Année universitaire 2019/2020

Exercices Python

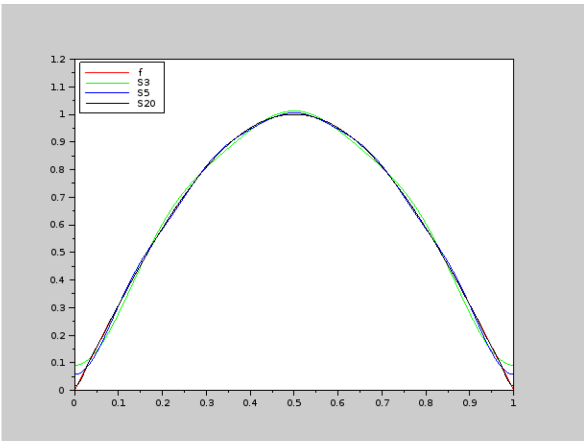
L3 mathématiques

Exercice 1

1. On considère la fonction 1-périodique $f(x) = |\sin(\pi x)|$. On considère la série de fonctions suivante, dite série de Fourier de f qui est donnée par

$$S_N f(x) = \frac{2}{\pi} - \sum_{n=1}^{N} \frac{4}{\pi(4n^2-1)} \cos(2\pi n x).$$

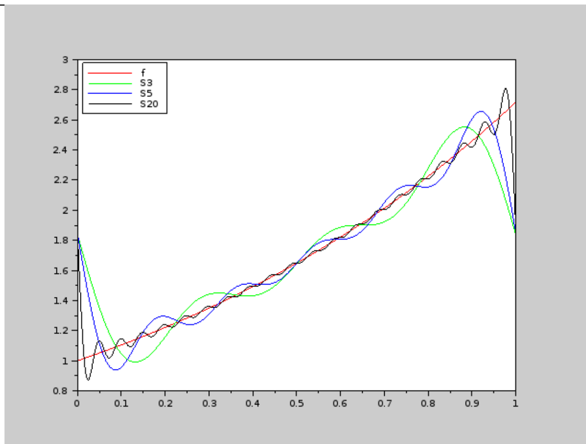
Tracer sur une même figure les graphes de f, S_3, S_5, S_9.



2. Visualiser en échelle logarithmique $N \mapsto \max_{0,1]}(|f - S_N f(x)|$. Que remarquez vous ? (On pourra utiliser la commande `loglog` de `matplotlib.pylab`.)
3. On considère la fonction 1-périodique $g(x) = e^{x-[x]}$. On sait que les sommes partielles de la série de Fourier de f sont données par

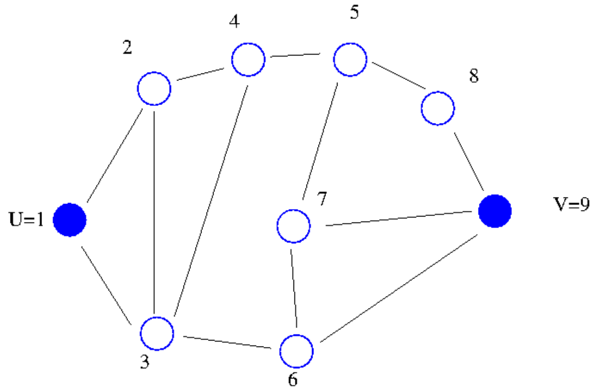
$$S_N g(x) = (e-1) + \sum_{n=1}^{N} \frac{2(e-1)}{1+4\pi^2 n^2} \cos(2\pi n x) + \sum_{n=1}^{N} \frac{-4\pi n(e-1)}{1+4\pi^2 n^2} \sin(2\pi n x).$$

Tracer sur une même figure les graphes de f, S_3, S_5, S_{20}.



Exercice 2

L'objectif de cet exercice est de résoudre le problème suivant : *Sébastien se rend régulièrement en train de la ville U à la ville V. Le réseau férrovière est représenté sur le graphe ci-dessous. Il fait toujours le trajet en 5 étapes, et veut faire à chaque fois un chemin différent. Combien de trajets pourra-t-il faire ?*



Si l'on essaie une énumération complète de tous les chemins de longueur 5 de U à V , on se rend vite compte qu'il est difficile de ne pas en oublier. Il faut trouver un moyen systématique de faire l'énumération. La solution, comme souvent, vient en cherchant un problème plus facile : chercher le nombre de chemins de longueur 1, puis 2, puis 3. . . Mais en échange de cette simplification, on va compliquer un peu : on va chercher les chemins joignant 2 villes I et J quelconques, et plus seulement U et V . Les chemins de longueur 1 qui joignent I à J se voient sur le graphe : ce sont les arêtes qui joignent I à J. On notera a_{ij} le nombre de ces chemins de longueur 1 : il vaut 0 si I n'est pas relié à J par une arête, 1 s'il y a une arête qui joint I à J. Comment compter les chemins de longueur 2 joignant I à J ? Un tel chemin est forcément formé d'un chemin de longueur 1 allant de I à une certaine ville K, puis d'un chemin de longueur 1 de la même ville K à J ; il faut bien sûr envisager toutes les possibilités pour K. Si b_{ij} désigne le nombre de chemins de longueur 2, on a donc :

$$b_{ij} = \sum_{k=1}^{p} a_{ik} a_{kj}$$

la somme étant prise sur tous les sommets du graphe. On reconnaît la formule usuelle du produit de matrices ! Si l'on numérote les villes de 1 à p, on peut regrouper les nombres a_{ij} définis ci-dessus en une matrice $A = (a_{ij})$. Si l'on note $B = (b_{ij})$ la matrice des chemins de longueur 2, on vient de voir que $B = A^2$, le produit étant le produit matriciel usuel. De même, si l'on note c_{ij} le nombre de chemins de longueur 3 de I à J, on voit que, puisqu'un tel chemin se décompose en un chemin de longueur 1 de I à K, suivi d'un chemin de longueur 2 de K à I, on a :

$$c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj}$$

d'où l'on déduit que la matrice $C = (c_{ij})$ vérifie $C = A.B = A^3$. Il est alors immédiat de généraliser : le nombre de chemins de longueur n qui joignent U à V est le terme d'indice u, v de la matrice A^n, où u et v désignent les numéros des sommets U et V.
La matrice A est appelée **Matrice d'adjacence**.

1. Construire sous Python la matrice A.
2. Calculer A^5 et répondre au problème initial.