

Examen de Science des Données, PHYS-231, 2024/25

Nom/Sciper:

Instructions:

- Durée de l'examen: 3 heures, 30.01.2025 de 15h15 à 18h15, salle AAC 1 37, SG 0213, SG 1138.
- Matériel permis: 2 pages (i.e. une page recto-verso ou deux pages recto) de notes personnelles (si imprimé, taille de police minimale 10 pt), papier, matériel pour écrire.
- Les problèmes peuvent être résolus dans n'importe quel ordre.
- Écrivez votre nom et prénom sur **toutes** les feuilles additionnelles que vous rendez. Vous pouvez utiliser la dernière page si vous avez besoin de plus d'espace.
- Le nombre total de points est 66.

Contenu

1	Questions de cours [18 points]	2
2	Régression linéaire pondérée [5 points]	6
3	Le problème du phare [8 points]	8
4	Distribution d'entropie maximale [4 points]	10
5	Échantillonnage direct dans la boule [6 points]	11
6	Chaîne de Markov à deux états [5 points]	12
7	Questions de programmation [14 points]	13
8	Implémentation d'une descente de gradient [6 points]	17

1 Questions de cours [18 points]

1. **Méthode des puissances itérées (3 points)** Considérons une matrice A symétrique de taille 3×3 ayant pour valeurs propres $5, 3, -1$. On applique la méthode des puissances itérées à A pour calculer le vecteur propre associé à la valeur propre dominante.
- (a) (1 point) Vous initialisez l'algorithme avec un vecteur initial donné. Expliquez pourquoi la méthode peut échouer à converger vers le vecteur propre dominant.
- (b) (1 point) On applique la méthode des puissances itérées à $B = A - \mu \mathbb{I}$ (\mathbb{I} est la matrice identité) avec $\mu = 3$. Quelle valeur propre domine ?
- (c) (1 point) Calculez le taux de convergence de la méthode des puissances itérées appliquée à la matrice A ainsi qu'à la matrice B . On rappelle que le taux de convergence est la constante a telle que l'erreur décroît en e^{-at} avec t le nombre d'itérations.

Solution: Answer: (a) If the starting vector has no component along the eigenvector of the largest magnitude eigenvalue (5), the method will fail. A solution is to ensure the initial vector has a non-zero projection along the dominant eigenvector.

(b) The shift changes the eigenvalues to 2, 0, -4 and the power method will hence converge to the eigenvector corresponding to the eigenvalue of largest absolute value, hence -4.

(c) The subleading terms will decrease as $|3/5|^t$ for A , $|2/4|^t$ for B .

2. **Propagation d'erreur (3 points)** On mesure deux variables aléatoires indépendantes x et y , respectivement d'écart type σ_x et σ_y . On s'intéresse à la quantité $z = \sqrt{x^2 + y^2}$.
- (a) (2 points) Écrire la formule pour l'écart-type de z , σ_z , en utilisant la propagation d'erreur. Quelle est l'hypothèse additionnelle principale qu'il faut pour que cette formule soit valide ?
- (b) (1 point) Pour $x = 4.0 \pm 0.2$ and $y = 3.0 \pm 0.2$, écrivez la valeur de z accompagnée de son erreur. Aide : si votre formule est correcte vous n'aurez pas besoin d'une calculatrice.

Solution: (a) Using error propagation: $\sigma_z^2 = \left(\frac{\partial z}{\partial x} \sigma_x\right)^2 + \left(\frac{\partial z}{\partial y} \sigma_y\right)^2$. With $z = \sqrt{x^2 + y^2}$, we compute:

$$\frac{\partial z}{\partial x} = \frac{x}{\sqrt{x^2 + y^2}}, \quad \frac{\partial z}{\partial y} = \frac{y}{\sqrt{x^2 + y^2}}.$$

Thus:

$$\sigma_z^2 = \left(\frac{x}{z} \sigma_x\right)^2 + \left(\frac{y}{z} \sigma_y\right)^2, \quad \text{where } z = \sqrt{x^2 + y^2}.$$

The main assumption here is that the standard deviations are small so that the Taylor expansion used to derive this formula is accurate. [1 point for the correct formula for σ_z , if partial derivatives are computed incorrectly or not computed in both (a) and (b) - 0.5 point; 1 point for correct assumption: independent variables x and y and/or small σ_x, σ_y]

(b) Substituting $x = 4.0$, $\sigma_x = 0.2$, $y = 3.0$, $\sigma_y = 0.2$:

$$z = \sqrt{4.0^2 + 3.0^2} = \sqrt{16 + 9} = 5.0,$$

$$\sigma_z^2 = \left(\frac{4.0}{5.0} \cdot 0.2\right)^2 + \left(\frac{3.0}{5.0} \cdot 0.2\right)^2,$$

$$\sigma_z^2 = (0.16)^2 + (0.12)^2 = 0.04, \quad \sigma_z = 10^{-2} * 2 * 5 = 0.2.$$

Final result: $z = 5.0 \pm 0.2$.

3. **Régression régularisée (2 points)** Considérons la perte de régression régularisée quadratique (ridge regression) :

$$\mathcal{L}(\beta) = \|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2,$$

où X est la matrice de donnée, y sont les étiquettes (labels), β sont les coefficients qui minimisent \mathcal{L} , et $\lambda > 0$ est le paramètre de régularisation.

- (a) (1 point) Expliquez brièvement comment accroître λ affecte la magnitude des coefficients.
- (b) (1 point) Expliquez brièvement comment choisir la valeur de λ qui donne la meilleure performance.

Solution: (a) Increasing λ shrinks the coefficients β , reducing their variance. (b) Look at the validation error as a function of λ and minimize it.

4. **Analyse en composantes principales (3 points)** On effectue une analyse en composantes principales (PCA) de la matrice normalisée et centrée $A \in \mathbb{R}^{n \times d}$, supposant $n \geq d$. Les valeurs singulières de A sont $\sigma_1, \sigma_2, \dots, \sigma_d$.

- (a) (2 points) Écrivez une formule pour la proportion de la variance (c'est-à-dire le carré de la norme de Frobenius de la quantité correspondante) qui est expliquée par les $k \leq d$ composantes principales. Aucune dérivation n'est demandée.
- (b) (1 point) Si $d = 3$, $\sigma_1 = 5$, $\sigma_2 = 3$, et $\sigma_3 = 2$, calculez la proportion expliquée par les deux premières composantes. Donnez la réponse en pourcentages arrondi au nombre entier le plus proche.

Solution:

- (a) The proportion of variance explained by the first k principal components is given by:

$$\text{Variance explained} = \frac{\sum_{i=1}^k \sigma_i^2}{\sum_{i=1}^d \sigma_i^2}.$$

The numerator represents the variance captured by the first k components, while the denominator is the total variance in the dataset. [2 points for correct formula, if only numerator is written – 1 point; if the answer is written in terms of Frobenius norm of matrix, the grade is not lowered; if the ratio of unexplained variance is found instead of explained, grade is not lowered. If the formula in (a) is not correct but correct formula is used in (b) then 1 point in (a)]

- (b) Substituting $\sigma_1 = 5$, $\sigma_2 = 3$, and $\sigma_3 = 2$:

$$\sum_{i=1}^3 \sigma_i^2 = 5^2 + 3^2 + 2^2 = 25 + 9 + 4 = 38.$$

The proportion of variance explained by $k = 2$ is:

$$\frac{\sigma_1^2 + \sigma_2^2}{\sum_{i=1}^3 \sigma_i^2} = \frac{25 + 9}{38} = \frac{34}{38} \approx 89\%.$$

5. **Mouvement brownien (4 points)** Une particule suit un mouvement brownien en une dimension. La position de la particule au temps t est notée $x(t)$; le processus satisfait l'équation de diffusion :

$$\frac{\partial P(x, t)}{\partial t} = D \frac{\partial^2 P(x, t)}{\partial x^2}.$$

où D est le coefficient de diffusion, et $P(x, t)$ est la densité de probabilité de la position de la particule au temps t . La particule part de $x = 0$ à $t = 0$.

- (a) (1 point) Écrivez explicitement $P(x, t)$. Une dérivation n'est pas demandée, mais P doit satisfaire l'équation différentielle.

Solution: (a) $P(x, t)$ is Gaussian with mean 0 and variance $2Dt$:

$$P(x, t) = \frac{1}{\sqrt{4\pi Dt}} e^{-x^2/(4Dt)}.$$

- (b) (1 point) Considérons une marche aléatoire symétrique sur un réseau unidimensionnel avec un pas spatial a et un pas temporel τ . À chaque pas, la particule saute à gauche ou à droite avec égale probabilité $1/2$. Écrivez la moyenne et la variance de la position de la particule x_n après n pas.

Solution: (b) Mean: $\langle x_n \rangle = 0$. Variance: $\langle x_n^2 \rangle = na^2$.

- (c) (1 point) Considérons la limite $a \rightarrow 0$ et $\tau \rightarrow 0$ avec un nombre de pas $n \rightarrow \infty$ tel que $t = n\tau$ est fixe. Donnez une relation entre a et τ qui ne mène pas à une limite triviale. À l'aide du théorème central limite, donnez la densité de probabilité $P(x, t)$ de la position de la particule au temps t .

Solution: (c) and (d) By defining $D = a^2/(2\tau)$, the position distribution converges to:

$$P(x, t) = \frac{1}{\sqrt{4\pi Dt}} e^{-x^2/(4Dt)}.$$

- (d) (1 point) En considérant les réponses à (a) et (c), donnez la constante de diffusion D en fonction de a et τ .

Solution: (c) and (d) By defining $D = a^2/(2\tau)$, the position distribution converges to:

$$P(x, t) = \frac{1}{\sqrt{4\pi Dt}} e^{-x^2/(4Dt)}.$$

6. Bilan détaillé et distribution stationnaire (3 points)

Considérons une chaîne de Markov avec des états $\{x_1, x_2, \dots, x_n\}$ et des probabilités de transition $p(x_i \rightarrow x_j)$ conçues pour échantillonner une distribution de probabilité $\pi^*(x_i)$.

- (a) (1 point) Écrivez la condition du bilan détaillé (detailed balance) par rapport à $\pi^*(x_i)$.

Solution: (a) Detailed Balance:

$$\pi^*(x_i)p(x_i \rightarrow x_j) = \pi^*(x_j)p(x_j \rightarrow x_i).$$

- (b) (1 point) Écrivez la définition d'une distribution stationnaire π^{st} pour cette chaîne de Markov.

Solution:

(b)

$$\pi^{\text{st}}(x_i) = \sum_{x_j} p(x_j \rightarrow x_i) \pi^{\text{st}}(x_j)$$

- (c) (1 point) Démontrez que le bilan détaillé garantit l'existence d'une distribution stationnaire.

Solution: (c) DB ensures that for the distribution $\pi^*(x)$ is stationary. By plugging the DB into the stationarity definition.

2 Régression linéaire pondérée [5 points]

1. (3 points) Considérez le cas d'une régression linéaire où chaque point du jeu de donnée est associé à un poids $\beta_i \in \mathbb{R}^+$. Ce cas est appelé régression linéaire pondérée. Trouvez une expression pour le paramètre $\hat{w} \in \mathbb{R}^d$ qui minimise la fonction de perte (loss function) suivante

$$\mathcal{L}(\vec{w}) = \frac{1}{n} \sum_{\mu=1}^n \beta_{\mu} \left(y_{\mu} - \vec{X}_{\mu} \cdot \vec{w} \right)^2 \quad (1)$$

où n est la taille du jeu de donnée, d est la dimension de chaque point du jeu de donnée, $y_1, \dots, y_n \in \mathbb{R}$ sont les valeurs de sortie et $\vec{X}_1, \dots, \vec{X}_n \in \mathbb{R}^d$ représentent les données d'entrée. Votre expression de \hat{w} doit être donnée sous forme matricielle. Vous pouvez utiliser la notation

$$B = \begin{pmatrix} \beta_1 & 0 & \dots & 0 \\ 0 & \beta_2 & 0 & \dots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \beta_n \end{pmatrix}. \quad (2)$$

Vous pouvez également supposer que $X^T B X$ est inversible, où $X = (\vec{X}_1, \dots, \vec{X}_n)^T$. Justifiez votre raisonnement.

Solution: The calculation is essentially the same as in section 3.5 of the lecture notes (2024 version).

The stationary points are situated where every partial derivatives is equal to 0:

$$\frac{\partial \mathcal{L}(\vec{w})}{\partial w_k} \stackrel{!}{=} 0 \quad \forall k = 1, \dots, d \quad (3)$$

This stationary point will be a minimum since $\mathcal{L}(\vec{w})$ is a parabola. The partial derivative read

$$\frac{\partial \mathcal{L}(\vec{w})}{\partial w_k} = \frac{-2}{n} \sum_{\mu=1}^n \beta_{\mu} \left(y_{\mu} - \sum_{i=1}^d X_{\mu i} w_i \right) X_{\mu k} \stackrel{!}{=} 0. \quad (4)$$

Splitting the sum we obtain

$$\sum_{\mu=1}^n \beta_{\mu} y_{\mu} X_{\mu k} \stackrel{!}{=} \sum_{\mu=1}^n \sum_{i=1}^d \beta_{\mu} X_{\mu i} w_i X_{\mu k}. \quad (5)$$

Rearranging the two sums in the right hand-side, the previous equation can be expressed as

$$\sum_{\mu=1}^n \beta_{\mu} y_{\mu} X_{\mu k} \stackrel{!}{=} \sum_{i=1}^d \left(\sum_{\mu=1}^n \beta_{\mu} X_{\mu k} X_{\mu i} \right) w_i. \quad (6)$$

Using the fact that $X_{\mu k} = [X^T]_{k\mu}$ the left-hand side becomes

$$\sum_{\mu=1}^n \beta_{\mu} y_{\mu} X_{\mu k} = \sum_{\mu=1}^n [X^T]_{k\mu} \beta_{\mu} y_{\mu} = [X^T B \vec{y}]_k \quad (7)$$

and the right-hand side becomes

$$\sum_{i=1}^d \left(\sum_{\mu=1}^n \beta_{\mu} X_{\mu k} X_{\mu i} \right) w_i = \sum_{i=1}^d \left(\sum_{\mu=1}^n [X^T]_{k\mu} \beta_{\mu} X_{\mu i} \right) w_i = \sum_{i=1}^d [X^T B X]_{ki} w_i = [[X^T B X] \vec{w}]_k. \quad (8)$$

Putting the obtained expressions together, we have (in matrix form)

$$X^T B \vec{y} \stackrel{!}{=} X^T B X \vec{w}. \quad (9)$$

Thus, if $X^T B X$ is invertible we have that

$$\hat{w} := \operatorname{argmin}_{\vec{w} \in \mathbb{R}^d} (\mathcal{L}(\vec{w})) = (X^T B X)^{-1} X^T B \vec{y}. \quad (10)$$

Shorter solutions can be obtained by directly doing the calculations in matrix form. One can also note that the change of variable $Z = X B^{\frac{1}{2}}$ and $\vec{v} = B^{\frac{1}{2}} \vec{y}$ brings the loss back to the same form as standard linear regression.

2. (1 point) Pour quelle(s) valeur(s) de β_1, \dots, β_n retrouve-t-on les paramètres \hat{w} de la régression linéaire standard ?

Solution: For $\beta_1 = \beta_2 = \dots = \beta_n$.

Additionally, they should not be equal to 0 (but this is not needed, as $X^T B X$ invertible implies that $B \neq 0$).

3. (1 point) Supposons que les valeurs de sortie sont générées par

$$y_\mu = X_\mu^T w^* + \epsilon_\mu \quad ; \quad \epsilon_\mu \sim \mathcal{N}(0, \Delta_\mu^2) \quad (11)$$

où les ϵ_μ sont tirée indépendamment d'une distribution normale de variance Δ_μ^2 centrée en 0. L'estimateur de maximum de vraisemblance (maximum likelihood estimator) \hat{w} est obtenu comme solution à un problème de régression linéaire pondérée avec des valeurs spécifiques de β_1, \dots, β_n . Donnez ces valeurs.

Solution:

$$\beta_\mu = \frac{1}{\Delta_\mu^2} \quad (12)$$

This result is directly presented in proposition 6 of section 10.2 of the lecture notes (2024 version).

3 Le problème du phare [8 points]

Il y a un phare près d'une côte rectiligne avec position α sur l'axe \vec{x} et à distance β de la plage (voir fig. 1). La distance β est connue, mais pas la position α .

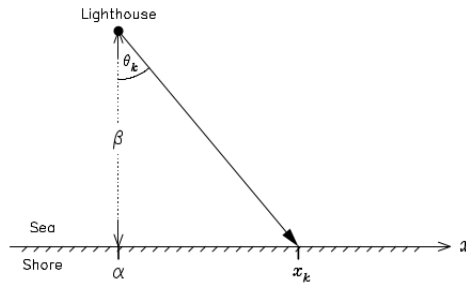


Figure 1: Schéma du problème. Lighthouse : phare ; shore : plage.

Le phare émet des impulsions lumineuses dans des directions aléatoires, uniformément sur tout l'angle 2π . Ces impulsions sont interceptées sur la côte par des photo-détecteurs, qui ne détectent que leur occurrence, mais pas l'angle dont elles viennent. N impulsions ont été enregistrées aux positions x_k , $k = 1, \dots, N$.

Le but est d'estimer la position α à partir des N observations $x = \{x_k\}_{k=1}^N$.

- (1 point) Sans écrire aucune probabilité, donnez une estimation naïve de α en fonction des données x .

Solution: A naive estimation of α is the average value of the data

$$\alpha = \frac{1}{N} \sum_{k=1}^N x_k.$$

- (2 points) Prouvez que la probabilité $P(x_k|\alpha, \beta)$ d'observer x_k conditionné à α et β est donnée par la distribution de Cauchy

$$P(x_k|\alpha, \beta) = \frac{\beta}{\pi [\beta^2 + (x_k - \alpha)^2]}.$$

Solution: Let's consider a uniform distribution over the angle θ_k , i.e. $P(\theta_k|\alpha, \beta) = \frac{1}{\pi}$ for $\theta_k \in (-\pi/2, \pi/2)$, and zero otherwise. Expressing the position x_k as a function of the angle θ_k

$$x_k = \beta \tan \theta_k + \alpha \quad \text{and} \quad \theta_k = \arctan((x_k - \alpha)/\beta).$$

Then, using the change of variable formula we get

$$P(x_k|\alpha, \beta) = \frac{d\theta_k}{dx_k} P(\theta_k|\alpha, \beta) = \frac{1}{\pi \beta (1 + (x_k - \alpha)^2/\beta^2)} = \frac{\beta}{\pi (\beta^2 + (x_k - \alpha)^2)}, \quad (13)$$

3. (2 points) Quelle est la moyenne de la distribution $P(x_k|\alpha, \beta)$ de la question 2 ? Indice : il suffit d'analyser l'intégrale correspondante pour $|x|$ grand. Qu'est-ce que cela implique quant à votre estimateur naïf de la question 1 ?

Solution: The mean is undefined (and hence the variance is infinite). Therefore, the naive estimation was not useful since the law of large numbers does not apply.

4. (2 points) Utilisez la formule de Bayes pour exprimer la probabilité $P(\alpha|\beta, x)$ que α prenne une valeur donnée conditionnée à la valeur de β et des N observations x . Supposez un a priori $P(\alpha|\beta)$ qui ne dépend pas du nombre de mesures.

Solution:

$$P(\alpha|\beta, \{x_k\}_{k=1}^N) \propto P(\{x_k\}_{k=1}^N|\alpha, \beta)P(\alpha|\beta) = \prod_{k=1}^N P(x_k|\alpha, \beta)P(\alpha|\beta).$$

5. (1 points) Écrivez l'estimateur de maximum de vraisemblance (maximum likelihood estimator) α^{ML} pour α sous la forme d'une équation dont α^{ML} est une solution.

Solution:

$$\alpha^{\text{ML}} = \arg \max_{\alpha} \log \prod_{k=1}^N P(x_k|\alpha, \beta) = \arg \min_{\alpha} \sum_{k=1}^N \log[\beta^2 + (x_k - \alpha)^2].$$

4 Distribution d'entropie maximale [4 points]

(4 points) Considérons une variable aléatoire $x \in \mathbb{R}$. On observe sa moyenne μ et sa variance Δ à partir de données. Dérivez la distribution de probabilité $P(x)$ qui maximise l'entropie en satisfaisant la contrainte que la moyenne et la variance sont égales à celles observées.

Solution: To solve the functional optimization problem we introduce the following Lagrangian

$$L(P(\cdot), \lambda_\mu, \lambda_\Delta) = - \int P(x) \log P(x) dx - \lambda_\mu \left(\mu - \int x P(x) dx \right) - \lambda_\Delta \left(\Delta - \int (x^2 - \mu^2) P(x) dx \right) - \lambda_n \left(1 - \int P(x) dx \right)$$

we now take the functional derivative of L w.r.t. $P(\cdot)$. This gives

$$\frac{\delta L}{\delta P(x)} = -\log P(x) - 1 + \lambda_\mu x + \lambda_\Delta x^2 + \lambda_n.$$

Setting this derivative to zero gives $P(x) = \exp[\lambda_n - 1 + \lambda_\mu x + \lambda_\Delta x^2]$ (A Gaussian!). The multipliers $\lambda_n, \lambda_\mu, \lambda_\Delta$ will take the values that satisfy the constraints respectively on normalization, mean, and variance. So that at the end

$$P(x) = \frac{1}{\sqrt{2\pi\Delta}} e^{-(x-\mu)^2/2\Delta}$$

5 Échantillonnage direct dans la boule [6 points]

On considère une méthode pour générer uniformément des échantillons dans la boule (ou disque) en dimension d de rayon $B_d(1)$, où $B_d(r) = \{x \in \mathbb{R}^d \text{ s.t. } \|x\| \leq r\}$. Pour tirer une variable aléatoire X uniformément distribuée dans la boule, on sépare la partie angulaire de la partie radiale.

Pour la partie angulaire, nous générons $W \in \mathbb{R}^d$ avec $W \sim \mathcal{N}(0, \mathbb{I}_d)$. Ensuite, nous normalisons W de sorte qu'il se situe à la surface de la boule en prenant $Z = W/\|W\|$. Z est uniformément distribué sur la surface de la boule de rayon 1. Nous devons maintenant échantillonner le rayon. Pour ce faire, nous devons d'abord calculer la distribution de la norme de X (c'est-à-dire la distribution de la norme d'un point aléatoire uniforme dans la boule).

1. (2 points) Soit X uniformément distribué dans la boule $B_d(1)$. Calculez $P(\|X\| < r)$. *Indice : le volume de la boule en dimension d est $\text{Vol}(B_d(r)) = C_d r^d$, où C_d est une constante qui dépend uniquement de d .*

Solution: $P(\|X\| < r) = \frac{\text{Vol}(B_d(r))}{\text{Vol}(B_d(1))} = r^d$, we used a ratio of volumes argument.

2. (2 points) Soit $U \sim \text{Uniform}([0, 1])$. Calculez la distribution cumulative $P(Y < y)$ de la variable aléatoire $Y = U^a$, où $a > 0$.

Solution: $P(Y < y) = P(U^a < y) = P(U < y^{1/a}) = y^{1/a}$.

3. (2 points) Quelle est la valeur de a qui fait correspondre la distribution de Y à celle de $\|X\|$? Combinez la variable aléatoire Z introduite ci-dessus et la variable aléatoire Y (avec un choix approprié de a) pour obtenir une variable aléatoire X uniformément distribuée dans la boule $B_d(1)$.

Solution: $a = 1/d$. $X = YZ = U^{1/d}Z$.

6 Chaîne de Markov à deux états [5 points]

Considérons la chaîne de Markov suivante à deux états

$$T = \begin{pmatrix} 1-p & q \\ p & 1-q \end{pmatrix}. \quad (14)$$

où T_{ab} est la probabilité d'aller de b à a et $0 \leq p, q < 1$.

1. (3 points) Trouvez, si elle existe, une distribution stationnaire de cette chaîne pour tout $0 \leq p, q < 1$.

Solution: A distribution π is stationary if $T\pi = \pi$, i.e. if it is an eigenvector of T . This gives the equation

$$\begin{pmatrix} 1-p & q \\ p & 1-q \end{pmatrix} \begin{pmatrix} \pi_0 \\ \pi_1 \end{pmatrix} = \begin{pmatrix} \pi_0 \\ \pi_1 \end{pmatrix} \Rightarrow \begin{cases} (1-p)\pi_0 + q\pi_1 = \pi_0 \\ p\pi_0 + (1-q)\pi_1 = \pi_1 \end{cases} \quad (15)$$

Assuming $p, q > 0$, we find $\pi_0 = \frac{q}{p}\pi_1$ from either of these two equations. Additionally, we need $\pi_0 + \pi_1 = 1$ to have a proper distribution. This gives $\frac{q}{p}\pi_1 + \pi_1 = 1$ so that $\pi_1 = \frac{p}{q+p}$ and $\pi_0 = \frac{q}{q+p}$.

For $p = 0, q > 0$, we find $\pi_1 = 0$ and thus $\pi_0 = 1$, so the solution above still applies. Similarly, for $p > 0, q = 0$, we obtain $\pi_1 = 1$ and thus $\pi_0 = 0$.

Finally, we only have the constraint $\pi_0 + \pi_1 = 1$ for the case $p = q = 0$. Thus, all $0 \leq \pi_0, \pi_1 < 1$ that satisfy this constraint are stationary distributions.

2. (2 points) Sous quelle(s) condition(s) cette chaîne est-elle ergodique ? Donnez une brève explication.

Solution: The chain is ergodic for $p, q > 0$, as there is always a positive probability to go from one state to another. If $p = 0$, then the chain is not ergodic, as there is no path from state 0 to state 1. Similarly, the chain is not ergodic for $q = 0$.

7 Questions de programmation [14 points]

Cette section contient des questions de programmation. On suppose que l'on a importé la bibliothèque numpy sous l'alias np.

1. (3 points) L'on dispose d'une matrice A symétrique. On suppose que A admet 0 pour valeur propre, que cette valeur propre n'est pas dégénérée et que ses autres valeurs propres ne sont pas dans l'intervalle $[-10^{-7}, 10^{-7}]$. L'on vient de calculer son spectre à l'aide de numpy : `spectre = np.linalg.eigvals(A)`. On veut obtenir l'indice (la position) i dans `spectre` de la valeur propre 0. Parmi les propositions suivantes (a), (b), (c), (d) lesquelles sont correctes ? Lesquelles sont correctes et efficaces ?

(a)

```
i = np.where(np.isclose(spectre, 0))
# np.isclose(a,b) retourne vrai si les deux éléments sont égaux à 10e-6 près,
# faux sinon. Elle agit de manière vectorisée sur a et b.
# np.where(a) retourne les indices i pour lesquels a[i] vaut vrai.
```

(b)

```
i = np.argmax(spectre==0)
```

(c)

```
i = np.argmin(np.abs(spectre))
# np.abs(a) retourne la valeur absolue, de manière vectorisée sur a.
```

(d)

```
for j in range(len(spectre)):
    eps=10e-7
    if np.abs(spectre[j])<eps:
        i = j
```

Solution: d et c sont correctes. c est plus efficace. Les autres propositions peuvent ne pas fonctionner, selon les erreurs numériques commises par `np.linalg.eigvals` et que A possède d'autres valeurs propres inférieures à $10e-6$.

2. (3 points) Soit `u` un tableau numpy de réels à une dimension de taille `n`. On souhaite en extraire les dix premiers éléments non-nuls, ou extraire tous les éléments non-nuls s'il y en a moins que dix. Quelles fonctions parmi `f1`, `f2`, `f3`, `f4` et `f5` accomplissent cela correctement ?

```
def f1(u):
    u_nonNul = u[u>0]
    return u_nonNul[1:10]

def f2(u):
    u_nonNul = u[u>0]
    return u_nonNul[:10]

def f3(u):
    u_nonNul = []
    for élément in u:
        if élément>0:
            u_nonNul.append(élément)
    return np.array(u_nonNul[1:11])

def f4(u):
    n = len(u)
    u_sortie = []
    i = 0
    while i<n and len(u_sortie)<10:
        if u[i]>0:
            u_sortie.append(u[i])
            i += 1
    return np.array(u_sortie)

def f5(u):
    indices_nonNul = u>0
    u_sortie = []
    for i in indices_nonNul:
        u_sortie.append(u[i])
        if len(u_sortie)==10:
            break
    return u_sortie
```

Solution: `f2` et `f4` sont correctes. `f1` et `f3` : problème d'indices ; `f5` : `indices_nonNul` est un masque, pas une liste d'indices.

3. (3 points) On considère une fonction `expérience` qui simule une expérience aléatoire. Appelée sans arguments, la fonction retourne un tuple de 2 réels. On souhaite appeler `N` fois `expérience` et retenir les `N` résultats dans un tableau numpy bidimensionnel `sorties` de taille $(N,2)$. Parmi les propositions suivantes (a) – (f) lesquelles fonctionnent ?

(a)
`sorties = np.array([[0,0]])`
`for n in range(N):`
 `sorties += np.array(expérience())`
`sorties = sorties[1,:]`

(b)
`sorties = np.array([])`
`for n in range(N):`
 `sorties.append(np.array(expérience()))`
`sorties = np.array(sorties)`

(c)
`sorties = [0]*N`
`for n in range(N):`
 `sorties[n] = expérience()`

(d)
`sorties = np.array([expérience()]*N)`

(e)
`sorties = []`
`for n in range(N):`
 `sorties.append(expérience())`
`sorties = np.array(sorties)`

(f)
`sorties = np.zeros((N,2))`
`for n in range(N):`
 `sorties[n,:] = expérience()`

Solution: e et f. a l'addition est une addition élément par élément ; dans b `append` sur un tableau numpy n'est pas définie (on ne peut pas changer la taille d'un tableau) ; c donne une liste ; dans d on n'appelle `expérience` qu'une fois.

4. (2 points) Vous exécutez le code suivant:

```
import numpy as np
seed=np.random.randint(0,1000)
rs1 = np.random.RandomState(seed=seed)
a, b = rs1.uniform(), rs1.uniform()
rs2 = np.random.RandomState(seed=seed)
c, d = rs2.uniform(), rs2.uniform()
print(a,b,c,d)
```

Est-ce que $a=b$ en général ? Est-ce que $a=c$ en général ?

Solution: In general $a \neq b$ and $a = c$.

5. (3 points) Vous êtes chargé d'écrire un code python qui calcule le volume d'une sphère de rayon r en utilisant une méthode de Monte-Carlo basée sur l'échantillonnage du cube. Vous demandez à chatgpt de résoudre ce problème, et vous obtenez le code suivant:

```
import random

def sphere_volume_monte_carlo(radius, num_samples):
    inside_sphere = 0
    for _ in range(num_samples):
        x = random.uniform(-radius, radius)
        y = random.uniform(-radius, radius)
        z = random.uniform(-radius, radius)
        distance_squared = x**2 + y**2 + z**2
        if distance_squared <= radius**2:
            inside_sphere += 1
    cube_volume = (2 * radius)**3
    sphere_volume_estimate = (inside_sphere / num_samples) * cube_volume
    return sphere_volume_estimate
```

Vous consultez la documentation pour trouver la définition suivante de la fonction `uniform` du module `random`: *la méthode `uniform()` retourne un nombre flottant entre les deux valeurs spécifiées (les deux incluses)*.

Ce code est-il correct ? Indiquez toute les erreurs. Une fois les erreurs corrigées, est-ce que ce code est efficace ? Expliquez brièvement pourquoi. Si le code n'est pas efficace, expliquez en mots (pas de code) comment l'améliorer.

Solution: The code is correct and contains no error. The code is not efficient as the loop is slow and could be executed in parallel. For instance, using `numpy`, the computations (generating the points, checking if they are inside the sphere, and counting the number of points inside the sphere) can be vectorized.

8 Implémentation d'une descente de gradient [6 points]

On souhaite calculer numériquement le minimiseur de la fonction scalaire f définie sur \mathbb{R}^3 par

$$f(x, y, z) = \log\left(\frac{1}{2} + \frac{1}{2}(x^2 + xy + x - 4)^2 + 0.1x + \frac{1}{4}y^4\right) + \frac{1}{4}z^4 + z.$$

On utilise l'algorithme de descente de gradient. On considère l'implémentation en python suivante.

```
import numpy as np
def f(x, y, z):
    return np.log(1/2+(x**2+x*y+x-4)**2/2+0.1*x+y**4/4)+z**4/4+z
def grad_f(x, y, z):
    arglog = 1/2+(x**2+x*y+x-4)**2/2+0.1*x+y**4/4
    return ((2*x+y+1)*(x**2+x*y+x-4)+0.1)/arglog, (x*(x**2+x*y+x-4)+y**3)/arglog, z**3+1

x, y, z = 0, 0, 0
eta = 0.05
arrêt = False

while not arrêt:
    maj = grad_f(x, y, z)
    if np.linalg.norm(maj)<10e-6:
        arrêt = True
    x, y, z -= eta*maj

print(f(x, y, z))
```

- (2 points) L'interpréteur python retourne une erreur à l'exécution du code. Quelle ligne du code pose problème ? comment la modifier pour rendre le code correct ?

Solution: Le problème vient de `x, y, z -= eta*maj` : `x, y, z` et `maj` sont des tuples et l'on ne peut pas les additionner ainsi. Il faut les convertir en vecteur, en faisant par exemple `x, y, z = np.array([x,y,z])-eta*np.array(maj)`.

- (2 points) Une fois cette erreur corrigée, le programme s'arrête après un temps fini et affiche `-1.1854956817925237`. Or l'on sait que $f(-3, 0.5, -1) \approx -1.83$. Quel est le problème ? Comment le corriger ?

Solution: La descente de gradient a convergé vers un minimum local. Il faut l'initialiser depuis un autre point, `(-3, 0.5, -1)` par exemple.

- (2 points) Un autre problème apparaît : le programme ne s'arrête plus. Quel en est la cause ?

Solution: Le pas de descente `eta` est devenu trop grand pour pouvoir atteindre le minimiseur global ; il faut prendre une valeur plus faible, comme 0.01.