# Mini-projet: Outils Mathématiques

## 1 Algèbre linéaire

**1. Manipulation de vecteurs** Considérons les vecteurs $\mathbf{u} = (1, -1, 3)$ et $\mathbf{v} = (2, 0, -1)$.

1. Calculs à effectuer à la main :
   - Le produit scalaire $\mathbf{u} \cdot \mathbf{v}$
   - Les normes $\|\mathbf{u}\|$ et $\|\mathbf{v}\|$
   - L'angle (en degrés) entre $\mathbf{u}$ et $\mathbf{v}$

**2. Projection et orthogonalité**

1. Calculez la projection orthogonale de $\mathbf{u}$ sur $\mathbf{v}$.
2. À l'aide des vecteurs $\mathbf{a} = (1, 0, 0)$ et $\mathbf{b} = (0, 1, 0)$, démontrez par le calcul qu'ils sont orthogonaux.

## 2 Implémentation Python

Pour chaque question, implémentez les solutions en Python en utilisant les fonctions `numpy.dot`, `numpy.linalg.norm`, `numpy.arccos` et `numpy.degrees`.

1. Utilisez ces fonctions pour vérifier vos résultats numériques sur $\mathbf{u}$ et $\mathbf{v}$.

## 3 Calcul matriciel

### Diagonalisation

1. Calculez à la main les valeurs propres et vecteurs propres de $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$.
2. Vérifiez vos résultats en Python avec la fonction `numpy.linalg.eig`.
3. Vérifier que la matrice $B = \begin{pmatrix} 4 & 2 \\ 2 & 3 \end{pmatrix}$ est symétrique définie positive.

# 4 Analyse complexe et fractales

Au début du 20e siècle, Pierre Fatou et Gaston Julia ont fondé la dynamique holomorphe, étudiant l'évolution de suites complexes par des applications itératives. Les ensembles de **Julia** et de **Mandelbrot**, aujourd'hui célèbres, présentent des structures fractales remarquables.

## 4.1 Fractale de Mandelbrot

Pour chaque point complexe $z = x + iy$ du plan, on définit la suite :

$$u_0 = 0, \quad u_{n+1} = u_n^2 + z$$

Les points $z$ pour lesquels $(u_n)$ ne diverge pas forment l'ensemble de Mandelbrot.

**Objectif :** Parcourir le plan complexe pour visualiser cet ensemble, en suivant l'algorithme ci-dessous:

$x_{min} \leftarrow -2$, $x_{max} \leftarrow 2$
$y_{min} \leftarrow -2$, $y_{max} \leftarrow 2$
$\delta x \leftarrow 0.01$, $\delta y \leftarrow 0.01$
$N_{max} \leftarrow 30$, $Seuil \leftarrow 2$
Créer une matrice $Mat$ adaptée
**for** $x$ de $x_{min}$ à $x_{max}$ avec pas $\delta x$ **do**
    **for** $y$ de $y_{min}$ à $y_{max}$ avec pas $\delta y$ **do**
        $z \leftarrow x + iy$
        $u \leftarrow 0$ ; $n \leftarrow 0$
        **while** $|u| \leq Seuil$ **et** $n < N_{max}$ **do**
            $u \leftarrow u^2 + z$, $n \leftarrow n + 1$
        **end while**
        $Mat[x, y] \leftarrow n$
    **end for**
**end for**
**Afficher** $Mat$ (exemple : `plt.imshow(Mat)`)

1. Implémentez cet algorithme en Python.
2. Modifiez $N_{max}$ pour comparer les résolutions d'image.

## 4.2 Fractale de Julia

Pour un complexe $c$ donné, la suite s'écrit :

$$u_0 = z, \quad u_{n+1} = u_n^2 + c$$

On obtient différents ensembles de Julia selon les valeurs de $c$.

**À faire :** Générer et comparer les ensembles de Julia pour :

1. $c = -1$
2. $c = -1 + 0.2312i$
3. $c = 0.112 - 0.64i$

## Références


- https://fr.wikipedia.org/wiki/Ensemble_de_Mandelbrot
- Documentation Python (https://numpy.org, https://matplotlib.org)