CC3: optimisation numérique

Exercice 1

1. Soit $\mathcal{D}$ une famille de vecteurs engendrant positivement $\mathbb{R}^n$. Montrer que pour tout $v \in \mathbb{R}^n$ non nul, il existe d dans $\mathcal{D}$ tel que $v^T d < 0$.
2. On note
$$\kappa = \min_{||v||=1} \max_{d \in \mathcal{D}} v^T d$$
Montrer que $\kappa > 0$.
3. On identifie $\mathbb{R}^2$ et l'ensemble des nombres complexes. Montrer que $\mathcal{D} = \{1, j, j^2\}$ est une famille de vecteurs engendrant positivement $\mathbb{R}^n$. Calculer la valeur de κ associée à cette famille.

Exercice 2

On note $S_k = \{y_k^0, ..., y_k^n\}$ le simplexe obtenu par la méthode de Nelder Mead à l'itération k associé aux valeurs suivantes pour la fonction f : $f_k^0 \leq f_k^1 \leq ... \leq f_k^n$.

On suppose que la fonction$f$ est minorée.

1. Prouver que la suite $(f_k^0)_{k\in\mathbb{N}}$ est convergente.
2. Si un nombre fini de 'rétrécissements' survient, prouver que toutes les suites $(f_k^i)_{k\in\mathbb{N}}$ sont convergentes ($0 \leq i \leq n$).
3. On définit le volume du simplexe :
$$V(S_k) = \frac{1}{n!} \det[y_k^0 - y_k^n, ...y_k^{n-1} - y_k^n]$$
Vérifier que pour $n = 2$ la définition de $V(S_k)$ correspond à l'aire du triangle (S_k).
4. Donner la valeur de $V(S_{k+1})$ en fonction de $V(S_k)$ quand un pas de rétrécissement se produit.

Exercice 3 Le script suivant a été écrit pour résoudre un problème de minimisation :

```
n = 2
lmbda = 50
mu = 5
Ngen = 200
sigma = 0.1
def f(x):
    return 100 * (x[1] - x[0] ** 2) ** 2 + np.sum((x[:-1] - 1) ** 2)
Ap = np.random.rand(mu, n + 1)
for i in range(mu):    Ap[i, n] = f(Ap[i, :n])
Ae = np.zeros((lmbda, n + 1))
BestX = np.zeros((Ngen, n))
for k in range(Ngen):
    tau = 0
    X = np.mean(Ap[:, :n], axis=0)
    val = f(X)
    for i in range(lmbda):
        Ae[i, :n] = X + sigma * np.random.normal(0, 1, size=n)
        Ae[i, n] = ---------------------------
        if Ae[i, n] < val:
            tau += 1 / lmbda
    sorted_indices = np.argsort(Ae[:, n])
    Ae = Ae[sorted_indices, :]
    Ap = Ae[:mu, :]
    BestX[k, :] = Ap[0, :n]
    sigma = np.exp(1 / 3 * (tau - 1 / 5) / (1 - 1 / 5)) * sigma
print(BestX[-1, :])
```

et affiche le résultat : `[0.98110579 0.96252734]`.

1. Expliquer la méthode utilisée par ce script pour résoudre le problème de minimisation.
2. Que représente les matrices `Ap` et `Ae` ? Décrire le détail d'une ligne de ces matrices.
3. Un morceau de ligne a malencontreusement été effacé et remplacé par des$------$. Reconstituer et expliquer cette ligne.
4. Que représente la variable `sigma` et comment va-t-elle évoluer au cours des itérations ?
5. Que vaut `a` si on tape dans Python : `a = np.argsort(np.array([3.3,4,2,6,1]))`?
6. La méthode est-elle élististe ? Justifier votre réponse.
7. Proposer et écrire une autre méthode de sélection.

Exercice 4

En vous inspirant du code précédent, écrire un script Python pour minimiser la fonction de Rosenbrock par la méthode du recuit simulé. On prendra comme température la suite décroissante

$$T_k = \frac{1}{\log(k)}$$

et on reprendra le principe d'adaptativté du code précédent.