienne (c'est à dire contenant des instructions du type « dessiner la droite perpendiculaire à la droite ...et passant par le point ... » ou « dessiner l'image du point A par la rotation de centre B et d'angle $\frac{\pi}{3}$ » ...)

1. Le package `tikz`. La syntaxe pour des figures élémentaires est facile à apprendre. On trouvera sur le site d'Alain Matthes une extension pour la géométrie euclidienne : `tkz-2d`.

<http://altermundus.fr/pages/download.html>

2. `METAPOST`, avec `geometriesyr` de Christophe Poulain pour la partie géométrie euclidienne. On trouvera `geometriesyr` et de nombreux exemples à l'adresse :

<http://melusine.eu.org/syracuse/poulecl/essaimphtml/>.

Notons que l'auteur met aussi à disposition à l'adresse <http://melusine.eu.org/lab/brevet> les annales du brevet des collèges de ces dernières années. Les nombreux dessins de géométrie de ces annales sont donc autant d'exemples pour se former à `metapost` et aux macros de C.Poulain.

3. `PSTricks`, avec `pst-eucl` de D.Rodriguez pour la partie géométrie euclidienne. On trouvera `pst-eucl` à l'adresse :

<http://texcatalogue.sarovar.org/entries/pst-eucl.html>.

Le package `pst-eucl` étant une extension de `PSTricks`, toutes les commandes `PSTricks` sont utilisables.

Notons qu'un collègue (Denis Vergès) met à disposition les annales du bacc de ces dernières années.

On les retrouvera en tapant « `apmep annales` » dans google.

L'outil utilisé pour les représentations graphiques est `PSTricks`. On a donc là de nombreux exemples sous la main...

4. `Asymptote`. On pourra consulter le site de Philippe Ivaldi :

<http://www.piprime.fr/asymptote/>

2.1.2 Avec des interfaces

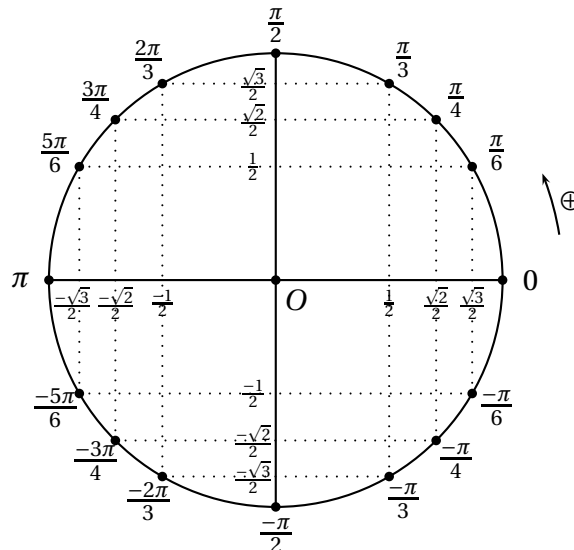
1. Pour débuter en douceur, vous pourrez aussi utiliser les possibilités d'export de `geogebra` (code `pstricks`, `tikz`, image `eps`, image `png`).
2. Vous pourrez jeter un oeil également à ce que propose le logiciel `pst+` :

<http://www.xmlmath.net/pstplus/index.html>

2.2 Dessinons

Exercice 6.

On aimerait représenter le cercle trigo ci-dessous :



On vous propose un bout de code de ce cercle, le compléter :

```
% pour centrer l'image dans la colonne de texte :
\begin{center}
% unité :
\psset{unit=3cm}
% début figure, déf du cadre la contenant :
\begin{pspicture}(-1,-1)(1,1)
% \pstGeonode : instruction du package pst-eucl
% \pstGeonode(coordonnées){nom du point} : dessine le point et place son nom.
% \pstGeonode[PosAngle=angle en degrés pour placer le nom du point
% par rapport à ce point (vu comme un cercle trigo)](coordonnées){nom du point}
% point O centre du cercle :
\pstGeonode[PosAngle=-45](0,0){O}
% point I de coord (1,0) :
\pstGeonode[PointName=0](1,0){I}
% le cercle de centre O passant par I (instruction de pst-eucl):
\pstCircleOA{O}{I}
% la mesure correspondant au point d'abscisse curviligne pi/6 :
\pstGeonode[PosAngle=30,PointName=\frac{\pi}{6}](1;30){A}
% tracé des deux "diamètres principaux" :
\pstGeonode[PosAngle=180,PointName=\pi](1;180){I'}
\pstGeonode[PosAngle=90,PointName=\frac{\pi}{2}](1;90){J}
\pstGeonode[PosAngle=-90,PointName=\frac{-\pi}{2}](1;-90){J'}
\pstLineAB{I'}{I}\pstLineAB{J'}{J}
% lignes pointillées de "projection" sur les axes :
\pstGeonode[PosAngle=-30,PointName=\frac{-\pi}{6}](1;-30){A'}
\pstLineAB[linestyle=dotted]{A}{A'}
\begin{scriptsize}
% abscisses :
\rput(0.866,-0.1){$\frac{\sqrt{3}}{2}$}
\end{scriptsize}
% flèche d'orientation trigo :
\pstGeonode[PointSymbol=none,PointName=none](1.25,0.2){P}
\pstGeonode[PointSymbol=oplus,PointName=none,dotscale=2](1.3,0.35){R}
```



```

\pstGeonode[PointSymbol=none,PointName=none](1.25,0.5){Q}
\pstArcOAB[arrows=->]{O}{P}{Q}
\end{pspicture}
\end{center}

```


Exercice 7.

Le code PSTricks :

```


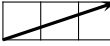
% unité de base :
\psset{unit=0.5cm}
% figure dans le rectangle de sommets opposés (0,0) et (2,2) :
\begin{pspicture}(0,0)(2,2)
% définition de la grille (quadrillage) :
\psgrid[gridlabels=0,gridwidth=0.4pt,subgriddiv=0](0,0)(2,2)
% représentation du vecteur :
\psline[linewidth=1.5\pslinewidth]{->}(0,0)(2,2)
\end{pspicture}

```

donne : 

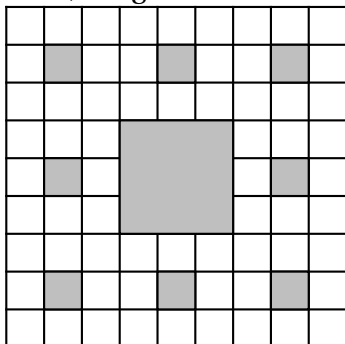
À l'aide de cet exemple, codez l'énoncé d'exercice ci-dessous puis donnez le code d'une figure servant de corrigé à l'exercice :

« Reproduire (à l'aide du quadrillage de votre

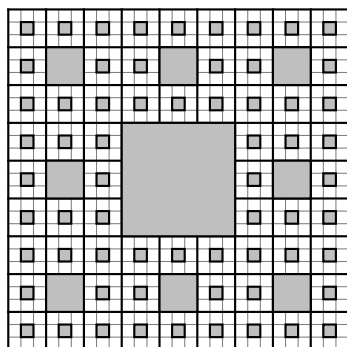
feuille) le vecteur \vec{i}  et le vecteur \vec{j} 
 puis représenter le vecteur $2\vec{i} - 3\vec{j}$. »

Exercice 8.

Sur le site de l'apmep de Bordeaux (<http://www.ac-bordeaux.fr/APMEP/>), on trouve codé en \LaTeX des annales du bac. On trouve notamment, dans le sujet de terminale L de Nouvelle-Calédonie de novembre 2005, la figure suivante :



Charger le code de la figure sur le site de l'apm, puis modifier ce code pour produire l'étape suivante de notre figure fractale :

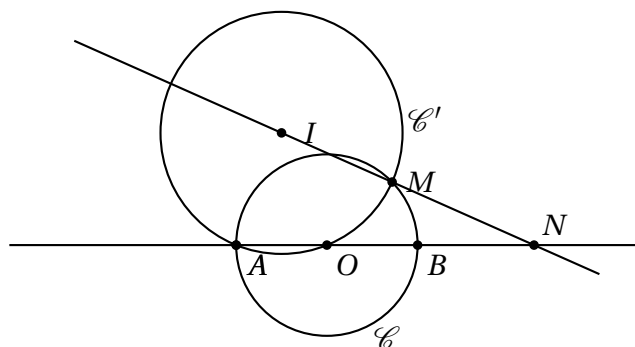


On trouvera le détail sur l'instruction de boucle multido à l'adresse suivante :
<http://texcatalogue.sarovar.org/entries/multido.html>.

Exercice 9.

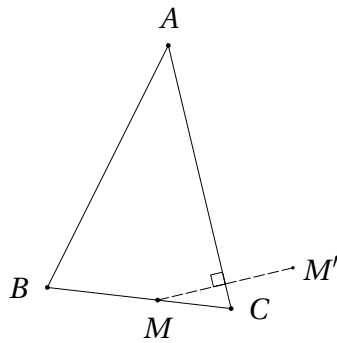
Codez la figure suivante en consultant la documentation de `pst-eucl`. Vous veillerez à donner un code tel qu'un simple changement final de coordonnées du point A ou du point B ne rende pas la figure fautive (c'est à dire en construisant la figure par ses propriétés géométriques plutôt que par des "positions absolues" des éléments de la figure).

On considère un cercle \mathcal{C} de centre O et de rayon 6. Soit A un point de \mathcal{C} et B le point de \mathcal{C} diamétralement opposé à A . Soit \mathcal{C}' un cercle, de centre un point I , de rayon 8 et passant par A et O . On nomme M le deuxième point d'intersection de \mathcal{C} et \mathcal{C}' . Les droites (IM) et (AB) se coupent en N .



Exercice 10. 1. Obtenir le code `pstricks` de la figure ci-dessous à l'aide de `geogebra`. Dans cette figure, ABC est un triangle isocèle en A . M est un point du segment $[BC]$, M' est l'image de M par la réflexion d'axe (AC) .

2. Coder la même figure à l'aide des instructions du package `pst-eucl`.



L'intérêt de définir la figure avec geogebra est bien sûr de la définir par ses propriétés géométriques invariantes...mais geogebra ne s'appuie pas sur pst-eucl pour le codage pstricks de la figure et on perd donc malheureusement la logique de la construction en exportant.



3 Des solutions

Exercice 1 ($\overrightarrow{\quad}$ / $\vec{\quad}$).

\overrightarrow{AB} est obtenu par \overrightarrow{AB} . On obtient \vec{AB} avec :

```
% préambule :
\usepackage{esvect}
% dans le texte :
$\vec{AB}$
```

Exercice 2 (carré scalaire).

```
\[ \left\| \overrightarrow{AB} \right\|^2
= \left\| z_B - z_A \right\|^2 \]
```

On observera la différence entre \overrightarrow{AB}^2 et $\|\overrightarrow{AB}\|^2$.

```
\[ \widehat{\left( \vec{AB}, \vec{AC} \right)} \equiv \alpha \pmod{2\pi} \]
```

Exercice 3 (caractère fantôme).

On peut écrire :

```
\[ \vec{AB} = \vec{\phantom{A} \dots D} = \vec{\phantom{A} \dots \phantom{D}} \]
```

La commande éfinit une boîte invisible dont la hauteur est celle de son argument.

Exercice 4.

Définition la plus simple :

```
1. \newcommand{\Oij}{\ensuremath{\left( O; \vec{i}, \vec{j} \right)}}
```

La commande `\ensuremath{}` permet d'utiliser `\Oij` dans un environnement maths comme en dehors d'un tel environnement.

2. Raffinons un peu :

```
\usepackage{ifthen} % pour la commande ifthenelse
\newcommand{\Oij}[1][ij]
{\ifthenelse
{\equal{#1}{ij}}
{\ensuremath{\left( O; \vec{i}, \vec{j} \right)}}
{\ifthenelse{\equal{#1}{uv}}
{\ensuremath{\left( O; \vec{u}, \vec{v} \right)}}
{\rep\`ere inconnu}}}
```

Avec cette nouvelle définition, que se passe-t-il avec les appels suivants : `\Oij`, `\Oij[uv]`, `\Oij[fk]` ?

3. Une autre proposition utilisant cette fois les définitions de macros en \TeX :

```
\def \reper(#1,#2,#3)%
{\ensuremath
{\left( #1 ; \overrightarrow{#2} , \overrightarrow{#3} \right)}}
}
```



qu'on utilisera ainsi : `\reper(O,\imath,\jmath)` ou `\reper(A,AB,AC)` qui donneront :

$$(O; \vec{i}, \vec{j}) \quad \left(A; \overrightarrow{AB}, \overrightarrow{AC} \right)$$

\LaTeX étant basé sur \TeX , les macros \TeX fonctionnent sous \LaTeX . On les utilisera avec précaution notamment à cause du comportement suivant :

- si une macro `\reper` est déjà définie, `\newcommand{\reper}` sera refusé à la compilation, cela évite d'écraser des commandes déjà définies (par exemple celles qui se trouvent dans les nombreux packages auxquels votre préambule fera appel),
- si une macro `\reper` est déjà définie, `\def \reper` se contentera d'écraser l'ancienne définition pour la remplacer par la nouvelle.

En \LaTeX , si on veut écraser une commande déjà existante, on écrira `\renewcommand{...}`.

Exercice 5.

```
1. \usepackage{pgffor}
   \usepackage{esvect}
   \newcommand*{\reper}[2]{
   \ensuremath{\left( #1
   \foreach \Vecteur in {#2} { ,\vv{\Vecteur}} \right) }}
```

Essayez avec `\reper{O}{i,j,k}`, `\reper{O}{AB,AC,AD}`, `\reper{O}{\imath,\jmath,k}`.

Une remarque sur des petits problèmes possibles même dans le monde \LaTeX . Redéfinir la macro précédente en :

```
\usepackage{pgffor}
\usepackage{esvect}
\newcommand*{\reper}[2]{
\ensuremath{\left( #1
\foreach \vecteur in {#2} { ,\vv{\vecteur}} \right) }}
```

(on a seulement remplacé `\Vecteur` par `\vecteur`)
 puis utiliser la ...

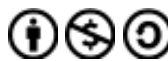
Explication : le package `esvect` définit une commande `\vecteur` et `pgffor` ne prend pas la peine de vérifier si la variable de boucle existe ou non. D'où un cafouillage entre `\vecteur` utilisé en variable de boucle et `\vecteur` utilisé dans `esvect`.

2. Autre proposition :

```
\makeatletter
\newcommand*{\reper}[2]{%
  \ensuremath{ \left( #1%
    \@for \basevecteur :=#2\do { , \overrightarrow{\basevecteur}%
    } \right) }}
\makeatother
```

Essayez avec : `\reper{O}{i,j}` puis avec `\reper{O}{i,j,k}`.

On trouvera le descriptif et la définition de `\@for` dans le fichier `source2e.tex`. La syntaxe de `\@for` est la suivante : `\@for <command>:=<list>\do{<code>}` où les éléments de la liste sont séparés par des virgules. C'est une commande « interne » du format \LaTeX .

**Une subtilité de $\@for$.**

Si dans un énoncé, on utilise beaucoup le repère $(A, \overrightarrow{AB}, \overrightarrow{AC}, \overrightarrow{AD})$, on pourra être tenté de définir :

$\newcommand*\rep}{AB,AC,AD}$ puis de faire appel à la macro \reper définie ci-avant avec $\reper{A}{\rep}$. Pas de problème, ça fonctionne. Mais maintenant, changez la définition de \reper en :

```
\makeatletter
\newcommand*\reper}[2]{%
  \ensuremath{\left( #1%
    \@for \basevecteur := #2\do { , \overrightarrow{\basevecteur}%
    } \right)}}
\makeatother
```

(on a juste ajouté un espace entre $:=$ et $\#2$). L'appel $\reper{A}{\rep}$ donne alors $(A, \overrightarrow{AB}, \overrightarrow{AC}, \overrightarrow{AD}) \dots$

Explication (schématique) : $\@for$ développe le token qui suit $:=$. Dans le premier cas, $\@for$ développe donc la commande \rep qui est ensuite interprété comme une liste de trois éléments. Dans le second cas $\@for$ développe le token "espace" (et donc ne fait rien) puis considère les éléments de la commande \reper d'un bloc (liste d'un seul élément, la commande n'étant pas développée les virgules ne sont pas vues et ne constituent plus ici des séparateurs d'élément de liste).

Exercice 6.

$\usepackage{pst-eucl}$.

```
\begin{center}
\psset{unit=3cm}
\begin{pspicture}(-1,-1)(1,1)
% point O centre du cercle :
\pstGeonode[PosAngle=-45](0,0){O}
% point I de coord (1,0) :
\pstGeonode[PointName=0](1,0){I}
% le cercle :
\pstCircleOA{O}{I}
% les angles "remarquables" :
\pstGeonode[PosAngle=30,PointName=\frac{\pi}{6}](1;30){A}
\pstGeonode[PosAngle=45,PointName=\frac{\pi}{4}](1;45){B}
\pstGeonode[PosAngle=60,PointName=\frac{\pi}{3}](1;60){C}
\pstGeonode[PosAngle=90,PointName=\frac{\pi}{2}](1;90){J}
\pstGeonode[PosAngle=120,PointName=\frac{2\pi}{3}](1;120){C_1}
\pstGeonode[PosAngle=135,PointName=\frac{3\pi}{4}](1;135){B_1}
\pstGeonode[PosAngle=150,PointName=\frac{5\pi}{6}](1;150){A_1}
\pstGeonode[PosAngle=180,PointName=\pi](1;180){I'}
\pstGeonode[PosAngle=-30,PointName=\frac{-\pi}{6}](1;-30){A'}
\pstGeonode[PosAngle=-45,PointName=\frac{-\pi}{4}](1;-45){B'}
\pstGeonode[PosAngle=-60,PointName=\frac{-\pi}{3}](1;-60){C'}
\pstGeonode[PosAngle=-90,PointName=\frac{-\pi}{2}](1;-90){J'}
\pstGeonode[PosAngle=-120,PointName=\frac{-2\pi}{3}](1;-120){C_1'}
\pstGeonode[PosAngle=-135,PointName=\frac{-3\pi}{4}](1;-135){B_1'}
\pstGeonode[PosAngle=-150,PointName=\frac{-5\pi}{6}](1;-150){A_1'}
```



```

% tracé des axes du repère :
\pstLineAB{I'}{I}\pstLineAB{J'}{J}
% lignes pointillées de "projection" sur les axes :
\pstLineAB[linestyle=dotted]{A}{A'}
\pstLineAB[linestyle=dotted]{B}{B'}
\pstLineAB[linestyle=dotted]{C}{C'}
\pstLineAB[linestyle=dotted]{C_1}{C_1'}
\pstLineAB[linestyle=dotted]{B_1}{B_1'}
\pstLineAB[linestyle=dotted]{A_1}{A_1'}
\pstLineAB[linestyle=dotted]{A}{A_1}
\pstLineAB[linestyle=dotted]{B}{B_1}
\pstLineAB[linestyle=dotted]{C}{C_1}
\pstLineAB[linestyle=dotted]{C'}{C_1'}
\pstLineAB[linestyle=dotted]{B'}{B_1'}
\pstLineAB[linestyle=dotted]{A'}{A_1'}
\begin{scriptsize} % (pour écrire petit)
% abscisses :
\rput(0.866,-0.1){$\frac{\sqrt{3}}{2}$}
\rput(0.707,-0.1){$\frac{\sqrt{2}}{2}$}
\rput(0.5,-0.1){$\frac{1}{2}$}
\rput(-0.9,-0.1){$\frac{-\sqrt{3}}{2}$}
\rput(-0.707,-0.1){$\frac{-\sqrt{2}}{2}$}
\rput(-0.5,-0.1){$\frac{-1}{2}$}
% ordonnées :
\rput(-0.1,0.866){$\frac{\sqrt{3}}{2}$}
\rput(-0.1,0.707){$\frac{\sqrt{2}}{2}$}
\rput(-0.1,0.5){$\frac{1}{2}$}
\rput(-0.1,-0.866){$\frac{-\sqrt{3}}{2}$}
\rput(-0.1,-0.707){$\frac{-\sqrt{2}}{2}$}
\rput(-0.1,-0.5){$\frac{-1}{2}$}
\end{scriptsize}
% flèche d'orientation trigo :
\pstGeonode[PointSymbol=none,PointName=none](1.25,0.2){P}
\pstGeonode[PointSymbol=oplus,PointName=none,dotscale=2](1.3,0.35){R}
\pstGeonode[PointSymbol=none,PointName=none](1.25,0.5){Q}
\pstArcOAB[arrows=->]{O}{P}{Q}
\end{pspicture}
\end{center}

```

Exercice 7.

```

\usepackage{pstricks}
Reproduire (à l'aide du quadrillage de votre feuille)
le vecteur  $\vec{i}$ 
\psset{unit=0.5cm}
\begin{pspicture}(0,0)(1,2)
% définition de la grille :

```

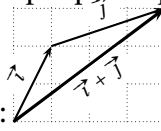


```

\psgrid [gridlabels=0,gridwidth=0.4pt,subgriddiv=0](0,0)(1,2)
% tracé du vecteur en trait un peu épais :
\psline [linewidth=1.5\pslinewidth]{->}(0,0)(1,2)
\end{pspicture}
et le vecteur  $\vec{j}$ 
\psset {unit=0.5cm}
\begin{pspicture}(0,0)(3,1)
% la grille :
\psgrid [gridlabels=0,gridwidth=0.4pt,subgriddiv=0](0,0)(3,1)
% le vecteur :
\psline [linewidth=1.5\pslinewidth]{->}(0,0)(3,1)
\end{pspicture}
puis représenter le vecteur  $2\vec{i}-3\vec{j}$ .

```

Le dessin de la solution s'attachera à distinguer, par exemple p_2 l'épaisseur, les données et le résultat.



Par exemple, de la façon suivante pour la somme $\vec{i} + \vec{j}$:

Le code :

```

% unité :
\psset {unit=0.5cm}
% début image, définition du cadre la contenant :
\begin{pspicture}(0,0)(3,3)
% la grille :
\psgrid [gridlabels=0,gridwidth=0.4pt,subgriddiv=0,griddots=10](0,0)(4,3)
% les vecteurs initiaux :
\psline [linewidth=1\pslinewidth]{->}(0,0)(1,2)
\psline [linewidth=1\pslinewidth]{->}(1,2)(4,3)
% le vecteur somme :
\psline [linewidth=1.5\pslinewidth]{->}(0,0)(4,3)
% les étiquettes :
% étiquette  $\vec{i}$  posée au point (0.5,1)
% à la distance 0.25pt de ce point dans la direction 63+90 degrés
% le texte est tourné de 63 degrés pour l'écriture
\begin{scriptsize}
\uput*{0.25}[153]{63}(0.5,1){ $\vec{i}$ }
\uput*{0.25}[108]{18}(2.5,2.5){ $\vec{j}$ }
\uput*{0.25}[-53]{37}(2,1.5){ $\vec{i}+\vec{j}$ }
\end{scriptsize}
\end{pspicture}

```

Exercice 8.

Le code proposé sur le site de l'apmep :

```

\psset {unit=0.5cm}
\begin{pspicture}(9,9)
\psgrid [subgriddiv=1,gridlabelcolor=white](0,0)(9,9)

```



```

\multido{\n=1+3,\d=2+3}{3}{\psframe[fillstyle=solid,fillcolor=lightgray](\n,1)(\d,2)}
\multido{\n=1+3,\d=2+3}{3}{\psframe[fillstyle=solid,fillcolor=lightgray](\n,4)(\d,5)}
\multido{\n=1+3,\d=2+3}{3}{\psframe[fillstyle=solid,fillcolor=lightgray](\n,7)(\d,8)}
\psframe[fillstyle=solid,fillcolor=lightgray](3,3)(6,6)
\end{pspicture}

```

Une proposition de code pour l'étape suivante :

on a imbriqué deux boucles.

On marque des petits carrés gris partout, ils disparaissent lorsqu'ils tombent dans un plus grand carré gris car ce plus grand carré gris est dessiné après et écrase le précédent.

```

\psset{unit=0.5cm}
\begin{pspicture}(9,9)
\psgrid[subgriddiv=3,gridlabelcolor=white](0,0)(9,9)
\multido{\rg=0.333+1,\rd=0.666+1}{9}{\multido{\rb=0.333+1,\rh=0.666+1}{9}
{\psframe[fillstyle=solid,fillcolor=lightgray](\rb,\rg)(\rh,\rd)}}
\multido{\n=1+3,\d=2+3}{3}{\psframe[fillstyle=solid,fillcolor=lightgray](\n,1)(\d,2)}
\multido{\n=1+3,\d=2+3}{3}{\psframe[fillstyle=solid,fillcolor=lightgray](\n,4)(\d,5)}
\multido{\n=1+3,\d=2+3}{3}{\psframe[fillstyle=solid,fillcolor=lightgray](\n,7)(\d,8)}
\psframe[fillstyle=solid,fillcolor=lightgray](3,3)(6,6)
\end{pspicture}

```

Exercice 9.

```

% \usepackage{pst-eucl} en préambule
% \usepackage{mathrsfs} pour la police calligraphique $\mathscr{C}$

% centrage de l'image par rapport à la colonne de texte :
\begin{center}
% unité du repère :
\psset{unit=2mm}
% définition du cadre contenant l'image par les
% coordonnées du coin inférieur gauche et du coin sup droit
\begin{pspicture}(-15,-6)(8,18)
% \pstGeonode(coordonnées){nom du point} : dessine le point et place son nom.
% \pstGeonode[PosAngle=angle en degrés pour placer le nom du point
% par rapport à ce point (vu comme un cercle trigo)](coordonnées){nom du point}
\pstGeonode[PosAngle=-45](-6,0){A}
\pstGeonode[PosAngle=-45](0,0){O}
\pstGeonode[PosAngle=-45](6,0){B}
\pstGeonode(-3,7.42){I}
% dessin du cercle de centre O passant par A :
\pstCircleOA{O}{A}
% dessin du cercle de centre I passant par A :
\pstCircleOA{I}{A}
% intersection du cercle de centre O passant par A et du cercle
% de centre I passant par A. Les points d'intersection sont nommés A et M :
\pstInterCC[PosAngleA=-45]{O}{A}{I}{A}{A}{M}
% tracé de la ligne droite (AB)

```



```
% nodesep= ... sert à faire 'dépasser' le tracé de part et d'autre des points
% sans cet argument, c'est donc le segment [AB] qui est tracé :
\pstLineAB[nodesep=-15]{A}{B}
\pstLineAB[nodesep=-15]{I}{M}
% intersection des droites (AB) et (IM).
% On appelle N le point d'intersection :
\pstInterLL[PosAngle=45]{A}{B}{I}{M}{N}
% On place des étiquettes indiquant les noms des points
% (4, -4.5) sont les coordonnées de l'étiquette C.
\uput[d](4, -4.5){$\mathscr{C}$}
\uput[d](6.5, 10){$\mathscr{C}'$}
% fin de l'environnement marquant le cadre pour l'image :
\end{pspicture}
% fin de l'environnement de 'centrage' :
\end{center}
```

Exercice 10.

```
% \usepackage{pst-eucl} en préambule
\psset{unit=1cm}
\begin{pspicture}(0,0)(6,6)
% A et B définis par coordonnées
\pstGeonode[PosAngle=90,dotscale=0.5](2,4){A}
\pstGeonode[PosAngle=180,dotscale=0.5](0,0){B}
% C défini comme troisième sommet pour avoir ABC isocèle en A
% angle en A facile à modifier
\pstRotation[PosAngle=0,RotAngle=40,dotscale=0.5]{A}{B}{C}
% on définit M sur le segment [BC] à l'aide d'une homothétie
\pstHomO[HomCoef=.60,dotscale=0.5,PosAngle=-90]{B}{C}{M}
% on trace maintenant les côtés du triangle :
\pstLineAB[linewidth=0.01cm]{A}{B}
\pstLineAB[linewidth=0.01cm]{C}{B}
\pstLineAB[linewidth=0.01cm]{A}{C}
% définition de M
\pstOrtSym[dotscale=0.5,CodeFig=true,CodeFigColor=black,
linewidth=0.01cm,dash=4pt 1pt,RightAngleSize=0.2,SegmentSymbol=none]{A}{C}{M}{M'}
\end{pspicture}
```