DEVOIR NUMÉRO 5 VERSION A

ECG2 MATHS APPLIQUÉES

Exercice

On considère la fonction f qui à tout couple (x, y) de $\mathbb{R}^2$ associe le réel :

$$f(x, y) = x^4 + y^4 - 2\,(x - y)^2$$

1. Justifier que f est de classe $\mathcal{C}^1$ sur $\mathbb{R}^2$.

2. **a.** Calculer les dérivées partielles d'ordre 1 de f.

b. Montrer que le gradient de f est nul si, et seulement si, on a : $\begin{cases} x^3 - x + y &= 0 \\ y^3 + x - y &= 0 \end{cases}$.

c. En déduire que f possède trois points critiques : $(0, 0)$, $(\sqrt{2}, -\sqrt{2})$, $(-\sqrt{2}, \sqrt{2})$.

3. **a.** Calculer les dérivées partielles d'ordre 2 de f.

b. Écrire la matrice hessienne de f en chaque point critique.

c. Déterminer les valeurs propres de chacune de ces trois matrices puis montrer que f admet un minimum local en deux de ses points critiques. Donner la valeur de ce minimum.

d. Déterminer les signes de $f(x, x)$ et $f(x, -x)$ au voisinage de $x = 0$.
Conclure quant à l'existence d'un extremum en le troisième point critique de f.

4. **a.** Pour tout (x, y) de $\mathbb{R}^2$, calculer $f(x, y) - (x^2 - 2)^2 - (y^2 - 2)^2 - 2(x + y)^2$.

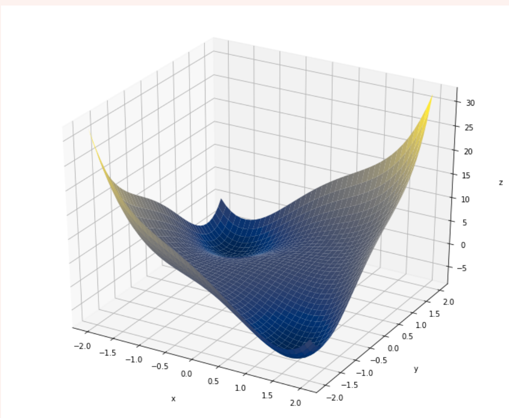
b. Que peut-on déduire de ce calcul quant au minimum de f ?

5. **a.** Compléter la deuxième ligne du programme suivant afin de définir la fonction f en Python.

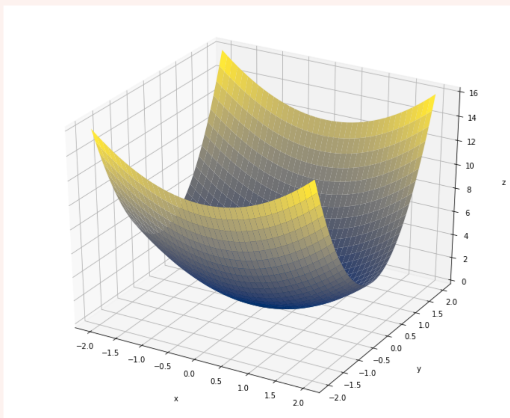
```
1 def f(x,y):
2     return ................
3
```

b. On utilise la fonction précédente pour tracer le graphe de f et Python renvoie l'une des trois nappes suivantes. Laquelle ? Justifier la réponse.

Graphe 1

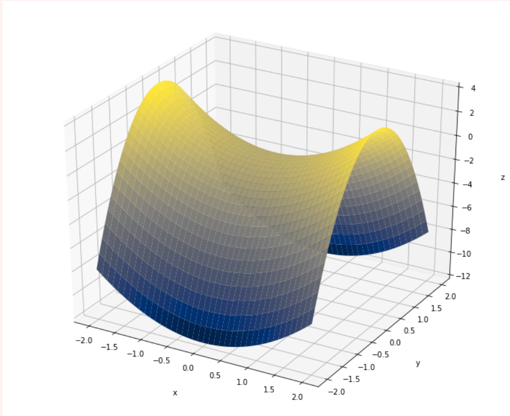


Graphe 2



Date: 30 Mars 2024 8h30-12h30.
http://louismerlin.fr.

Graphe 3



Exercice 2

On considère une variable aléatoire X suivant la loi normale $\mathcal{N}(0,\sigma^2)$ où σ est strictement positif. On rappelle que la fonction f_X qui a tout réel x associe $\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{x^2}{2\sigma^2}\right)$ est une densité de X et on note F_X la fonction de répartition de X, définie pour tout $x\in\mathbb{R}$, par

$$F_X(x)=\int_{-\infty}^{x} f_X(t)dt.$$

1. Montrer que pour tout $x\in\mathbb{R}$, $F_X(-x)=1-F_X(x)$.

2. On pose $Y=|X|$ et on admet que Y est une variable aléatoire.

a. Montrer que la fonction de répartition de Y, notée F_Y est définie par

$$F_Y(x)=\begin{cases} 2F_X(x)-1 & \text{si } x\geqslant 0\\ 0 & \text{si } x<0\end{cases}.$$

b. En déduire que Y est une variable à densité et donner une densité f_Y de Y.

c. Montrer que Y possède une espérance et que l'on a $E(Y)=\sigma\sqrt{\frac{2}{\pi}}$.

3. On suppose, dans cette question seulement, que σ est inconnu et on se propose de l'estimer. Soit n un entier supérieur ou égal à 1. On considère une échantillon $(Y_1,\cdots,Y_n)$ composé de variables aléatoires mutuellement indépendantes et ayant toutes la même loi que Y.

On note S_n la variable aléatoire définie par $S_n=\frac{1}{n}\sum_{k=1}^{n}$.

a. Montrer que S_n est un estimateur de σ, donner la valeur de son espérance puis proposer un estimateur (noté T_n) de σ, construit de façon affine à partir de S_n et tel que $E(T_n)=\sigma$.

b. Rappeler la valeur du moment d'ordre 2 de X, puis déterminer $E(Y^2)$, $V(Y)$ et $V(S_n)$.

c. Déterminer la variance de T_n et en déduire que T_n est un estimateur convergent de σ.

4. On rappelle qu'en Python, la commande rd.normal(m,s,n) simule n variables aléatoires indépendantes qui suivent toutes la loi normale de paramètres m et s^2, et les place dans un tableau à n colonnes. Compléter le script suivant afin qu'il simule les variables S_n et T_n pour des valeurs de n et σ rentrées par l'utilisateur du programme.

```
def simul_ST(n,sigma) :
  X = .....    ## simulation de X_1,...X_n
  Y = .....    ## simulation de Y_1,...Y_n
  S = .....
  T = .....
```

German Tank Problem

le contexte. En été 1943, les Alliés essaient de percer le bloc de l'Axe en créant un nouveau front via l'Italie. Ils rencontrent un nouveau type de char allemand, le bien nommé *Sonderkraftfahrzeug 171*, plus connu sous le nom de *Panther*.

Ce char est mieux équipé et plus performant que ceux rencontrés jusqu'alors. Il a été conçu en réponse à l'excellent *T-34* utilisé par les soviétiques sur le front de l'Est.

Sans rentrer dans les détails de fonctionnement de cette subtile et sympathique machine, il peut percer les défenses et détruire la majorité des tanks alliés.

Néanmoins, malgré sa puissance théorique, celui-ci ne peut avoir un réel impact sur l'issue de la guerre que si le nombre d'unités produites est suffisant. Il apparaît crucial pour les Alliés de déterminer ou plutôt d'*estimer* combien de *Panther* étaient produits. La tâche fût confié à [la] *Economic Warfare Division of the American Embassy in London.*

La modélisation. On suppose que l'ennemi produit une série de chars immatriculés par des entiers commençant par 1. En plus de cela, quelle que soit la date de production du char, ses années de service, ou encore son numéro de série, la distribution des numéros d'immatriculation est considérée comme uniforme dès l'instant où on mène l'analyse.

Dans notre modélisation, les allemands disposent de N tanks numérotés de 1 à N. Les forces alliées observent aléatoirement, uniformément et "avec remise" n numéros de séries $(X_1, X_2, \cdots, X_n)$ et cherchent à estimer le paramètre N.

On considère dans tout le problème un n-échantillon $(X_1, X_2, \cdots, X_n)$ de la loi uniforme $\mathcal{U}(\llbracket 1, N \rrbracket)$ et un première idée serait de considérer la moyenne des valeurs observées. On commence donc par poser

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i.$$

1. Que vaut l'espérance $E(\bar{X}_n)$ de $\bar{X}_n$?

Il serait *pratique* qu'en moyenne la variable aléatoire choisie pour estimer N renvoie N. Expliciter alors une variable aléatoire T_n construite à partir de $\bar{X}_n$ telle que $E(T_n) = N$.

2. Calculer la variance $V(T_n)$ et montrer, à l'aide de l'inégalité de Bienaymé-Tchebychev que, pour tout $\varepsilon > 0$,

$$\lim_{n \to +\infty} P(|T_n - N| > \varepsilon) = 0.$$

Ce résultat affirme, comme le dit le cours, que l'estimateur T_n converge (dans un certain sens) vers N, c'est-à-dire que si n est assez grand (si on dispose de suffisamment de données), la valeur approchée de N obtenue avec la définition de T_n appliquée à l'observation est proche de N.

On introduit alors le nouvel estimateur, c'est-à-dire une nouvelle fonction du n-échantillon

$$M_n = \max(X_1, \cdots, X_n).$$

3. Calculer, pour tout $k \in \mathbb{N}$, $P(M_n \leqslant k)$.

4. Soit Y une variable aléatoire à valeurs dans $\llbracket 1, N \rrbracket$. Montrer que

$$E(Y) = \sum_{k=0}^{N-1} P(Y > k).$$

5. Montrer alors que

$$E(M_n) = N - \sum_{k=0}^{N-1} \left(\frac{k}{N} \right)^n.$$

6. Vérifier que, pour tout $k \in \llbracket 0, N-1 \rrbracket$,

$$0 \leqslant \left(\frac{k}{N} \right)^n \leqslant N \int_{k/N}^{(k+1)/N} t^n dt.$$

7. En déduire que
$$N - \frac{N}{n+1} \leqslant E(M_n) \leqslant N$$
puis que
$$\lim_{n \to +\infty} E(M_n) = N.$$
Dans le cours, nous avions dit que l'estimateur M_n est asymptotiquement sans biais.

Si l'estimateur M_n parait naturel, il a clairement un défaut : il sous-estime nécessairement N puisqu'il renverra toujours une valeur inférieure ou égale à N. On va donc essayer de lui apporter une légère correction.

Commençons par introduire le numéro du plus petit tank observé $m_n = \min(X_1, \cdots, X_n)$.

Comme N est inconnu, on ne connaît pas l'écart entre N et M_n, mais il paraît raisonnable de penser qu'il y a (en moyenne) autant de tanks *non observés* entre M_n et N qu'entre 1 et m_n. Entre le plus petit numéro observé et le tank avec le numéro de série 1, il y a $m_n - 1$ numéros de tanks. On pense alors ajouter la correction
$$\tilde{M_n} = M_n + (m_n - 1).$$

8. En s'inspirant des calculs précédents pour M_n, déterminer $E(m_n)$ sous forme d'une somme que l'on ne cherchera pas à simplifier.

9. Montrer maintenant que $\tilde{M_n}$ vérifie $E(\tilde{M_n}) = N$.

10. Comparaison des estimateurs. On considère le programme Python suivant. Que fait-il ?

```
import numpy as np
import numpy.random as rd

def mystere(N,n) :
  T = []
  M = []
  for j in range(1000):
    X=[rd.randint(1,N+1) for k in range (n)]
    T.append(2*np.mean(X)-1)
    M.append(np.max(X)+np.min(X)-1)
  t=np.mean(T)
  m=np.mean(M)
  return [t,m]
```

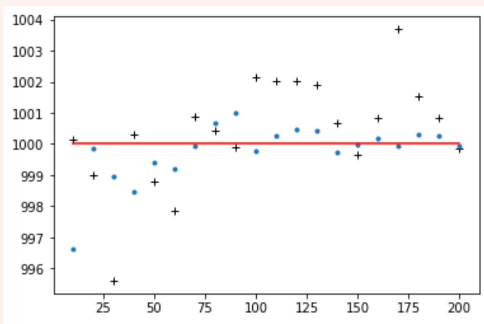
On ajoute les instructions ci-dessous dont l'exécution permet d'obtenir la figure ci-contre. Interpréter. Quel estimateur semble le plus performant ?

```
import matplotlib.pyplot as plt

T = []
M = []
N = 1000
x = [10*n for n in range (1,21)]
for n in x :
  [t,m] = mystere(N,n)
  T.append(t)
  M.append(m)

plt.plot(x,T,'k+')
plt.plot(x,M,'.')
plt.plot(x,[N for k in x], 'red')
plt.show()
```

Affichage des plots.



11. Les services secrets ont envoyé la liste des numéro de séries des tanks observés. À partir de ces données, choisir l'estimateur le plus performant et donner une estimation du nombre de tanks ennemis.

```
X = [14, 44, 50, 101, 117, 127, 134, 139, 165, 188, 192, 201, 204, 215, 234,
    243, 244, 253, 269, 269, 282, 287, 288, 322, 345]
```