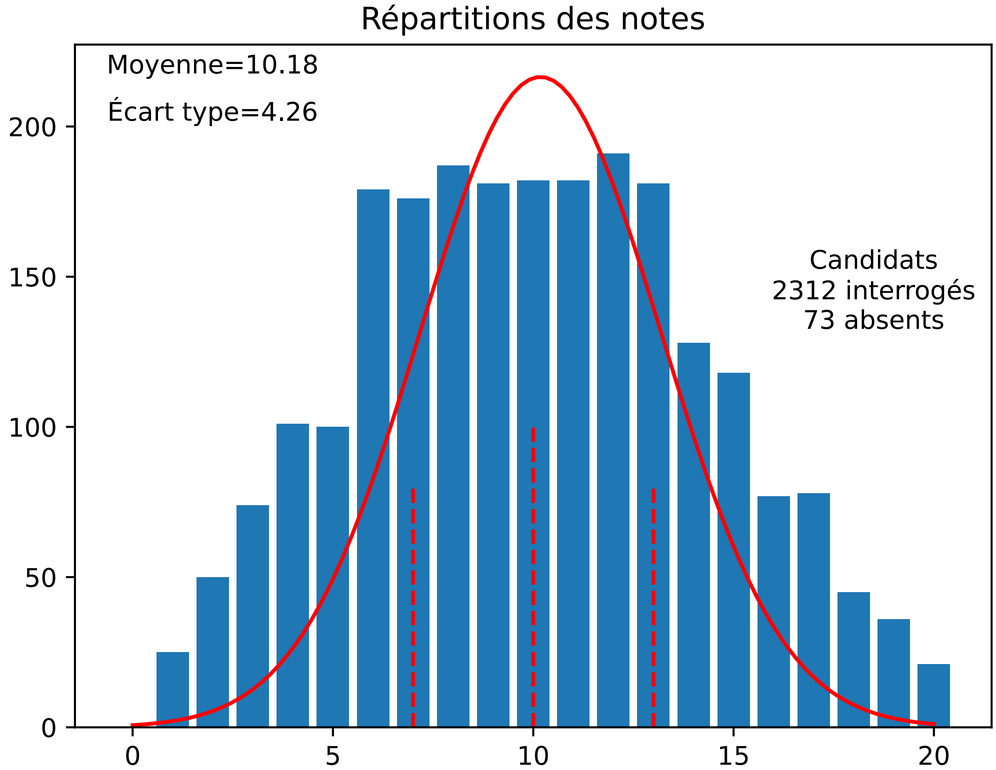
Épreuve de mathématiques pratiques et informatique
Rapport du jury – Session 2024

1 Modalités de l'épreuve

Le format de cette épreuve est le même que celui de la session passée. Il se compose d'un seul oral. Deux phases importantes sont à distinguer : la phase de préparation et l'oral en tant que tel.
Pendant les 40 minutes de la phase de préparation, les candidats prennent connaissance de la question de cours et d'un exercice. La question de cours et l'exercice n'ont aucun lien entre eux. La question de cours ne doit pas prendre beaucoup de temps lors de la préparation. L'exercice est composée de plusieurs questions autour d'un même thème, mais pouvant faire appel à différents domaines des mathématiques, certaines questions permettront d'illustrer les thèmes abordés ou de conjecturer des résultats à l'aide de l'outil informatique. Un ordinateur est donc mis à disposition de chaque candidat pendant sa préparation ainsi qu'une clé USB sur laquelle il enregistre ses programmes ; un formulaire Python est également disponible et les candidats peuvent s'y référer autant qu'ils le souhaitent.
L'oral dure également 40 minutes. Cependant, il se divise en trois étapes distinctes : la restitution de la question de cours, l'exposé et un échange sur l'exercice préparé et, enfin, un dernier exercice non préparé d'informatique.
La restitution de la question de cours ne dure pas plus de 2 minutes. Il peut être demandé aux candidats de préciser leur réponse, mais aucune indication sur l'exactitude de ses propos n'est fournie. Aucune justification ou démonstration des résultats liés à la question de cours n'est attendue.
Ensuite, pendant 28 minutes environ, le candidat expose ses résolutions ou ses pistes de résolution des questions de l'exercice préparé. Cet exposé est l'occasion d'un échange avec l'examinateur. L'examinateur peut demander de préciser les énoncés utilisés, de fournir les étapes d'une démonstration d'un théorème ou d'une proposition, mais aussi fournir des indications pour discuter des questions non résolues, voire non abordées pendant la préparation.
Les dix dernières minutes de l'oral sont consacrées à un exercice d'informatique dont l'énoncé, composé de deux questions, est fourni au candidat. Il est possible, si le temps le permet, de proposer une question supplémentaire au candidat.
Le candidat n'a pas à se soucier de la durée de chaque étape de l'oral. L'examinateur s'en charge. Ainsi il sera demandé à tout candidat venant avec un chronomètre de ranger ce dernier.
Pendant l'oral, le candidat est libre d'utiliser l'ordinateur mis à sa disposition ou d'écrire au tableau son ou ses programme(s) informatique(s).
Dans un souci d'équité, les candidats convoqués sur une même demi-journée sont interrogés sur les mêmes sujets. Les horaires de convocation prennent en compte cette contrainte et certains candidats sont ainsi invités à patienter dans une salle prévue à cet effet sans pouvoir communiquer avec l'extérieur, mais avec la possibilité de lire, de réviser et d'échanger entre eux sans bruit par respect des autres candidats.
Les sujets sont écrits dans le respect du programme officiel et en essayant, autant que possible, d'aborder les différents grands thèmes du programme : algèbre, analyse, probabilités, algorithmique.

2 Éléments statistiques



3 L'oral

3.1 Remarques générales

- Comme les années précédentes, nous avons encore une fois fortement apprécié que les candidats soient bien préparés à l'épreuve. Nous tenons une nouvelle fois à remercier les enseignants de BCPST, qui tiennent vraisemblablement compte des remarques issues des rapports précédents dans la préparation de leurs étudiants.
- Si le niveau des candidats est globalement correct, il est cependant à remarquer une très grande hétérogénéité : quelques candidats maîtrisent parfaitement les éléments du programme et sont clairement très à l'aise avec les notions abordées dans les sujets traités mais d'autres candidats (dans une proportion plus importante que l'année précédente) sont en grande difficulté avec les notions de base en mathématiques, le raisonnement logique, et peinent à mobiliser leurs connaissances.
- En général les méthodes classiques et les raisonnements fréquemment rencontrés au cours des deux années de préparation, que ce soit en algèbre, en analyse ou en probabilités, semblent bien assimilés par la majorité des candidats. Cependant cette assimilation s'apparente trop souvent à un apprentissage par cœur et donc une restitution sans aucune compréhension des enchaînements logiques ou sans aucune référence aux énoncés du cours. Certains candidats repèrent quelques mots clés dans les questions de l'énoncé et fournissent alors une démonstration conforme certainement à une démonstration rencontrée au cours de leur formation mais trop souvent sans lien avec l'énoncé proposé. Cette méthode de travail est à bannir complètement pendant la préparation aux concours. Nous ne pouvons qu'inciter les candidats à reprendre les exercices et entraînements effectués pendant leur préparation, à comprendre les étapes des différents raisonnements, à déterminer les énoncés utilisés et à vérifier clairement les hypothèses, mais pas à les apprendre par cœur !
- Comme indiqué depuis quelques années, les calculs représentent trop souvent des obstacles infranchissables pour bon nombre de candidats, les simplifications sont faites bien trop tardivement et les calculs, trop longs et menés sans discernement. Un temps d'analyse du résultat

obtenu à la fin d'un calcul est toujours profitable. Ainsi une erreur de signe entraînant une densité de probabilité négative ne sera pas préjudiciable et sera rapidement indiquée par l'examinateur si le candidat souligne l'incohérence de son résultat.

- Un échantillon de sujets donnés en 2024 est publié sur le site du jury en complément de ce rapport.
- Les exercices peuvent sembler très et même trop longs : ils le sont. La résolution complète de l'exercice n'est pas un attendu de l'examinateur ainsi une note maximale peut être obtenue sans avoir traité la totalité de l'exercice. Par contre, un raisonnement rigoureux et argumenté, reposant sur des connaissances solides est attendu, davantage que d'avoir donné tous les résultats (même justes) trop vite et sans explication convaincante. Mais il ne s'agit pas non plus de prendre trop de temps dans des explications inutiles, dans ce cas l'examinateur invite le candidat à accélérer.

3.2 Question de cours

Les questions de cours sont des éléments très classiques du programme et, en général, la formulation du programme officiel est conservée. Une réponse précise et rigoureuse est attendue, les hypothèses doivent apparaître clairement et les liens logiques explicités. Les candidats peuvent soit écrire cet énoncé au tableau, soit l'exposer oralement (attention alors à la précision des propos et du vocabulaire utilisé)
Aucune démonstration ni détail de calculs ne sont attendus ici.
Un temps très court est consacré à cette partie de l'oral, pas plus de 2 minutes.
Les questions de cours choisies n'ont pas pour but de piéger les candidats, mais bien de leur montrer l'importance d'un apprentissage précis et rigoureux des notions abordées pendant toute l'année.
Cependant, ces questions de cours ne sont réussies qu'à 68 %, un pourcentage en baisse par rapport à l'année dernière.
Nous ne pouvons qu'inciter les candidats à préparer cette première étape de l'oral pendant les deux années de préparation en vérifiant par différents moyens et, à tout instant, leurs connaissances du cours. Par exemple, il est assez surprenant de remarquer qu'une majorité des candidats interrogés ne connait pas la définition du gradient mais sait qu'un point critique se définit à l'aide du gradient et donne alors correctement le système vérifié par les coordonnées de ce point. Il serait judicieux lors de sa préparation qu'un candidat s'interroge sur la signification exacte des mots employés et qu'il n'agisse pas par automatisme.
L'effet n'est pas très positif quand la question de cours, première partie de l'oral, est imprécise ou bâclée, voire complètement farfelue.

3.3 Exercice préparé

Une résolution complète de l'exercice n'est pas un attendu, mais un exposé précis et rigoureux des questions résolues et une présentation des démarches suivies, des recherches faites dans les autres cas sont très appréciés.
L'examinateur rythme l'oral en incitant le candidat à poursuivre telle ou telle piste, ou à passer rapidement sur une question, en n'en donnant que les grandes étapes ou les arguments indispensables pour en développer une autre, il est également souvent amené à vérifier les connaissances du candidat en lui demandant de définir précisément les objets manipulés. Il est alors assez étonnant de constater que, dans le cadre de l'exercice, une simple question de cours sous la forme d'une définition pose de réels problèmes. Rappelons que toute résolution nécessite une bonne connaissance de la définition : ainsi, pour la recherche du noyau d'une matrice, donner directement un système sans faire le lien avec la définition du noyau ne peut pas satisfaire.
Dans l'échange entre le candidat et l'examinateur, la capacité du candidat à utiliser les indications de l'examinateur est prise en compte dans l'évaluation.
Nous invitons les candidats à bien lire l'ensemble de l'énoncé, à identifier les questions qui le rassurent, les questions qui peuvent prendre du temps par la réalisation de calculs et à les traiter dans le temps de préparation.

- L'algèbre semble être un chapitre peu compris ni apprécié des candidats. A nouveau, trop de candidats travaillent par automatisme et en apprenant des résolutions par cœur. Une référence aux définitions et aux théorèmes devrait être plus systématique. Partir sur la résolution d'un système pour déterminer un noyau sans faire le lien avec la définition du noyau, chercher les valeurs propres et les vecteurs propres pour prouver qu'une matrice est diagonalisable sans savoir à quoi mènera une telle recherche montre un manque sérieux de compréhension des outils utilisés et de recul par rapport à ces notions. Ainsi certains candidats sont très étonnés de découvrir qu'une matrice diagonale est diagonalisable.
  Nous ne pouvons qu'inciter les candidats à revenir très régulièrement lors de leur apprentissage aux définitions, aux propriétés, aux propositions et aux théorèmes. Il est important de prendre conscience qu'il n'existe pas une seule et une unique résolution à une question, mais un grand nombre et que chaque résolution a son intérêt, mais que certains arguments seront présents dans toutes les résolutions car ils sont indispensables. L'identification de tels arguments ne peut se faire que par une connaissance parfaite du cours et une manipulation variée des notions (en particulier jongler entre la matrice et l'endomorphisme associé).
  De nombreuses confusions sont à noter d'abord entre application linéaire, stabilité par combinaison linéaire, mais aussi entre application linéaire et espace vectoriel, ou encore entre les notions d'algèbre et d'analyse (la bijectivité est une notion commune, mais dont les théorèmes d'algèbre ne sont pas valables en analyse et vice versa, le théorème de la bijection monotone ne peut pas s'appliquer en algèbre).
  Notons que la définition de $\mathbb{R}_n[X]$ est peu connue, mais les candidats savent globalement manipuler des objets de cet ensemble.
  Les propriétés du produit scalaire sont trop souvent inconnues ou assez farfelues : $\langle x, y\rangle = |x|\,|y|$ ou $\langle x, y\rangle = 0 \Longleftrightarrow x = 0$ ou $y = 0$.
- Les probabilités représentent une partie du programme sur laquelle les candidats sont bien plus à l'aise. De nombreux réflexes sont en place et en général, ils s'appuient sur des connaissances. Utiliser la formule des probabilités totales, déterminer une espérance, une variance, montrer qu'une fonction est une densité ou encore utiliser le lien entre une fonction de répartition et une densité sont des aptitudes qu'une bonne partie des candidats ont bien acquises. Remarquons cependant que la notion de système complet d'événements est assez mal connue et son utilité assez incertaine.
  Pour certains candidats, il existe des variables aléatoires discrètes finies n'admettant pas d'espérance, ce qui les oblige à prouver la convergence d'une série qui est en fait une somme finie. A nouveau un peu plus de recul permettrait aux candidats d'éviter de se retrouver dans une situation assez délicate.
  La notion d'indépendance reste aussi très compliquée pour beaucoup : toutes les variables aléatoires ne sont pas indépendantes entre elles, en particulier peu de variables aléatoires sont indépendantes à elles-mêmes. La définition fait trop souvent défaut aux candidats qui tentent en général de prouver l'indépendance de deux variables aléatoires par un baratin sans beaucoup de sens.
  Il serait souhaitable que les candidats identifient clairement les quelques éléments du cours permettant de déterminer la probabilité d'une réunion finie ou infinie d'événements et la probabilité d'une intersection finie ou infinie. De même il est indispensable de connaître parfaitement les lois usuelles, leurs espérances et leurs variances.
  Une grande confusion existe chez de nombreux candidats entre la définition de la probabilité conditionnelle et la formule des probabilités composées mais aussi entre événements et probabilités, ou encore entre variables aléatoires et probabilités ou entre événements et variables aléatoires.
  A chaque fois qu'il est utilisé, le produit de convolution donnant une densité d'une somme de variables aléatoires indépendantes est donné. Cependant de nombreux candidats peinent à utiliser la formule donnée dans l'énoncé.
  Beaucoup d'énoncés de probabilités nécessitent une mise en situation un peu longue car elle se doit d'être très précise et sans ambiguïté. Il est alors très important que le candidat s'approprie

bien tous les éléments de l'énoncé et, pour cela, que l'énoncé soit lu plusieurs fois, que les candidats n'hésitent pas à revenir sur le début de l'énoncé quand un résultat est incohérent. Des dessins de la situation sont toujours les bienvenus ; traiter des cas pour des valeurs faibles ne peut qu'aider les candidats à mieux saisir l'énoncé.

- En analyse de nombreux théorèmes et méthodes sont plutôt bien acquis par les candidats. Cependant, leurs maîtrises sont trop souvent mises en défaut par des problèmes simples de calculs. En particulier l'utilisation de parenthèses n'est pas une option et modifie sérieusement le résultat. Le symbole de multiplication noté $\times$ se transforme aussi facilement en un signe $+$ ou en une variable x alors que très souvent aucun symbole ou un simple point pourrait très bien faire l'affaire et éviterait toute confusion. Les candidats doivent conserver une concentration et une attention dans leurs calculs, sans les précipiter, ni les détailler à outrance les rendant alors indigestes.
  Les notions de base en analyse ne sont pas toujours acquises, ainsi les expressions des dérivées et des primitives de fonctions usuelles sont très souvent confondues, et peu pensent à vérifier leur expression par le calcul.
  Une grande imprécision dans le vocabulaire est constatée. Dire que «x passe de l'autre côté» n'a pas de sens ; il serait plus correct de préciser de «x est additionné à chaque membre de l'égalité» par exemple. Ou encore il est préférable de dire que «en composant par l'exponentielle» au lieu de «on passe à l'exponentielle» .
  Les fonctions polynomiales restent peu maîtrisées en particulier la représentation graphique des fonctions polynomiales de degré 2 ou 3, la détermination des racines d'un polynôme de degré 2 quand il est écrit sous une forme factorisée ou quand il est une identité remarquable.
- En informatique, une très grande majorité des candidats se montre à l'aise avec l'écriture d'une fonction, la détermination du maximum sans utiliser la fonction `max`, le tri d'une liste, la simulation d'une variable aléatoire de Bernoulli, ou d'une variable aléatoire qui suit une loi binomiale ou géométrique.
  La méthode avec les sommes de Riemann pour déterminer une valeur approchée d'une intégrale est connue, mais peu, dans ce cadre, font le lien avec le théorème associé.
  Trop de candidats semblent cependant considérer que toute fonction doit contenir une boucle `for`.
  Contrairement à ce que pensent trop de candidats, quand une fonction a comme argument une liste `L` alors cette fonction ne doit pas commencer par `L=[]`, car ceci écraserait la liste prise en argument.
  Le tracé d'une fonction avec `matplotlib` n'est que trop peu acquis et donc trop peu utilisé. Les candidats peuvent représenter une fonction à l'aide du logiciel Python à tout moment de l'exercice si ce tracé permet d'illustrer leur propos.
  Les questions d'informatique doivent être traitées sur l'ordinateur pendant la préparation et le code enregistré sur la clé USB mise à disposition. Écrire du code au brouillon pour le recopier au tableau représente une perte de temps qui ne peut qu'être préjudiciable au candidat.

3.4 Exercice non préparé

Cet exercice est le plus souvent un exercice d'algorithmique. La première question est une question assez proche du cours et ne demande pas d'appréhender de nouvelles notions. Il est demandé au candidat de ne lire que cette première question et d'y répondre avant d'aborder la deuxième question.
En très grande majorité, la première question est bien traitée et ne pose pas de difficulté.
Nous pouvons remarquer cependant que les chaînes de caractères sont souvent mal connues des candidats et qu'ils hésitent à les manipuler.
Les candidats utilisent très peu les listes en compréhension.
Dans cet exercice, à nouveau, certains candidats se réfugient dans un apprentissage par cœur et fournissent des morceaux de programmes ou des programmes entiers qui, au mieux, ne respectent pas les notations imposées par l'énoncé ou ne répondent pas du tout aux questions posées.
La candidat choisit s'il expose son programme au tableau ou l'écrit à l'ordinateur. Notons que lors de cette session, $64,6$ % des candidats ont choisi l'ordinateur. L'utilisation de l'ordinateur peut donner

l'impression de plus facilement exécuter et tester le programme. Cependant il est possible de faire une vérification des programmes (qui ne comportent que quelques lignes) au tableau. Que le candidat choisisse l'ordinateur ou le tableau, il lui est souvent demandé quels tests simples pourraient être faits pour vérifier l'exécution du programme, puis éventuellement de vérifier son programme sur des exemples qui en éprouvent les limites.
Rappelons que l'absence de messages d'erreurs lors de l'exécution d'un programme ne signifie pas que le programme est correct et encore moins qu'il réponde à la question posée dans l'énoncé. Trop souvent les candidats ne prennent pas en compte les messages d'erreurs ; pourtant à la lecture de ces derniers, ils pourraient assez facilement dans la majorité des cas corriger eux-mêmes leur programme.

4 Formulaire Python

Le formulaire suivant est mis à la disposition des candidats dans la salle de préparation et dans la salle d'oral. Cette liste n'est bien entendu qu'indicative, et les candidats sont libres d'utiliser d'autres modules ou fonctions de leur choix s'ils les connaissent.
Nous attirons l'attention des candidats sur la fonction `randint`, dont le comportement n'est pas exactement le même selon qu'elle est appelée à l'aide de la bibliothèque random, ou de numpy.random.

PYTHON
AGRO-VETO
2024

Listes

`[]`	Créer une liste vide	`L.pop(k)`	Renvoie le $k^{\text{ème}}$ élément de la liste `L` et l'enlève de `L`
`[a]*n`	Créer une liste avec n fois l'élément `a`	`L.remove(a)`	Enlève une fois la valeur `a` de la liste `L`
`L.append(a)`	Ajoute l'élément `a` à la fin de la liste `L`	`max(L)`	Renvoie le plus grand élément de la liste `L`
`L1 + L2`	Concatène les deux listes `L1` et `L2`	`min(L)`	Renvoie le plus petit élément de la liste `L`
`len(L)`	Renvoie le nombre d'éléments de la liste `L`	`sum(L)`	Renvoie la somme de tous les éléments de la liste `L`

Numpy

`import numpy as np`

`np.array()`	Transforme une liste en matrice `numpy`
`np.linspace(a,b,n)`	Crée une matrice ligne de n valeurs uniformément réparties entre a et b (inclus)
`np.zeros([n,m])`	Crée la matrice nulle de taille $n \times m$
`np.eye(n)`	Crée la matrice identité de taille n
`np.diag(L)`	Crée la matrice diagonale dont les termes diagonaux sont les éléments de la liste `L`
`np.transpose(M)`	Renvoie la transposée de M
`np.dot(M,P)`	Renvoie le produit matriciel MP
`np.sum(M)`	Renvoie la somme de tous les éléments de M
`np.prod(M)`	Renvoie le produit de tous les éléments de M
`np.max(M)`	Renvoie le plus grand élément de M
`np.min(M)`	Renvoie le plus petit élément de M
`np.shape(M)`	Renvoie dans un couple le format de la matrice M
`np.size(M)`	Renvoie le nombre d'éléments de M
`np.arange(a,b,eps)`	Renvoie la liste des flottants de a à b de pas constant eps

Numpy.linalg

`import numpy.linalg as la`

`la.inv(M)`	Renvoie l'inverse de la matrice M si elle est inversible
`la.eigvals(M)`	Renvoie la liste des valeurs propres de M
`la.eig(M)`	Renvoie un couple `L,P` où `L` est la liste des valeurs propres de M et `P` la matrice de passage associée
`la.matrix_rank(M)`	Renvoie le rang de M

Random

`import random as rd`

`rd.random()`	Simule une réalisation d'une variable $X \hookrightarrow \mathcal{U}([0,1])$
`rd.randint(a,b)`	Simule une réalisation d'une variable $X \hookrightarrow \mathcal{U}([\![a,b]\!])$
`rd.gauss(0,1)`	Simule une réalisation d'une variable $X \hookrightarrow \mathcal{N}(0,1)$
`rd.choice(L)`	Choisit aléatoirement un élément de la liste `L`

Math

`import math as m`

`m.atan(x)`	Renvoie $\arctan(x)$	`m.sqrt(x)`	Renvoie $\sqrt{x}$ si $x \geq 0$
`m.floor(x)`	Renvoie $\lfloor x \rfloor$	`m.log(x)`	Renvoie $\ln(x)$ si $x > 0$
`m.factorial(n)`	Renvoie $n!$ si $n \in \mathbb{N}$	`m.exp(x)`	Renvoie e^x

Logique

`a == b`	Teste l'égalité « $a = b$ »
`a != b`	Teste « $a \neq b$ »
`a < b`	Teste « $a < b$ »
`a <= b`	Teste « $a \leq b$ »
`a > b`	Teste « $a > b$ »
`a => b`	Teste « $a \geq b$ »
`not A`	Renvoie la négation de A
`A and B`	Renvoie « A et B »
`A or B`	Renvoie « A ou B »
`True`	Constante booléenne « Vrai »
`False`	Constante booléenne « Faux »

Matplotlib.pyplot

`import matplotlib.pyplot as plt`

`plt.plot(X,Y,'+-r')` ---- Génère la courbe des points définis par les listes `X` et `Y` (abscisses et ordonnées) avec les options :

- symbole : `'.'` point, `'o'` rond, `'h'` hexagone, `'+'` plus, `'x'` croix, `'*'` étoile, ...
- ligne : `'-'` trait plein, `'--'` pointillé, `'-.'` alterné, ...
- couleur : `'b'` **bleu** , `'r'` **rouge** , `'g'` **vert** , `'c'` **cyan** , `'m'` **magenta** , `'k'` **noir** , ...

`plt.bar(X,Y)`	Génère l'histogramme des points définis par les listes `X` et `Y` (abscisses et ordonnées)
`plt.axis('equal')`	Rend le repère orthonormé
`plt.xlim(xmin,xmax)`	Fixe les bornes de l'axe des abscisses
`plt.ylim(ymin,ymax)`	Fixe les bornes de l'axe des ordonnées
`plt.show()`	Affiche le graphique

Cette liste est non exhaustive. Les candidats sont libres d'utiliser les commandes de leur choix.