

Share



Email destinataire :

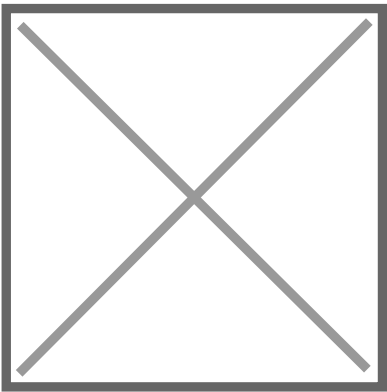
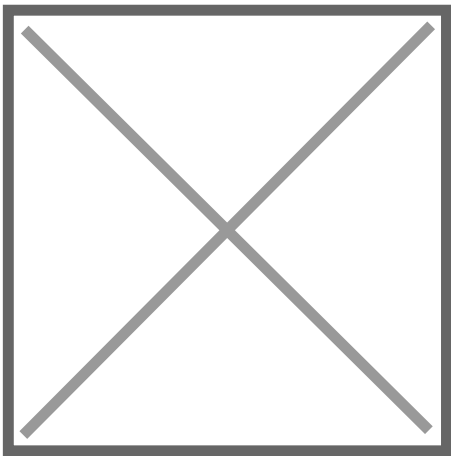
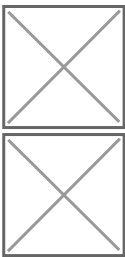
Vous devez renseigner un email valide

Envoyer

Une erreur est survenue. Merci de réessayer ultérieurement
Le mail de partage a bien été envoyé.

Add to Compare

M1 Distributed, Parallel, and Quantum Computing (1st year Master's DiPaQ)



Master's degree
Specialisation Computer Science
Enrolment type
Full-time academic programmes
Language(s) of instruction

English

The **M1 DiPaQ** master's program focuses on topics in **high-performance (HPC), distributed, and quantum computing** to master the use of **large-scale computational systems**. Students learn to **design fast, scalable, and robust solutions** that address the computational demands of applications in **artificial intelligence (AI), big data analytics, and quantum and scientific computing**. The program prepares graduates for careers in engineering and PhD or doctoral studies.

The curriculum includes **twelve core disciplinary courses** in **high performance, parallel, distributed, and quantum computing**. In addition, students take **supplementary courses** from other master tracks to learn the fundamentals of **AI** and **data science**. Finally, a **project course (TER)**, an **M1 summer internship or summer school** on DiPaQ-related themes, and a **course on sustainable development** completes the program.

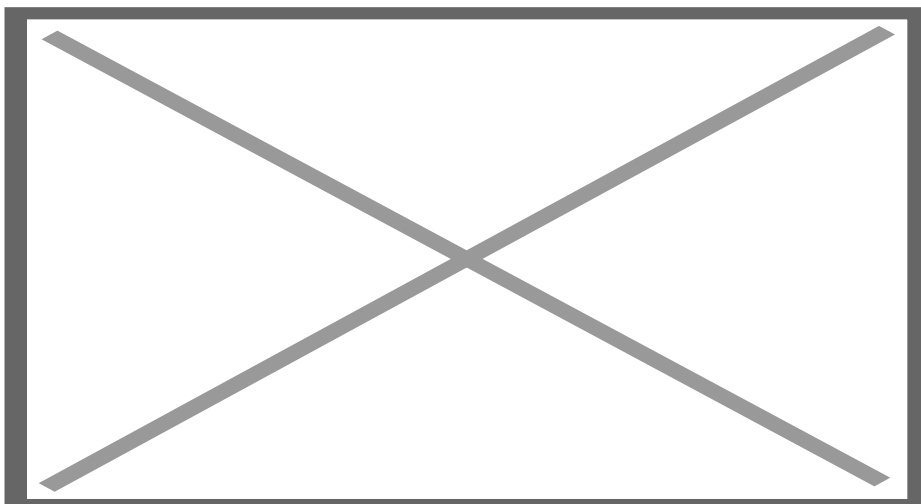
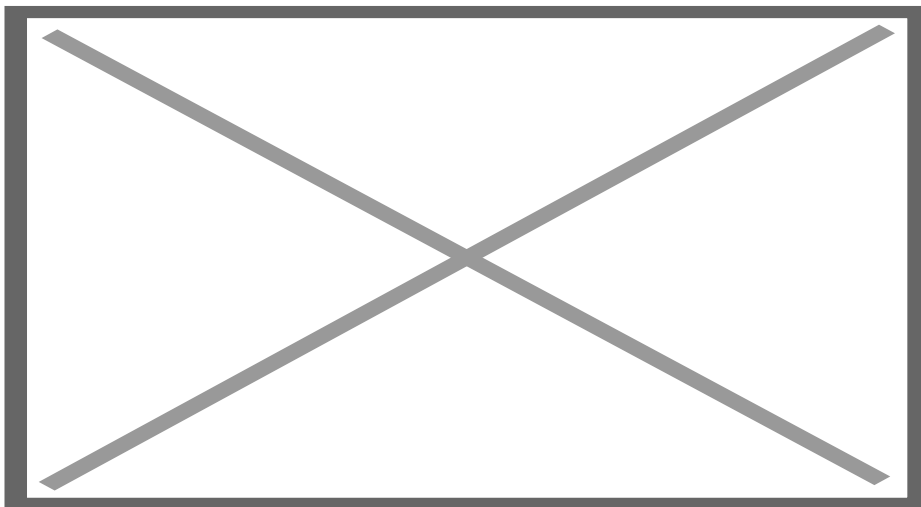
The program's **official language is English**; all courses are taught in **English**. Most of our faculty are also fluent in **French**; hence interaction in French is possible in the courses and assignments (homework, exams, etc.).

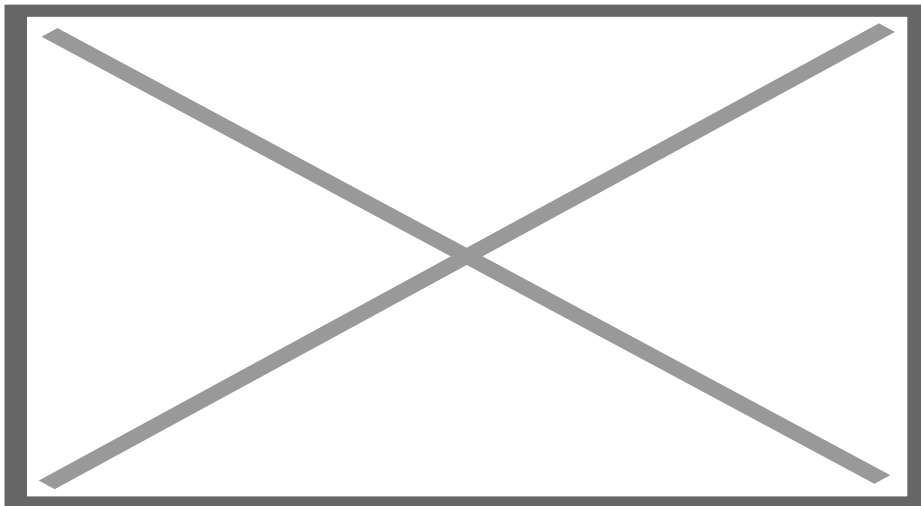
The program is closely integrated within the Paris-Saclay ecosystem of research laboratories and industrial partners.

M1 DiPaQ graduates can either pursue **M2 DiPaC**, focusing on **HPC** and **distributed computing** with specialization in **AI/big data analytics** or **hybrid classical/quantum computing**, or apply for **M2 QMI**, specializing on **quantum information technologies**.

[Télécharger la fiche détaillée](#)

[Apply Now](#)





■ ■

Information

- [Overview](#)
- [Application requirements](#)
- [Schedule](#)

Présentation

Skills

M1 DiPaQ students will gain **competencies and skills** in:

- Understanding current and emerging challenges in distributed, parallel, and quantum systems; assessing their impact on application domains in AI and data science.
- Designing, proving, and analyzing distributed, parallel, and quantum algorithms and protocols, with complexity analyses in time, memory, communication, and energy.
- Mastering the fundamentals of quantum algorithms and complexity; programming quantum circuits with standard libraries.
- Applying parallel programming models and performance engineering on distributed supercomputers and multicore machines.
- Designing and analyzing distributed algorithms with theoretical guarantees.
- Learning the basics of data analysis and AI for large-scale learning tasks.
- Performing code profiling and tracing, diagnosing bottlenecks, and tuning performance.
- Practicing software engineering for HPC using modern C++, profiling, testing, continuous integration, and reproducibility.
- Using version control with Git and maintaining rigorous documentation.
- Communicating technical content and scoping projects in research and industry contexts.
- Adopting responsible, reliable, and sustainable computing practices.

Objectives

Computer systems are evolving toward higher efficiency and richer functionality across three major, interconnected scientific fields:

- **Distributed systems** deliver connectivity and dependable operation across the Internet, clouds, clusters, and sensors, addressing hard problems in synchronization, security, concurrency, and robustness.
- **High-performance and parallel computing (HPC)** tackles intensive workloads in science and AI by exploiting supercomputing architectures and rigorous performance engineering.
- **Quantum computing** provides algorithms and hardware that exploit quantum parallelism to achieve gains unreachable by classical paradigms.

The **M1 DiPaQ master's program** provides students with a deep knowledge of these three fields through both advanced theoretical courses and extensive practice of programming techniques. **Distributed systems** deal with protocols and algorithms that allow connectivity and efficient functionality for network based systems, like Internet, Cloud, sensor networks, computing clusters, blockchain distributed systems, and even microbiological circuits. For these systems, the challenges include synchronization, security, concurrency and robustness. Similar issues arise in the field of **HPC** which aims at efficiently solving computationally intensive problems in applied science or artificial intelligence. **HPC** pushes modern parallel computer architectures to their limits by using various forms of parallelism, data representations and code optimization. So does **distributed computing** by means of various methods of communication and algorithmics. They thereby draw the frontier between what can be achieved within the realm of classical Computer Science, and what will only be accessible through a new paradigm: that of **Quantum Computing**. Quantum Computing and Quantum Information feature novel algorithms and protocols, bringing radical performance gains,

together with their own set of conceptual and technical challenges.

M1 DiPaQ establishes the foundations in **HPC**, **distributed systems**, and **quantum computing** for **large-scale systems** and **applications** in **AI** and **applied science**, with strong ties to the **Paris-Saclay** research and industry ecosystem leveraging these technologies. Students master the fundamentals of these three interconnected disciplines and thereby learn to design fast, scalable, and robust solutions for real-world computational challenges in **big data analytics**, **AI**, **scientific simulations**, and **quantum-enabled workflows** with following objectives:

Knowledge objectives:

- Efficient parallel algorithms and programming on distributed and multi-core parallel machines with vector processing units,
- Advanced C++ programming, debugging, profiling, and performance tuning of HPC kernels
- Large-scale distributed algorithms and systems: replication, consensus, consistency, robustness
- Fundamentals of quantum technologies, algorithms, and programming
- Basics of AI, data science, and optimization for high performance data analytics, machine learning, and scientific computing

Skill objectives:

- Building high-quality parallel and distributed algorithms and software that meet latency, throughput, and performance targets on modern HPC architectures.
- Developing and analyzing scalable, robust algorithms with theoretical guarantees on scalability, consensus, termination, and fault tolerance.
- Optimizing HPC algorithms and code across the stack: complexity, memory locality, vectorization, communication, I/O, and networks.

Resources and practice:

- Access to university clusters and partner supercomputers for hands-on labs, course projects, code development, and tuning.
- Use of open-source toolchains and libraries widely adopted by the HPC community.
- Acquiring modern HPC software engineering practices with advanced C++, IDEs, version control (git), documentation, and continuous integration.

Débouchés

Career prospects

After a Master's or Master's + PhD: engineer (R&D, control, production, etc.)

After a Master's or Master's + PhD: researcher or lecturer

After a Master's + PhD: researcher or lecturer

After a Master's or Master's + PhD: engineer (Research & Development, control, production, etc.)

Ingenieur R&D

R&D Project Manager

Chef de projet

Development Manager

Consulting

Information Systems Manager

Research & Development Engineer

Research and Innovation Manager

Project Manager

Validation/qualification Manager

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

Research Engineer

Head of mission / projects

coordinateur d'expertises scientifiques

Head of development

Consultant.e

After Master and PhD : reseacher or assistant professor or professor

Further Study Opportunities

In particular, this track prepares students for applied research activities, usually undertaken after a PhD, whether in academia or in R&D.

Head of development

Chef·fe de projet/de mission

R&D researcher or data modelling and analysis expert in leading-edge companies or laboratories.

Consultant

Engineering, research and development

Engineering methods and industrialisation

Master 2

Calendar

Start of programme (indicative date)

04/09/2026

End of programme (indicative date)

30/09/2027

Admission

Admission Route

Computer Science

Mathématiques

Physics

Double Licence Mathématique - Physique

Capacity

Available Places

20

Target Audience and Entry Requirements

Prior studies in computer science or a related discipline are desirable. However, students from other fields such as mathematics or physics who have foundational knowledge in computer science (algorithms and programming) can also apply to the **M1 DiPaQ** master's program.

A limited number of scholarships (Eiffel, IDEX, Quantum Saclay) are available for exceptional candidates.

Période(s) de candidature

Inception Platform

From 15/01/2026 to 16/03/2026

Supporting documents

Compulsory supporting documents

Course selection sheet.

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

Motivation letter.

A letter detailing the motivation and reasons for willing to study quantum, parallel and distributed computing in the DiPaQ master's program in the light of previous studies and experiences as well as future career plans.

All transcripts of the years / semesters validated since the high school diploma at the date of application.

Grades of all courses since high school.

Curriculum Vitae.

A CV detailing all previous studies, internships, trainings, work experience (if any), distinctions and awards, as well as other personal interests and activities.

Additional supporting documents

Copy diplomas.

Letter of recommendation or internship evaluation.

Document at your convenience.

VAP file (obligatory for all persons requesting a valuation of the assets to enter the diploma).

only needed in case you have officially validated your prior professional experience to count as equivalent to a university degree

Supporting documents :

- Residence permit stating the country of residence of the first country
- Or receipt of request stating the country of first asylum
- Or document from the UNHCR granting refugee status
- Or receipt of refugee status request delivered in France
- Or residence permit stating the refugee status delivered in France
- Or document stating subsidiary protection in France or abroad
- Or document stating temporary protection in France or abroad.

Programme

[M1 DiPaQ] Disciplinary courses

Subjects	ECTS	Semester	Lecture	directed study	practical class	Lecture/directed study	practical class	directed study/practical class
[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

Semester: Semestre 1

Detail

Lecture: 10.5

Practical study : 10.5

Language(s) of instruction

Anglais

Remote teaching

non

Prerequisites

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/contents

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

Subjects	ECTS	Semester	Lecture	directed study	practical class	Lecture/directed study	Lecture/practical class	directed study/practical class	co
[M1 DiPaQ] Calcul haute performance sur architectures multicœurs	3	Semestre 1	10.5		10.5				

[M1 DiPaQ] Calcul haute performance sur architectures multicœurs

ECTS : 3

Semester: Semestre 1

Detail

Lecture: 10.5

Practical study : 10.5

Language(s) of instruction

Anglais

Remote teaching

non

Prerequisites

Prerequisites:

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

Programme/contents

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

Subjects	ECTS	Semester	Lecture	directed study	practical class	Lecture/directed study	Lecture/practical class	directed study/practical class	0
[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

Semester: Semestre 2

Detail

Lecture: 10.5

Practical study : 10.5

Language(s) of instruction

Anglais

Remote teaching

non

Prerequisites

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/contents

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

Subjects	ECTS	Semester	Lecture	directed study	practical class	Lecture/directed study	Lecture/practical class	directed study/practical class	
[M1 DiPaQ] Programmation C++ avancée	3	Semestre 1	10.5		10.5				

[M1 DiPaQ] Programmation C++ avancée

ECTS : 3

Semester: Semestre 1

Detail

Lecture: 10.5

Practical study : 10.5

Language(s) of instruction

Anglais

Remote teaching

non

Prerequisites

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/contents

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

Subjects	ECTS	Semester	Lecture	directed study	practical class	Lecture/directed study	Lecture/practical class	directed study/practical class	0
[M1 DiPaQ] Optimisation pour le calcul scientifique	3	Semestre 2							

[M1 DiPaQ] Optimisation pour le calcul scientifique

ECTS : 3

Semester: Semestre 2

Language(s) of instruction

Anglais

Remote teaching

non

Prerequisites

Prerequisites:

- Linear algebra
- Multivariable calculus
- Basics in quantum computing

Programme/contents

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.

Subjects	ECTS	Semester	Lecture	directed study	practical class	Lecture/directed study	Lecture/practical class	directed study/practical class	credits
[M1 DiPaQ] Initiation à l’algorithmique et à la programmation quantique	3	Semestre 1	10.5		10.5				6

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

ECTS : 3

Semester: Semestre 1

Detail

Lecture: 10.5

Practical study : 10.5

Language(s) of instruction

Anglais

Remote teaching

non

Prerequisites

Prerequisites:

- Familiarity with linear algebra
- Exposure to programming in Python

Programme/contents

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

Learning objectives

Learning objectives:

Subjects	ECTS	Semester	Lecture	directed study	practical class	Lecture/directed study	Lecture/practical class	directed study/practical class	0
[M1 DiPaQ] Fondements de l'informatique quantique	3	Semestre 2	10.5	10.5					

[M1 DiPaQ] Fondements de l'informatique quantique

ECTS : 3

Semester: Semestre 2

Detail

Lecture: 10.5

Directed study: 10.5

Language(s) of instruction

Anglais

Remote teaching

non

Prerequisites

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/contents

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.

Subjects	ECTS	Semester	Lecture	directed study	practical class	Lecture/directed study	Lecture/practical class	directed study/practical class	0
[M1 DiPaQ] Optimisation combinatoire quantique	3	Semestre 2	10.5	10.5					

[M1 DiPaQ] Optimisation combinatoire quantique

ECTS : 3

Semester: Semestre 2

Detail

Lecture: 10.5

Directed study: 10.5

Language(s) of instruction

Anglais

Remote teaching

non

Prerequisites

Prerequisites:

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

Programme/contents

Course program, plan, content:

Total duration: 21 hours (7 sessions × 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

Subjects	ECTS	Semester	Lecture	directed study	practical class	Lecture/directed study	Lecture/practical class	directed study/practical class	
[M1 DiPaQ] Algorithmes distribués robustes	3	Semestre 1	10.5	10.5					

[M1 DiPaQ] Algorithmes distribués robustes

ECTS : 3

Semester: Semestre 1

Detail

Lecture: 10.5

Directed study: 10.5

Language(s) of instruction

Anglais

Remote teaching

non

Prerequisites

Prerequisites:

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

Programme/contents

Course program, plan, content:

Total duration: 21 hours (6 sessions × 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

Learning objectives

Learning objectives:

Subjects	ECTS	Semester	Lecture	directed study	practical class	Lecture/directed study	Lecture/practical class	directed study/practical class	
[M1 DiPaQ] Algorithmes distribués auto-stabilisants	3	Semestre 2	10.5	10.5					

[M1 DiPaQ] Algorithmes distribués auto-stabilisants

ECTS : 3

Semester: Semestre 2

Detail

Lecture: 10.5

Directed study: 10.5

Language(s) of instruction

Anglais

Remote teaching

non

Prerequisites

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/contents

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.

Learning objectives

Learning objectives:

Self-stabilization is a versatile technique used to overcome any transient failure in a system. The

Subjects	ECTS	Semester	Lecture	directed study	practical class	Lecture/directed study	Lecture/practical class	directed study/practical class	
[M1 DiPaQ] Graphes et algorithmes	3	Semestre 1	10.5	10.5					

[M1 DiPaQ] Graphes et algorithmes

ECTS : 3

Semester: Semestre 1

Detail

Lecture: 10.5

Directed study: 10.5

Language(s) of instruction

Anglais

Remote teaching

non

Prerequisites

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/contents

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

Subjects	ECTS	Semester	Lecture	directed study	practical class	Lecture/directed study	Lecture/practical class	directed study/practical class	
[M1 DiPaQ] Algorithmes probabilistes	3	Semestre 2	10.5	10.5					

[M1 DiPaQ] Algorithmes probabilistes

ECTS : 3

Semester: Semestre 2

Detail

Lecture: 10.5

Directed study: 10.5

Language(s) of instruction

Anglais

Remote teaching

non

Prerequisites

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/contents

Course program, plan, content:

Total duration: 21 hours (6 sessions × 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,

[M1 DiPaQ] Specialization courses

Subjects	ECTS	Semester	Lecture	directed study	practical class	Lecture/directed study	Lecture/practical class	directed study/practical class
[DKAI] MATHEMATICS FOR DATA SCIENCE	3	Semestre 1	12	4.5	4.5			

[DKAI] MATHEMATICS FOR DATA SCIENCE

ECTS : 3

Semester: Semestre 1

Detail

Lecture: 12

Practical study : 4.5

Directed study: 4.5

Language(s) of instruction

Anglais

Remote teaching

non

Prerequisites

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

Learning objectives

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>

Subjects	ECTS	Semester	Lecture	directed study	practical class	Lecture/directed study	Lecture/practical class	directed study/practical class
[AI] Hands on machine learning with scikitlearn	3	Semestre 1	9	12				
■								

[AI] Hands on machine learning with scikitlearn

ECTS : 3

Semester: Semestre 1

Detail

Lecture: 9

Directed study: 12

Language(s) of instruction

Anglais

Remote teaching

non

Learning objectives

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

Type of assessment

Evaluation Continue non Intégrale

[AI] Machine Learning Basics 1	3	Semestre 1
--	---	------------

[AI] Machine Learning Basics 1

ECTS : 3

Semester: Semestre 1

Language(s) of instruction

Anglais

Remote teaching

non

Prerequisites

- (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).

Learning objectives

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,

Christoph Bishop, 2006 (available online for free)

Subjects	ECTS	Semester	Lecture	directed study	practical class	Lecture/directed study	Lecture/practical class	directed study/practical class
Impacts Environnementaux du numérique	3	Semestre 2	10.5	10.5				

Impacts Environnementaux du numérique

ECTS : 3

Semester: Semestre 2

Detail

Lecture: 10.5

Directed study: 10.5

Language(s) of instruction

Anglais

Remote teaching

non

Programme/contents

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.

Learning objectives

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.

Type of assessment

Evaluation Continue Intégrale

[\[M1 DiPaQ\]](#)

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

6

Semestre 2

Subjects	ECTS	Semester	Lecture	directed study	practical class	Lecture/directed study	Lecture/practical class	directed study/practical class
----------	------	----------	---------	----------------	-----------------	------------------------	-------------------------	--------------------------------



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

ECTS : 6

Semester: Semestre 2

Detail

Project : 21

Language(s) of instruction

Anglais

Remote teaching

non

Type of assessment

Evaluation Terminale

[\[M1 DiPaQ\] Stage ou école d'été](#) 6 Semestre 2



[M1 DiPaQ] Stage ou école d'été

ECTS : 6

Semester: Semestre 2

Language(s) of instruction

Anglais

Remote teaching

non

Type of assessment

Evaluation Terminale

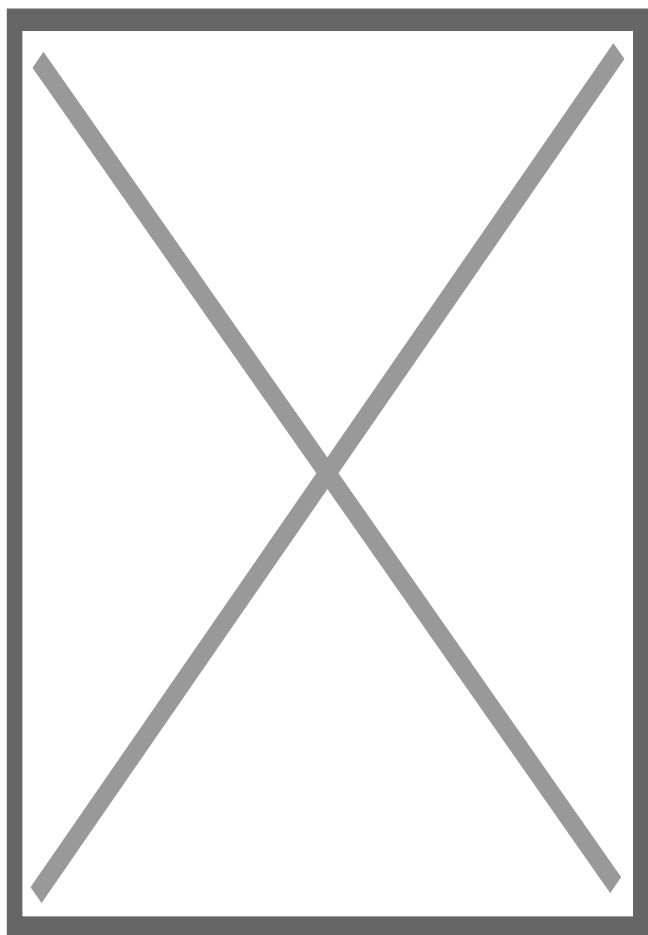
Teaching Location(s)

ORSAY

GIF SUR YVETTE

Training campus

Javascript est requis pour visualiser cette carte.



Orsay Bures

Orsay / Bures-sur-Yvette

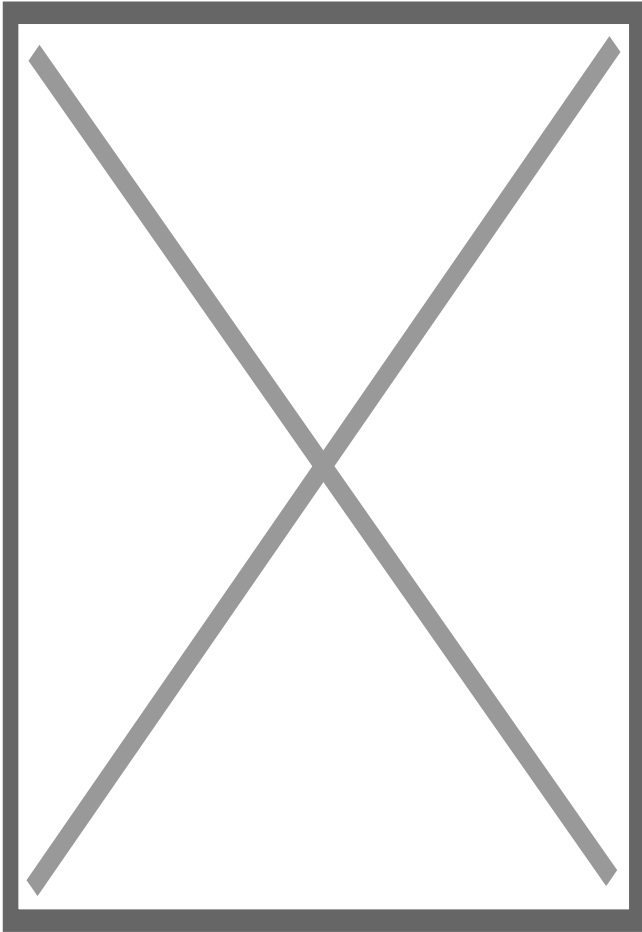
Bus 4607, 4626

RER B Orsay ville ou Bures-sur-Yvette

[Voir](#)

Library

Community center



Saclay Corbeville

Saclay / Palaiseau

Bus 4606, 4627, 5154, 9105, 9108

RER B Palaiseau, Palaiseau villebon ou Lozère

[Voir](#)

Student restaurant (CROUS)

Student residence

Sports facilities

Contact

- **Thomas GALLEY**

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