1 Pyzo

L'interface proposée à l'oral est Pyzo. Bien penser à s'habituer à l'interface avant l'oral, en particulier l'éditeur et la console. Savoir comment interpréter le script complet, ou bien seulement une partie de son code. Ne pas oublier d'enregistrer régulièrement son script.

2 Les aide-mémoire

Il est important de se familiariser avec les aide-mémoire, et de ne pas les découvrir le jour de l'oral. Comprendre en particulier ce que sont les arguments et les valeurs renvoyer par les différentes fonctions. Ne pas avoir de doute sur la façon d'importer un module, ou simplement une fonction de module.

3 Utiliser l'aide intégrée

Savoir utiliser l'aide intégrée à l'aide de `help`. Indiquer la fonction entre guillemet permet de ne pas charger le module. S'habituer aux différentes rubriques de l'aide, et savoir ce qu'est un argument optionnel d'une fonction.

```
>>> help("numpy.random.randint")

Help on built-in function randint in numpy.random:

numpy.random.randint = randint(...) method of numpy.random.mtrand.RandomState instance
    randint(low, high=None, size=None, dtype=int)

    Return random integers from `low` (inclusive) to `high` (exclusive).

    Return random integers from the "discrete uniform" distribution of
    the specified dtype in the "half-open" interval [`low`, `high`). If
    `high` is None (the default), then results are from [0, `low`).

    Parameters
    ----------
    low : int or array-like of ints
        Lowest (signed) integers to be drawn from the distribution (unless
        ``high=None``, in which case this parameter is one above the
        *highest* such integer).
    high : int or array-like of ints, optional
        If provided, one above the largest (signed) integer to be drawn
        from the distribution (see above for behavior if ``high=None``).
        If array-like, must contain integer values
    size : int or tuple of ints, optional
        Output shape.  If the given shape is, e.g., ``(m, n, k)``, then
        ``m * n * k`` samples are drawn.  Default is None, in which case a
        single value is returned.
    dtype : dtype, optional
        Desired dtype of the result. Byteorder must be native.
        The default value is int.

    Returns
    -------
    out : int or ndarray of ints
        `size`-shaped array of random integers from the appropriate
        distribution, or a single such random int if `size` not provided.
```

```
Examples
--------
>>> np.random.randint(2, size=10)
array([1, 0, 0, 0, 1, 1, 0, 0, 1, 0]) # random

Generate a 2 x 4 array of ints between 0 and 4, inclusive:
>>> np.random.randint(5, size=(2, 4))
array([[4, 0, 2, 1], # random
       [3, 2, 2, 0]])
```

Réalisation de tracés

- Tracé de la grille `matplotlib.pyplot.grid()`
- Repère orthonormal `matplotlib.pyplot.axis('equal')`
- Tracé de ligne brisé, de fonctions, de courbe paramétrée `matplotlib.pyplot.plot`
- Tracé de surface `numpy.meshgrid`, `mpl_toolkits.mplot3d.Axes3D`
- Tracé de lignes de niveau `numpy.meshgrid`, `matplotlib.pyplot.contour`

Analyse numérique

- Résoudre (numériquement) une équation comme $x^5 + x^2 + 1 = 0$ `scipy.optimize.fsolve`
- Résoudre (numériquement) un système comme $\begin{cases} x^2 - y^2 = 1 \\ x + 2y - 3 = 0 \end{cases}$ `scipy.optimize.root`
- Calculer (numériquement) une intégrale comme $\displaystyle\int_0^{+\infty} \mathrm{e}^{-t^2}\,\mathrm{d}t$ `scipy.integrate.quad`
- Résoudre (numériquement) une éq. diff. comme $y' = x + y^2$ `scipy.integrate.odeint`
- Résoudre (numériquement) une éq. diff. comme $y'' + xy' - y^3 = \cos x$ `scipy.integrate.odeint`
- Résoudre (numériquement) un syst. diff. comme $\begin{cases} x' = -x - y \\ y' = x - y \end{cases}$ `scipy.integrate.odeint`
- Calculer avec des nombres complexes comme $1 + \mathrm{i}$ `1+1j`

Polynômes

- Définir un polynôme comme $1 + X + X^2$ `X = numpy.polynomial.Polynomial([0,1])` `P=1+X+X**2`

La liste des opérations proposée ici est très incomplète, se référer à l'aide-mémoire

Calcul matriciel

- Créer des matrices, des vecteurs comme $\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{pmatrix}$ `numpy.array`
- Faire des produits de matrices `A.dot(B)`

- Utiliser la transposée d'une matrice `A.T`
- Obtenir les éléments propres d'une matrice `numpy.linalg.eig`

La liste des opérations proposée ici est très incomplète, se référer à l'aide-mémoire

Probabilités

- Simuler 20 lancers de dé `numpy.random.randint(1,7,20)`
- Simuler 20 valeurs d'une va suivant $\mathscr{U}(\llbracket 1, 7 \rrbracket)$ `numpy.random.randint(1,7,20)`
- Simuler une épreuve de Bernoulli de paramètre .2 `numpy.random.binomial(1,.2)`
- Simuler 42 valeurs d'une va suivant $\mathscr{G}(.7)$ `numpy.random.geometric(.7,42)`

La liste des opérations proposée ici est très incomplète, se référer à l'aide-mémoire

4 Exemples de sujets

641.1

Centrale II

On note :

$$F(x) = \int_0^{+\infty} \frac{\mathrm{e}^{-xt}}{1+t}\,\mathrm{d}t$$

(a) Tracer F. Faire des conjectures sur son signe, son sens de variations, sa limite en $+\infty$, son domaine de définition.

(b) Vérifier toutes les conjectures.

(c) c1. Montrer que f vérifie une équation différentielle du premier ordre à coefficients constants.
c2. Montrer que F est de classe $\mathcal{C}^\infty$.

(d) d1. Montrer que $F(x) = \mathrm{e}^x \displaystyle\int_x^{+\infty} \frac{\mathrm{e}^{-t}}{t}\,\mathrm{d}t$.
d2. Déterminer la limite de F en 0.

(e) Trouver un équivalent de F en 0.

Il y avait une erreur d'énoncé à la question 4. J'ai perdu du temps de préparation, mais pas de temps d'oral car l'examinateur l'a dit rapidement. Sinon, il était très sympathique et aidait pour la dernière question.

641.2

Centrale II

On note (E) l'équation différentielle :

$$\operatorname{sh}^5 x y' + 2 \operatorname{ch} x \operatorname{sh}^2 x y = \operatorname{ch}^3 x$$

On pourra utiliser `scipy.integrate.odeint`.

(a) a1. Quelle est la nature de cette équation différentielle ? Sur quel(s) domaine(s) l'étudier ?
a2. Résoudre l'équation homogène sur les domaines précédents.

(b) b1. Tracer avec Python la solution vérifiant $y(1) = 1$, sur $[1, 10]$, en utilisant un pas de 0.01. Conjecturer le comportement de la solution au voisinage de 0.

b2. On souhaite vérifier la conjecture. Tracer à rebours cette solution sur $[0.1, 1]$, toujours avec un pas de 0.01.

(c) c1. Déterminer une primitive de $t \mapsto (t-1)\mathrm{e}^t$.

c2. Déterminer la solution sur $\mathbb{R}_+^*$ de (E) qui satisfait la condition $y(1) = 1$.

c3. Existe-t-il une solution de (E) continue sur $\mathbb{R}$?

Il y avait trois ou quatre autres questions que je n'ai pas eu le temps de lire.

641.3

Soit $n \geqslant 1$ un entier et $a \neq 0$ un réel. On considère la matrice carrée d'ordre n :

$$A_{n,a} = \begin{pmatrix} 0 & \frac{1}{a} & 0 & \cdots & 0 \\ a & 0 & \frac{1}{a} & \ddots & \vdots \\ 0 & a & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & \frac{1}{a} \\ 0 & \cdots & 0 & a & 0 \end{pmatrix}$$

(a) Écrire une fonction qui étant donnés un entier $n \geqslant 1$ et un réel a non nul renvoie la matrice $A_{n,a}$.

(b) Donner des valeurs approchées décimales des valeurs propres de $A_{n,a}$ pour $3 \leqslant n \leqslant 8$ et a dans $\{-2, -1, 1, 2, 3\}$.

(c) Soit $(P_n)_{n\geqslant 1}$ la suite de polynômes définie par :

$$P_1 = X \qquad P_2 = X^2 - 1 \qquad \forall n \in \mathbb{N},\ P_{n+2} = XP_{n+1} - P_n$$

c1. Calculer les coefficients de $P_3, \ldots, P_8$.

c2. Donner des valeurs approchées des racines de $P_3, \ldots, P_8$.

c3. Conjecturer un lien entre P_n et A_n, a et le démontrer.

(d) Les matrices $A_{n,a}$ sont-elles inversibles ? diagonalisables ?

(e) Trouver un segment de $\mathbb{R}$ contenant toutes les valeurs propres de $A_{n,a}$ pour n entier et $a \in \mathbb{R}^*$.

641.4

Centrale II

Pour traiter ce sujet le candidat est invité à utiliser l'ordinateur à sa disposition, équipé de Python/Pyzo et de Scilab. Si on utilise Python, on pourra effectuer les suivants :

```
import numpy as np, numpy.linalg as alg
```

Pour une matrice $A \in \mathcal{M}_{n,p}(\mathbb{R})$ (où n et $p \in \mathbb{N}^*$), on note $A^\top$ la matrice transposée de A. On note également $\mathrm{Im}(A) = \{AX \mid X \in \mathcal{M}_{p,1}(\mathbb{R})$ et $\mathrm{Ker}(A) = \{X \in \mathcal{M}_{p,1}(\mathbb{R}) | AX = 0\}$, il s'agit respectivement de l'image et du noyau de l'application linéaire : $u_A\ :\ X \in \mathcal{M}_{p,1}(\mathbb{R}) \mapsto AX \in \mathcal{M}_{n,1}(\mathbb{R})$.
On note aussi pour $D \in \mathcal{M}_{n,1}(\mathbb{R})$, les systèmes linéaires $\Sigma_{A,D}\ :\ AX = D$ et $\Sigma'_{A,D}\ :\ A^\top AX = A^\top D$; il s'agit en fait d'équations matricielles d'inconnue $X \in \mathcal{M}_{p,1}(\mathbb{R})$.

A. Exemples numériques. On considère $A_1 = \begin{pmatrix} 1 & 0 & 1 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$, $D_1 = \begin{pmatrix} 0 \\ 5 \\ 10 \\ 5 \end{pmatrix}$, $A_2 = \begin{pmatrix} 0 & 2 \\ 0 & 1 \end{pmatrix}$, $D_2 = \begin{pmatrix} 10 \\ 5 \end{pmatrix}$ et $A_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$, $D_3 = \begin{pmatrix} 10 \\ 0 \\ 5 \end{pmatrix}$. Afin d'alléger les notations, on note respectivement Σ_i et Σ'_i les systèmes Σ_{A_i,D_i} et Σ'_{A_i,D_i}, pour $i \in \{1,2,3\}$.

(a) Résoudre les systèmes Σ_i et Σ'_i pour $i \in \{1,2,3\}$. *On pourra s'aider du logiciel.*

(b) b1. Écrire une fonction `rangs(A)` qui prend en argument un tableau `numpy` représentant une matrice A et qui renvoie la liste de longueur 2 contenant les rangs des matrices A et $A^\top A$.

b2. Tester la fonction précédente avec les matrices A_i pour $i \in \{1,2,3\}$. Que peut-on conjecturer ?

(c) Pour $i \in \{1,2,3\}$, comparer $\operatorname{Ker}(A_i)$ et $\operatorname{Ker}({A_i}^\top A_i)$. Conjecture ?

Désormais, on fixe $n, p \in \mathbb{N}^*$, $A \in \mathcal{M}_{n,p}(\mathbb{R})$ et $D \in \mathcal{M}_{n,1}(\mathbb{R})$ et on note respectivement Σ et Σ' les systèmes $\Sigma_{A,D}$ et $\Sigma'_{A,D}$.
Pour $q \in \mathbb{N}^*$, on munit $\mathcal{M}_{q,1}(\mathbb{R})$ du produit scalaire usuel défini par $\langle X, Y\rangle = X^\top Y$, en identifiant les matrice de $\mathcal{M}_1(\mathbb{R})$ avec leur unique coefficient et on note $\|\cdot\|$ la norme euclidienne associée.

B. Relation entre rang de A et de $A^\top A$.

(d) Montrer que $\operatorname{Ker}(A) = \operatorname{Ker}(A^\top A)$.

(e) En déduire une relation entre le rang de A et celui de $A^\top A$.

On dit que $X_0 \in \mathcal{M}_{p,1}(\mathbb{R})$ est une pseudo-solution du système Σ : $AX = D$ lorsque :

$$\forall X \in \mathcal{M}_{p,1}(\mathbb{R}),\ \|AX_0 - D\| \leqslant \|AX - D\|$$

Enfin, on introduit $D_0 \in \mathcal{M}_{n,1}(\mathbb{R})$, le projeté orthogonal de D sur le sous-espace vectoriel $\operatorname{Im}(A)$.

C. Existence de pseudo-solutions.

(f) Montrer que, si $X_0 \in \mathcal{M}_{p,1}(\mathbb{R})$ est une solution de Σ, alors X_0 est une pseudo-solution de Σ.

(g) Montrer que : $\forall Y \in \operatorname{Im}(A),\ \|Y - D\|^2 = \|Y - D_0\|^2 + \|D_0 - D\|^2$.

(h) Soit $X_0 \in \mathcal{M}_{p,1}(\mathbb{R})$. Établir que $AX_0 = D_0$ si et seulement si X_0 est une pseudo-solution de Σ.

(i) En déduire l'existence d'une pseudo-solution de Σ.

D. Calculs des pseudo-solutions et condition d'unicité.

(j) Soit $X_0 \in \mathcal{M}_{p,1}(\mathbb{R})$ telle que $AX_0 = D_0$. Montrer que :

$$\forall X \in \mathcal{M}_{p,1}(\mathbb{R}),\ \langle X, A^\top AX_0 - A^\top D\rangle = 0$$

(k) Montrer que les pseudo-solutions de Σ sont les solutions de Σ'.

(l) À quelle condition portant sur $\operatorname{rg}(A)$ le système Σ admet-il une unique pseudo-solution ?

641.5

Centrale II

On s'intéresse au lancer d'une pièce, pour laquelle on a une probabilité $p \in]0,1[$ d'obtenir pile. On note $q = 1 - p$. On effectue une succession de lancers indépendants et on définit l'événement :

$E_n =$ « Deux piles successifs n'ont pas été obtenus après le $n^{\text{ième}}$ lancer »

On note $P(E_n) = p_n$.

(a) Définir une fonction `En(n,p)` qui simule l'expérience et renvoie `True` si E_n est réalisé, `False` sinon.

(b) b1. Montrer que, pour tout $n \geqslant 0$:

$$p_{n+2} = qp_{n+1} + pqp_n$$

b2. Montrer que :

$B =$ « Deux piles successifs ont été obtenus »

est un événement presque sûr.

b3. On note T la v.a. qui vaut n lorsque deux piles successifs apparaissent pour la première fois au $n^{\text{ième}}$ lancer. Modifier la première fonction pour simuler T.

(c) c1. Calculer la fonction génératrice de T en fonction de p.

c2. Montrer que T admet une espérance et calculer $E(T)$.

c3. Proposer une méthode pour obtenir numériquement $E(T)$, et vérifier pour quelques valeurs de p l'espérance calculée à la question précédente.

Examinateur presque inaudible, mais bienveillant et poli. Il a écouté toutes mes propositions et me guidait vers la plus adéquate. Il semblait apprécier l'emploi de termes rigoureux de proba. Il a gentiment essayé de me piéger sur le rayon de convergence de G_X, dommage pour lui.