

TPs EDP <sup>a</sup>Travaux Pratiques N° 1 (*Distanciel*)

## Algorithmique : mosaïques

---

<sup>a</sup>. Version du 22 octobre 2020

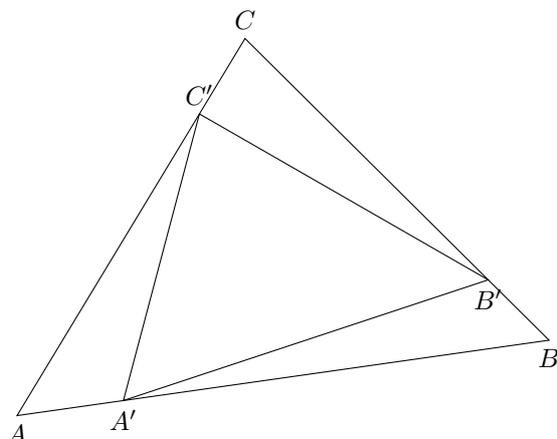
---

**Remarques**

---

- Le premier exercice est **à faire** sur feuille (durée 30mn). Il vous faudra le scanner ou prendre en photo lorsque nous vous le dirons et nous le faire parvenir (nous vous précisons les modalités lors du TP)
- Dans l'ensemble des exercices de ce TP, on dispose de la fonction algorithmique `plot([x1 x2], [y1 y2])` permettant de représenter le segment reliant les points  $A_1 = (x_1, y_1)$  et  $A_2 = (x_2, y_2)$ . La syntaxe est la même sous Matlab/Octave.
- Dans tous les exercices il faudra user (voir abuser) de la création de fonctions

## EXERCICE 1



Soit  $T$  un triangle de sommets  $A$ ,  $B$  et  $C$ . A partir de ce triangle on peut construire un nouveau triangle de sommets  $A'$ ,  $B'$  et  $C'$  vérifiant

$$\overrightarrow{AA'} = x\overrightarrow{AB}$$

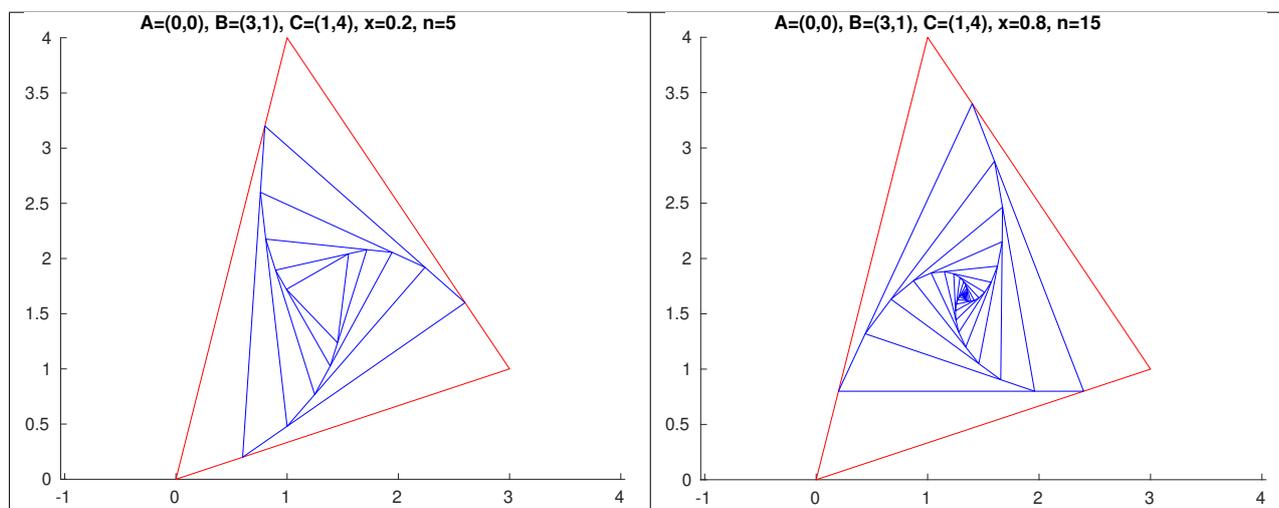
$$\overrightarrow{BB'} = x\overrightarrow{BC}$$

$$\overrightarrow{CC'} = x\overrightarrow{CA}$$

avec  $x$  un réel compris strictement entre 0 et 1.

L'objectif est, pour un  $x$  fixé, d'itérer  $n$  fois ce processus de construction en partant à chaque itération du dernier triangle construit et de représenter l'ensemble des triangles.

**Q. 1** *Ecrire une fonction **Algorithmique triangles** permettant à partir des trois sommets  $A, B, C$  d'un triangle initial quelconque non réduit à une droite ou à un point, de représenter ce triangle ainsi que les  $n$  triangles obtenus par le processus de construction décrit ci-dessus avec un  $x$  donné dans  $]0, 1[$ . On dispose pour cela de la fonction `plot([xA, xB], [yA, yB])` permettant de tracer le segment  $[A, B]$  du plan avec  $A = (x_A, y_A)$  et  $B = (x_B, y_B)$ . Voici deux exemples d'utilisation de cette fonction :*



### Instructions pour la suite

Deux archives compressées de même contenu au format **zip**

[www.math.univ-paris13.fr/~cuvelier/docs/Enseignements/MACS2/TPs-EDP/20-21/TP1/CodesFournis\\_Mosaiques.zip](http://www.math.univ-paris13.fr/~cuvelier/docs/Enseignements/MACS2/TPs-EDP/20-21/TP1/CodesFournis_Mosaiques.zip)

et au format **tar.gz**

[www.math.univ-paris13.fr/~cuvelier/docs/Enseignements/MACS2/TPs-EDP/20-21/TP1/CodesFournis\\_Mosaiques.tar.gz](http://www.math.univ-paris13.fr/~cuvelier/docs/Enseignements/MACS2/TPs-EDP/20-21/TP1/CodesFournis_Mosaiques.tar.gz)

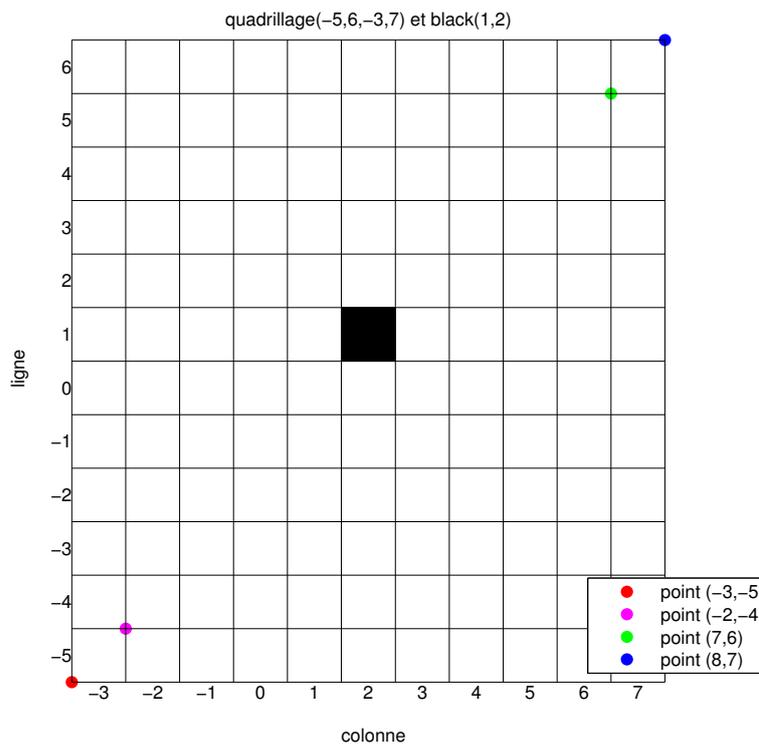
sont disponibles en ligne. Il faut télécharger une des archives et la décompresser dans un répertoire. Cette archive contient la fonction `PrintNumbers` et le programme `Quadrillagefigure`. Dans le programme `Quadrillagefigure` les appels aux fonctions manquantes `black` et `Quadrillage` ont été mis en commentaire. Ce programme va vous permettre, en autres, de valider/tester la fonction `Quadrillage` que vous allez écrire.

## EXERCICE 2

Ecrire les fonctions graphiques `Quadrillage` et `black` :

- la fonction `Quadrillage(imin,imax,jmin,jmax)` permet de générer un quadrillage pour les lignes `imin` à `imax` et les colonnes `jmin` à `jmax`. Cette fonction trace uniquement les traits noirs horizontaux et verticaux du quadrillage (utilisation de la fonction `plot` de Matlab).
- la fonction `black(i,j)` remplit en noir la *case* en ligne `i` et colonne `j` du quadrillage (utilisation de la fonction `fill` de Matlab).

Voici un exemple d'utilisation de la commande `Quadrillage(-5,6,-3,7)` générant uniquement les traits noirs de la figure :

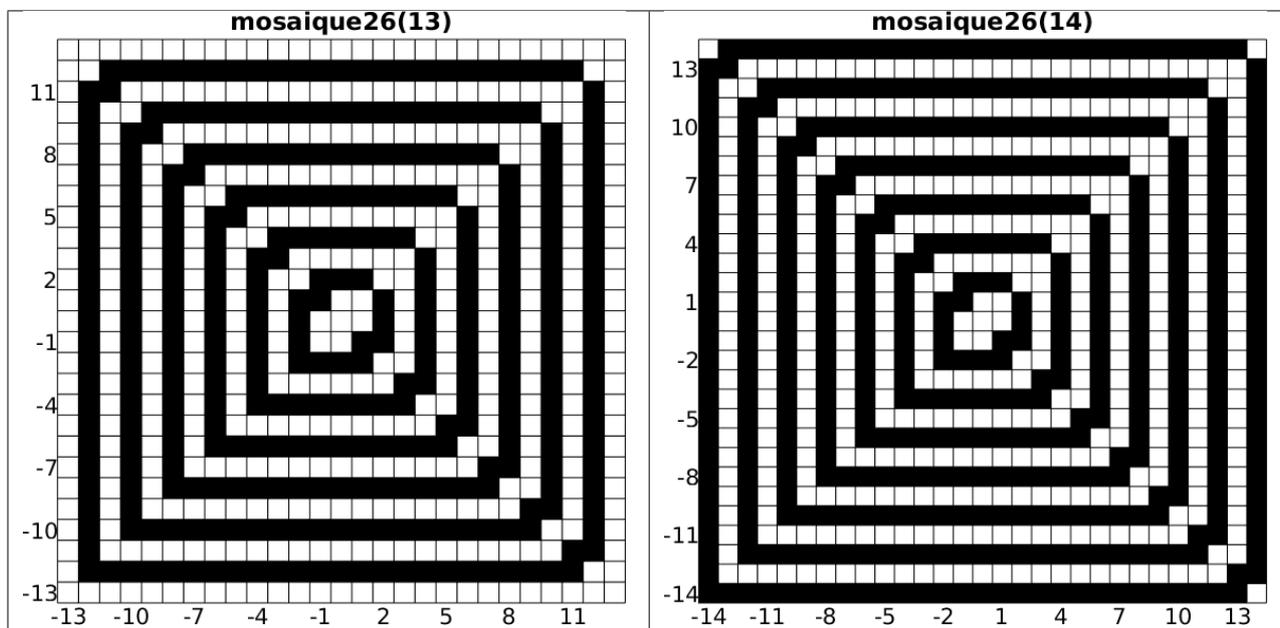


On peut noter que les coordonnées des points sont exprimées dans le plan classique  $xOy$ . On doit tester/valider cette fonction avec le programme `Quadrillagefigure` fourni pour obtenir la figure précédente. Le carré noir en ligne 1 et colonne 2 a été représenté à l'aide de la commande `black(1,2)`. La numérotation des lignes et des colonnes a été réalisée par la fonction `PrintNumbers` fournie dans l'archive.

Il est primordial que ces deux fonctions soient parfaitement valides pour la suite du TP!

### EXERCICE 3

On dispose, des fonctions Matlab [Quadrillage](#) et [black](#) de l'exercice 2, ainsi que de la fonction [PrintNumbers](#) fournie. Ecrire une fonction Matlab [mosaïque26](#) de paramètre  $n \in \mathbb{N}$ ,  $n \geq 3$ , permettant de créer des figures sur le quadrillage de lignes  $-n$  à  $n$  et de colonnes  $-n$  à  $n$ . Voici deux exemples, avec  $n = 13$  et  $n = 14$ , des figures que l'on souhaite représenter :



### EXERCICE 4

On dispose, des fonctions Matlab [Quadrillage](#) et [black](#) de l'exercice 2, ainsi que de la fonction [PrintNumbers](#) fournie. Ecrire une fonction [mosaïque18](#) de paramètre  $n \in \mathbb{N}$ ,  $n \geq 5$ , permettant de créer des figures sur le quadrillage de lignes 1 à  $n$  et de colonnes 1 à  $n$ . Voici deux exemples, avec  $n = 21$  et  $n = 22$ , des figures que l'on souhaite représenter sachant que la ligne  $n$  suit nécessairement la séquence *blanc, noir, noir, noir, blanc, noir, noir, noir,...*

