Année universitaire 2026/2027

2eme année de licence - Mathematique-Informatique - Talents (3A) - Maths-Info - FI

Responsables pédagogiques :

- Juliette BOUHOURS
- Denis PASQUIGNON

Crédits ECTS : 38

LES OBJECTIFS DE LA FORMATION

- Connaissances de base en mathématiques,
- Connaissance de base en informatique,
- Connaissance de base en économie.

Le parcours Talents permet aux étudiants sportifs de haut niveau ou artistes (musique, danse, arts, art dramatique) de suivre exactement le programme de L1-L2 tout en bénéficiant d'un aménagement : étalement des cours sur six semestres au lieu de quatre. Ils poursuivent en Licence de Mathématiques ou en Licence d'Informatique.

MODALITÉS D'ENSEIGNEMENT

Les Modalités des Contrôles de Connaissances (MCC) détaillées sont communiquées en début d'année.

Le programme de cours est identique à celui suivi par l'ensemble des étudiantes et les étudiants en L1 et L2, avec une progressivité du cursus organisée de manière adaptée sur trois ans. Des options sont fléchées pour permettre une valorisation du talent sportif, artistique ou entrepreneurial. Les étudiantes et les étudiants sont tenus de suivre au minimum deux demi-journées de cours par semaine et bénéficient d'un tutorat pour le reste des enseignements. Les examens sont communs à toute la promotion et la présence est obligatoire. Les enseignements des deux premières années de Licence MIDO sont organisés, dans le parcours Talents, en trois années et six semestres S1 à S6. Les semestres S3 et S4 sont communs pour les parcours Talents Mathématiques-Économie et Mathématiques-Informatique.

Langue d'enseignement : Français

POURSUITE D'ÉTUDES

Cette formation peut être naturellement prolongée par la Licence Mathématiques Appliquées ou Informatique des Organisations puis par un Master dans le département MIDO.

PROGRAMME DE LA FORMATION

- Semestre 5T
  - UE obligatoires Maths-Info
    - Algèbre linéaire 3
    - Algorithmique et programmation 3
    - Architecture des ordinateurs
    - Programmation C
- Semestre 6T
  - UE obligatoires Maths-Info
    - Algèbre 4 et méthodes numériques
    - Analyse 4

- Functional programming
- Programmation système

DESCRIPTION DE CHAQUE ENSEIGNEMENT

SEMESTRE 5T

UE obligatoires Maths-Info

Algèbre linéaire 3

ECTS : 8

Enseignant responsable : Guillaume LEGENDRE (https://dauphine.psl.eu/recherche/cvtheque/legendre-guillaume)

Langue du cours : Français

Volume horaire : 78

Description du contenu de l'enseignement :

1. Réduction des endomorphismes : diagonalisation et trigonalisation.
2. Formes bilinéaires.3. Formes quadratiques.
4. Espaces euclidiens : produit scalaire, norme euclidienne, orthogonalité, isométries vectorielles et endomorphismes auto-adjoints.

Compétences à acquérir :

Réduction des endomorphismes, formes bilinéaires et quadratiques, espaces euclidiens.

Algorithmique et programmation 3

ECTS : 4

Enseignant responsable : Denis CORNAZ (https://www.lamsade.dauphine.fr/~cornaz/)

Langue du cours : Français

Volume horaire : 49

Description du contenu de l'enseignement :

Chacun des points suivants sera présenté et expérimenté en langage Python :

1. Algorithmes et fonctions logarithmes : logarithmes naturels dans les appels récursifs ou dans les boucles type série harmonique, preuves courtes des propriétés de base des logarithmes. Notations asymptotiques et arrondis récursifs.
2. Complexité : algorithmes en T(n)=aT(n-b) + poly(n), et application aux implémentations exponentielle/linéaire de Fibonacci et à l'algorithme d'Euler-Bachet-Bezout.
3. Récursivité de la forme T(n)=aT(n/b) + poly(n): (rappel tri fusion), preuve courte du "Master Theorem", calcul rapide de complexité à partir du cas n puissance de b.
4. Performance des algorithmes : application du "Master Theorem" à la conception d'algorithmes de multiplication rapide d'entiers (Karatsuba), et de matrices (Strassen).
5. Force brute : algorithmes énumératifs, application à la résolution de systèmes d'équations et aux placements de reines sur échiquiers nxn.
6. Complexités des Tris : variétés du concept de complexité (pire cas, moyenne, structure des données) avec les algorithmes classiques de tri (rappel: insertion, dénombrement, tas)

Compétences à acquérir :

Fondements mathématiques de la complexité algorithmique et idées précises, avec connaissances profondes des exemples emblématiques, de ses paradigmes centraux. Maîtrise des mécanismes de base du langage Python.

Architecture des ordinateurs

ECTS : 4

Langue du cours : Français

Volume horaire : 39

Description du contenu de l'enseignement :

Histoire de l'informatique. Représentation des nombres et arithmétique.Circuits logiques. Structure générale d'un ordinateur.L'unité centrale : instructions, registres, pipeline, interruptions. L'assembleur.Les mémoires : hiérarchie, mémoire électronique, mémoire cache, mémoire de masse. Les entrées/sorties.Performances d'un ordinateur.

Compétences à acquérir :

Comprendre la structure interne d'un ordinateur à travers l'étude de ses différents composants : microprocesseur, mémoire, entrées/sorties acquérir les notions de base en langage machine : instruction, adressage, assembleur.

Programmation C

ECTS : 4

Enseignant responsable : Florian SIKORA (https://www.lamsade.dauphine.fr/~sikora/)

Langue du cours : Français

Volume horaire : 49

Description du contenu de l'enseignement :

Types et expression.
Structures de contrôle.
Fonctions.
Tableaux et pointeurs.
Structures.
Préprocesseur.
Entrées/sorties.

Compétences à acquérir :

Apprentissage du langage C de base et évolué.

SEMESTRE 6T

UE obligatoires Maths-Info

Algèbre 4 et méthodes numériques

ECTS : 4

Enseignant responsable : Guillaume BONNET (https://dauphine.psl.eu/recherche/cvtheque/bonnet-guillaume)

Langue du cours : Français

Volume horaire : 0

Description du contenu de l'enseignement :

1. Résolution numérique de systèmes linéaires (méthodes directes et itératives).
2. Calcul numérique de valeurs propres (méthode de la puissance).
3. Résolution numérique d'équations scalaires non linéaires (méthodes d'encadrement et de point fixe, méthode de la sécante).
4. Interpolation polynomiale.
5. Formules de quadrature interpolatoires.

Compétences à acquérir :

Présentation de méthodes numériques de résolution et d'éléments d'analyse numérique. Mise en œuvre : utilisation de Python, NumPy et Jupyter (travaux pratiques et projet).

Analyse 4

ECTS : 6

Enseignant responsable : Francois SIMENHAUS (https://dauphine.psl.eu/recherche/cvtheque/simenhaus-francois)

Langue du cours : Français

Volume horaire : 0

Description du contenu de l'enseignement :

1. Espaces métriques. Exemples : espaces euclidiens, espaces vectoriels normés. 2. Boules ouvertes, fermées, sphères. 3. Parties bornées. 4. Suites : convergence, bornitude, unicité de la limite. Suites extraites, valeurs d'adhérence. 5. Ouvert, voisinage. Fermé, point adhérent. Intérieur, adhérence, frontière. 6. Caractérisations séquentielles. 7. Compacité (au sens de Bolzano-Weierstrass). 8. Densité, exemples. 9. Restrictions à une partie. 10. Complétude : suites de Cauchy et définition d'un espace de Banach. 11. Convergence normale dans un Banach. 12. Exemple de l'exponentielle de matrice (TD). 13. Comparaison des topologies, distances, normes. Normes équivalentes. Exemples de normes non équivalentes (TD). 14. Limite en un point. Propriétés. 15. Continuité. Caractérisation séquentielle. 16. Image réciproque d'un ouvert, fermé. 17. Compacité et continuité. 18. Applications (bi)linéaires continues, norme. Exemple d'applications linéaires non continues (TD). 19. Connexité et connexité par arcs. 20. Dimension finie : équivalence des normes. Complétude. 21. Convergence des coordonnées. Caractérisation des compacts. 22. Calcul différentiel élémentaire en dimension finie (pas de différentielle) : 23. Dérivées partielles d'ordre 1 ou 2, fonctions de classe C1 ou C2.

Compétences à acquérir :

Notions de Topologie : savoir démontrer qu'un ensemble est ouvert, fermé, borné ; calculer l'intérieur, l'adhérence, la frontière dans des cas simples ; savoir étudier les suites à valeurs dans Rn ou des espaces de matrices ; savoir utiliser la compacité en dimension finie, la notion d'ensemble dense, savoir utiliser la continuité pour montrer qu'un ensemble est ouvert, fermé ; savoir utiliser la caractérisation séquentielle de la continuité ; savoir étudier la norme d'applications (bi)linéaires en dimension finie ; savoir calculer des dérivées partielles.

Functional programming

ECTS : 4

Enseignant responsable : Andre ROSSI (https://www.lamsade.dauphine.fr/~arossi/)

Langue du cours : Français

Volume horaire : 58

Description du contenu de l'enseignement :

The goal of this course is to familiarize students with the principles of functional programming using the Haskell language. Functional programming is a modern programming paradigm that allows the rapid and reliable design of complex applications. Functional programming concepts are currently prevalent in most modern programming languages, such as Java, C++, JavaScript, etc. The goal of this course is to help students master them using a purely functional language (Haskell). In addition, the course covers the Haskell type system, functors, applicatives and monads and let the students practice these notions with the Glasgow Haskell Compiler.

Compétences à acquérir :

This class covers the main principles of functional programming, like high-order functions, partial application and anonymous functions. Type systems that allow the manipulation of functions (the notion of Currying), recursion in the context of types, infinite data types, associated data structures and their manipulation are also covered. Functors, applicatives, monads and IO monads are also part of the program. All these topics are developed with programming exercises in Haskell.

Bibliographie, lectures recommandées :

https://learnyouahaskell.com/chapters

Programmation système

ECTS : 4

Langue du cours : Français

Volume horaire : 33

Description du contenu de l'enseignement :

Rôle du système d’exploitation et de son interface de programmation.
Étude et mise en pratique de l’utilisation d’un système Unix.
Étude et mise en pratique de la programmation Shell.
Étude des principaux appels systèmes de l’interface Posix (gestion de fichiers, processus).
Réalisation d’exercices simples mettant en œuvre chacun de ces appels système.

Réalisation d’un exercice complet combinant tous ces appels système.

Compétences à acquérir :

Ce cours est orienté vers l'utilisation du système d'exploitation par le développeur. Il s'agit donc d'étudier l'interface de programmation d'un système d'exploitation, l'interface Posix des systèmes Unix en l'occurrence. On vise ainsi à donner un sens concret à la notion de système et à son utilisation par les développeurs. Le cours comporte une partie pratique importante d'utilisation du système Linux et de sa programmation.

Document susceptible de mise à jour - 03/11/2026
Université Paris Dauphine - PSL - Place du Maréchal de Lattre de Tassigny - 75775 PARIS Cedex 16