Devoir d'informatique à rendre Lundi 01 Mars 2021

Si travail en binôme : exo 1 : se répartir 1-2 et 3-4-5; se répartir exo 2 et exo 3.

Exercice 1

On considère un polygone $A_0A_1 \ldots A_n$ non croisé (deux segments distincts ne se recoupent pas). On peut calculer l'aire $\mathcal{A}$ de ce polygone par la formule de Green-Riemann qui s'écrit ici :

$$\mathcal{A} = \int_\gamma x\mathrm{d}y = \sum_{k=0}^{n} \int_{[A_k\, A_{k+1}]} x\,\mathrm{d}y = \sum_{k=0}^{n} \int_0^1 (x_k + t\,(x_{k+1} - x_k))\,(y_{k+1} - y_k)\,\mathrm{d}t = \sum_{k=0}^{n} \mathbf{1/2\,(y_{k+1} - y_k)\,(x_{k+1} + y_k)}$$

où le sommet A_k a pour coordonnées (x_k, y_k) et $A_{n+1} = A_0$.

La formule donne un résultat positif si les sommets sont parcourus dans le sens trigonométrique (en gardant l'intérieur du polygone sur sa gauche en permanence), sinon le résultat est négatif.

1) Ecrire une fonction **calcul_aire_polygone(pts)** qui prend en argument une liste de points en coordonnées cartésiennes et qui calcule l'aire du polygone. Chaque point étant codé comme une liste de deux nombres flottants.

2) Tester la fonction sur des figures simples puis sur le polygone régulier à n côtés et de rayon 1 (l'aire doit s'approcher de π quand n augmente).

On souhaite déterminer si deux segments $[AB]$ et $[CD]$ se croisent. On peut procéder ainsi :

On forme les vecteurs $\overrightarrow{AB}$ et $\overrightarrow{CD}$ et on calcule le déterminant de ces deux vecteurs. Il est nul ssi les deux vecteurs sont colinéaires, ssi les droites (AB) et (CD) sont sécantes.

- cas $\overrightarrow{AB}$ et $\overrightarrow{CD}$ non colinéaires : le point d'intersection de (AB) et (CD) est de la forme $A + x\overrightarrow{AB}$ et aussi de la forme $C + y\,\overrightarrow{CD}$. On doit donc résoudre le système linéaire $\boldsymbol{x\overrightarrow{AB} - y\overrightarrow{CD} = C - A = \overrightarrow{AC}}$. Les réels x et y peuvent se calculer par les formules de Cramer une fois calculées les coordonnées de $\overrightarrow{AB}$, $\overrightarrow{CD}$ et $\overrightarrow{AC}$. Le point d'intersection sera sur $[AB]$ et sur $[CD]$ ssi $x \in [0,1]$ et $y \in [0,1]$.
- cas $\overrightarrow{AB}$ et $\overrightarrow{CD}$ colinéaires : les droites (AB) et (CD) sont parallèles.
  1. Si $\overrightarrow{AB}$ et $\overrightarrow{AC}$ sont non colinéaires, les droites (AB) et (CD) sont strictement parallèles et les segments $[AB]$ et $[CD]$ ne se coupent pas.
  2. Si $\overrightarrow{AB}$ et $\overrightarrow{AC}$ sont colinéaires, les 4 points A, B, C et D sont alignés. On écrit $\overrightarrow{AC} = k_1\,\overrightarrow{AB}$ et $\overrightarrow{AD} = k_2\,\overrightarrow{AB}$. Les segments $[AB]$ et $[CD]$ sont disjoints ssi les deux réels k_1 et k_1 sont soit tous les deux > 1, soit tous les deux < 0. On peut calculer k_1 et k_2 par un produit scalaire :

$$\overrightarrow{AC}.\,\overrightarrow{AB} = k_1\,\overrightarrow{AB}.\overrightarrow{AB}, \text{ d'où } k_1 = \frac{\overrightarrow{AC}.\,\overrightarrow{AB}}{\|\overrightarrow{AB}\|^2} \quad \text{et de même, } \overrightarrow{AD}.\,\overrightarrow{AB} = k_2\,\overrightarrow{AB}.\overrightarrow{AB} \text{ avec } k_2 = \frac{\overrightarrow{AD}.\,\overrightarrow{AB}}{\|\overrightarrow{AB}\|^2}$$

(rappel : le produit scalaire de $\vec{u} = (x, y)$ et $\vec{v} = (x', y')$ est le réel $xx' + yy'$ et $\det(\vec{u}, \vec{v}) = xy' - x'y$)

3) Ecrire une fonction **croisement_segments(S1, S2)** qui prend en argument deux segments, S1 et S2 étant chacun une liste de deux points, un point étant une liste de deux nombre flottants comme dans la question 1. La fonction retourne True si les segments se croisent et False sinon. La fonction **affichera aussi les deux segments** et le point d'intersection éventuel.
On pourra reprendre les fonctions écrites en début d'année pour calculer les coordonnées d'un vecteur défini par deux points, pour calculer un déterminant ou un produit scalaire. Les tests d'égalité entre nombres flottants se feront à ε près, la précision étant à spécifier.

4) Faire des tests avec par exemple le jeu de données suivant :
croisement_deux_segments([[1,0] , [2,3]] , [[-1,1] , [4,2]])   # droite y = 3(x-1) et y = 1/5(x+1)+1
croisement_deux_segments([[1,0] , [2,3]] , [[-1,1] , [3/2,3/2]])   # point intersection (3/2,3/2) à l'extrémité de S2.
croisement_deux_segments([[1,0] , [2,3]] , [[1,0.5] , [5/2,5]])   # y = 3(x-1) et y = 3x-2.5
croisement_deux_segments([[1,0] , [2,3]] , [[0,-3] , [2/3,-1]]) # segments sur même droite mais disjoints
croisement_deux_segments([[1,0] , [2,3]] , [[0,-3] , [4/3,1]]) # segments se recoupent
croisement_deux_segments([[1,0] , [2,3]] , [[0,-3] , [2.5,9/2]]) # [CD] contient [AB]
croisement_deux_segments([[1,0] , [2,3]] , [[4,9] , [2.5,9/2]]) # segments sur même droite mais disjoints

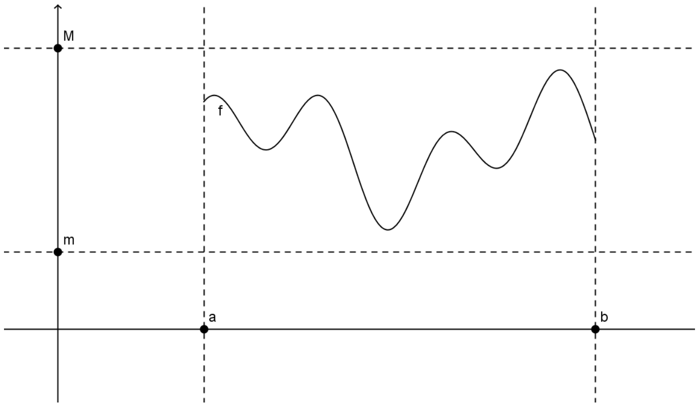
5) Ecrire une fonction **est_polygone_noncroise(pts)** prenant en argument une liste de points définissant un polygone et qui retourne True si le polygone est non croisé et qui retourne False sinon.

Exercice 2

But : on veut estimer $\int_a^b f(t)\,\mathrm{d}t$ où f est une fonction continue sur $[a, b]$ telle que $m \leq f \leq M$.

Principe de la méthode

On suppose $m \leq f \leq M$ et donc $0 \leq f - m \leq M - m$.
On effectue un tirage aléatoire de N valeurs de x dans $[a, b]$ et N valeurs de y dans $[m,M]$. Les tirages sont indépendants et sont faits selon la loi uniforme. Pour chaque valeur du couple (x, y) on teste si $m \leq y \leq f(x)$.



On note,

- D le domaine situé entre la droite $y = m$ et la courbe de f : $D = \{(x, y), x \in [a, b], y \in [m, f(x)]\}$
- s le nombre de succès, c.a.d le nombre de tirages où $m \leq y \leq f(x)$.

Si N est grand et les tirages aléatoires, on peut penser que le taux de succès est dans le même rapport que l'aire de D rapportée à l'aire du rectangle où ont été faits les tirages :

$$\frac{s}{N} \approx \frac{\text{aire de } D}{\text{aire du rectangle } [a,b] \times [0,M]} = \frac{\int_a^b f(t)\,\mathrm{d}t - m(b-a)}{(M-m) \times (b-a)}$$

On en déduit une estimation de $\boldsymbol{\int_a^b f(t)\,\mathrm{d}t \approx \frac{s}{N}(M-m)(b-a) + m(b-a)}$.

C'est une méthode dite de Monte-Carlo.

1) Ecrire une fonction **mc_integrale(f, bx, by, N)** où bx est la liste des bornes en abscisses (donc de la forme $[a, b]$, $a < b$) et by la liste des bornes en ordonnées, de la forme [m, M], f la fonction à intégrer et N le nombre de tirages.
La fonction effectue N tirages à l'aide d'une boucle for et calcule s. Elle retourne l'estimation de l'intégrale.

> Pour obtenir un **nombre flottant aléatoire dans le segment $[a, b]$**, on pourra écrire
> a + (nombre aléatoire entre 0 et $b - a$) = a + (nombre aléatoire entre 0 et 1) × $(b - a)$
> = ***a* + rd.random()*(*b* – *a*)**

2) Tester sur les intégrales suivantes :

$$\int_0^{\pi} \frac{\sin(t)}{t}\,\mathrm{d}t \approx 1.851937052 \quad \text{et} \quad \int_0^2 t\ln(t)\,\mathrm{d}t = 2\ln(2) - 1 \approx 0.386294361$$

> **Module random** (à importer sous la forme : import random as rd)
> rd.random() : retourne un nombre flottant aléatoire dans [0,1] selon la loi uniforme sur [0,1]
> rd.randint(a,b) : retourne un entier aléatoire entre a et b, extrémités comprises (loi uniforme sur ⟦a, b⟧)
> rd.choice(liste) : retourne aléatoirement un des termes de la liste (loi uniforme sur l'ensemble des termes de la liste)
> rd.sample(liste, n) : renvoie n éléments distincts choisis parmi les termes de la liste

Exercice 3

Ecrire une fonction **anagramme(ch1, ch2)** qui prend en argument deux chaînes de caractères et qui retourne vrai si les deux chaînes ont même longueur et si on peut permuter les caractères de l'une pour former l'autre. Par exemple, les mots 'coursiers', 'souscrire', 'croiseurs', 'sourciers' sont des anagrammes.