PTSI — **Informatique – Partie Modélisation Scientifique**

Présentation du problème. Nous allons étudier dans cette partie le système dynamique de n vortex planaires qui est un système dynamique à n corps tout comme l'ensemble des planètes du système solaire. Un système dynamique à n corps est un système qui évolue dans le temps, et où chaque corps exerce une "force" sur chacun des autres, les n corps interagissent ainsi entre eux formant un mouvement général complexe difficile à prédire. Un des buts du scientifique dans ce type de système est de déterminer les configurations d'équilibre dans lesquelles le système est immobile, et ensuite d'étudier le caractère stable ou instable de ces configurations.

Nous considérons l'écoulement d'un fluide incompressible et non visqueux sur une couche *plane* d'épaisseur infinitésimale. Un vortex est un tourbillon ponctuel que l'on peut se représenter comme une tornade ou un cyclone. A chaque vortex est attaché un paramètre, la *vorticité* λ, qui correspond à la vitesse avec laquelle le fluide tourne sur lui-même, la vorticité est donc un nombre réel positif ou négatif suivant si le vortex tourne dans le sens direct ou non. L'étude des vortex planaires a des applications dans des domaines aussi variés que la météorologie ou les cristaux liquides.

On considère n vortex $v_1, \dots, v_n$ de vorticités identiques avec $\lambda = 1$. L'énergie H du système est conservée, et est donnée par la somme **double** suivante :

$$H = -\sum_{1 \leq i < j \leq n} \ln d_{ij}$$

où d_{ij} est la distance entre les vortex v_i et v_j. Chaque vortex v_i est assimilé à un point de coordonnées (x_i, y_i) dans un repère orthonormé $(O; \vec{i}, \vec{j})$. On considère les points $A(1,1)$ et $B(1,-1)$.

1. Ecrire une fonction `dist(V1,V2)` qui calcule la distance entre deux vortex v_1 et v_2, où $V1 = [x_1, y_1]$ et $V2 = [x_2, y_2]$ sont les listes à deux éléments des coordonnées des vortex v_1 et v_2.
2. Ecrire une fonction `testdist(L)` qui prend en argument la liste $L = [V_1, V_2, \dots, V_n]$ avec $V_i = [x_i, y_i]$ la liste des coordonnées du vortex v_i et qui renvoie `True` si les positions des n vortex sont distinctes et `False` dans le cas contraire.
3. Ecrire une fonction `energie(L)` qui calcule l'énergie H d'une configuration donnée où $n =$`len(L)` est le nombre de vortex, et $L = [V_1, V_2, \dots, V_n]$ avec $V_i = [x_i, y_i]$ la liste des coordonnées du vortex v_i. On n'oubliera pas d'importer correctement la fonction logarithme népérien ln. Déterminer la complexité de cette fonction.

   On recherche les configurations d'équilibre de vortex, pour cela on recherche les minimums de la fonction H.
4. On considère une liste $h = [h_1, \dots, h_p]$ constituée de p valeurs de l'énergie H pour p différentes configurations. Ecrire une fonction `minliste(h)` qui détermine le minimum d'une liste h donnée.
5. On considère une matrice carrée h de taille p donnée sous la forme d'une liste de p listes à p éléments, le coefficient h_{ij} de la matrice h correspond alors à `h[i-1][j-1]`. Alternativement, vous pouvez - si vous le souhaitez - considérer la matrice h comme un tableau (*array*) Numpy. Ecrire une fonction `minmat(h)` qui détermine le minimum des coefficients h_{ij} d'une matrice carrée h donnée.
6. Dans cette question, on considère deux vortex v_1 et v_2. Le vortex v_1 est fixé à l'origine O sans nuire à la généralité, et le vortex v_2 reste à l'intérieur du carré $\mathcal{C}$ de côté $[AB]$ et de centre l'origine. Ainsi :

   $$x_1 = 0 \ , \qquad y_1 = 0 \ , \qquad -1 \leq x_2 \leq 1 \ , \qquad -1 \leq y_2 \leq 1 \ .$$

   On discrétise le problème en considérant les $11^2 = 121$ positions suivantes pour le vortex v_2 :

   $$x_2 = -1 + \frac{2}{10} i \ , \qquad y_2 = -1 + \frac{2}{10} j$$

   avec $0 \leq i \leq 10$ et $0 \leq j \leq 10$.
   Ecrire une fonction `deuxvortex()` qui détermine le minimum de l'énergie H pour toutes ces configurations.
7. Dans cette question, on considère trois vortex v_1, v_2, v_3. Le vortex v_1 est fixé à l'origine O, et les vortex v_2 et v_3 restent à l'intérieur du carré $\mathcal{C}$. On réalise N tirages aléatoires pour déterminer chacune des valeurs de x_2, y_2, x_3, y_3, on obtient ainsi une liste de N valeurs de l'énergie H. Ecrire une fonction `troisvortex(N)` qui détermine le minimum de l'énergie H pour ces N configurations. On importera la fonction `random` du module `random`.