Classes préparatoires aux grandes écoles

Programme de mathématiques de la classe de BCPST
2nde année

Programme de mathématiques pour la classe BCPST2

I – Préambule

Objectifs de la formation

En classe de BCPST2 l'objectif est, dans le cadre d'un approfondissement de la formation, d'amener l'étudiant à intégrer les différentes étapes permettant de résoudre un problème exprimable de façon mathématique. L'enjeu est la reformulation et la résolution de problèmes issus de contextes ou de réalités a priori non mathématiques (provenant souvent d'autres disciplines).

Ainsi sont mises en jeu diverses compétences. Certaines ont déjà été envisagées en première année (BCPST1), et sont consolidées en seconde année :

1. Engager une recherche, définir une stratégie.
2. Modéliser un phénomène à l'aide du langage mathématique.
3. Représenter, changer de registre.
4. Raisonner, démontrer, argumenter…
5. Calculer (symboliquement ou numériquement avec une calculatrice ou un ordinateur), maîtriser le formalisme mathématique.
6. Communiquer à l'écrit et à l'oral.

D'autres constituent des objectifs plus spécifiquement approfondis en seconde année, dans la perspective des concours :

— Identifier un problème sous différents aspects ;
— Mobiliser des connaissances scientifiques pertinentes ;
— Critiquer ou valider un modèle ou un résultat.

Buts visés

Le programme de mathématiques de BCPST2 approfondit celui de BCPST1, ce qui se traduit par les enjeux suivants.

— Consolider les acquis mathématiques de BCPST1, notamment en matière de calcul et raisonnement. Par souci de clarté, il a été choisi de numéroter de manière compatible les têtes de chapitre des programmes de BCPST1 et de BCPST2.
— Généraliser et compléter les concepts introduits en BCPST1.
— Mettre un accent particulier sur la notion de modélisation, où se confrontent les mathématiques et les autres sciences, notamment dans le cadre des T.I.P.E.

Équilibre entre compétences

Les différentes compétences sont développées puis évaluées (au cours de l'année puis lors des concours) en veillant à leur équilibre. On prend garde en particulier à ne pas surdévelopper une compétence par rapport à une autre.

Les capacités en calcul par exemple (point 5 ci-dessus), lorsqu'elles sont propres aux mathématiques, restent relativement simples, l'objectif n'étant pas ici d'aboutir à une virtuosité technique. On attend, en la matière, une maîtrise solide des calculs, concepts et théorèmes mathématiques, dans des situations courantes, sans pour autant négliger les autres compétences.

Contenu

Le programme de seconde année combine des révisions du programme de première année, des approfondissements de certaines parties et des nouveautés.

Les résultats mentionnés dans le programme seront admis ou démontrés selon les choix didactiques faits par le professeur ; pour certains résultats, marqués comme « admis », la présentation d'une démonstration en classe est déconseillée.

L'**analyse** apparait sous forme de révisions, de nouveautés (séries et intégrales généralisées) ou de compléments (équations différentielles). C'est ainsi que les séries sont introduites comme outil de base des probabilités, tandis que l'étude des intégrales généralisées est insérée dans la mise en place des variables aléatoires à densité ; l'usage de ces outils est limité aux contextes probabilistes et aux démarches de modélisation ; on évitera les développements artificiels ou purement techniques à ce propos.

En **algèbre linéaire**, le passage de $\mathbf{K}^n$ aux espaces vectoriels généraux permet d'élargir le champ d'action et de donner une vision géométrique des espaces de fonctions. Ce cadre plus systématique permet de donner un sens à l'étude des bases et changements de base qui sont fondamentaux pour aborder les valeurs propres et vecteurs propres des applications linéaires et des matrices ; cette dernière approche se limite à la diagonalisation pour s'en tenir à des phénomènes simples. En vue de nombreuses applications (optimisation, analyse de données), est proposée une présentation du produit scalaire dans $\mathbf{R}^n$, du théorème de projection orthogonale et du théorème spectral. La notion de sous-espaces supplémentaires ne figure pas au programme, mais dans bien des situations le théorème de la projection orthogonale fournit une approche similaire tout en permettant un calcul effectif.

L'étude des **probabilités** est donc un enjeu majeur du programme de seconde année. Le but de ce parcours est de mettre en place, de la manière la plus efficace possible, un contexte opérationnel permettant d'utiliser aussi bien des variables aléatoires discrètes prenant une infinité de valeurs (amenant notamment les lois géométrique et de Poisson) que des variables aléatoires à densité (dites « continues »), avec un accent particulier sur les variables gaussiennes. Pour maintenir le programme dans un volume raisonnable, les couples de variables aléatoires ne sont abordés que pour les variables discrètes, ce qui évite d'avoir à aborder les intégrales doubles. Les démarches de simulation de variables aléatoires sont fortement encouragées.

Quelques théorèmes limites en probabilités ainsi que la construction précise d'un **test d'hypothèse** en découlant (comparaison d'une moyenne ou d'une proportion expérimentale à sa valeur théorique) offrent un environnement propice à la simulation numérique et permettent aux étudiants qui en ont le besoin pour leurs TIPE d'aller plus loin sur ces questions.

La variété des modèles ainsi mis en place, combinés avec les différents théorèmes limites proposés, permet d'aborder de nombreuses applications dans les domaines les plus divers ; l'évocation de ces contextes applicatifs est un élément important de la formation et fait partie des buts visés. Comme dans le programme de première année, on signale par un symbole $\rightleftharpoons$ certaines situations particulières où un lien avec d'autres enseignements scientifiques est encouragé, permettant de donner corps aux démarches de modélisation et d'application pratique des mathématiques.

En prolongement des programmes de première année en mathématiques et informatique, le programme encourage la **démarche algorithmique** et le recours aux **outils informatiques** ; le maniement de ces outils fait partie intégrante de la formation et a toute sa place dans l'évaluation en cours d'année et lors des concours.

Pour ce qui concerne les **révisions**, la proposition de consolider les compétences acquises en première année par quelques exercices ne doit pas être prise dans un sens restrictif : des approches numériques, pouvant s'appuyer sur le programme d'informatique ou recourir à des outils logiciels ou des calculatrices, peuvent tout aussi bien renforcer la maîtrise des concepts et de leurs applications.

II – Programme de seconde année

La répartition en chapitres proposée ci-dessous (ainsi que l'agencement des chapitres de révisions) est fournie à titre indicatif et ne constitue pas une progression figée ou obligatoire. Les impératifs pédagogiques liés à la préparation aux concours peuvent justifier une organisation différente, sous réserve de maintenir une structure cohérente.

Révisions 1 – Suites

Exercices et situations illustrant le programme de première année (Analyse 1 et Analyse 5).
⇌ Exemples en lien avec le programme d'informatique.

Révisions 2 – Fonctions et dérivées

Exercices et situations illustrant le programme de première année (Analyse 2, Analyse 3, Analyse 6, Analyse 7, Analyse 9).
⇌ Exemples en lien avec le programme d'informatique.

Révisions 3 – Intégrales

Exercices et situations illustrant le programme de première année (Analyse 8).
⇌ Exemples en lien avec le programme d'informatique.

Révisions 4 – Equations différentielles

Exercices et situations illustrant le programme de première année (Analyse 4) ⇌ Exemples en lien avec le programme d'informatique.

Révisions 5 – Fonctions de deux variables

Exercices et situations illustrant le programme de première année (Analyse 10).

Analyse 1 – Séries réelles

Contenus	Commentaires
Sommes partielles, convergence d'une série, somme d'une série convergente.	La série est notée $\sum_{n\geq n_0} u_n$ ou plus succinctement $\sum u_n$. En cas de convergence, la somme de la série est notée $\sum_{n=n_0}^{+\infty} u_n$. La terminologie de « famille sommable » n'est pas donnée. La notion de reste d'une série est hors programme.
Combinaison linéaire de séries convergentes.	

Contenus (suite)	**Commentaires**
Thèorèmes de convergence pour deux séries à termes positifs u_n et v_n : • théorème de comparaison si $u_n \leq v_n$ à partir d'un certain rang, • si $u_n \sim v_n$, alors les séries $\sum u_n$ et $\sum v_n$ sont de même nature.	Tout autre critère de convergence est hors programme. Les résultats relatifs aux restes et sommes partielles sont hors programme.
Convergence et somme de la série géométrique $\sum\limits_{n\geq 0} q^n$ (pour $\lvert q\rvert < 1$) et des séries « dérivées » $\sum\limits_{n\geq 1} nq^{n-1}$ et $\sum\limits_{n\geq 2} n(n-1)q^{n-2}$.	
Convergence et somme de la série exponentielle $\sum\limits_{n\geq 0} \dfrac{x^n}{n!}$.	Résultat admis.
Convergence de $\sum\limits_{n\geq 1} \dfrac{1}{n^2}$ et divergence de $\sum\limits_{n\geq 1} \dfrac{1}{n}$.	L'étude générale des séries de Riemann est hors programme.
Convergence absolue.	La convergence absolue est présentée comme une condition suffisante pour obtenir la convergence de la série. En vue des applications probabilistes, on admet que la valeur de la somme d'une série absolument convergente ne dépend pas de l'ordre d'énumération de ses termes. L'étude de séries semi-convergentes est hors programme.

Analyse 2 – Intégrales généralisées

Contenus	**Commentaires**
Convergence d'une intégrale généralisée (ou impropre) d'une fonction continue sur un intervalle I semi-ouvert ou ouvert.	La convergence est traduite en termes de limites portant sur une primitive. La terminologie de « fonction intégrable » n'est pas donnée. Les notations $\int_I f$, $\int_I f(t)\mathrm{d}t$, $\int_a^b f$, $\int_a^b f(t)\mathrm{d}t$ pourront, selon le contexte, désigner l'intégrale généralisée ou sa valeur.
Cas d'une fonction définie sur un intervalle et continue sur cet intervalle sauf éventuellement en un nombre fini de points.	Cas particulier d'une fonction prolongeable par continuité en un point.
Propriétés des intégrales convergentes : linéarité, relation de Chasles, positivité, stricte positivité (f positive non nulle), croissance.	La démonstration de la stricte positivité n'est pas exigible.
Adaptation de l'intégration par parties aux intégrales généralisées.	On souligne la nécessité de confirmer la convergence de tous les termes apparaissant dans une telle formule.
Adaptation de la formule de changement de variable pour les intégrales généralisées.	Si la fonction φ est de classe $\mathscr{C}^1$ et strictement monotone sur un intervalle d'extrémités a et b ayant des limites $\alpha = \lim_a \varphi$ et $\beta = \lim_b \varphi$ et si f est continue sur l'intervalle d'extrémités α et β, alors les intégrales $\int_\alpha^\beta f(x)\mathrm{d}x$ et $\int_a^b f(\varphi(t))\varphi'(t)\mathrm{d}t$ sont de même nature, et ont la même valeur lorsqu'elles convergent.
Cas des fonctions paires ou impaires.	
Théorèmes de convergence pour deux fonctions positives f et g : • théorème par comparaison si $f \leq g$, • si $f(x) \underset{x\to b}{\sim} g(x)$, alors les intégrales généralisées en b $\int_a^b f$ et $\int_a^b g$ sont de même nature.	Tout autre critère de convergence est hors programme. Tout résultat sur la nature des intégrales de Riemann devra être démontré.

Contenus (suite)	Commentaires
Convergence absolue d'une intégrale généralisée.	La convergence absolue est présentée comme une condition suffisante pour obtenir la convergence de l'intégrale. Les intégrales semi-convergentes sont hors programme.
L'intégrale $\int_{-\infty}^{+\infty} \mathrm{e}^{-x^2/2}\mathrm{d}x$ converge et vaut $\sqrt{2\pi}$.	La valeur de cette intégrale est un résultat admis.

Analyse 3 – Équations différentielles scalaires autonome d'ordre 1

Contenus	Commentaires
Exemples de résolution d'équations différentielles autonomes du type $y'(t) = F(y(t))$, F étant une fonction continue sur un intervalle et à valeurs réelles.	Aucune théorie générale ne doit être faite. Toute étude devra être entièrement guidée. $\rightleftharpoons$ On se limite ici à quelques exemples issus de la biologie des populations ou de la cinétique chimique (modèles malthusien, logistique, de Gompertz). $\rightleftharpoons$ Lien avec l'informatique : programmation de la méthode d'Euler. Dans un énoncé, la méthode d'Euler sera rappelée.

Révisions 6 – Nombres complexes

Exercices et situations illustrant le programme de première année (Outils 3, Outils 4).

Révisions 7 – Systèmes linéaires et matrices

Exercices et situations illustrant le programme de première année (Algèbre linéaire 1 et 2).

Algèbre – Polynômes

Contenus	Commentaires
a) Polynômes, règles de calcul.	
Retour sur les polynômes réels : notation X pour l'application $x \mapsto x$ et réécriture d'un polynôme avec cette notation. On introduit les polynômes à coefficients dans $\mathbf{C}$. Notation X pour l'application $x \mapsto x$.	On remarque que les règles de calcul avec X prolongent les règles de calculs dans $\mathbf{R}$ ou $\mathbf{C}$.
Les opérations usuelles (combinaison linéaire, produit, composée) sur les polynômes fournissent des polynômes.	
Unicité de l'écriture des polynômes : un polynôme est nul si, et seulement si, tous ses coefficients sont nuls.	En conséquence, deux polynômes sont égaux si, et seulement si, ils ont les mêmes coefficients.
Coefficient dominant et degré d'un polynôme.	On convient que le polynôme nul est de degré $-\infty$.
Degré d'une somme, d'un produit de polynômes. Notations $\mathbf{R}[X], \mathbf{C}[X], \mathbf{R}_n[X], \mathbf{C}_n[X]$.	
b) Racines et factorisation.	
Définition d'une racine α d'un polynôme P : $P(\alpha) = 0$.	
Un nombre réel ou complexe α est racine d'un polynôme P si, et seulement si, il existe un polynôme Q tel que $P = (X - \alpha)Q$.	La division euclidienne des polynômes est hors programme.
Généralisation à plusieurs racines distinctes. Le nombre de racines distinctes d'un polynôme non nul est majoré par son degré.	

Contenus (suite)	Commentaires
Ordre de multiplicité d'une racine.	La caractérisation de la multiplicité d'une racine à l'aide des polynômes dérivés n'est pas un attendu du programme.
Cas des polynômes réels : si α est racine, $\overline{\alpha}$ est aussi racine.	
Théorème de d'Alembert-Gauss. Factorisation dans $\mathbf{C}[X]$.	Ce théorème est admis. La factorisation dans $\mathbf{R}[X]$ est hors programme.

Algèbre linéaire 1 – Espaces vectoriels

Ce chapitre reprend les concepts présentés en première année dans un cadre limité ($\mathbf{K}^n$) et les adapte brièvement à d'autres espaces, de dimension finie ou non.

La notion de somme de sous-espaces vectoriels n'est pas au programme.

On travaille uniquement dans des **K**-espaces vectoriels, **K** désignant **R** ou bien **C**. Lorsqu'un espace est un **C**-espace vectoriel, le considérer comme un **R**-espace vectoriel n'est pas un attendu du programme. Il n'est pas dans l'esprit du programme de rentrer dans des détails techniques comme parler de **R**-base, **C**-base, **R**-dimension, **C**-dimension.

Contenus	Commentaires
a) Structure vectorielle	
Structure d'espace vectoriel. Règles de calcul.	On met plus particulièrement en valeur les espaces vectoriels suivants : $\mathbf{K}^n$, $\mathscr{M}_{n,p}(\mathbf{K})$, l'ensemble des applications définies sur un intervalle I à valeurs dans $\mathbf{K}$, $\mathbf{K}[X]$, $\mathbf{K}_n[X]$. L'étude d'espaces de suites n'est pas un attendu du programme.
Combinaison linéaire d'une famille finie de vecteurs.	
Sous-espaces vectoriels.	
Intersection d'un nombre fini de sous-espaces vectoriels.	
Sous-espace vectoriel engendré par une famille finie de vecteurs.	On introduit la notation $\mathrm{Vect}(x_1, x_2, \ldots, x_k)$.
Famille génératrice finie d'un espace vectoriel (sous réserve d'existence).	
Famille libre finie. Famille liée finie.	
Exemple fondamental de famille libre : toute famille finie de polynômes non nuls de degrés deux à deux distincts est libre.	
Base finie d'un espace vectoriel (sous réserve d'existence). Coordonnées d'un vecteur dans une base.	
Matrice des coordonnées d'une famille finie de vecteurs dans une base.	
Bases canoniques de $\mathbf{K}^n$ et $\mathbf{K}_n[X]$.	D'autres exemples peuvent être proposés, mais les attendus du programme se limitent aux cas mentionnés.
b) Dimension	
On dit que E est de dimension finie s'il possède une famille génératrice finie. De toute famille génératrice finie d'un espace E non réduit au vecteur nul on peut extraire une base. Toutes les bases d'un espace vectoriel de dimension finie non réduit au vecteur nul E ont le même cardinal ; ce nombre commun est appelé dimension de E. Par convention, l'espace vectoriel réduit au vecteur nul est de dimension 0.	
Dans un espace vectoriel de dimension $n \geq 1$: • Toute famille libre peut se compléter en une base.	

Contenus (suite)	Commentaires
• Toute famille libre a au plus n éléments. • Une famille libre ayant n éléments est une base. • Toute famille génératrice a au moins n éléments. • Une famille génératrice ayant n éléments est une base.	Compte tenu des objectifs pédagogiques, la plupart de ces énoncés doivent être admis, mais on peut montrer comment certains de ces résultats peuvent en impliquer d'autres.
Si F est un sous-espace vectoriel de E, alors F est de dimension finie et $\dim F \leq \dim E$. Si les deux dimensions sont égales, alors $F = E$.	
Rang d'une famille finie de vecteurs.	Ce rang peut se calculer comme le rang de la matrice des coordonnées de la famille dans n'importe quelle base.

Algèbre linéaire 2 – Applications linéaires et matrices

Le passage aux espaces vectoriels quelconques pousse à redéfinir les notions liées aux applications linéaires. Il convient de faire cette adaptation avec une certaine brièveté afin de garder tout le temps requis pour traiter des exemples.

On travaille dans $\mathbf{K} = \mathbf{R}$ ou $\mathbf{C}$.

Contenus	Commentaires
a) Applications linéaires	
Application linéaire, endomorphisme, isomorphisme. Espaces isomorphes.	On introduit les notations $\mathscr{L}(E,F)$ et $\mathscr{L}(E)$, mais leur étude n'est pas un attendu du programme.
Opérations sur les applications linéaires : addition, multiplication par un scalaire, composition, réciproque. Propriétés de ces opérations.	Notation f^n pour $n \in \mathbf{N}$.
Noyau. Lien avec l'injectivité.	On montre que le noyau est un sous-espace vectoriel de l'espace de départ.
Image. Lien avec la surjectivité.	On montre que l'image est un sous-espace vectoriel de l'espace d'arrivée.
b) Cas de la dimension finie	
Détermination d'une application linéaire par l'image d'une base.	
Une application linéaire est un isomorphisme si, et seulement si, l'image d'une base est une base.	Tout espace de dimension n est isomorphe à $\mathbf{K}^n$.
Rang d'une application linéaire.	
Théorème du rang.	Résultat admis.
Pour une application linéaire entre deux espaces de même dimension finie, il y a équivalence entre l'injectivité, la surjectivité et la bijectivité.	On soulignera, à travers un exemple, que ce n'est pas le cas en dimension infinie. Toutefois, aucun exemple ne sera exigible des étudiants.
c) Matrices et applications linéaires	
Matrice d'une application linéaire d'un espace vectoriel de dimension finie dans un espace vectoriel de dimension finie, une base ayant été choisie dans chacun d'eux.	
Matrice de la somme de deux applications linéaires, du produit par un scalaire d'une application linéaire, de la composée de deux applications linéaires, de l'application réciproque.	On montre qu'un endomorphisme est bijectif si, et seulement si, sa matrice, dans une base quelconque, est inversible, et qu'il suffit pour cela de disposer d'une matrice inverse à gauche ou à droite.
Définitions du noyau et de l'image d'une matrice. Lien entre noyau et image d'une matrice et d'une application linéaire représentée par cette matrice dans des bases.	Toute identification entre vecteur de $\mathbf{K}^n$ et sa représentation matricielle dans une base, même la base canonique, est à éviter.

Contenus (suite)	Commentaires
d) Changement de base	
Changement de base. Matrice de passage. Action d'un changement de base sur les coordonnées d'un vecteur. Action d'un changement de base sur la matrice d'un endomorphisme.	
Matrices semblables.	On met en valeur l'intérêt des matrices semblables pour le calcul des puissances. On ne parlera pas de matrices équivalentes.

Algèbre linéaire 3 – Valeurs propres, vecteurs propres

Contenus	Commentaires
a) Éléments propres	
Valeurs propres, vecteurs propres, sous-espaces propres d'un endomorphisme. Valeurs propres, vecteurs propres, sous-espaces propres d'une matrice carrée.	On appelle spectre de l'endomorphisme f (respectivement de la matrice A) l'ensemble des valeurs propres de f (respectivement de A). En dimension finie, on fait le lien entre les éléments propres d'un endomorphisme et ceux d'une matrice qui le représente dans une base.
Les valeurs propres d'une matrice triangulaire sont les éléments diagonaux de cette matrice.	
b) Diagonalisation	
Une famille finie de vecteurs propres associés à des valeurs propres distinctes est libre. Une famille finie obtenue par juxtaposition de bases de sous-espaces propres associés à des valeurs propres distinctes est libre. En dimension finie, endomorphisme diagonalisable. Matrice diagonalisable. Un endomorphisme en dimension n ou une matrice carrée $n \times n$ est diagonalisable si, et seulement si, la somme des dimensions des sous-espaces propres est égale à n.	Un endomorphisme en dimension n ou une matrice carrée $n \times n$ admet au plus n valeurs propres deux à deux distinctes et la somme des dimensions des sous-espaces propres est inférieure ou égale à n.
Un endomorphisme en dimension n ou une matrice carrée $n \times n$ ayant n valeurs propres distinctes est diagonalisable.	On fait observer que les sous-espaces propres sont de dimension 1. La notion de polynôme annulateur est hors programme.

Révisions 7 – Géométrie

Exercices et situations illustrant le programme de première année (Géométrie 1).

Géométrie – Produit scalaire dans $\mathbf{R}^n$

Ce chapitre propose une extension modeste des notions de géométrie euclidienne à l'espace euclidien de dimension n, avec la notion de projection orthogonale sur un sous-espace et une application aux statistiques.

Contenus	Commentaires
a) Produit scalaire dans $\mathbf{R}^n$	
Produit scalaire usuel dans $\mathbf{R}^n$. Écriture matricielle. Bilinéarité.	

Contenus (suite)	**Commentaires**
Norme euclidienne. Inégalité de Cauchy-Schwarz et inégalité triangulaire. Cas d'égalité.	Le recours à l'inégalité de Cauchy-Schwarz devra être précisé.
Vecteurs orthogonaux.	Définition de deux matrices colonnes orthogonales.
Une famille de vecteurs non nuls deux à deux orthogonaux est libre.	
Théorème de Pythagore.	
Bases orthonormales de l'espace $\mathbf{R}^n$ ou d'un sous-espace de $\mathbf{R}^n$.	On souligne le fait que le produit scalaire et la norme se calculent de la même manière dans toutes les bases orthonormales. Les algorithmes d'orthonormalisation ne sont pas au programme.
b) Projection orthogonale	
Orthogonal $F^\perp$ d'un sous-espace vectoriel F de $\mathbf{R}^n$.	
L'ensemble $F^\perp$ est un sous-espace vectoriel de $\mathbf{R}^n$ et, pour tout $x \in \mathbf{R}^n$, il existe un unique couple $(x_F, x_{F^\perp}) \in F \times F^\perp$ vérifiant $x = x_F + x_{F^\perp}$.	On rappelle que les notions générales de sommes de sous-espaces vectoriels et de projections ne sont pas au programme. On admet qu'il existe une base orthonormale du sous-espace F dès que F n'est pas réduit au vecteur nul.
On appelle projection orthogonale sur le sous-espace F de $\mathbf{R}^n$ l'application p qui à tout $x \in \mathbf{R}^n$ associe x_F.	Écriture du projeté orthogonal d'un vecteur de $\mathbf{R}^n$ dans une base orthonormale de F.
La projection orthogonale sur le sous-espace F est l'endomorphisme p de $\mathbf{R}^n$ vérifiant $p \circ p = p$, $\mathrm{Im}(p) = F$ et $\mathrm{Ker}(p) = F^\perp$.	
Relation $\dim F + \dim F^\perp = n$.	
Distance entre deux vecteurs de $\mathbf{R}^n$.	
Définition de la distance d'un vecteur à une partie non vide de $\mathbf{R}^n$. Cas de la distance d'un vecteur à un sous-espace de $\mathbf{R}^n$.	
Interprétation en termes de projection orthogonale.	Interprétation de l'ajustement affine par la méthode des moindres carrés en termes de projection sur un sous-espace de dimension 2. La démonstration n'est pas exigible. Les coefficients de la droite de meilleure approximation au sens des moindres carrés devront être rappelés.
c) Théorème spectral	
Deux vecteurs propres associés à des valeurs propres distinctes d'une matrice symétrique réelle sont orthogonaux.	
Toute matrice symétrique réelle est diagonalisable en base orthonormale.	La démonstration de ce thorème est hors programme. On fera remarquer qu'il existe aussi des bases de diagonalisation non orthonormales. Les étudiants devront être guidés pour la construction effective d'une base orthonormale de vecteurs propres.

Probabilités 1 – Concepts de base des probabilités et des variables aléatoires

Ce chapitre étend le cadre des probabilités qui avait été posé en première année (Probabilités 1) pour aborder une situation plus générale, se prêtant à la définition des variables aléatoires discrètes ou à densité.

Les séries ont été introduites comme un outil pour donner tout leur sens aux probabilités et variables aléatoires discrètes. En dehors de questions probabilistes, les séries ne doivent être utilisées que de manière exceptionnelle et en lien avec des démarches de modélisation.

Contenus	Commentaires
a) Compléments ensemblistes et notion de probabilité	
Définition de $\bigcap_{n=0}^{+\infty} A_n$ et $\bigcup_{n=0}^{+\infty} A_n$.	
Notion de tribu.	On convient de nommer événements les éléments d'une tribu. Une tribu $\mathcal{T}$ (ou σ-algèbre) sur Ω est une partie de $\mathcal{P}(\Omega)$ contenant Ω, stable par passage au complémentaire et telle que, pour toute suite (B_n) d'événements, la réunion des B_n est un événement. Aucune question sur les tribus ne doit être proposée dans une épreuve de mathématiques.
Définition d'une probabilité sur $(\Omega, \mathcal{T})$.	On met en valeur l'axiome de σ-additivité $P\left(\bigcup_{n=0}^{+\infty} B_n\right) = \sum_{n=0}^{+\infty} P(B_n)$ pour des suites (B_n) d'événements deux à deux incompatibles, et on fait remarquer que la série $\sum_{n\geq 0} P(B_n)$ converge.
Définition d'un événement négligeable, d'un événement presque sûr.	On distingue l'événement impossible (resp. certain) des événements négligeables (resp. presque sûrs).
Révisions et extensions à ce nouveau cadre des propriétés des probabilités et des définitions vues en première année, en particulier :	
• Une suite d'événements (A_n) est un système complet d'événements si les A_n sont deux à deux incompatibles et si leur réunion est égale à Ω.	Pour une telle suite, on a $\sum_{n=0}^{+\infty} P(A_n) = 1$.
• Formule des probabilités totales : si (A_n) est un système complet d'événements, alors, pour tout événement B, la série $\sum_{n\geq 0} P(A_n \cap B)$ converge et $P(B) = \sum_{n=0}^{+\infty} P(A_n \cap B)$.	Cette formule reste valable dans le cas d'une suite (A_n) d'événements deux à deux incompatibles et tels que $\sum_{n=0}^{+\infty} P(A_n) = 1$; on dira dans ce cas que le système est quasi-complet. Interprétation en termes de probabilités conditionnelles, avec la convention suivante : si $P(A_n) = 0$, alors on pose $P(A_n) P_{A_n}(B) = 0$.
• Indépendance de deux événements. Indépendance (mutuelle) de n événements, d'une suite d'événements.	
b) Variables aléatoires réelles	
On nomme variable aléatoire réelle sur $(\Omega, \mathcal{T})$ toute application X de Ω dans $\mathbf{R}$ telle que, pour tout $a \in \mathbf{R}$, l'ensemble $\{\omega \in \Omega : X(\omega) \leq a\}$, noté $(X \leq a)$, soit un événement.	Aucune vérification du fait qu'une fonction est une variable aléatoire ne sera demandée dans une épreuve de mathématiques.
Si I est un intervalle de $\mathbf{R}$, alors $(X \in I) = \{\omega \in \Omega : X(\omega) \in I\}$ est un événement.	Résultat admis.
Fonction de répartition : $F_X : t \mapsto P(X \leq t)$. Croissance, limites en $\pm\infty$.	
Deux variables X et Y sont dites indépendantes si pour tous intervalles I et J, $P(X \in I \cap Y \in J) = P(X \in I)\, P(Y \in J)$.	
Généralisation au cas de n variables aléatoires, puis d'une suite de variables aléatoires.	

Probabilités 2 – Variables aléatoires réelles discrètes

L'ensemble de ce chapitre donne l'occasion de revoir, par le biais d'exercices, les lois de probabilités finies présentées dans le programme de première année (Probabilités 2).

Contenus	Commentaires
a) Variables aléatoires réelles discrètes	

Contenus (suite)	Commentaires
Une variable aléatoire réelle est dite discrète si l'ensemble $X(\Omega)$ de ses valeurs est inclus dans un sous-ensemble $\mathcal{N}$ de $\mathbf{R}$ indexé par une partie de $\mathbf{N}$.	On pourra utiliser le terme dénombrable mais ce terme n'est pas exigible. On met en valeur le système complet d'événements formé des événements $(X = x)$ pour $x \in \mathcal{N}$. On souligne la validité de la formule des probabilités totales obtenue.
Loi de probabilité et fonction de répartition d'une variable aléatoire discrète.	On décrit les représentations graphiques de ces deux fonctions. Les étudiants doivent savoir déterminer la loi d'une variable aléatoire à partir de sa fonction de répartition.
Si $(x_i)_{i\in\mathbf{N}}$ est une suite de réels deux à deux distincts et $(p_i)_{i\in\mathbf{N}}$ une suite de réels positifs tels que $\sum_{i\geq 0} p_i$ converge et a pour somme 1, alors il existe une variable aléatoire réelle discrète X vérifiant $P(X = x_i) = p_i$ pour tout entier naturel i.	On tolère qu'une variable aléatoire issue d'une expérience aléatoire puisse ne pas être définie sur un événement de probabilité nulle. $\rightleftharpoons$ En lien avec l'informatique : simulation d'une variable aléatoire discrète dont la loi est imposée, construite à partir d'une variable aléatoire uniforme.
b) Indépendance Deux variables aléatoires discrètes X et Y sont indépendantes si, et seulement si, $P(X = x, Y = y) = P(X = x)P(Y = y)$ pour tout $(x, y) \in X(\Omega) \times Y(\Omega)$. Généralisation : indépendance (mutuelle) de n variables aléatoires ; d'une suite de variables aléatoires. Propriétés de l'indépendance mutuelle : • Si $X_1, X_2, \ldots, X_n$ sont indépendantes, toute sous-famille l'est aussi.	
• Lemme des coalitions : si $X_1, \ldots, X_n, X_{n+1}, \ldots, X_{n+p}$ sont indépendantes, alors $u(X_1, \ldots, X_n)$ et $v(X_{n+1}, \ldots, X_{n+p})$ sont indépendantes.	On observera que cette propriété peut s'étendre à un nombre fini de fonctions s'appliquant à une partition des variables, et en particulier au cas de $(u_1(X_1), u_2(X_2), \ldots, u_n(X_n))$.
c) Espérance et variance	
Espérance. Propriétés (linéarité, positivité, croissance).	La linéarité de l'espérance est admise.
Théorème de transfert.	Ce résultat peut être admis.
Généralisation des propriétés et des définitions vues en première année, en particulier : • Inégalité de Markov. Inégalité de Bienaymé-Tchebychev. • Variance et moments d'une variable aléatoire. • Écart-type $\sigma(X)$ d'une variable aléatoire X. • Formule de König-Huygens $V(X) = E(X^2) - E(X)^2$. • Variance de $aX + b$. Notion de variable centrée réduite.	
• Si X est une variable aléatoire admettant une variance non nulle, $X^* = \dfrac{X - E(X)}{\sigma(X)}$ est une variable centrée réduite.	X^* est appelée variable centrée réduite associée à X.
• Si X et Y sont indépendantes, espérance de XY et variance de $X + Y$.	Résultat sur l'espérance admis. Généralisation au cas de n variables aléatoires indépendantes.
d) Lois usuelles discrètes Loi de Poisson. Espérance, variance.	
Loi géométrique. Espérance, variance. Propriété d'invariance temporelle ou d'absence de mémoire de la loi géométrique.	On présente la loi géométrique comme loi du nombre d'épreuves nécessaires pour obtenir le premier succès dans une suite illimitée d'épreuves de Bernoulli indépendantes et de même paramètre. $\rightleftharpoons$ Exemples de situations expérimentales modélisées par une loi géométrique.

Probabilités 3 – Couples de variables aléatoires discrètes

Ce chapitre permet, par le maniement de sommes de séries, d'appréhender les phénomènes liés aux couples de variables aléatoires : lois conjointes, lois marginales, indépendance. Cependant, le théorème de transfert est énoncé dans le seul cas des couples de variables aléatoires discrètes finies, et les séries doubles ne sont au programme.

Contenus	Commentaires
a) Couples de variables aléatoires réelles discrètes	
Couple (X,Y) de deux variables aléatoires discrètes. Loi conjointe.	L'événement $((X=x)\cap(Y=y))$ est également noté $(X=x,Y=y)$.
Lois marginales.	
Lois conditionnelles.	L'espérance conditionnelle n'est pas un attendu du programme.
b) Exemples de variable aléatoire de la forme $u(X,Y)$	
Sur des exemples simples, recherche de la loi de $u(X,Y)$, le couple (X,Y) ayant une loi conjointe connue.	On s'intéressera en particulier au maximum et au minimum de deux ou de n variables aléatoires indépendantes.
Cas particulier de la somme de deux variables discrètes à valeurs dans $\mathbf{N}$.	Les deux variables ne sont pas nécessairement indépendantes.
Loi de la somme de deux variables indépendantes suivant des lois de Poisson.	Généralisation au cas de n variables.
Théorème de transfert : espérance de $u(X,Y)$ à partir de la loi de (X,Y) quand X et Y sont des variables aléatoires discrètes finies.	Ce résultat peut être admis.
c) Covariance	
Covariance, formule de König-Huygens $\text{Cov}(X,Y)=E(XY)-E(X)E(Y)$ et calcul effectif quand X et Y sont discrètes finies.	Le calcul effectif de $E(XY)$ au moyen d'une série double n'est pas au programme.
Variance de $X+Y$.	On remarquera qu'en cas d'indépendance $\text{Cov}(X,Y)=0$, mais que la réciproque est fausse.

Probabilités 4 – Variables aléatoires à densité

Contenus	Commentaires
a) Variables aléatoires admettant une densité	
On appelle densité de probabilité une fonction f définie sur $\mathbf{R}$, positive, dont l'intégrale généralisée sur $\mathbf{R}$ converge et vaut 1.	Dans le cadre du programme, l'intégrale généralisée n'est définie que pour des fonctions continues sauf éventuellement en un nombre fini de points.
On dit qu'une variable aléatoire réelle X est à densité s'il existe une densité de probabilité f telle que, pour tout $x\in\mathbf{R}$: $F_X(x)=\displaystyle\int_{-\infty}^{x} f(t)\,\mathrm{d}t$.	Une telle fonction, qui n'est pas unique, est appelée densité de X.
F_X est dérivable en tout point de continuité x de f et $F'_X(x)=f(x)$	Ce résultat peut être admis. Dans ce contexte, donner la loi d'une variable aléatoire X, c'est justifier que X admet une densité et en donner une.
Si f est une densité de probabilité, alors il existe une variable aléatoire X dont f est une densité.	Résultat admis.

Contenus (suite)	**Commentaires**
X admet une densité si, et seulement si, sa fonction de répartition F_X est continue sur $\mathbf{R}$ et de classe $\mathscr{C}^1$ sur $\mathbf{R}$ sauf éventuellement en un nombre fini de points.	Ce résultat peut être admis. On insistera sur les représentations graphiques de la fonction de densité et de la fonction de répartition, en faisant le lien avec les histogrammes de variables aléatoires finies. Les étudiants doivent savoir déterminer la loi d'une variable aléatoire à partir de sa fonction de répartition. Sur des exemples simples, recherche de la loi de $u(X)$, X ayant une densité donnée.
b) Indépendance Propriétés de l'indépendance mutuelle : • Si $X_1, X_2, \ldots, X_n$ sont indépendantes, toute sous-famille l'est aussi. • Lemme des coalitions : si $X_1, \ldots, X_n, X_{n+1}, \ldots, X_{n+p}$ sont indépendantes, alors $u(X_1, \ldots, X_n)$ et $v(X_{n+1}, \ldots, X_{n+p})$ sont indépendantes.	On observera que cette propriété peut s'étendre à un nombre fini de fonctions, et en particulier au cas de $(u_1(X_1), u_2(X_2), \ldots, u_n(X_n))$. Exemples de recherche de la loi du minimum et du maximum de deux ou de n variables aléatoires indépendantes.
c) Espérance Espérance. Propriétés. Notion de variable centrée.	La linéarité de l'espérance est admise. Par extension, on pourra appliquer la linéarité de l'espérance à des variables aléatoires, qu'elles soient discrètes ou à densité, sans savoir si leur résultante est discrète ou à densité.
Théorème de transfert : si X est une variable aléatoire à densité et u est une fonction définie sur un intervalle I contenant $X(\Omega)$, continue sauf éventuellement en un nombre fini de points, alors $u(X)$ admet une espérance si, et seulement si, $\int_I u(x) f(x)\,\mathrm{d}x$ est absolument convergente. Le cas échéant, $E(u(X)) = \int_I u(x) f(x)\,\mathrm{d}x$.	Résultat admis. On pourra appliquer ce théorème sans savoir si $u(X)$ est une variable aléatoire discrète ou à densité.
Propriétés : • Inégalité de Markov. Inégalité de Bienaymé-Tchebychev.	On pourra appliquer ce théorème dès lors que la variable aléatoire admet une variance, sans savoir si elle est discrète ou à densité.
• Variance et moments. • Écart-type $\sigma(X)$ d'une variable aléatoire X. • Formule de König-Huygens $V(X) = E(X^2) - E(X)^2$. • Variance de $aX + b$. Notion de variable centrée réduite. • Si X est une variable aléatoire admettant une variance non nulle, $X^* = \dfrac{X - E(X)}{\sigma(X)}$ est une variable centrée réduite.	X^* est appelée variable centrée réduite associée à X.
• Si X et Y sont indépendantes, espérance de XY et variance de $X + Y$.	Résultat sur l'espérance admis. Par extension, on pourra appliquer ces formules à des variables aléatoires, qu'elles soient discrètes ou à densité, sans savoir si leurs résultantes XY et $X+Y$ sont discrètes ou à densité. Généralisation au cas de n variables aléatoires indépendantes.
d) Lois usuelles Loi uniforme : densité, fonction de répartition, espérance, variance.	

Contenus (suite)	**Commentaires**
Loi exponentielle : densité, fonction de répartition, espérance, variance. Propriété d'invariance temporelle ou d'absence de mémoire : $P(X \geq s+t \mid X \geq s) = P(X \geq t)$ et on donne quelques exemples d'expériences donnant du sens à cette propriété.	$\rightleftharpoons$ Une variable aléatoire de loi exponentielle peut être simulée à partir d'une variable aléatoire suivant la loi uniforme sur $]0,1[$.
Loi normale (ou gaussienne) centrée et réduite : densité, espérance et variance.	$\rightleftharpoons$ On obtient les valeurs de la fonction de répartition (notée souvent Φ) et de sa réciproque au moyen de la calculatrice ou d'une bibliothèque associée à un langage de programmation. Un échantillon de valeurs utiles devra être rappelé. $\rightleftharpoons$ Une variable aléatoire de loi normale peut être simulée à partir d'une variable aléatoire suivant la loi uniforme sur $]0,1[$.
Loi normale de paramètres μ et σ^2 : densité, espérance et variance.	
Si X suit une loi normale, alors $aX+b$ aussi si $a \neq 0$.	Pour une variable de loi $\mathcal{N}(\mu,\sigma^2)$, on se ramènera le plus souvent à la variable centrée réduite associée.
e) Sommes de variables aléatoires à densité indépendantes	
Loi de la somme de deux variables indépendantes à densité.	Le résultat est admis. La formule du produit de convolution devra être rappelée en cas de besoin. La démonstration de la convergence de l'intégrale, le cas échéant, n'est pas attendue des étudiants.
Somme de deux variables aléatoires normales indépendantes.	Le calcul montrant la normalité de la somme n'est pas un attendu du programme. On généralise le résultat au cas de n variables gaussiennes indépendantes.

Probabilités 5 – Théorèmes limites

Contenus	**Commentaires**
a) Loi faible des grands nombres	
La moyenne empirique d'un n-uplet de variables aléatoires $(X_1,\dots,X_n)$, notée M_n, est définie par $M_n = \frac{1}{n}\sum_{i=1}^{n} X_i$.	
Loi faible des grands nombres pour des variables aléatoires mutuellement indépendantes.	La définition générale de la convergence en probabilité n'est pas un objectif du programme.
b) Convergence en loi	
Définition de la convergence en loi d'une suite de variables aléatoires (X_n) vers une variable aléatoire X. Cas particulier où les X_n prennent leurs valeurs dans $\mathbf{N}$.	
Convergence en loi d'une suite de variables aléatoires de lois binomiales vers une variable aléatoire de loi de Poisson.	Approximations qui en découlent. Les critères d'approximation devront être explicités.
Théorème central limite (première forme) : si $(X_n)_{n\geq 1}$ est une suite de variables aléatoires indépendantes de même loi, admettant une espérance μ et une variance σ^2 non nulle, alors $(M_n^*)_{n\geq 1}$ converge en loi vers une variable aléatoire suivant la loi normale centrée réduite. Cas de la loi binomiale : théorème de de Moivre-Laplace.	Théorème admis. On rappelle que $M_n^* = \dfrac{M_n-\mu}{\frac{\sigma}{\sqrt{n}}}$ est la variable aléatoire centrée réduite associée à M_n. $\rightleftharpoons$ On illustre numériquement cette convergence.
L'écart-type empirique d'un n-uplet de variables aléatoires $(X_1,\dots,X_n)$, noté S_n, est défini par $S_n^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i-M_n)^2$.	

Contenus (suite)	Commentaires
Théorème central limite (seconde forme) : Si $(X_n)_{n\geq 1}$ est une suite de variables aléatoires indépendantes de même loi, admettant une espérance μ et une variance, alors $\left(\frac{M_n-\mu}{\frac{S_n}{\sqrt{n}}}\right)_{n\geq 1}$ converge en loi vers une variable aléatoire suivant la loi normale centrée réduite.	Théorème admis. Une autre version de ce théorème, impliquant l'écart-type empirique corrigé S'_n défini par ${S'_n}^2 = \frac{1}{n-1}\sum\limits_{i=1}^{n}(X_i - M_n)^2$, pourra être donnée.
c) Introduction aux tests Test de conformité à la moyenne.	On traitera le cas particulier d'une proportion par majoration de l'écart-type. Les notions de risque α ou β, de puissance ne sont pas au programme. $\rightleftharpoons$ En lien avec l'informatique, mécanisme et simulation de tests statistiques.

Classes préparatoires aux grandes écoles

Filière scientifique

Voie Biologie, chimie, physique et sciences de la Terre (BCPST)

Annexe 4

Programmes d'informatique

$1^{\text{ère}}$ et 2^{nde} années

Table des matières

Introduction au programme

Les objectifs du programme Le programme d'informatique de BCPST s'inscrit entre deux continuités : en amont avec les programmes rénovés du lycée, en aval avec les enseignements dispensés dans les grandes écoles, et plus généralement les poursuites d'études universitaires. Il a pour objectif la formation de futurs ingénieures et ingénieurs, vétérinaires, enseignantes et enseignants, chercheuses et chercheurs et avant tout des personnes informées, capables de gouverner leur vie professionnelle et citoyenne en pleine connaissance et maîtrise des techniques et des enjeux de l'informatique et en la nourrissant par les habitudes de la démarche scientifique.
Le présent programme a pour ambition de poser les bases d'un enseignement cohérent et mesuré d'une science informatique encore jeune et dont les manifestations technologiques connaissent des cycles d'obsolescence rapide. On garde donc à l'esprit :

- de privilégier la présentation de concepts fondamentaux pérennes sans s'attacher outre mesure à la description de technologies, protocoles ou normes actuels;
- de donner aux futurs diplômées et diplômés les moyens de réussir dans un domaine en mutation rapide et dont les technologies qui en sont issues peuvent sauter brutalement d'un paradigme à un autre très différent;
- de préparer les étudiantes et étudiants à tout un panel de professions et de situations de la vie professionnelle qui les amène à remplir tour à tour une mission d'expertise, de création ou d'invention, de prescription de méthodes ou de techniques, de contrôle critique des choix opérés ou encore de décision en interaction avec des spécialistes;

Compétences visées Ce programme vise à développer les six grandes compétences suivantes :

analyser et modéliser un problème ou une situation, notamment en utilisant les objets conceptuels de l'informatique pertinents (table relationnelle, graphe, dictionnaire, etc.) ;

imaginer et concevoir une solution, décomposer en blocs, se ramener à des sous-problèmes simples et indépendants, adopter une stratégie appropriée, décrire une démarche, un algorithme ou une structure de données permettant de résoudre le problème;

décrire et spécifier les caractéristiques d'un processus, les données d'un problème, ou celles manipulées par un algorithme ou une fonction;

mettre en œuvre une solution, par la traduction d'un algorithme ou d'une structure de données dans un langage de programmation ou un langage de requête;

justifier et critiquer une solution, en développant des processus d'évaluation, de contrôle, de validation d'un code que l'on a produit;

communiquer à l'écrit ou à l'oral, présenter des travaux informatiques, une problématique et sa solution; défendre ses choix; documenter sa production et son implémentation.

La pratique régulière de la résolution de problèmes par une approche algorithmique et des activités de programmation qui en résultent constitue un aspect essentiel de l'apprentissage de l'informatique. Les exemples ou les exercices d'application peuvent être choisis au sein de l'informatique elle-même ou en lien avec d'autres champs disciplinaires.

Sur les partis pris par le programme Ce programme impose aussi souvent que possible des choix de vocabulaire ou de notation de certaines notions. Les choix opérés ne présument pas la supériorité de l'option retenue. Ils ont été précisés dans l'unique but d'aligner les pratiques d'une classe à une autre et d'éviter l'introduction de longues définitions récapitulatives préliminaires à un exercice ou un problème. De même, ce programme nomme aussi souvent que possible l'un des algorithmes possibles parmi les classiques qui répondent à un problème donné. Là encore, le programme ne défend pas la prééminence d'un algorithme ou d'une méthode par rapport à un autre mais il invite à faire bien plutôt que beaucoup.

Sur les langages et la programmation L'enseignement du présent programme repose sur un langage de manipulation de données (SQL) ainsi que le langage de programmation Python, pour lequel une annexe liste de façon limitative les éléments qui sont exigibles des étudiants. La poursuite de l'apprentissage du langage Python est vue en particulier par les étudiants pour adopter immédiatement une bonne discipline de programmation tout en se concentrant sur le noyau du langage plutôt que sur une interface de programmation applicative (API) pléthorique.

Mode d'emploi Ce programme a été rédigé par semestre pour assurer une certaine homogénéité de la formation. Le premier semestre permet d'asseoir les bases de programmation vues au lycée et les concepts associés. L'organisation de la progression au sein des semestres relève de la responsabilité pédagogique de la professeure ou du professeur et le tissage de liens entre les thèmes contribue à la valeur de son enseignement. Les notions étudiées lors d'un semestre précédent sont régulièrement revisitées tout au long des deux années d'enseignement. Le programme est présenté sous forme de tableaux à deux colonnes. Les premières colonnes - notions et thèmes - présentent les attendus exigibles du programme. Les secondes colonnes présentent des listes, sans aucun caractère impératif, d'exemples d'activités qui peuvent être proposées aux étudiants ainsi que des commentaires.

1 Programme du semestre 1

Les séances de travaux pratiques du premier semestre poursuivent les objectifs suivants :
- consolider l'apprentissage de la programmation en langage Python qui a été entrepris dans les classes du lycée ;
- mettre en place un environnement de travail ;
- mettre en place une discipline de programmation : spécification précise des fonctions et programmes, annotations et commentaires, jeux de tests ;

La consolidation du langage Python portera principalement sur
- les variables
- les expressions et instructions
- les instructions conditionnelles
- les fonctions
- les instructions itératives
- la manipulation de quelques trsuctures de données

Rappelons que l'annexe liste les éléments du langage Python qui sont exigibles des étudiants.

Le tableau ci-dessous présente les thèmes qui sont abordés lors de ces séances. L'ordre de ces thèmes n'est pas impératif.

Aucune connaissance relative aux modules éventuellement rencontrés lors de ces séances n'est exigible des étudiants.

Thèmes	**Exemples d'activité. *Commentaires.***
Algorithmes élémentaires opérant par boucles simples.	Calculs de sommes et produits. Calculs des termes d'une suite récurrente (ordre 1, ordre supérieur), liste des termes, chaînes de caractères.
Algorithme opérant par boucles dans un tableau unidimensionnel.	Recherche d'un élément. Recherche du maximum, du second maximum. *Manipulations élémentaires d'un tableau unidimensionnel (indexation, extraction, etc.).*
Lecture et écriture dans un fichier texte.	
Utilisation de modules, de bibliothèques.	Calculs statistiques sur des données. Représentation graphique (histogrammes, etc.).
Algorithmes opérant par boucles imbriquées.	Recherche d'un facteur (ou d'un mot) dans un texte. Recherche des deux valeurs les plus proches dans un tableau. Manipulations élémentaires des tableaux à deux dimensions (indexation et extraction, etc.). Calculs de la somme, du produit et de la transposée d'une matrice. *On en profitera pour introduire la bibliothèque « NUMPY ».*
Recherche dichotomique.	Recherche de valeurs approchées d'une racine d'une équation algébrique. Recherche dichotomique dans un tableau trié. *On met en évidence une accélération du processus.*
Fonctions récursives.	Factorielles, suites récurrentes. Algorithme d'exponentiation rapide. Dessins de fractales. *On évite de se cantonner à des fonctions mathématiques.*
Matrices de pixels et images.	Obtention d'une image en niveaux de gris, image miroir, négatif. Algorithmes de rotation, de réduction ou d'agrandissement. Modification d'une image par convolution : flou, détection de contour, etc. *0n pourra utiliser la bibliothèque « PIL ».*
Tris.	Algorithmes naïfs : tri par insertion, par sélection. Tri par comptage. Application aux statistiques : tri d'une série statistique, recherche de la médiane (éventuellement des quartiles). *0n pourra faire le lien entre le tri par comptage et la recherche des effectifs d'apparition dans une liste.*

2 Programme du semestre 2

On approfondit, via les leçons et travaux pratiques, le travail entrepris au premier semestre concernant la discipline et les méthodes de programmation.

2.1 Méthodes de programmation et dictionnaire

Même si on ne parle pas de preuve d'algorithme, on insistera sur l'importance des tests dans la mise au point des programmes.

Notions	**Exemples d'activité. *Commentaires.***
Identifiants et valeurs. Objets mutables et non mutables, portée d'un identifiant, effets de bord.	*On mettra en évidence le phénomène d'aliasing et son impact dans le cas d'objets mutables (listes).*
Dictionnaires, clés et valeurs. Usage des dictionnaires en programmation Python. Syntaxe pour l'écriture des dictionnaires. Parcours d'un dictionnaire.	Nombre d'éléments distincts dans une liste, construction d'un index.

2.2 Bases de données

On se limite volontairement à une description applicative des bases de données en langage SQL. Il s'agit de permettre d'interroger une base présentant des données à travers plusieurs relations.

Notions	**Exemples d'activité. *Commentaires.***
Vocabulaire des bases de données : tables ou relations, attributs ou colonnes, domaine, schéma de tables, enregistrements ou lignes, types de données.	*On présente ces concepts à travers de nombreux exemples. On s'en tient à une notion sommaire de domaine : entier, flottant, chaîne ; aucune considération quant aux types des moteurs SQL n'est au programme. Aucune notion relative à la représentation des dates n'est au programme ; en tant que de besoin on s'appuie sur des types numériques ou chaîne pour lesquels la relation d'ordre coïncide avec l'écoulement du temps. Toute notion relative aux collations est hors programme ; en tant que de besoin on se place dans l'hypothèse que la relation d'ordre correspond à l'ordre lexicographique usuel.* `NULL` *est hors programme.*
Clé primaire, clé étrangère	*On se limite au cas où une clé primaire est associée à un unique attribut.*
Requêtes `SELECT` avec simple clause `WHERE` (sélection), projection, renommage `AS`. Utilisation des mots-clés `DISTINCT` et `ORDER BY`.	*Les opérateurs au programme sont* `+`, `-`, `*`, `/` *(on passe outre les subtilités liées à la division entière ou flottante),* `=`, `<>`, `<`, `<=`, `>`, `>=`, `AND`, `OR`, `NOT`. *D'autres mots-clés comme* `OFFSET` *et* `LIMIT` *pourront être utilisés mas sleur maîtrise n'est pas au programme.*
Jointures T_1 `JOIN` T_2 … `JOIN` T_n `ON` ϕ.	*On présente les jointures en lien avec la notion de relations entre tables. On se limite aux équi-jointures :* ϕ *est une conjonction d'égalités.*
Agrégation avec les fonctions `MIN`, `MAX`, `SUM`, `AVG` et `COUNT`, y compris avec `GROUP BY`.	*Pour la mise en œuvre des agrégats, on s'en tient à la norme SQL99. Les requêtes imbriquées ne sont pas au programme.*

Mise en œuvre
La création de tables et la suppression de tables au travers du langage SQL sont hors programme. La mise en œuvre effective se fait au travers d'un logiciel permettant d'interroger une base de données à l'aide de requêtes SQL comme *MySQL* ou *SQLite*. Récupérer le résultat d'une requête à partir d'un programme n'est pas un objectif. Sont hors programme : la notion de modèle logique *vs* physique, les bases de données non relationnelles, les méthodes de modélisation de base, les fragments DDL, TCL et ACL du langage SQL, les transactions, l'optimisation de requêtes par l'algèbre relationnelle.

2.3 Graphes

Il s'agit de définir le modèle des graphes, leurs représentations et leurs manipulations.
On s'efforce de mettre en avant des applications importantes et si possibles modernes : réseau de transport, graphe du web, réseaux sociaux, bio-informatique. On précise autant que possible la taille typique de tels graphes.

Notions	Exemples d'activité. *Commentaires.*
Vocabulaire des graphes. Graphe orienté, graphe non orienté. Sommet (ou nœud) ; arc, arête. Boucle. Chemin d'un sommet à un autre. Connexité dans les graphes non orientés. Matrice d'adjacence. Graphe $G = (S, A)$.	*On présente l'implémentation des graphes à l'aide de listes d'adjacence (rassemblées par exemple dans une liste ou dans un dictionnaire). On n'évoque ni multi-arcs ni multi-arêtes.*
Pondération d'un graphe. Étiquettes des arcs ou des arêtes d'un graphe.	*On motive l'ajout d'information à un graphe par des exemples concrets.*
Parcours d'un graphe. Parcours en largeur.	*La file sera représentée par une liste. On pourra évoquer le problème de cette représentation naïve en terme d'efficacité mais aucune connaissance sur d'autres représentations plus performante n'est au programme.*

2.4 Méthodes numériques

Cette section propose des sujets de travaux pratiques venant compléter le programme du deuxième semestre.

Thèmes	Exemples d'activité. *Commentaires.*
Méthode des rectangles.	Comparaison avec d'autres méthodes : méthode de Newton, méthode des trapèzes.
Simulation de variables aléatoires suivant des lois usuelles : Bernoulli, binomiale, uniforme. Estimation d'une probabilité, estimation de l'espérance et de la variance.	Simulation d'expériences et de variables aléatoires. Simulation d'une variable aléatoire à l'aide de sa fonction de répartition. *La justification de cette estimation sera donnée en deuxième année*

3 Programme des semestres 3 et 4

Le programme de deuxième année est constitué de séances de travaux pratiques qui poursuivent les objectifs suivants :

— approfondir les connaissances acquises au deuxième semestre notamment sur les bases de données et sur les graphes ;
— programmer des méthodes numériques vues en mathématiques ;
— mettre en application des notions vues en statistiques et probabilités dont les tests statistiques ;
— programmer de nouveaux algorithmes sur des applications en lien avec d'autres disciplines, notamment la biologie.

Comme au premier semestre, les tableaux ci-dessous présentent les thèmes qui sont abordés lors de ces séances. L'ordre de ces thèmes n'est pas impératif.

Aucune connaissance relative aux modules éventuellement rencontrés lors de ces séances n'est exigible des étudiants.

Les deux parties du programme représentent un volume horaire équivalent.

3.1 Méthodes numériques et statistiques

Thèmes	**Exemples d'activités** *Commentaires*
Méthode d'Euler explicite pour la résolution approchée d'équation différentielle ordinaire d'ordre 1.	Modèle logistique, comparaison avec d'autres méthodes de résolution approchée (méthode de Heun par exemple). *L'impact du pas de discrétisation sur la qualité des résultats et sur le temps de calcul est mis en évidence.* *Les résultats obtenus peuvent être comparés avec une résolution exacte ou avec une fonction de résolution approchée fournie par un module.*
Simulation d'une variable aléatoire de loi géométrique à l'aide de la loi de Bernoulli. Estimation de l'espérance, de la loi d'une variable aléatoire à partir de simulation.	On pourra simuler des variables aléatoires à l'aide de la loi uniforme et de la réciproque de la fonction de répartition : cas discret et continu. Exemples de la loi de Poisson, de la loi exponentielle et de la loi normale. Exemples de chaînes de Markov. Fonction de répartition empirique.
Illustration numérique de convergence en lois. Simulation d'une variable aléatoire suivant une loi de Poisson à l'aide d'une loi binomiale. Intervalle de confiance pour le paramètre d'une loi de Bernoulli.	Illustration du théorème central limite. *La notion théorique d'intervalle de confiance n'est pas au programme*
Approfondissement sur les statistiques : simulation et mécanisme de tests. Simulation de variables aléatoires de lois de khi-2 et de Student. Elaboration d'un test de moyenne, dit de Student, sur un petit échantillon gaussien.	Elaboration d'un test de la moyenne, cas particulier d'une proportion. On affichera les histogrammes obtenus. On pourra visualiser la convergence des lois de Student vers une loi normale. *La connaissance des lois de Student et du khi-2 n'est pas un attendu du programme. On pourra utiliser les simulateurs intégrés dans Python. La mise en place de test statistique (hormis le test de conformité à la moyenne) doit être accompagné, les outils mathématiques sous-jacents ne sont pas au programme*

On pourra compléter cette partie par une étude sur un autre sujet ou approfondir un des thèmes précédents. Les thèmes suivants ont été choisis par leur lien possibles avec des applications possible en SVT mais ils ne sont que des exemples proposés et ne sont pas des attendus exigibles du programme.

Exemples de thèmes libres	**Exemples d'activités** *Commentaires*
Complément sur la méthode d'Euler.	Méthode d'Euler explicite pour la résolution approchée de système d'équations différentielles d'ordre 1 et d'équation différentielle d'ordre 2. Modèle de Lotka-Volterra, modèle SIR, système différentiel issu d'une équation de cinétique chimique, équation d'un oscillateur harmonique amorti. *Une programmation vectorisée, même si elle peut être proposée, n'est pas un attendu du programme.*
Méthode du pivot de Gauss.	Résolution d'un système linéaire. *Une version sans recherche de pivot peut être proposée en premier puis une recherche de pivot partiel peut être programmée.* *On pourra mettre en évidence l'impact de la taille du système sur le temps de calcul.* *La programmation sans aide n'est pas un attendu du programme.*
Autour des équations aux dérivées partielles.	Équation de la chaleur, couplage réaction-diffusion. *Aucune connaissance sur les équations aux dérivées partielles n'est exigible.*

3.2 Approfondissements des concepts informatiques

Thèmes	**Exemples d'activités** *Commentaires*
Révisions et approfondissements sur les graphes. Parcours en profondeur. Plus court chemin dans un graphe pondéré.	Algorithme de Dijkstra. Problème du voyageur de commerce. Algorithme glouton, colonie de fourmis, recuit simulé. *L'objectif est de reprendre la structure de graphe à travers des approfondissements.*
Révisions des tris naïfs et exemple de tri récursif.	Tri fusion ou tri rapide. On pourra visualiser à l'aide de graphique l'accélération du temps de calcul. *La programmation du tri rapide en place n'est pas un objectif du programme. L'objectif sur les deux années est qu'un étudiant sache programmer un tri de son choix de façon autonome.*
Révisions et approfondissements sur les bases de données : révision sur les jointures et agrégations	*A partir d'une base de données comprenant 3 ou 4 tables ou relations, on approfondit la notion de jointure interne en lien avec la notion d'associations entre entités.* *Pour la notion d'agrégation, on présente quelques exemples de requêtes imbriquées.* *On marque la différence entre WHERE et HAVING sur des exemples.*

L'objectif de cette dernière partie est à travers l'étude d'un thème, sur plusieurs séances, de permettre aux étudiants de gagner en autonomie (choix de la représentation des données, choix du découpage en fonctions, etc).

Les thèmes suivants ont été choisis par leurs applications possibles en lien avec la SVT mais ils ne sont que des exemples proposés et ne sont pas des attendus exigibles du programme.

Exemples de thèmes libres	**Exemples d'activités** *Commentaires*
Informatique pour la génétique.	Recherche de motif : recherche exacte d'une ou plusieurs séquences courtes dans un génome. Algorithme naïf, algorithme de Rabin-Karp. Alignement de séquences en présence d'erreurs de séquençage ou de mutations. Distance d'édition, algorithmes de Needleman-Wunsch, de Smith-Waterman. Construction d'arbres phylogénétiques.
Images.	Segmentation d'images : comment partitionner une image en zones connexes ? *La structure union-find peut être introduite dans ce but.*
Apprentissage et classification.	Algorithme des k-moyennes. Classification par k plus proches voisins. *On pourra travailler autant sur des données réelles que sur des données issues de simulations (clusters gaussiens par exemple).*

A Langage Python

Cette annexe liste limitativement les éléments du langage Python (version 3 ou supérieure) dont la connaissance est exigible des étudiants. Aucun concept sous-jacent n'est exigible au titre de la présente annexe.
Aucune connaissance sur un module particulier n'est exigible des étudiants.
Toute utilisation d'autres éléments du langage que ceux que liste cette annexe, ou d'une fonction d'un module, doit obligatoirement être accompagnée de la documentation utile, sans que puisse être attendue une quelconque maîtrise par les étudiants de ces éléments.

Traits généraux

- Principe d'indentation.
- Portée lexicale : lorsqu'une expression fait référence à une variable à l'intérieur d'une fonction, Python cherche la valeur définie à l'intérieur de la fonction et à défaut la valeur dans l'espace global du module.

Types de base

- Opérations sur les entiers (`int`) : `+`, `-`, `*`, `**`, avec des opérandes positifs.
- Opérations sur les flottants (`float`) : `+`, `-`, `*`, `/`, `**`.
- Opérations sur les booléens (`bool`) : **`not`**, **`or`**, **`and`**.
- Comparaisons `==`, `!=`, `<`, `>`, `<=`, `>=`.

Types structurés

- Structures indicées immuables (chaînes de caractères) : `len`, accès par indice positif valide, concaténation `+`, répétition `*`, tranche.
- Listes : création par compréhension `[`*e* **`for`** *x* **`in`** *s*`]`, par `append` successifs ; `len`, accès par indice positif valide ; concaténation `+`, répétition `*`, tranche, copie ; `pop` en dernière position.
- Dictionnaires : création, accès, insertion, `len`, `copy`.

Structures de contrôle

- Instruction d'affectation avec `=`.
- Instruction conditionnelle : **`if`**, **`elif`**, **`else`**.
- Boucle **`while`** (sans **`else`**), **`return`** dans un corps de boucle.
- Boucle **`for`** (sans **`else`**) et itération sur `range(`*a*`,` *b*`)`, une chaîne de caractères une liste, un dictionnaire au travers des méthodes `keys` et `items`.
- Définition d'une fonction **`def`** $f(p_1,\ \dots,\ p_n)$, **`return`**.

Divers

- Introduction d'un commentaire avec `#`.
- Utilisation simple de `print`, sans paramètre facultatif.
- Importation de modules avec **`import`** *module*, **`import`** *module* **`as`** *alias*, **`from`** *module* **`import`** $f, g, \dots$