Calcul Numérique TP2

Partie 1: Application directe du cours

Dans cette première partie, vous n'avez pas besoin de l'ordinateur. Un papier et un crayon seront suffisants.

Exercice 1: Dérivées partielles

Après avoir écrit leur domaine de définition, déterminer les dérivées partielles premières et secondes, quand elles existent, et donner la matrice Hessienne des fonctions suivantes:

- [ ] $f(x, y) = xy^2 + 3x^2$
- [ ] $f(x, y) = \dfrac{x}{y} + y$
- [ ] $f(x, y) = e^{xy}$
- [ ] $f(x, y) = \sqrt{xy}$
- [ ] $f(x, y) = x^3 e^y + \ln(xy) + y^2 \ln(x)$
- [ ] $f(x, y, z) = x^2 yz$
- [ ] $f(x, y, z) = x^3 + xy + y^2 + 3z^2 x$

Exercice 2: Mineurs principaux

Pour les matrices suivantes, donner les mineurs principaux, ainsi que les mineurs principaux diagonaux. Puis dire si ces matrices sont définies ou semie-définies, positives ou négatives.

- [ ] $A = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$
- [ ] $B = \begin{bmatrix} -2 & 2 \\ 2 & -4 \end{bmatrix}$
- [ ] $C = \begin{bmatrix} 2 & 2 & 1 \\ 2 & 3 & 0 \\ 1 & 0 & 2 \end{bmatrix}$
- [ ] $D = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$

Pour les plus motivés, vous pouvez créer deux fonctions Python, itératives, permettant de déterminer si une matrice est définie positive ou négative.

Partie 2: Optimisation de fonction

On considère dans cette partie plusieurs fonctions:

1) $f(x, y) = xy$

2) $g(x, y) = 3xy - x^3 - y^3$

3) $h(x, y) = x^2 + \sin(y)$

Exercice 1: recherche des optima

On cherche les optima de cette fonction si ils existent, et on souhaite les caractériser: si ce sont des maximum, minimum locaux, globaux.

- [ ] Déterminer les optima possibles et les caractériser (à faire sur papier). On détaillera les CPO et CSO.

Exercice 2: Visualisation

- [ ] Tracer la surface de ces fonctions en vous aidant du code Python vu en cours et rappelé ci-dessous.
- [ ] Confirmer les valeurs des optima obtenues précédemment en les visualisant sur la surface.

In [ ]:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

def f(x,y):
    return ...

debut = ??
fin   = ??
N     = ??
x = np.linspace(debut, fin, N)
y = np.linspace(debut, fin, N)
x, y = np.meshgrid(x,y)
z = f(x,y)
fig = plt.figure()
#pour tracer la surface:
axes = fig.add_subplot(111, projection='3d')
axes.plot_surface(x, y, z)
axes.set_title('z = f(x,y)')
plt.show()
```

3) Tracer des vues en coupes pertinentes afin de voir les optima en 2D.

4) Tracer des isoclines pertinentes de la fonction.

In [ ]:

```

```