

Les nombres premiers et les spirales

La spirale d'Ulam

Pour manipuler et explorer la spirale d'Ulam, vous pouvez utiliser le fichier `spirale_ulam.py`. En modifiant `etape`, vous pouvez changer le niveau de zoom. Et en modifiant `debut`, vous pouvez changer la valeur au centre.

EXERCICE 1 :

- 1) En modifiant la valeur au centre, chercher celle qui permet d'avoir le plus de nombres premiers sur la diagonale allant du coin bas gauche au coin haut droit.
- 2) Vérifier que cela correspond à la formule $n \times n + n + 41$.
- 3) Vérifier aussi que cette diagonale est composée de deux des demi-droites trouvées dans l'exercice 4 de la feuille papier.

EXERCICE 2 : En faisant une nouvelle fonction dans votre fichier `nombres_preiers.py` ou en générant une liste bien choisie, déterminer la première valeur de n qui ne donne pas un nombre premier avec la formule d'Euler.

EXERCICE 3 (optionnel) : Écrire une fonction `rendement_formule_euler` qui prend un entier `n_limite` et qui renvoie la proportion des nombres générés par la formule d'Euler avec $n \leq n_limite$ qui sont premiers.

EXERCICE 4 : Vérifier les résultats des exercices 2 et 3 de la feuille papier.

La spirale d'Archimède

Pour manipuler et explorer la spirale d'Archimède, vous pouvez utiliser le fichier `spirale_archimede.py`.

En modifiant `etape`, vous pouvez changer le niveau de zoom.

EXERCICE 5 : Explorer la spirale en changeant la valeur de la variable `etape` pour changer le niveau de zoom et retrouver les images se trouvant sur la dernière page de la feuille papier.

Dans le fichier, il est possible d'afficher les spirales générées par un nombre donné et les restes modulo ce nombre. Pour cela, il faut changer la valeur de `modulo` pour choisir le nombre par lequel on divise et mettre dans la liste `restes` les restes pour lesquels on veut afficher les spirales.

EXERCICE 6 : Vérifier les résultats trouvés à l'exercice 6.

EXERCICE 7 : Dans le fichier `nombres_preiers.py`, écrire une fonction `nombre_points_sur_spirale` qui prend un entier `reste`, un entier `modulo`, une liste de nombres premiers `liste_preiers` et qui renvoie le nombre de nombres premiers de la liste qui sont congrus à `reste modulo modulo`.

```
>>> nombre_points_sur_spirale(1, 6, premiers_preiers2(100))
11
>>> nombre_points_sur_spirale(3, 6, premiers_preiers2(100))
1
```

Pour trouver les nombres générant les autres spirales observées à d'autres niveaux de zoom, il faut chercher les valeurs qui permettent de se rapprocher encore plus d'un tour complet que 6.

EXERCICE 8 : Dans le fichier `nombres_premiers.py`, recopier le code de la fonction suivante et chercher les autres entiers qui pourraient générer les spirales observées. On pourra mettre la limite à 1000

```
def cherche_modulo(limite):
    meilleur_ecart = abs(2*math.pi-6)
    liste_meilleurs = [6]
    for i in range(liste_meilleurs[-1]+1, limite):
        ecart = i % (2*math.pi)
        if ecart < meilleur_ecart:
            meilleur_ecart = ecart
            liste_meilleurs.append(i)
    return liste_meilleurs
```

EXERCICE 9 : Parmi les valeurs trouvées à l'exercice précédent, trouver ceux qui engendrent les spirales observées. On pourra par exemple changer la valeur de modulo dans `spirale_archimede.py` et mettre restes à [0].

EXERCICE 10 : En vous aidant des résultats trouvés dans l'exercice 6 de la feuille papier, pour les 2 valeurs générant les spirales observées, trouver les restes qui correspondent aux spirales contenant un nombre potentiellement infini de nombres premiers.