

Partager



Email destinataire :

Vous devez renseigner un email valide

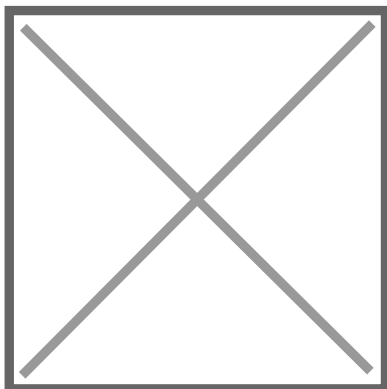
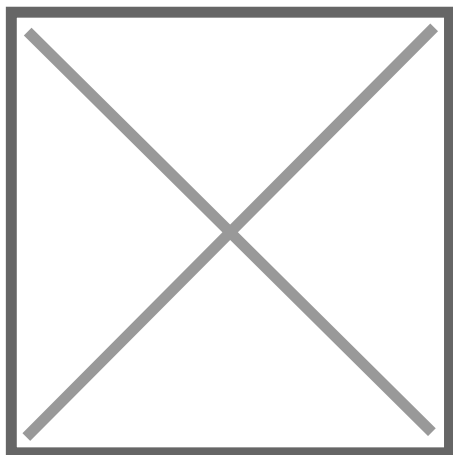
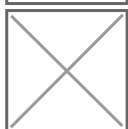
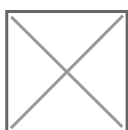
Envoyer

Une erreur est survenue. Merci de réessayer ultérieurement

Le mail de partage a bien été envoyé.

Ajouter au comparateur

Distributed, Parallel, and Quantum Computing (M1 DiPaQ)



Master

Mention Informatique

Régime(s) d'inscription

Formation initiale

Langue(s) d'enseignement

Anglais

Le parcours **M1 DiPaQ** se concentre sur le **calcul à haute performance (HPC), distribué et quantique** afin de maîtriser l'utilisation de **systèmes de calcul à grande échelle**. Les étudiants apprennent à concevoir des **solutions rapides, évolutives et robustes** répondant aux exigences de calcul d'applications en **intelligence artificielle (IA), analyse de données massives, calcul quantique et scientifique**. Les diplômés poursuivent des carrières en ingénierie et R&D ou des études doctorales.

Le programme comprend **douze cours fondamentaux** en **calcul haute performance, parallèle, distribué et quantique**. Des cours d'autres parcours apportent les bases de l'**IA** et de la **science des données**. Un **projet** (TER), un **stage M1** ou une **école d'été**, et un **cours sur le développement durable** complètent le cursus.

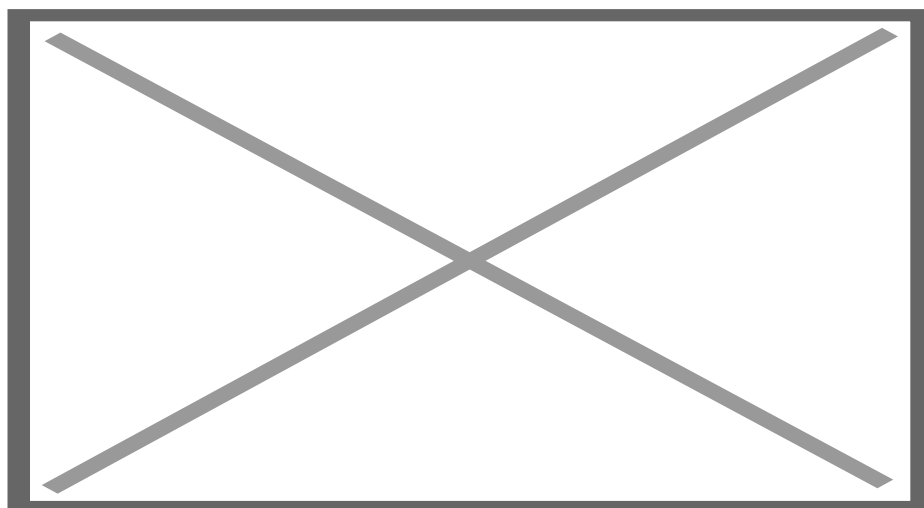
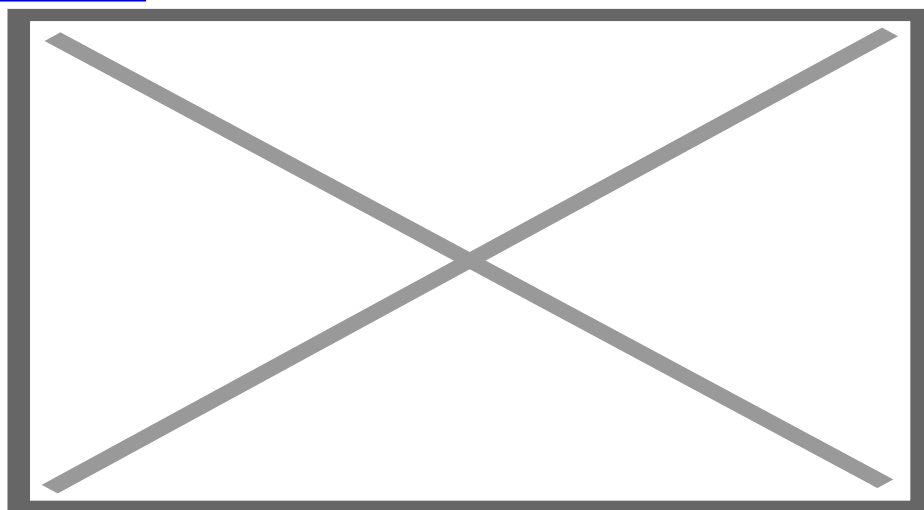
La **langue officielle** du programme est l'**anglais** ; tous les cours sont dispensés en anglais. La plupart de nos enseignants maîtrisent également le **français** ; des interactions en français sont donc possibles dans les cours et les évaluations (devoirs, examens, etc.).

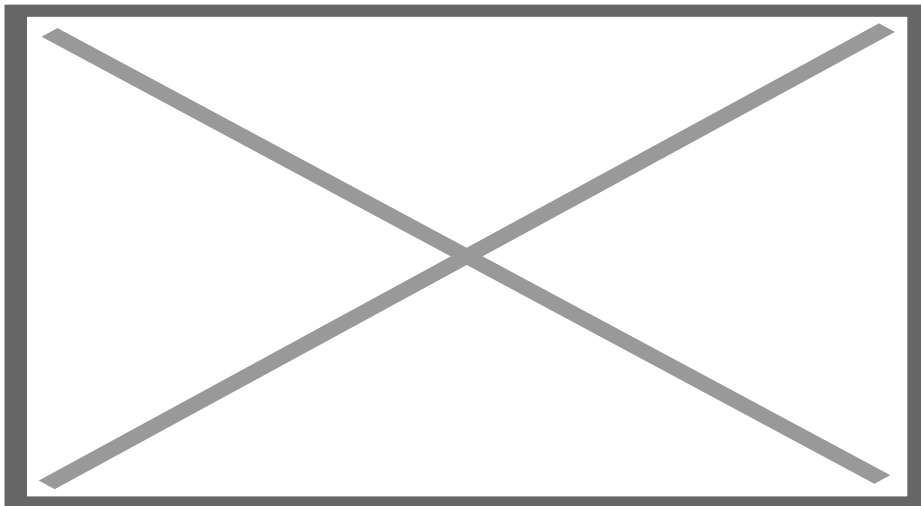
Le programme est étroitement intégré à l'écosystème Paris-Saclay de laboratoires de recherche et de partenaires industriels.

Les diplômés du **M1 DiPaQ** peuvent intégrer le **M2 DiPaC**, axé sur le **HPC** et le **calcul distribué** avec une spécialisation en **IA/analyse de données massives** ou en **calcul hybride classique/quantique**, ou candidater au **M2 QMI**, spécialisé dans les **technologies de l'information quantique**.

[Télécharger la fiche détaillée](#)

[Je candidate](#)





■ ■

Informations

- [Présentation](#)
- [Admission](#)
- [Programme](#)

Présentation

Compétences

Les étudiants du **M1 DiPaQ** acquerront des **compétences** et **savoir-faire** en :

- Compréhension des enjeux actuels et émergents des systèmes distribués, parallèles et quantiques ; évaluation de leur impact sur les domaines d'application en IA et science des données.
- Conception, preuve et analyse d'algorithmes et de protocoles distribués, parallèles et quantiques, avec analyses de complexité en temps, mémoire, communication et énergie.
- Maîtrise des fondamentaux des algorithmes et de la complexité quantiques ; programmation de circuits quantiques avec des bibliothèques standard.
- Mise en œuvre de modèles de programmation parallèle et d'ingénierie de la performance sur des supercalculateurs distribués et des machines multicœurs.
- Conception et analyse d'algorithmes distribués avec garanties théoriques.
- Acquisition des bases de l'analyse de données et de l'IA pour des tâches d'apprentissage à grande échelle.
- Réalisation de profilage et de traçage de code, diagnostic des goulots d'étranglement et réglage des performances.
- Pratiques d'ingénierie logicielle pour le HPC avec le C++ moderne, le test de code, l'intégration continue et la reproductibilité.
- Utilisation du contrôle de version avec Git et tenue d'une documentation rigoureuse.
- Communication technique et cadrage de projets dans des contextes de recherche et d'industrie.
- Adoption de pratiques de calcul responsables, fiables et durables.

Objectifs pédagogiques de la formation

Les systèmes informatiques évoluent vers une plus grande efficacité et des fonctionnalités plus riches au croisement de trois grands domaines scientifiques interconnectés :

- **Les systèmes distribués** assurent la connectivité et un fonctionnement fiable à l'échelle d'Internet, des clouds, des grappes et des capteurs, en s'attaquant à des problèmes difficiles de synchronisation, de sécurité, de concurrence et de robustesse.
- **Le calcul parallèle à haute performance (HPC)** traite des charges intensives en science et en IA en exploitant des supercalculateurs avec une ingénierie de performance rigoureuse.
- **Le calcul quantique** fournit des algorithmes et du matériel exploitant le parallélisme quantique pour atteindre des gains inaccessibles aux paradigmes classiques.

Le **master M1 DiPaQ** apporte aux étudiants une connaissance approfondie de ces trois domaines à travers des cours théoriques et une pratique intensive des techniques de programmation. Les **systèmes distribués** traitent des algorithmes qui permettent la connectivité et une fonctionnalité efficace pour les systèmes en réseau, comme Internet, le cloud, les réseaux de capteurs, les supercalculateurs, le blockchain, et les circuits microbiologiques. Pour ces systèmes, les défis incluent la synchronisation, la sécurité, la concurrence et la robustesse. Des problématiques similaires se posent en **HPC**, qui vise à résoudre efficacement des problèmes intensifs en calcul en sciences appliquées ou en IA. Le **HPC** pousse les architectures parallèles jusqu'à leurs limites en mobilisant diverses formes de parallélisme, des structures de données et l'optimisation du code. Il en va de même pour le **calcul distribué**, au moyen de diverses méthodes de communication et d'algorithmique. Ils tracent ainsi la frontière entre ce qui peut être atteint dans le cadre de l'informatique

classique et ce qui ne sera accessible qu'au moyen d'un nouveau paradigme : celui du **calcul quantique**. Le calcul quantique et l'information quantique introduisent de nouveaux algorithmes et protocoles, apportant des gains de performance d'un ordre inédit, tout en posant leurs propres défis conceptuels et techniques.

Le **M1 DiPaQ** établit les fondations en **HPC**, en **systèmes distribués** et en **calcul quantique** pour les **systèmes à grande échelle** et les **applications** en **IA** et en **science appliquée**, avec de forts liens avec l'écosystème de recherche et industriel de **Paris-Saclay** qui mobilise ces technologies. Les étudiants maîtrisent les fondamentaux de ces trois disciplines interconnectées et apprennent ainsi à concevoir des solutions rapides, évolutives et robustes pour relever des défis de calcul concrets en **analyse de données massives**, en **IA**, en **simulations scientifiques**, et en flux de travail compatibles avec l'**informatique quantique**, avec les objectifs suivants :

Objectifs de connaissances :

- Algorithmes et programmation parallèles efficaces sur des machines parallèles, distribuées, et multicœurs avec unités de traitement vectoriel.
- Programmation C++ avancée, débogage, profilage et optimisation des performances de noyaux HPC.
- Algorithmes et systèmes distribués à grande échelle : réplication, consensus, cohérence, robustesse.
- Fondamentaux des technologies, des algorithmes et de la programmation quantiques.
- Bases de l'IA, de la science des données et de l'optimisation pour l'apprentissage automatique et le calcul scientifique à grande échelle avec le HPC.

Objectifs de compétences :

- Concevoir des algorithmes et logiciels parallèles et distribué sur les architectures de HPC.
- Développer et analyser des algorithmes distribués évolutifs et robustes avec des garanties théoriques.
- Optimiser le code HPC: complexité, localité mémoire, vectorisation, communication, E/S et réseaux.

Ressources et pratique :

- Accès aux supercalculateurs partenaires,
- Utilisation de chaînes d'outils et de bibliothèques HPC
- Acquisition de pratiques de développement avec C++ avancé, IDE, git, profilage, documentation, intégration continue.

Débouchés

Professionnels

Après un Master ou Master + Doctorat : ingénieur (R&D, contrôle, production...)

Après un Master ou Master + Doctorat : chercheur ou enseignant-chercheur

Après Master + Doctorat : chercheur ou enseignant-chercheur

Après un Master ou Master + Doctorat : ingénieur (recherche et développement, contrôle, production...)

Ingenieur R&D

Responsable de projets R&D

Chef de projet

Chargé de développement

Consultant

Responsable de systèmes d'information

Ingénieur.e recherche & développement

Chargé.e de recherche et innovation

Chargé.e de projet

Chargé.e de validation/qualification

enseignant.e-chercheur.se (après un doctorat)

ingénieur.e de recherche

Chargé de mission / projets

coordinateur d'expertises scientifiques

Chargé.e de développement

Consultant.e

After Master and PhD : reseacher or assistant professor or professor

Poursuite d'études

Cette orientation prépare en particulier aux activités de recherche appliquée, généralement exercées après un doctorat, que ce soit dans le milieu académique ou en R&D.

Chargé.e de développement

Chef.fe de projet/de mission

Chercheur/chercheuse en R&D ou expert.e en modélisation et analyse de données dans des entreprises ou laboratoires de pointe.

Consultant.e

Ingénierie études, recherche et développement

Ingénierie méthodes et industrialisation

Master 2

Agenda

Début de la formation (date indicative)

04/09/2026

Fin de la formation (date indicative)

30/09/2027

Admission

Voie d'accès

Informatique

Mathématiques

Physique

Double Licence Mathématique - Physique

Capacité d'accueil

Places

20

Public visé et prérequis

Un cursus général en informatique est souhaitable. Toutefois, les titulaires d'une licence dans un autre domaine scientifique (par exemple en mathématiques ou en physique) disposant de solides bases en informatique (algorithmique et programmation) peuvent postuler au master **M1 DiPaQ**.

Un nombre limité de bourses (Eiffel, IDEX, Quantum Saclay) est disponible pour des candidats exceptionnels.

Période(s) de candidature

Plateforme Inception

Du 15/01/2026 au 16/03/2026

Plateforme MonMaster

Du 17 février au 16 mars 2026

Pour connaître la plateforme sur laquelle vous devez candidater, vous trouverez plus de renseignements sur la page [Candidater à nos masters](#). Vous trouverez ci-dessous la liste des pièces justificatives demandées sur la plateforme Inception.

Pièces justificatives

Obligatoires

Choix de parcours.

<https://cirrus.universite-paris-saclay.fr/s/jDS4JG5znqpeqRi>

Lettre de motivation.

Lettre détaillant la motivation et les raisons pour vouloir étudier l'informatique quantique, parallèle et distribuée dans le programme de master DiPaQ, en rapport avec des études et des expériences précédentes ainsi que des plans de carrière futurs.

Tous les relevés de notes des années/semestres validés depuis le BAC à la date de la candidature.

Tous les relevés de notes depuis le BAC.

Curriculum Vitae.

CV détaillant toutes les études antérieures, les stages, les formations suivies, les expériences professionnelles (si pertinent), les distinctions/récompenses ainsi que les intérêts et activités personnelles.

Facultatives

Copie diplômes.

Lettre de recommandation ou évaluation de stage.

Pièce libre.

Dossier VAPP (obligatoire pour toutes les personnes demandant une validation des acquis pour accéder à la formation) <https://www.universite-paris-saclay.fr/formation/formation-continue/validation-des-acquis-de-lexpérience>.

nécessaire uniquement si vous avez officiellement validé votre expérience professionnelle antérieure pour être considérée comme équivalente à un diplôme universitaire

Document justificatif des candidats exilés ayant un statut de réfugié, protection subsidiaire ou protection temporaire en France ou à l'étranger (facultatif mais recommandé, un seul document à fournir) :

- Carte de séjour mention réfugié du pays du premier asile
- OU récépissé mention réfugié du pays du premier asile
- OU document du Haut Commissariat des Nations unies pour les réfugiés reconnaissant le statut de réfugié
- OU récépissé mention réfugié délivré en France
- OU carte de séjour avec mention réfugié délivré en France
- OU document faisant état du statut de bénéficiaire de la protection subsidiaire en France ou à l'étranger.

Programme

[M1 DiPaQ] UE disciplinaires obligatoires

Matières	ECTS	Semestre	Cours	TD	TP	Cours- TD	Cours- TP	TD- TP	A distance	Projet	Tutorat
----------	------	----------	-------	----	----	--------------	--------------	-----------	---------------	--------	---------

[\[M1 DiPaQ\]](#)

[Introduction à](#)

[l'algorithmique et](#)

[à la](#)

[programmation](#)

[parallèle](#)

3

Semestre 1 10.5

10.5



[M1 DiPaQ] Introduction à l'algorithmique et à la programmation parallèle

ECTS : 3

Semestre calendaire : Semestre 1

Détail du volume horaire

Cours magistraux : 10.5

Travaux pratiques : 10.5

Langue d'enseignement

Anglais

Enseignement à distance

non

Prérequis

Prerequisites:

- **Basic algorithms and complexity analysis** – familiarity with algorithm design, asymptotic analysis, and common data structures.
- **Programming in C/C++** – ability to write, compile, and debug programs in C or C++.
- **Basic computer architecture** – understanding of CPU, memory hierarchy, and instruction execution.

Programme / plan / contenus

Course program, plan, content:

Total duration: 21 hours (7 sessions × 3 hours)

This course provides an in-depth study of high-performance computing (HPC) techniques targeting modern multicore processors on shared-memory architectures. Students learn how hardware architecture influences software performance and how to exploit shared-memory systems effectively. The course examines advanced SIMD vector instructions, cache-aware optimizations, NUMA behavior, efficient sparse and irregular data processing, and modern task-based parallelization strategies. Students also gain hands-on experience using foundational high-performance linear algebra libraries such as BLAS and LAPACK.

Multicore and shared-memory architecture:

- Multicore and manycore processor organization
- Memory hierarchy and cache behavior
- Cache coherency protocols
- Data races, memory ordering, and synchronization challenges

NUMA behavior and memory locality

[\[M1 DiPaQ\]](#)

[Calcul haute](#)

[performance sur](#) 3

Semestre 1 10.5

10.5

[architectures](#)

[multicœurs](#)



[M1 DiPaQ] Calcul haute performance sur architectures multicœurs

ECTS : 3

Semestre calendaire : Semestre 1

Détail du volume horaire

Cours magistraux : 10.5

Travaux pratiques : 10.5

Langue d'enseignement

Anglais

Enseignement à distance

non

Prérequis

Prerequisites:

- Basic C/C++ programming
- Basics of computer architecture
- Familiarity with basic linear algebra concepts
- [M1 DiPaQ] Introduction to parallel algorithms and programming

Programme / plan / contenus

Course program, plan, content:

Total duration: 21 hours (7 sessions × 3 hours)

This course builds upon the basic concepts of parallel programming introduced in "[M1 DiPaQ] Introduction to parallel algorithms and programming" and explores more advanced parallelization and code optimization techniques to write efficient code for multicore parallel architectures. The material covered in the course includes:

- An overview of the multicore processor architecture
- Writing efficient low-level code using SIMD instructions
- Memory and cache optimizations
- Advanced code optimization techniques (loop unrolling, branch optimizations, ...)
- Task-based parallelization
- Hybrid parallelization combining SIMD and multicore parallelism
- Parallelization and optimization approaches for dense linear algebra
- Parallelization and optimization approaches for sparse irregular computations
- Introduction to high performance linear algebra libraries (BLAS, LAPACK)

[\[M1 DiPaQ\]](#)
[Programmation](#)
[MPI pour les](#)
[architectures à](#)
[mémoire](#)
[distribuée](#)

3

Semestre 2 10.5

10.5

[M1 DiPaQ] Programmation MPI pour les architectures à mémoire distribuée

ECTS : 3

Semestre calendaire : Semestre 2

Détail du volume horaire

Cours magistraux : 10.5

Travaux pratiques : 10.5

Langue d'enseignement

Anglais

Enseignement à distance

non

Prérequis

Prerequisites:

- Basic knowledge of programming in C or Fortran is required.
- Familiarity with parallel programming concepts, computer architecture is recommended but not mandatory.
- Basic knowledge of Unix is ??recommended
- Students should be comfortable with compiling programs in a Linux environment.
- Prior exposure to performance optimization or HPC tools will be beneficial but is not a strict prerequisite.

Programme / plan / contenus

Course program, plan, content:

Total duration: 21 hours (7 sessions × 3 hours)

- **Session 1 (Lecture + Lab):** Introduction; MPI environment (with exercise); introduction to supercomputer, module and SLURM environment.
- **Session 2 (Lecture + Lab):** Point-to-point communications (with exercise)
- **Session 3 (Lecture + Lab):** Collective communications (with exercise).
- **Session 4 (Lecture + Lab):** Communication model (with exercise).
- **Session 5 (Lecture + Lab):** Derived types (with exercise).
- **Session 6 (Lecture + Lab):** Communicators (with exercise).
- **Session 7 (Lecture + Lab):** Open session based on questions and finalization of projects.

Objectifs d'apprentissage

Learning objectives:

[M1 DiPaQ]				
Programmation	3	Semestre 1	10.5	10.5
C++ avancée				



[M1 DiPaQ] Programmation C++ avancée

ECTS : 3

Semestre calendaire : Semestre 1

Détail du volume horaire

Cours magistraux : 10.5

Travaux pratiques : 10.5

Langue d'enseignement

Anglais

Enseignement à distance

non

Prérequis

Prerequisites:

- Prior experience in programming, preferably in C, Fortran, or Python
- Basic familiarity with computer usage and development tools such as text editors, compilers, and command-line interfaces

Programme / plan / contenus

Course program, plan, content:

Total duration: 21 hours (7 sessions × 3 hours)

- Introduction to C++ and development environment – compiling, linking, running programs, and basic debugging
- Fundamental data types and control structures – variables, constants, arithmetic, logic, loops, and conditionals
- Functions and modular programming – defining functions, parameter passing, recursion, and scope
- Object-oriented programming concepts – classes, objects, constructors/destructors, encapsulation
- Inheritance, polymorphism, and virtual functions – designing reusable and extensible code
- Standard library basics – containers (vector, map, list), iterators, and algorithms
- Memory management and pointers – pointers, references, dynamic allocation, smart pointers
- File I/O and exception handling – reading/writing files, exceptions, and program robustness
- Basic program optimization and debugging techniques – performance considerations, common pitfalls, and testing

Objectifs d'apprentissage

Learning objectives:

- Understand and apply the core concepts of the

[M1 DiPaQ]

Optimisation pour 3
le calcul
scientifique

Semestre 2



[M1 DiPaQ] Optimisation pour le calcul scientifique

ECTS : 3

Semestre calendaire : Semestre 2

Langue d'enseignement

Anglais

Enseignement à distance

non

Prérequis

Prerequisites:

- Linear algebra
- Multivariable calculus
- Basics in quantum computing

Programme / plan / contenus

Course program, plan, content:

Total duration: 21 hours (7 sessions $\times$ 3 hours)

- Introductions and background (convexity, differentiability, optimality conditions, convergence rates ...)
- Continuous optimization (first order methods: Gradient methods, line search, Acceleration)
- Continuous optimization (second order methods: Newton methods including Quasi-Newton, secant, IRLS)
- Conjugate Gradient Methods
- Linear conjugate gradient
- Solution of linear systems
- Least-Squares Problems
- Linear least Squares (Normal equations, CholeskyQR, SVD-based)
- Non-linear least squares (Gauss-Newton)
- Convexity, Optimality Conditions, and Duality: Convex Functions and Convex Sets, First- and Second-Order Optimality Conditions, Lagrangian, Duality Theory, Gradient methods, Physics Informed neural networks for optimization problems.
- Quantum Approaches for optimization problems: Variational Quantum Eigensolver (VQE), Quantum Approximate Optimization Algorithm (QAOA), Grover's search for optimization problems.

Objectifs d'apprentissage

Learning objectives:

- At the end of this course, students will be able to:

[\[M1 DiPaQ\]](#)

[Initiation à](#)

[l'algorithmique et](#) 3

[à la](#)

[programmation](#)

[quantique](#)

Semestre 1 10.5

10.5



[M1 DiPaQ] Initiation à l'algorithmique et à la programmation quantique

ECTS : 3

Semestre calendaire : Semestre 1

Détail du volume horaire

Cours magistraux : 10.5

Travaux pratiques : 10.5

Langue d'enseignement

Anglais

Enseignement à distance

non

Prérequis

Prerequisites:

- Familiarity with linear algebra
- Exposure to programming in Python

Programme / plan / contenus

Course program, plan, content:

Total duration: 21 hours (6 sessions × 3.5 hours + final exam session)

- **Session 1 (Lecture):** Introduction to quantum computation in the context of the quantum co-processor model. Notion of unitary, measure, quantum circuit.
- **Session 2 (Lecture + Lab):** lab session for practice, followed by the presentation of Deutsch-Jozsa algorithm.
- **Session 3 (Lecture + Lab):** lab session for practice, followed by a discussion on circuit and oracle synthesis.
- **Session 4 (Lecture + Lab):** Quantum Fourier Transform and Phase Estimation: theory and implementation.
- **Session 5 (Lecture + Lab):** Shor's factoring algorithm: theory and implementation
- **Session 6 (Lecture + Lab):** Quantum Amplification (aka Grover's algorithm): theory and implementation
- **Session 7:** Final written exam

Objectifs d'apprentissage

Learning objectives:

By the end of this course, students will be able to:

- Understand the quantum co-processor model and the general structure of quantum algorithms

[\[M1 DiPaQ\]](#)
[Fondements de](#)
[l'informatique](#)
[quantique](#)

3	Semestre 2	10.5	10.5
---	------------	------	------



[M1 DiPaQ] Fondements de l'informatique quantique

ECTS : 3

Semestre calendaire : Semestre 2

Détail du volume horaire

Cours magistraux : 10.5

Travaux dirigés : 10.5

Langue d'enseignement

Anglais

Enseignement à distance

non

Prérequis

Prerequisites:

- All the necessary reminders will be given. However, the student should make sure that he knows the basics in linear algebra: vectors, matrices, inner product, norm.
- Moreover, and again even if all the necessary formal objects will be defined and introduced properly, it is a huge plus if the student has a relatively well advanced level in mathematics, let's say the equivalent of either 2 years of "Math Sup / Mat spé", or of an L2 in pure mathematics in some university (and even better if it is an L3), or of an L2 or L3 in theoretical computer science (usually, students from purely applied computer science have difficulties to follow the course), or an L3 of fundamental physics, with enough mathematics for physics.

Programme / plan / contenus

Course program, plan, content:

The course is composed of 5+1 sessions of 3h30min each, which, with the 2h final-exam session, covers a T2. Lecture 6 is not compulsory material to learn for the final exam.

Lecture 1: Postulates of quantum theory

- Their meaning;
- The necessary mathematics;
- Dirac and Coecke notations for these mathematics;
- Several basic results: no-cloning; no-distinguishability.

Lecture 2: Useful linear-algebra theorems

Lecture 3: Density matrices and quantum operations

Lecture 4: Quantum error correction

Several variants of the 3-qubit code; fundamental theorems of quantum error correction.

[\[M1 DiPaQ\]](#)
[Optimisation](#)
[combinatoire](#)
[quantique](#)

3	Semestre 2	10.5	10.5
---	------------	------	------

[M1 DiPaQ] Optimisation combinatoire quantique

ECTS : 3

Semestre calendaire : Semestre 2

Détail du volume horaire

Cours magistraux : 10.5

Travaux dirigés : 10.5

Langue d'enseignement

Anglais

Enseignement à distance

non

Prérequis

Prerequisites:

- Familiar with linear algebra
- Basics of linear optimization
- Basic of quantum computing

Programme / plan / contenus

Course program, plan, content:

Total duration: 21 hours (7 sessions $\times$ 3 hours)

- **Session 1 (Lecture + Lab):** Foundations of Combinatorial Optimization: Introduction to Combinatorial Optimization; Problem Formulation: Linear Programming, Integer Programming; Simplex Method: Formulation, Efficiency, Duality.
- **Session 2 (Lecture + Lab):** Combinatorial optimization: Reformulations, Bounds, Lagrangian relaxation; Metaheuristics: Simulated annealing, Variable Neighborhood search.
- **Session 3 (Lecture + Lab): Dynamic Programming:** Dynamic programming in combinatorial optimization; Application to knapsack and lot-sizing problems; Bellman's Principle.
- **Session 4 (Lecture + Lab): Introduction to Quantum Computing for Optimization:** Basics of quantum computing (qubits, gates, superposition, entanglement); reformulation of combinatorial optimization problems into their quantum counterparts.
- **Session 5 (Lecture + Lab): Quantum Optimization Algorithms:** Quantum Annealing: Applications in Combinatorial Optimization; Variational Quantum Algorithms (VQAs): QAOA for Optimization.
- **Session 6 (Lecture + Lab): Quantum linear programming:** Introduction to Quantum Circuit Optimization, applications, hybrid optimization

[\[M1 DiPaQ\]](#)
[Algorithmes](#)
[distribués](#)
[robustes](#)

3	Semestre 1	10.5	10.5
---	------------	------	------



[M1 DiPaQ] Algorithmes distribués robustes

ECTS : 3

Semestre calendaire : Semestre 1

Détail du volume horaire

Cours magistraux : 10.5

Travaux dirigés : 10.5

Langue d'enseignement

Anglais

Enseignement à distance

non

Prérequis

Prerequisites:

- Basic notions of algorithms and algorithm analysis (complexity measures, asymptotic notations)
- Familiarity with basic notions of networking and operating systems is recommended

Programme / plan / contenus

Course program, plan, content:

Total duration: 21 hours (6 sessions $\times$ 3.5 hours + 1 exam session)

- **Session 1:** Introduction to distributed algorithms and presentation of the synchronous message-passing distributed model that we will use in the course; complexity measures and other definitions
- **Session 2:** Leader election on a ring: LeLann, Chang and Roberts algorithm and its variants.
- **Session 3:** Leader election on a ring: Hirschberg and Sinclair algorithm, Time-slice algorithm, impossibility of LE in an anonymous ring.
- **Session 4:** Crash consensus and consensus with link failures.
- **Session 5:** Byzantine consensus.
- **Session 6:** Network synchronization and distributed BFS tree construction.
- **Session 7:** Final exam

Objectifs d'apprentissage

Learning objectives:

Distributed algorithms are the building blocks of distributed systems and applications, such as the Internet, the Internet of Things, the Cloud, Bitcoin, etc. In addition to the difficulties induced by the geographical distribution and dynamic behavior of their components, these real systems are subject to failures: crashes,

[\[M1 DiPaQ\]](#)

[Algorithmes](#)

[distribués auto-](#)
[stabilisants](#)

3

Semestre 2 10.5 10.5



[M1 DiPaQ] Algorithmes distribués auto-stabilisants

ECTS : 3

Semestre calendaire : Semestre 2

Détail du volume horaire

Cours magistraux : 10.5

Travaux dirigés : 10.5

Langue d'enseignement

Anglais

Enseignement à distance

non

Prérequis

Prerequisites:

- Basic notions of algorithms and algorithm analysis (complexity measures, asymptotic notations)
- Familiarity with basic notions of graph theory and elementary graph algorithms
- Familiarity with basic notions of networking and operating systems are recommended
- Some knowledge on distributed algorithms and systems is recommended

Programme / plan / contenus

Course program, plan, content:

Total duration: 21 hours (6 sessions × 3 hours + 1 presentations/exam session)

- **Session 1:** Introduction to self-stabilization.
- **Session 2:** Self-stabilizing Census algorithm.
- **Session 3:** Transient fault detection technique, tree construction example
- **Session 4:** Self-stabilizing shortest-path tree construction
- **Session 5:** Dijkstra's token circulation.
- **Session 6:** Proof methods of self-stabilizing algorithms.
- **Session 7:** Students' presentations on self-stabilizing algorithms and an oral and written examination on the course material.

Objectifs d'apprentissage

Learning objectives:

Self-stabilization is a versatile technique used to overcome any transient failure in a system. The focus of this module is on distributed systems such as the Internet, networks of mobile agents or sensors, and their applications such as the Internet of Things, the Cloud,

[\[M1 DiPaQ\]](#)
[Graphes et](#)
[algorithmes](#)

3

Semestre 1 10.5 10.5

[M1 DiPaQ] Graphes et algorithmes

ECTS : 3

Semestre calendaire : Semestre 1

Détail du volume horaire

Cours magistraux : 10.5

Travaux dirigés : 10.5

Langue d'enseignement

Anglais

Enseignement à distance

non

Prérequis

Prerequisites:

- We will recall the important notions about the structure of a graph in the first session; it will be important to be familiar with them for the next sessions.
- Students should know what an algorithm is, how to prove that it is correct, and how to compute its complexity, in time and in space. They should be able to use classical algorithmic strategies, such as greedy algorithms and dynamic programming. Cf. Cormen Introduction to Algorithms if needed: [https://edutechlearners.com/download/Introduction_to_algorithms-3rd%20Edition.pdf](https://edutechlearners.com/download/Introduction_to_algorithms-3rd Edition.pdf)
- Students should be familiar with the most classical data structures and know how to use them: static/dynamic arrays, (doubly) linked lists, binary search trees, hashmaps, ...
- Students should be able to articulate and formalize a reasoning. There will be an important focus on the quality of formulation during the course.

Programme / plan / contenus

Course program, plan, content:

Total duration: 21 hours (6 sessions $\times$ 3.5 hours + 3h exam session). Each chapter lasts between 1 and 2 sessions. The first 6 sessions are composed of lectures and exercise sessions, and the 7th is dedicated to the exam.

- **Chapter 1: Graph Theory basics.** Introduction of basic structural notions on graphs: (induced) subgraphs, minors, heredity, ...; of classical graph parameters: radius, diameter, minimum/maximum degree, degeneracy, maximum average degree, clique/independence number, Hall ratio, ...; and of some relations linking those parameters. Focus on

[\[M1 DiPaQ\]](#)
[Algorithmes](#)
[probabilistes](#)

3

Semestre 2 10.5 10.5



[M1 DiPaQ] Algorithmes probabilistes

ECTS : 3

Semestre calendaire : Semestre 2

Détail du volume horaire

Cours magistraux : 10.5

Travaux dirigés : 10.5

Langue d'enseignement

Anglais

Enseignement à distance

non

Prérequis

Prerequisites:

- Background in algorithms and data structures.
- Basic knowledge of discrete mathematics and probability theory is recommended but not mandatory.
- Familiarity with asymptotic analysis (Big-O notation) and complexity concepts.
- Programming experience in a general language (e.g., Python, C++, or Java) is recommended.

Programme / plan / contenus

Course program, plan, content:

Total duration: 21 hours (6 sessions $\times$ 3.5 hours)

- **Session 1 (Lecture + Lab):** Introduction to randomization (Motivation and examples: why randomization?) Review of basic probability concepts (expectation, variance, independence). Case study: Randomized QuickSort and its expected complexity.
- **Session 2 (Lecture + Lab):** Probabilistic analysis tools: linearity of expectation and indicator variables, the coupon collector problem and conditional probability laws.
- **Session 3 (Lecture + Lab):** Probabilistic Bounds and Applications (Markov, Chebyshev, and Chernoff bounds). The balls-into-bins paradox and implications for load balancing. Verification problems and Bayesian classification examples.
- **Session 4 (Lecture + Lab):** Introduction to Online Algorithms (The ski-rental problem, Online paging and caching algorithms, Basics of load balancing in online settings).
- **Session 5 (Lecture + Lab):** Hashing and Streaming Algorithms (Randomized hashing and collision analysis, universal hashing and expected performance, introduction to streaming algorithms).

[M1 DiPaQ] UE disciplinaires d'ouverture

Matières	ECTS	Semestre	Cours	TD	TP	Cours- TD	Cours- TP	TD- TP	A distance	Projet	Tutorat
[DKAI] MATHEMATICS FOR DATA SCIENCE	3	Semestre 1	12	4.5	4.5						

[DKAI] MATHEMATICS FOR DATA SCIENCE

ECTS : 3

Semestre calendaire : Semestre 1

Détail du volume horaire

Cours magistraux : 12

Travaux pratiques : 4.5

Travaux dirigés : 4.5

Langue d'enseignement

Anglais

Enseignement à distance

non

Prérequis

Students will be able to handle mathematical concepts underlying tools used by practitioners

Objectifs d'apprentissage

This class aims at teaching/reminding mathematical basis useful in data science:

1. vector spaces, linear transformations,
2. matrices, linear systems,
3. norms, orthogonality,
4. eigenvalues, singular value decomposition,
5. tensors (notions), multivariable calculus
6. etc

Optional exercises will be posted on my webpage (on eCampus, when restored):

<https://perso.ensta-paris.fr/~bonazzoli/teaching.html>

Reading recommendations:

To complement the lectures, you can find plenty of good material on the web about Linear Algebra and Multivariable Calculus, such as:

- Lectures "Introduction to Probability, Statistics, and Machine Learning" by Samuel S. Watson from Brown University (lectures 2, 3, 4),
<https://data1010.github.io/class/>

- Lectures "Mathematics for Machine Learning" by Garrett Thomas from University of California, Berkeley,
<https://gwthomas.github.io/docs/math4ml.pdf>

- Book "Mathematics for Machine Learning" by M. P. Deisenroth, A. A. Faisal, C. S. Ong, published by Cambridge University Press, <https://mml-book.github.io/book/mml-book.pdf>

- Notes from Zico Kolter (updated by Chuong Do)

Matières	ECTS	Semestre	Cours	TD	TP	Cours- TD	Cours- TP	TD- TP	A distance	Projet	Tutorat
[AI] Hands on machine learning with scikitlearn	3	Semestre 1	9	12							
■											

[AI] Hands on machine learning with scikitlearn

ECTS : 3

Semestre calendaire : Semestre 1

Détail du volume horaire

Cours magistraux : 9

Travaux dirigés : 12

Langue d'enseignement

Anglais

Enseignement à distance

non

Objectifs d'apprentissage

A practical oriented class, where students apply ML techniques to simple illustrative examples and then to tackle competitive challenges. It will start with an introduction to present (refresh) the ML landscape. Classes will then be articulated to successively focus on the major concepts of practical ML.

Outline:

1. Introduction/refresher on ML
2. Working with real data
3. Discover and visualize the data to gain insights
4. Prepare the data for processing
5. Select and train models
6. Fine-tune models

Recommation of reading:

- Géron (2019) Hands-on Machine Learning with Scikit-Learn, Keras, and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems
- VanderPlas (2017) Python Data Science Handbook: Essential Tools for Working with Data

Nature de l'évaluation

Evaluation Continue non Intégrale

[\[AI\] Machine Learning Basics 1](#)

3

Semestre 1

[AI] Machine Learning Basics 1

ECTS : 3

Semestre calendaire : Semestre 1

Langue d'enseignement

Anglais

Enseignement à distance

non

Prérequis

- (T1A, Applied Stats): Maximum Likelihood Estimate and related notions

- (T1B, Maths for DS): Basic Linear Algebra, well mastered (tensor product, diagonalization)

- python, numpy.

Ideally, too:

- (OPT9, Hands On ML): it's a good complement to this class, very good to master sklearn. Here we'll look inside the algos of sklearn or others.

+ Check out previous years exam to see the level of math-mastery that is expected (of course some of it is covered in this course, but you need good basics).

Objectifs d'apprentissage

This course is algorithms-oriented, i.e. we will sketch the great principles of ML at first, and then focus on how algorithms work in practice, including all necessary mathematical aspects.

They are the basic building blocks of more advanced algorithms.

1 Gradient Descent, Linear Regression from scratch

2 Classification with a single layer Perceptron, from scratch. Geometrical interpretation, a word on SGD/mini-batch learning. Discussion on the choice of the loss function, or activation functions. OVR multi-class scheme (quickly)

3 overfitting, train/validation/test split, K-fold CV, regularization: in general, L2, L1.

4 MAP, Bayesian interpretation of Ridge Regression or Lasso one.

5 feature maps ("Kernel trick"), PCA (from scratch, seen as variance maximization), PCA as pre-processing (dimensional reduction)

6 Kernels, Kernelized perceptron, SVM in the separable case (some details omitted). Exercise: Lasso regularization from scratch (pseudo-gradients quickly introduced)

Recommation of reading:

Pattern Recognition and Machine Learning, Christopher Bishop, 2006 (available online for free)

Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow

Matières	ECTS	Semestre	Cours	TD	TP	Cours- TD	Cours- TP	TD- TP	A distance	Projet	Tutorat
Impacts Environnementaux du numérique	3	Semestre 2	10.5	10.5							



Impacts Environnementaux du numérique

ECTS : 3

Semestre calendaire : Semestre 2

Détail du volume horaire

Cours magistraux : 10.5

Travaux dirigés : 10.5

Langue d'enseignement

Anglais

Enseignement à distance

non

Programme / plan / contenus

5 hours of lectures on direct and indirect impacts, and several tutorials including an IT equipment inventory, the analysis of CSR reports from major digital groups, assessment of the specific impacts of AI, and a poster session based on scientific articles on the subject.

Objectifs d'apprentissage

During this course, students will explore the topic of environmental damages linked to digital technology and various ways to assess and limit these damages. For those attending the course, the aim is to subsequently be able to think critically as computer engineers and know what steps to take to assess the impact of their hardware purchases and software developments and how to reduce this impact.

Nature de l'évaluation

Evaluation Continue Intégrale

[\[M1 DiPaQ\]](#)

[Travaux d'étude et
de recherche \(TER\)](#)

6

Semestre 2

21



[M1 DiPaQ] Travaux d'étude et de recherche (TER)

ECTS : 6

Semestre calendaire : Semestre 2

Détail du volume horaire

Projet : 21

Langue d'enseignement

Anglais

Enseignement à distance

non

Nature de l'évaluation

Evaluation Terminale

[\[M1 DiPaQ\] Stage
ou école d'été](#)

6

Semestre 2



[M1 DiPaQ] Stage ou école d'été

ECTS : 6

Semestre calendaire : Semestre 2

Langue d'enseignement

Anglais

Enseignement à distance

non

Nature de l'évaluation

Evaluation Terminale

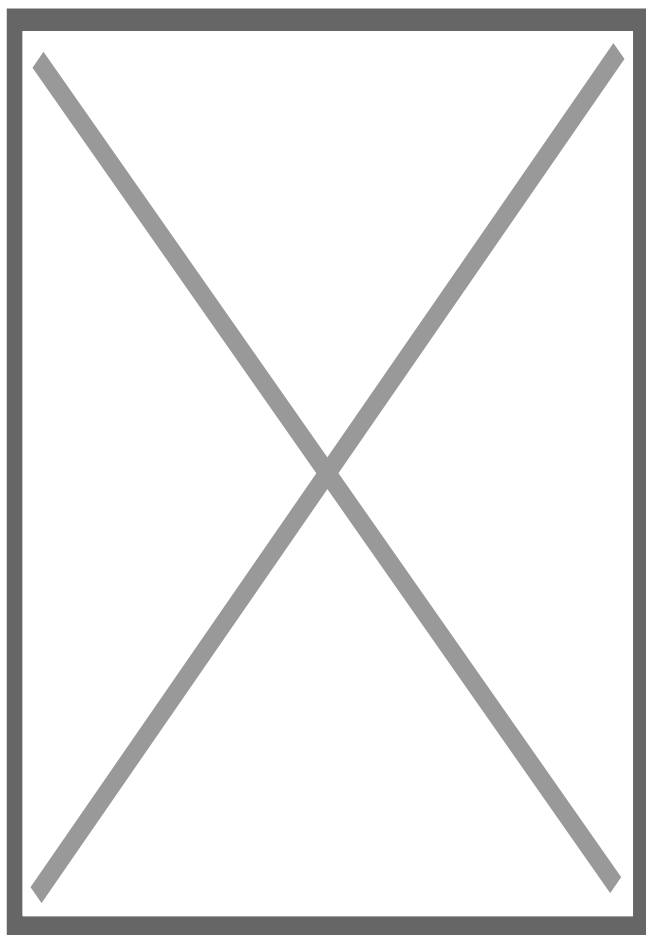
Lieu(x) d'enseignement

ORSAY

GIF SUR YVETTE

Campus de la formation

Javascript est requis pour visualiser cette carte.



Orsay Bures

Orsay / Bures-sur-Yvette

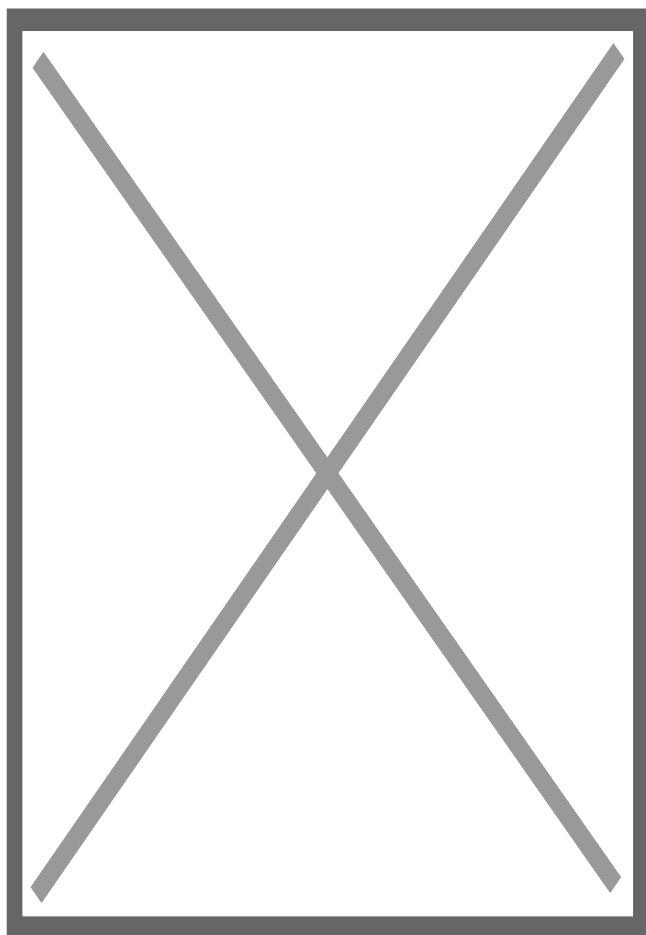
Bus 4607, 4626

RER B Orsay ville ou Bures-sur-Yvette

[Voir](#)

Bibliothèque

Maison des associations



Saclay Corbeville

Saclay / Palaiseau

Bus 4606, 4627, 5154, 9105, 9108

RER B Palaiseau, Palaiseau villebon ou Lozère

[Voir](#)

Restaurant CROUS

Résidence étudiante

Equipements sportifs

Contact(s)

- **Thomas GALLEY**

Liens

[Blog Destination études supérieures : rejoignez la révolution de l'ordinateur d...](#)

<http://www.universite-paris-saclay.fr/formation/master/informatique/distributed-parallel-and-quantum-computing-m1-dipaq>