FORMATION INTERUNIVERSITAIRE EN INFORMATIQUE À L'ÉCOLE NORMALE SUPÉRIEURE

Année scolaire 2012/2013

Département d'informatique – ENS, 45 rue d'Ulm, F-75230 Paris Cedex 05

http://diplome.di.ens.fr

Édition du 19 septembre 2012

Table des Matières

1. Les études d'informatique à l'ENS et le diplôme de l'École normale supérieure

1.1. Présentation du diplôme de l'École normale supérieure

Suite à la mise en place du système universitaire européen LMD (Licence, Master, Doctorat), l'École normale supérieure a créé en 2005 un diplôme d'établissement intitulé diplôme de l'École normale supérieure, qui complète le cursus universitaire. Sa finalité est d'offrir une variété de parcours conjuguant une formation d'excellence dans une discipline principale avec une ouverture à la fois souple et ambitieuse dans d'autres disciplines.

Le diplôme est ouvert à des étudiants, issus des classes préparatoires aux grandes écoles et des universités françaises ou étrangères, désireux de recevoir la même formation que les élèves normaliens (fonctionnaires stagiaires ou boursiers de la section internationale). Les étudiants font l'objet d'une procédure de sélection spécifique (cf.4 *Conditions et procédures d'admission*).

Le diplôme de l'ENS est délivré au terme d'une scolarité d'une durée de trois ans (en règle générale[1]) pendant laquelle chaque étudiant fait valider :

- un cursus universitaire de haut niveau sanctionné par l'obtention d'un master dans une discipline qui constitue la spécialité principale du diplôme. En règle générale[1], ce cursus comprend une troisième année de licence (L3) et les deux années du master (M1 et M2), chacune des trois années correspondant à la validation d'un total de 60 unités ECTS (*European Credit Transfer System*) ;
- des formations supplémentaires validées par 36 unités ECTS au minimum sur l'ensemble des trois années. Celles-ci peuvent (i) relever de la spécialité principale (enseignements suivis dans la discipline du master), (ii) constituer la spécialité secondaire du diplôme (enseignements coordonnés dans une autre discipline) ou (iii) exploiter toute la diversité de l'offre pédagogique proposée par l'ENS aux étudiants.

1.2. L'informatique dans le diplôme de l'École normale supérieure

Dans le cadre du diplôme de l'ENS, les études d'informatique prennent deux formes distinctes selon que cette discipline constitue ou non la spécialité principale de l'étudiant :

- **Élèves et étudiants informaticiens** : le Département d'informatique est la structure pédagogique et scientifique à laquelle sont rattachés tous ceux qui envisagent l'informatique comme discipline principale. Les étudiants inscrits auprès du Département d'informatique suivent la *filière informatique du diplôme de l'ENS*, qui constitue une formation spécifique de haut niveau comprenant notamment une troisième année de licence (L3) et les deux années d'un master (M1 et M2).
- **Élèves et étudiants relevant d'un autre département scientifique** *:* le Département d'informatique propose également une offre de formation aux étudiants inscrits dans d'autres disciplines auprès des différents départements de l'ENS. Il peut s'agir, soit d'un ensemble cohérent d'enseignements en informatique ayant vocation à constituer la spécialité secondaire de l'étudiant, soit de formations ponctuelles que l'étudiant fait valider pour son diplôme en accord avec son tuteur et les responsables de ces enseignements.

Des passerelles existent entre les différents départements de l'ENS. Sous réserve de l'accord des directions des études concernées, une réorientation peut être envisagée en cours de scolarité pour les élèves de la filière informatique, soit vers d'autres disciplines au sein du diplôme de l'ENS, soit vers d'autres formations universitaires en dehors de ce cadre.

[1] L'intégration d'étudiants français ou étrangers peut aussi s'envisager en deuxième année de scolarité, c'est-à-dire à l'entrée du master : le diplôme de l'ENS est alors délivré au terme des deux années du master (M1 et M2).

2. Responsables de la formation

Directeur des études :

Patrick Cousot

Directeur des enseignements :

Jean Vuillemin

Secrétariat :

Isabelle Delais
École normale supérieure
Département d'informatique
45, rue d'Ulm – 75230 Paris cedex 05
Tél : +33 (0) 1 44 32 20 45
Fax : +33 (0) 1 44 32 20 75
Mél : diplome AT di.ens.fr ou isabelle.delais AT ens.fr

Page d'accueil : http://diplome.di.ens.fr

3. Objectifs et débouchés

3.1. Objectifs de la filière informatique du diplôme de l'ENS

Les étudiants inscrits dans la filière informatique du diplôme sont rattachés statutairement au Département d'informatique de l'ENS. Celui-ci leur propose, au titre de leur spécialité principale, un parcours de type universitaire aux effectifs réduits (une vingtaine d'étudiants par promotion, en incluant les élèves normaliens et les boursiers de la section internationale) dans lequel est dispensée une formation originale d'informaticiens possédant une bonne connaissance générale des mathématiques pures et appliquées dans des secteurs variés. Un encadrement renforcé permet un rythme plus rapide et une réflexion plus approfondie que dans d'autres formations. Les enseignements sont complétés par des stages de recherche obligatoires.

Objectifs des études en informatique à l'ENS :

- l'intégration, dans un cursus d'excellence en informatique, des élèves normaliens, des boursiers de la section internationale et des étudiants retenus par le Département d'informatique pour suivre leurs études dans le cadre du diplôme de l'ENS (cf.4 *Conditions et procédures d'admission*) ;
- une formation à et par la recherche, visant à assurer une professionnalisation exigeante, qui se traduit par un enseignement d'un très haut niveau scientifique. Les stages à l'étranger, ciblés et obligatoires, permettent en outre une ouverture internationale dans les domaines choisis ;
- une orientation de chaque élève ou étudiant, respectueuse de la diversité des profils et des choix personnels, rendue possible par une structure aussi flexible que possible. À cette fin, chaque étudiant de la filière informatique est suivi par un tuteur, à titre individuel, tout au long de son parcours ; il est invité à suivre également quelques enseignements d'ouverture à d'autres disciplines qui personnalisent ce parcours de formation.

3.2. Débouchés de la filière informatique du diplôme de l'ENS

Tout étudiant titulaire du diplôme de l'ENS aura acquis un master recherche. Un étudiant qui a suivi la filière informatique du diplôme peut donc commencer une thèse de doctorat en mathématiques ou en informatique, qu'il achèvera en principe au terme de deux ou trois années de travail de recherche à l'issue de la scolarité. Il peut également commencer immédiatement une carrière non académique.

À moyen terme, une fois la thèse de doctorat éventuelle achevée, les débouchés possibles après 1, 2 ou 3 ans sont les suivants :

- chercheur en informatique dans un grand organisme de recherche publique (CNRS, CEA, INRIA, ONERA, CNES, etc.) ;
- enseignant-chercheur à l'université en France ou à l'étranger ;
- chercheur en informatique dans l'industrie (France Telecom, EADS, etc.) ;
- ingénieur informaticien dans l'industrie en France ou à l'étranger ;
- enseignant dans les classes préparatoires aux grandes écoles, plus généralement, dans l'enseignement post-baccalauréat (IUT, CNAM, etc.).

4. Conditions et procédures d'admission

4.1. Les élèves de l'ENS et boursiers de la section internationale

Ayant acquis 120 unités ECTS (niveau L2) en classes préparatoires aux grandes écoles et réussi le concours d'entrée à l'ENS, les élèves normaliens désireux de suivre un cursus en informatique commencent leurs études en informatique à l'ENS en s'inscrivant en première année de scolarité (L3).

Les boursiers de la section internationale accèdent également aux études en informatique à l'ENS, soit au niveau L3, soit directement au niveau M1 selon les études effectuées au préalable dans leur université d'origine.

4.2. Les étudiants du diplôme de l'ENS

Tout étudiant issu, soit d'une classe préparatoire aux grandes écoles, soit d'une université française ou d'un établissement universitaire étranger, justifiant directement ou par équivalence de 120 unités ECTS (niveau L2) ou 180 unités ECTS (niveau L3) est autorisé à se porter candidat au diplôme de l'ENS pour y effectuer des études en informatique.

Pour la filière informatique comme dans les autres disciplines représentées au sein du diplôme de l'ENS, la règle générale est de recruter les étudiants au niveau de la troisième année de licence (L3). L'intégration peut toutefois s'effectuer au niveau du master (M1), notamment pour les étudiants issus des universités étrangères.

4.3. Candidature à la filière informatique du diplôme de l'ENS

Les dates exactes (entre la mi-avril et la mi-juillet) et les modalités de candidature sont précisées chaque année à l'URL suivante :

http://diplome.di.ens.fr/Candidature.html

N.B. Bourses et Hébergement :

Les possibilités d'hébergement à l'ENS pour les étudiants admis à préparer le Diplôme de l'ENS sont extrêmement réduites. Les candidats concernés doivent s'en préoccuper sans attendre les résultats des jurys.

Les demandes de bourse et d'hébergement du Crous doivent être effectuées avant le 30 avril environ chaque année.

Les étudiants de 1ere année en informatique à l'ENS seront inscrits en L3 d'informatique à l'université de Paris 7.

Les étudiants de 2ème année en informatique à l'ENS seront inscrits en M1 d'Informatique à l'ENS de Paris.

5. Inscriptions, tutorat et programmes d'études

A la mi-septembre, les étudiants admis à préparer le diplôme de l'ENS en informatique et les élèves qui souhaitent faire des études en informatique se présentent au secrétariat du département d'informatique pour procéder aux inscriptions administratives et pédagogiques et pour qu'un tuteur leur soit affecté.

5.1. Inscriptions administratives et pédagogiques

Les élèves/étudiants de la filière informatique doivent s'acquitter, pendant chacune des trois années de leur scolarité, d'une double inscription (i) au diplôme universitaire correspondant au cursus de l'année en cours et (ii) au diplôme d'établissement qu'est le diplôme de l'ENS :

	Cursus universitaire	**Diplôme de l'ENS**
1ère année :	inscription en L3 à l'université Paris 7	inscription à l'ENS
2e année :	inscription en M1 (M.P.R.I.) à l'ENS	inscription à l'ENS
3e année :	inscription en M2 (M.P.R.I.) à l'ENS	inscription à l'ENS

Cursus universitaire : après leur admission à suivre les études en informatique à l'ENS, les élèves/étudiants s'inscrivent - via le secrétariat d'enseignement du département d'informatique de l'ENS - en licence (L3) à l'université Paris 7 qui leur délivre le diplôme de licence à l'issue de la première année de scolarité.

Au cours des deux années suivantes, les élèves/étudiants s'inscrivent au Master parisien de recherche en informatique (M.P.R.I.) à l'ENS qui leur délivre le diplôme de master au terme de leur scolarité. Il est également possible de s'inscrire dans un autre master (comme le master recherche spécialité « Mathématiques appliquées — mathématiques/vision/apprentissage » de l'École normale supérieure de Cachan).

Diplôme de l'ENS : l'élève/étudiant s'inscrit au « diplôme de l'École normale supérieure » au bureau des inscriptions du service de la scolarité. Le secrétariat du département d'informatique fournit la liste des documents nécessaires.

5.2. Tutorat

Un tuteur, enseignant ou chercheur au département d'informatique, est affecté à chaque élève/étudiant. Le rôle du tuteur est d'aider l'élève/étudiant dans l'organisation de ses études, de le conseiller pour ses stages, ses recherches, son orientation.

Il est recommandé de rencontrer régulièrement son tuteur et pas seulement au moment des programmes d'études et des bilans.

L'élève/étudiant peut demander à changer de tuteur.

5.3. Programmes d'études

Chaque année, l'élève/étudiant s'engage sur un programme d'études (ou contrat d'études) annuel qui est déposé auprès de la direction des études de l'ENS après signature du tuteur et du directeur des études du Département d'informatique. Ce document contient les enseignements obligatoires du cursus choisi, mais aussi les cours et les activités qui pourront être comptabilisés pour le diplôme, les stages, etc.

Le département d'informatique peut aussi demander des programmes d'études complémentaires.

6. Organisation de la formation pédagogique

Les études en informatique dispensées dans le cadre du diplôme de l'ENS sont organisées sur trois années, correspondant aux années universitaires L3 (Licence), M1 et M2 (Master). L'obtention des diplômes universitaires requiert la validation de 60 ECTS par année. A la fin de sa scolarité à l'ENS, l'élève/étudiant qui a obtenu un master-recherche et validé, par ailleurs, des enseignements supplémentaires à hauteur de 36 unités ECTS recevra le diplôme de l'École normale supérieure à condition d'avoir effectué les inscriptions requises. On rappelle qu'aucun enseignement validé dans le cadre d'un diplôme universitaire national (licence ou master) ne peut l'être une seconde fois dans le cadre des enseignements supplémentaires du diplôme de l'ENS.

Les enseignements supplémentaires du diplôme de l'ENS se répartissent en trois catégories :

- **un module de langue (3 ECTS)** à valider en en première, deuxième ou troisième année**.** Il existe à l'ENS une structure intitulée ECLA (Espace des cultures et langues d'ailleurs), spécialisée dans l'enseignement des langues qui propose un grand choix de modules de langues vivantes et en particulier, **des modules d'anglais pour scientifiques qui sont vivement recommandés aux élèves/étudiants d'informatique**. L'élève/étudiant peut être dispensé de ce module de langue s'il effectue un stage de 6 mois ou plus dans un pays non francophone.
- **12 ECTS de modules obligatoires de la filière informatique**. Ces modules sont à choisir parmi les modules de niveau M1 ou M2 non retenus pour valider le M.P.R.I, ou parmi les modules de mathématiques de niveau M1 ou plus.
- des modules laissés au libre choix de l'élève/étudiant, **dont 12 ECTS au minimum hors informatique**. Ces modules sont choisis dans l'offre de formation d'autres départements de l'ENS ou dans d'autres formations universitaires, avec l'accord de leurs responsables pédagogiques, sous le contrôle du tuteur et du directeur des études du Département d'informatique.

Il est très fortement recommandé de valider chaque année 12 ECTS d'enseignements supplémentaires.

6.1. Première année

La **validation de la licence (L3) d'informatique**, dans le cadre du partenariat avec l'université Paris 7, nécessite l'obtention de 60 ECTS, répartis en 48 ECTS de cours de niveau L3 (premier et deuxième semestre) et M1 (deuxième semestre) et 12 ECTS de stage. Les enseignements sont organisés et donnés à l'ENS. Ils sont régulièrement renouvelés pour suivre de près l'actualité scientifique. La diversité des sujets traités et celle de l'origine des enseignants permettent une grande variété de débouchés potentiels.

Les élèves/étudiants valident également des enseignements supplémentaires pour le diplôme de l'ENS (au moins 12 ECTS recommandés par année).

En fin de première année, les élèves/étudiants effectuent un stage en laboratoire (universitaire et industriel) avec une priorité donnée à la province.

À la fin de la première année, la commission des études, en partenariat avec l'université Paris 7, statue sur l'obtention par l'élève/étudiant du diplôme de licence et sur son admission en seconde année du diplôme de l'ENS.

A partir de septembre 2012, des **filières info-maths et maths-info** sont proposées en première année. Les élèves qui la choisiront seront inscrits en **L3 de Mathématiques OU en L3 d'Informatique et valideront une seule licence.** (*cf. 7.1.4 Filières Info-maths et Maths-info de 1ère année*).

6.2. Deuxième année

La deuxième année est constituée, au premier semestre, de cours de niveau M2 pour 30 ECTS, et au second semestre d'un stage de recherche de 5 mois environ, en laboratoire à l'étranger, comptant 30 ECTS.

En parallèle sont proposés des mini-cours de niveau recherche assurés par des spécialistes (le plus souvent par des professeurs étrangers invités à l'ENS). Les élève/étudiants valident également des enseignements supplémentaires pour le diplôme de l'ENS (au moins 12 ECTS recommandés par année).

La commission des études se réunit à nouveau en fin de seconde année pour statuer sur la validation par l'élève/étudiant de la première année du master (M1) et sur son admission en troisième année du diplôme de l'ENS.

6.3. Troisième année

Durant cette troisième année, l'élève/étudiant achève son master en suivant au premier semestre des cours de niveau M2 pour 30 ECTS, et effectue au second semestre un stage de recherche de 5 mois minimum, en France ou à l'étranger, validant 30 ECTS. Les élèves/étudiants valident également des enseignements supplémentaires pour le diplôme de l'ENS (au moins 12 ECTS recommandés par année).

L'année se termine le plus fréquemment par le choix d'un directeur et d'un sujet de thèse de doctorat. À ce niveau, les élèves/étudiants s'intègrent progressivement dans un laboratoire de recherche. Afin de faciliter l'insertion dans le milieu de la recherche, il est souvent judicieux de passer tout ou partie de cette année dans un laboratoire de province ou d'un autre pays européen.

La commission des études du master statue en fin d'année sur l'obtention du diplôme de master par l'élève/étudiant qui est alors proposé pour l'obtention du diplôme à la direction des études de l'ENS.

Si l'élève/étudiant s'est acquitté, en outre, de la validation d'enseignements supplémentaires à hauteur de 36 unités ECTS sur l'ensemble de sa scolarité, l'ENS lui délivre le « diplôme de l'École normale supérieure » avec une spécialité principale qui correspond à celle de son master et, le cas échéant, une spécialité secondaire dans une autre discipline (cf. *Présentation du diplôme de l'ENS*).

6.4. Stages

Outre les stages obligatoires de L3, M1 et M2, il est possible à partir de la 2ème année de scolarité de la filière informatique de faire une année de stage à l'étranger.

NB. Il est nécessaire de préparer les stages plusieurs mois à l'avance pour obtenir l'accord des directeurs des études, effectuer les démarches d'obtention de visa, obtenir la signature des conventions de stage et des ordres de mission et parfois obtenir une année de césure (étudiants) ou un congé sans traitement (élèves).

7. Cours de l'année universitaire 2012-2013

Pour chaque cours dans la liste ci-dessous, sont indiqués le nom du professeur responsable et le nombre d'ECTS (*European Credit Transfer System*).

Les coordonnées de la plupart des enseignants se trouvent sur l'annuaire Intranet de l'ENS ou sur Internet.

Une description plus détaillée de chaque cours se trouve dans la dernière partie de cette plaquette.

7.1. Première année : licence (L3).

7.1.1 Premier semestre de la licence (L3) d'informatique

Les 4 cours suivants sont obligatoires :

ALGORITHMIQUE ET PROGRAMMATION (6 ECTS)
Jacques Stern, Claire Mathieu

LANGAGES DE PROGRAMMATION ET COMPILATION (6 ECTS)
Jean-Christophe Filliâtre

LANGAGES FORMELS, CALCULABILITÉ ET COMPLEXITÉ (6 ECTS)
Eugène Asarin

SYSTÈME DIGITAL: DE L'ALGORITHME AU CIRCUIT (6 ECTS)
Jean Vuillemin

L'élève/étudiant doit suivre et valider au moins 1 cours de mathématiques ou maths-info suivants en 1ère année :

ALGÈBRE 1 (12 ECTS) – 1er semestre
Olivier Biquard

INTÉGRATION ET PROBABILITÉS (12 ECTS) – 1er semestre
Zhan SHI

STRUCTURES ET ALGORITHMES ALÉATOIRES (6 ECTS) – 1er semestre
Anne Bouillard, Pierre Bremaud

ANALYSE COMPLEXE ET HARMONIQUE (12 ECTS) – 2e semestre
Wendelin Werner

APPRENTISSAGE STATISTIQUE (12 ECTS) - 2e semestre
Gilles Stoltz
Cours spécifique aux filières maths-info et info-maths

7.1.2 Deuxième semestre de la licence (L3) d'informatique

Le cours d'informatique suivant est obligatoire :

SYSTÈMES ET RÉSEAUX (6 ECTS)
Marc Pouzet

L'élève/étudiant doit choisir et valider au moins 2 cours d'informatique, de niveau M1, parmi les cours suivants. L'un de ces cours peut-être remplacé par un cours de mathématiques.

BASES DE DONNÉES (6 ECTS)
Serge Abiteboul
(Ce cours a lieu à l'ENS de Cachan)

BASES GÉOMÉTRIQUES DE L'INFORMATIQUE (6 ECTS)
Michel Pocchiola, Jean Ponce

GÉNIE LOGICIEL ET CLOUD COMPUTING (6 ECTS)
Joannes Vermorel

L'INFORMATIQUE SCIENTIFIQUE PAR LA PRATIQUE (6 ECTS)
David Naccache

INITIATION À LA CRYPTOLOGIE (6 ECTS)
Jacques Stern, **David Naccache**, **Damien Vergnaud**

LOGIQUE ET INFORMATIQUE (6 ECTS)
Jean Goubault-Larrecq
*(Ce cours **fortement recommandé** a lieu à l'ENS de Cachan)*

RÉSEAUX DE COMMUNICATION (6 ECTS)
Anna Busic

THÉORIE DE L'INFORMATION ET CODAGE (6 ECTS)
Marc Lelarge

TRAITEMENT DU SIGNAL (6 ECTS)
Stéphane Mallat

7.1.3 Stage

L'élève/étudiant doit effectuer un stage d'initiation à la recherche en informatique de 2 à 3 mois dans un laboratoire de recherche public ou privé, en France (de préférence en province) ou en Europe, entre début juin et fin août 2013.
Le stage (qui comprend aussi la rédaction d'un rapport et une soutenance) comptera 12 ECTS pour la licence (L3).
Les stages de L3 des dernières années : http://www.di.ens.fr/~vergnaud/stages.html

7.1.4 Filières Info-Maths et Maths-Info de 1ère année (à compter de septembre 2012)

Les élèves normaliens ou de la sélection internationale qui choisissent une de ces filières seront inscrits et valideront une seule licence : **L3 d'informatique ou L3 de mathématiques.**

Ces filières sont organisées conjointement par la FIMFA et le Département d'informatique de l'ENS. Elles permettent :
- aux élèves motivés de poursuivre une double formation en informatique et en mathématiques.
- aux élèves encore indécis de repousser d'une année le choix entre ces deux disciplines.

--- **Info-maths pour les élèves inscrits en L3 d'informatique et rattachés au Département d'Informatique (DI):**

- Cours d'informatique :

-- 4 cours d'informatique au 1er semestre :

- Algorithmique et programmation
- Langages formels, calculabilité et complexité
- Langages de programmation et de compilation
- Structures et algorithmes aléatoires

-- 2 cours d'informatique au 2ème semestre

-- + le cours spécifique à ces filières : Apprentissage au 2ème semestre

- Cours de mathématiques :

-- 2 cours au choix parmi les 5 ci-dessous :

- Logique (1er semestre)
- Intégration et probabilités (1er semestre)
- Algèbre I (1er semestre)
- Algèbre II (2ème semestre)
- Processus aléatoires (2ème semestre)

-- + Analyse complexe et harmonique (2ème semestre)

- Stage (12 ECTS) et Exposé du cursus maths-informatique (12 ECTS)

Ce travail personnel bi-disciplinaire, encadré par un enseignant de chaque discipline, consiste en :

- un travail bibliographique comparable à l'exposé de première année du cursus Mathématiques au cours du second semestre, sous la houlette d'un enseignant de mathématiques et/ou d'un enseignant d'informatique, sur un sujet relié à celui du stage,

- un stage niveau L3 dans un laboratoire d'informatique,

- la rédaction d'un mémoire en deux parties et une soutenance en présence d'enseignants des deux disciplines.

Les élèves qui voudront poursuivre en mathématiques en deuxième année devront obtenir l'accord du département de mathématiques et rattraper les cours fondamentaux non validés en première année.

--- **Maths-Info pour les élèves inscrits en L3 de mathématiques et rattachés au Département de Mathématiques (DMA):**

- Cours de mathématiques :

-- 3 cours au choix parmi les 5 ci-dessous :

- Logique (1^{er} semestre)
- Intégration et probabilités (1^{er} semestre)
- Algèbre I (1^{er} semestre)
- Algèbre II ($2^{ème}$ semestre)
- Processus aléatoires ($2^{ème}$ semestre)

-- + le cours spécifique à ces filières : Apprentissage au 2ème semestre

-- + Analyse complexe et harmonique ($2^{ème}$ semestre)

- Cours d'informatique :

-- 2 cours d'informatique au 1er semestre :

- Langages formels, calculabilité et complexité
- Algorithmique et programmation **ou** Langages de programmation et de compilation

-- + 1 cours d'informatique au 2ème semestre

- Stage (12 ECTS) et Exposé du cursus maths-informatique (12 ECTS)

Ce travail personnel bi-disciplinaire, encadré par un enseignant de chaque discipline, consiste en :

- un travail bibliographique comparable à l'exposé de première année du cursus Mathématiques au cours du second semestre, sous la houlette d'un enseignant de mathématiques et/ou d'un enseignant d'informatique, sur un sujet relié à celui du stage,

- un stage niveau L3 dans un laboratoire d'informatique,

- la rédaction d'un mémoire en deux parties et une soutenance en présence d'enseignants des deux disciplines.

Les élèves qui voudront poursuivre en informatique en deuxième année devront obtenir l'accord du département d'informatique et rattraper certains cours obligatoires de licence ainsi que Logique si ce cours n'a pas été validé en première année.

7.2. Deuxième année : master (M1)

7.2.1 Premier semestre

A compter de 2012-2013, l'élève/étudiant en M1 d'informatique doit suivre et valider au moins 1 cours de mathématiques sauf s'il a déjà validé 2 cours de mathématiques dont Logique en 1re année.

Le cours Logique est obligatoire sauf si validé en 1re année

LOGIQUE (12 ECTS)
Martin Hils

L'élève/étudiant en M1 d'informatique est fortement encouragé à suivre et à valider 1 cours de maths ou maths-info supplémentaire parmi :

ALGÈBRE 1 (12 ECTS)
Olivier Biquard

INITIATION À LA MODÉLISATION ET À LA SIMULATION NUMÉRIQUE (12 ECTS)
Erwan Faou - David Lannes

INTÉGRATION ET PROBABILITÉS (12 ECTS)
Zhan SHI

STATISTIQUE (12ECTS)
Gérard Biau

STRUCTURES ET ALGORITHMES ALÉATOIRES (6 ECTS) – 1er semestre
Anne Bouillard, Pierre Bremaud

TOPOLOGIE ET CALCUL DIFFERENTIEL (12ECTS)
Patrick Bernard

Dans le cadre de son diplôme de master M1, l'élève/étudiant doit valider 30 ECTS de cours au 1er semestre.
En plus des cours de mathématiques, l'élève/étudiant doit choisir des cours d'informatique dans la liste suivante ou dans les cours du MPRI (Master Parisien de Recherche en Informatique).
Il est également possible, avec l'accord du tuteur et du directeur des études, de choisir les cours dans d'autres masters comme par exemple le master MVA (Mathématiques, Vision, Apprentissage) de l'ENS de Cachan.
Les élèves/étudiants sont fortement encouragés à suivre et à valider 1 ou 2 cours supplémentaires de cette liste pour le diplôme de l'ENS.

ALGORITHMES ARITHMÉTIQUES POUR LA CRYPTOLOGIE (3ECTS) COURS MPRI 2-12-2
François Morain

ALGORITHMIQUE DISTRIBUÉE POUR LES RÉSEAUX (3ECTS) COURS MPRI 2-18-1
Pierre Fraigniaud

ALGORITHMIQUE POUR LES GRAPHES PLONGÉS (3ECTS) COURS MPRI 2-38-1
Eric Colin de Verdière, Claire Mathieu

CATÉGORIES, LAMBDA-CALCULS (6ECTS) COURS MPRI 1-20
Paul-André Melliès

COMPLEXITÉ AVANCÉE (6ECTS) COURS MPRI 1-17
Jean Goubault-Larrecq

FONDEMENTS SUR LA MODÉLISATION DES RÉSEAUX (3ECTS) COURS MPRI 2-17-1
François Baccelli & Jean Mairesse

INTERPRÉTATION ABSTRAITE : APPLICATION À LA VÉRIFICATION ET À L'ANALYSE STATIQUE (6 ECTS) COURS MPRI 2-6
Patrick Cousot, Radhia Cousot

INTRODUCTION A LA VISION ARTIFICIELLE (4ECTS) COURS INFO
Jean Ponce

MÉTHODES MATHÉMATIQUES POUR LES NEUROSCIENCES (4ECTS)
COURS INFO & **MVA** & UPMC MATHS ET APPLICATIONS
Olivier Faugeras

PLANIFICATION DE MOUVEMENT EN ROBOTIQUE ET EN ANIMATION GRAPHIQUE : DU CONTINU AU COMBINATOIRE VIA LA COMMANDABILITÉ DES SYSTÈMES (6ECTS) COURS MPRI 1-19
Jean-Paul Laumond

PROTOCOLES CRYPTOGRAPHIQUES: PREUVES FORMELLES ET CALCULATOIRES (6ECTS) COURS MPRI 2-30
Hubert Comon-Lundh, David Pointcheval

RECONNAISSANCE D'OBJETS ET VISION ARTIFICIELLE (4ECTS) COURS INFO & **MVA**
Ivan Laptev, Cordelia Schmid, Josef Sivic

SÉMANTIQUE, LANGAGES ET ALGORITHMES POUR LA PROGRAMMATION MULTICORE (3ECTS) COURS MPRI 2-37-1
Albert Cohen

SYSTÈMES SYNCHRONES (3ECTS) COURS MPRI 2-23-1
Marc Pouzet, Jean Vuillemin

TECHNIQUES EN CRYPTOGRAPHIE ET CRYPTANALYSE (3ECTS) COURS MPRI 2-12-1
Michel Abdalla, Phong Nguyen, Vadim Lyubashevsky

7.2.2 Deuxième semestre : stage

L'élève/étudiant doit effectuer un stage de 5 mois environ à l'étranger pour valider 30 ECTS dans le cadre de son diplôme M1.

7.3. Troisième année : master (M2)

7.3.1 Premier semestre

L'élève/étudiant doit valider 30 ECTS de cours du MPRI de niveau M2. Il est possible de valider certains cours d'une autre formation universitaire, avec l'accord de son tuteur et l'accord de Pierre-Alain Fouque qui représente l'ENS de Paris auprès de la commission des études du MPRI.

A la place du M2 du MPRI, l'élève/étudiant peut préparer et valider le M2 du Master MVA (Mathématiques, Vision, Apprentissage) de l'ENS Cachan avec l'accord de son tuteur et l'accord du directeur des études du département d'informatique.

L'élève/étudiant doit également valider des enseignements supplémentaires pour valider le diplôme de l'ENS, s'il n'a pas déjà acquis les 36 ECTS requis.

7.3.2 Deuxième semestre : stage

Pour le M2 du MPRI, l'élève/étudiant doit effectuer un stage d'environ 5 mois, en France ou à l'étranger, qui comptera 30 ECTS.

Pour le M2 du MVA, le stage obligatoire est d'environ 4 mois et a lieu entre avril et septembre.

8. Enseignements d'informatique du diplôme de l'ENS (hors filière informatique)

8.1. L'informatique comme « spécialité secondaire » du diplôme de l'ENS

Un élève/étudiant inscrit dans une autre filière du diplôme que la filière informatique (et donc rattaché à un autre département de l'ENS que le Département d'informatique) peut choisir de valider un ensemble cohérent d'enseignements d'informatique pour constituer la spécialité secondaire de son diplôme. Cet ensemble cohérent d'enseignements doit représenter un total d'au moins 24 ECTS.

8.2. L'informatique dans le diplôme de l'ENS

Les élèves/étudiants possédant déjà des notions de base en informatique peuvent suivre des cours de L3 au premier semestre.

Le Département d'Informatique propose également un cours d'initiation à la programmation ouvert à tous :

INITIATION À LA PROGRAMMATION POUR NON-INFORMATICIENS (3ECTS) INFO
Damien Vergnaud

(Cours enseigné au 2ème semestre)

Ce cours est ouvert aux élève/étudiants de toutes les disciplines, littéraires comme scientifiques. Aucune connaissance préalable en programmation n'est requise. Le cours n'est pas orienté à priori vers une application particulière. Il s'adaptera aux besoins des participants. Il sera utile au non informaticien qui aura un jour à programmer rapidement une simulation, mais aussi à toute personne souhaitant comprendre comment sont faits les programmes informatiques.

- Python en ligne de commande (la calculatrice, les variables, les types, …)
- Programmation (scripts, conditions, boucles, fonctions)
- Calcul scientifique en Python (Scipy/Numpy/Pylab)
- Calcul efficace : programmation vectorielle (comme Matlab)

Pour plus de renseignements sur ce cours, consulter : http://www.di.ens.fr/~vergnaud/initPython.html

Programme des cours de l'année 2012/2013

Algèbre 1

(Olivier Biquard)

1) Groupes, action d'un groupe sur un ensemble. Groupe symétrique. Sous-groupes distingués et groupes quotients, produits semi-directs de groupes et extensions de groupes. Groupe des éléments inversibles d'un groupe cyclique, applications arithmétiques.

2) Groupes et géométrie : groupe linéaire, groupe orthogonal, groupes classiques. Formes quadratiques. Formes hermitiennes et formes alternées.

3) Algèbre multilinéaire : produit tensoriel, algèbre tensorielle, algèbre symétrique, algèbre extérieure.

4) Éléments de théorie des représentations des groupes finis, théorie des caractères.

Cours Maths : 12 ECTS
Voir la page de ce cours: http://www.math.ens.fr/enseignement/fiche_cours.html?cours=41#

Algèbre 2

(Olivier Debarre)

Cours Maths : 12 ECTS
Voir la page de ce cours sur : http://www.math.ens.fr/enseignement/fiche_cours.html?cours=46#

Algorithmes arithmétiques pour la cryptologie

(François Morain)

L'objectif du cours est de présenter aux étudiants les concepts et les outils de la cryptologie moderne à clefs publiques, dont les briques de bases mathématiques se trouvent dans les corps finis, et, de plus en plus, dans les courbes algébriques (elliptiques ou hyperelliptiques). Le cours se propose de présenter la théorie algorithmique des nombres, alliance de la théorie des nombres classique et de la théorie de la complexité, avec pour objectifs les applications à la cryptologie. Ce cours se veut comme présentant également les fondements mathématiques du cours 2-12-1. Il forme un tout cohérent avec les autres cours du MPRI traitant de cryptographie, comme par exemple les cours 2-30, 2-13-1 et 2-13-2.

Prérequis spécifiques

Nous attendons que les élèves aient déjà suivis un cours d'introduction à la cryptologie.
Les résultats de base sur les corps finis devront être connus.
Remarque: Aucune connaissance préalable n'est demandée sur les courbes algébriques.

Prérequis généraux

Ces prérequis ne sont pas spécifiques à la cryptologie et sont déjà essentiellement inclus dans la liste générale.

On aura besoin des notions de classes de complexité, de machine de Turing, de problèmes NP. Un minimum de connaissance en algèbre et en probabilité sera aussi requis. Enfin les outils algorithmiques de base doivent être maîtrisés.

Pour plus de renseignements sur ce cours MPRI 2-12-2 (3 ECTS),
consulter sa page : https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-2-12-2

Algorithmique distribuée pour les réseaux

(Pierre Fraigniaud)

L'*algorithmique distribuée* consiste à concevoir et analyser des algorithmes dédiés à un ensemble d'entités autonomes dont l'action conjointe doit contribuer à la réalisation d'une tâche commune.

Le champ d'applications de l'algorithmique distribuée est si vaste qu'il serait vain d'en proposer une liste exhaustive. A eux seuls, le domaine des réseaux (Internet bien sûr, mais aussi les systèmes pair-à-pair, les réseaux sociaux, les réseaux sans fil, les réseaux mobiles, etc.) et celui des multi-processeurs (machine multi-coeurs, grilles de calcul, etc.) fournissent déjà une source immense d'applications potentielles. On peut également citer de nombreux autres cadres d'applications, dont évidemment la biologie avec l'étude de différents systèmes naturellement distribués (nuée d'oiseaux, colonie de fourmis, population de bactéries, etc.).

Ce cours, couplé au cours 2.18-2, a pour objectif de fournir les bases essentielles à la conception, la compréhension, le contrôle, et l'analyse de systèmes tels que ceux listés ci-dessus.

Pour plus de renseignements sur ce cours MPRI 2-18-1 (3 ECTS),
consulter sa page : https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-2-18-1

Algorithmes pour les graphes plongés

(Eric Colin de Verdière, Claire Mathieu)

Le thème du cours est l'étude des algorithmes exacts et approchés pour les graphes plongés, c'est-à-dire les graphes planaires et les graphes dessinés sans croisements sur une surface. Il se situe à la frontière de l'algorithmique "classique" des graphes et de la géométrie algorithmique, et combine plusieurs directions de recherche actuelle qui partagent des techniques communes :

- algorithmes exacts pour les graphes planaires ;
- algorithmes d'approximation pour les graphes planaires ;
- algorithmes pour les graphes plongés sur les surfaces, utilisant des méthodes topologiques.

(cours MPRI : 3 ECTS)
Pour plus de détails sur ce cours (MPRI 2-38-1), merci de consulter :
https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-2-38-1

Algorithmique et programmation

(Jacques Stern, Claire Mathieu, Maribel Fernandez et Damien Vergnaud)

Le cours présente les bases sur les structures de données et les principes de conception des algorithmes ainsi qu'un certain nombre de développements plus avancés. On attend des étudiants un minimum de connaissances algorithmiques. Chaque séance est organisée en deux parties, la première consacrée aux connaissances de base et la seconde à un résultat plus avancé (ou exceptionnellement plusieurs).

Algorithmes : conception et évaluation

- cours de base : terminaison, complexité, stratégies de programmation,
- cours avancé : bin packing, allocation dynamique de mémoire.

Première partie : algorithmique des structures de données

- Tri et hachage
  - cours de base : exemples de tris, hachage, collisions, hachage ouvert,
  - cours avancé : tri Shell.
- Recherche de motifs
  - cours de base : Rabin-Karp, Knuth-Morris-Pratt,
  - cours avancé : algorithmes de bio-informatique.
- Arbres
  - cours de base : arbres de recherche, exemples,
  - cours avancé : tas fusionnables (tas binomiaux, tas de Fibonacci).
- Graphes
  - cours de base : Fermeture transitive, composantes connexes, plus courts chemins,
  - cours avancé : valeurs propres et graphe d'expansion.
- Flots
  - cours de base : Ford-Fulkerson, Edmonds-Karp,
  - cours avancé : Flots unitaires, Dinic, couplages maximaux
- Réductions
  - cours de base : introduction à P, NP, NP-complétude,
  - cours avancé : preuves de NP-complétude par réductions.

Deuxième partie : algorithmique numérique et symbolique

- Entiers
  - cours de base : multiplication, exponentiation,
  - cours avancé : tests de primalité.
- Transformation de Fourier rapide
  - cours de base : FFT, complexité,
  - cours avancé : multiplication rapide.
- Programmation linéaire
  - cours de base : simplexe, complexité,
  - cours avancé : méthode de l'ellipsoïde.
- Algèbre linéaire et géométrie des nombres
  - cours de base : décomposition LUP, moindres carrés,
  - cours avancé : réseaux à coordonnées entières; algorithme LLL.
- Factorisation des Polynômes
  - cours de base : polynômes à coefficients entiers, pgcd, polynômes binaires,
  - cours avancé : algorithme de Berlekamp, Cantor-Zassenhaus.
- Systèmes d'équations polynomiales
  - cours de base : algorithmes de base standard,
  - cours avancé : complexité exp-space.

Page web du cours 2011-2012 : http://www.di.ens.fr/~bouillaguet/teaching.html

(Cours Info : 6 ECTS)

Analyse complexe et harmonique

(Wendelin Werner)

- Introduction: Fonctions harmoniques, fonctions harmoniques conjuguées, fonctions holomorphes discrètes.
- Fonctions holomorphes, formule de Cauchy et ses (nombreuses) applications, fonctions analytiques.
- Transformations conformes: Théorème de Riemann, exemples, métrique de Poincaré.
- Fonctions méromorphes, factorisation de fonctions entières, théorème d'Hadamard et applications.
- Quelques considérations sur la fonction Gamma, sur la fonction Zeta et sur les fonctions elliptiques.

Références:
E.M. Stein, R. Shakarchi, Complex Analysis, Princeton University Press,
L.V. Ahlfors, Complex Analysis, 3rd Ed., McGraw-Hill

(Cours Maths : 12 ECTS)
Voir les mises à jour de ce cours sur :
http://www.math.ens.fr/enseignement/fiche_cours.html?cours=45#

Apprentissage statistique

(Francis Bach, Olivier Catoni)

Cours spécifique aux filières de L3 Info-Maths et Mahs-Info

Ce cours portera sur l'analyse de données de grande dimension, signaux, images, bioinformatique, données économiques, les domaines d'applications dans lesquels d'importants volumes de données sont collectées et demandent à être analysées, classées, corrélées sont nombreux.

L'objectif du cours est de présenter les théories et algorithmes majeurs de l'apprentissage statistique. Les méthodes abordées reposeront en particulier sur des arguments d'analyse convexe et les inégalités de déviations non asymptotiques. Les séances de TDs (dont plus de la moitié seront réalisées sur machines) donneront lieu à des implantations simples des algorithmes vus en cours et à une application à différents domaines comme la bioinformatique ou la vision.

(Cours Maths : 12 ECTS)
Voir les mises à jour de ce cours sur : http://www.fimfa.ens.fr/spip.php?article2

Page de ce cours : http://www.math.ens.fr/cours-apprentissage/

ET :http://www.math.ens.fr/enseignement/fiche_cours.html?cours=62#

Bases de données

(Serge Abiteboul)

1. Introduction: Bases de données et SGBD
2. Modèle relationnel: Algèbre et calcul relationnels, théorème d'équivalence
3. Langages utilisés en pratique, SQL
4. Gestion de fichiers, structures d'accès: Arbres B, hachage
5. Optimisation de requêtes
6. Concurrence et transactions: Sérialisabilité, verrouillage à deux phases, estampillage et pannes
7. Gestion de données distribuées
8. Contraintes d'intégrité

Page web du cours: http://abiteboul.com/2011/DBCOURSE/

(Cours Info : 6 ECTS)
Ce cours a lieu à l'ENS de Cachan.

Bases géométriques de l'Informatique

(Michel Pocchiola et Jean Ponce)

Ce cours introduit les bases géométriques et algorithmiques des domaines de l'informatique où la géométrie joue un rôle fondamental, en particulier la géométrie algorithmique et la vision artificielle. La première partie du cours est consacrée à la géométrie discrète et aux objets, techniques et applications de la géométrie algorithmique. On y développe en particulier l'étude des polyèdres convexes, des arrangements d'hyperplans et des techniques de randomisation. La seconde partie du cours est consacrée à une présentation concrète de notions élémentaires de géométrie projective et de géométrie différentielle et de leur application à la modélisation de systèmes de caméras en vision artificielle.

1. Cônes, polyèdres et polytopes, treillis des faces d'un polyèdre, polytopes cycliques et théorème de la borne supérieure, diagrammes de Voronoi, algorithmes et applications.

2. Arrangements d'hyperplans, niveaux, théorème de la zone, cuttings, algorithmes et applications.

3. Hypergraphes géométriques, théorie des epsilon-nets, algorithmes et applications.

4. Caméras euclidiennes, affines, et projectives : perspective centrale et projection parallèle ; éléments de géométrie affine et projective ; projection et projection inverse de points et de droites.

5. Ensembles de caméras : géométrie épipolaire ; tenseur trifocal ; étalonnage projectif ; mouvement affine ou projectif; étalonnage euclidien : la conique absolue de Chasles et ses cousines.

6. Les surfaces euclidiennes lisses et leurs silhouettes : éléments de géométrie différentielle descriptive ; le théorème de Koenderink; les graphes d'aspects.

Bibliographie :

[1] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry:* Algorithms and Applications. Springer-Verlag, Berlin,Germany, 2nd edition, 2000.

[2] D.A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*.Prentice Hall, 2003.

[3] J. E. Goodman and J. O'Rourke, editors. *Handbook of Discrete and Computational Geometry*. CRC Press, 2nd edition, 2004.

[4] J. Matousek. *Lectures on Discrete Geometry*. Number 212 in Graduate texts in Mathematics. Springer-Verlag, 2002.

[5] K. Mulmuley. *Computational Geometry: An Introduction Through RandomizedAlgorithms*. Prentice Hall, Englewood Cliffs, NJ, 1994.

(Cours Info : 6 ECTS)

Catégories, lambda-calculs

(Paul-André Melliès)

Ce cours s'intéresse à la syntaxe et à la sémantique des langages de programmation, à partir du lambda-calcul. On rappellera les principaux théorèmes syntaxiques du lambda-calcul: confluence, standardisation, résultats de terminaison. Puis on étudiera les modèles du lambda-calcul : pour ce faire, le langage de la théorie des catégories sera utilisé.

Plus généralement, les catégories servent à interpréter bien des extensions du lambda-calcul (avec références, exceptions, etc.), ainsi qu'à comprendre et structurer des notions de concurrence (notamment la notion de bisimulation). Le cours fournit une introduction assez générale et complète au formalisme catégorique, et l'applique à la sémantique des langages de programmation.

Interpréter un langage dans un modèle s'apparente à une compilation, et les modèles offrent ainsi des occasions de retour sur la syntaxe : machines abstraites pour l'exécution des programmes, preuves de propriétés de programmes. Dans le même ordre d'idées, ce sont des observations sur un modèle du lambda-calcul qui ont conduit Girard à la logique linéaire, munie de connecteurs exprimant un contrôle sur l'usage des hypothèses vues comme ressources, ou bien plus récemment Thomas Ehrhard au lambda-calcul différentiel, qui relie de manière originale substitution et... formule de Taylor.

Support de cours :
- Domains and Lambda-calculi. R. Amadio et P.-L. Curien. Cambridge University Press, 1998.
- Categorical semantics of linear logic. P.-A. Melliès. Paru dans la collection Panorama et Synthèse, Société Mathématique de France, 2009.

Et aussi :
- Semantics of programming languages. C. Gunter. MIT Press, 1992.
- Categories, types and structures. A. Asperti and G. Longo. MIT Press, 1991 (épuisé, mais disponible sur la page web de Giuseppe Longo (di.ens.fr).
- Theories of programming languages. J. Reynolds. Cambridge University Press, 1992.

Pour le lambda-calcul :
- The Lambda-calculus. H. Barendregt. North Holland, 1984.
- Lambda-calcul, types et modèles. J.-L. Krivine. Masson, 1990.

Pour les catégories, lire les premiers chapitres d'un livre tel que:
- Toposes, Triples and Theories. M. Barr and C. Wells. Springer, 1985.
- Sheaves in Geometry and Logic: a first introduction to topos theory. S. Mac Lane and Ieke Moerdjik. Springer, 1992.

Pour plus de renseignements sur ce cours MPRI 1-20, consulter les pages :
http://www.pps.jussieu.fr/~mellies/mpri-ens.html

(Cours Info : 6 ECTS)

Complexité avancée

(Jean Goubault-Larrecq)

La théorie de la complexité va bien au-delà de celle de la NP-complétude. Le but de ce cours est d'aller regarder un certain nombre d'autres constructions fondamentales de la théorie de la complexité: complexité en espace, notions de machines alternantes, ou randomisées. On y verra quelques théorèmes fascinants: l'équivalence du temps alternant et de l'espace déterministe par exemple, ou le théorème IP=PSPACE de Shamir.

Description du cours

- la hiérarchie polynomiale, les machines alternantes, la classe PSPACE. QBF est PSPACE-complet.
- classes de complexité alternantes, jeux. Les théorèmes de Chandra-Kozen-Stockmeyer: AL=P, AP=PSPACE. Réductions en espace logarithmique. HORNSAT est P-complet.
- l'accessibilité dans les graphes orientés est NL-complète. Le théorème d'Immerman-Szelepcsényi: la non-accessibilité dans les graphes orientés est aussi NL-complète. Donc NL=coNL.
- classes de complexité randomisées: RP, coRP, BPP, ZPP. Réduction d'erreur. La classe P/poly. Le théorème de Bennett-Gill: BPP ⊆ P/poly. Le théorème de Karp-Lipton; si NP ⊆ BPP alors PH s'effondre au niveau 2. Le théorème de Sipser et Gács: BPP est au niveau 2 de la hiérarchie polynomiale.
- Jeux entre Arthur et Merlin. Les classes MA et AM. Le théorème de Babai: MA ⊆ AM, la hiérarchie Arthur-Merlin s'effondre. BP.NP = AM. Les jeux entre Arthur et Merlin via l'alternance entre quantificateurs E et ∃. Preuves interactives. GRAPH-NON-ISOMORPHISM est dans IP [1].
- techniques de hachage universel, GRAPH-NON-ISOMORPHISM est dans AM (preuve directe), l'erreur peut être ramenée à zéro si x in L pour tout langage L de AM, AM est au niveau 2 de la hiérarchie polynomiale. Théorème de Goldwasser-Sipser: IP [k] ⊆ AM [k+1]. Théorème de Boppana-Håstad-Zachos: si coNP ⊆ AM alors PH s'effondre au niveau 2. Conséquence pour GRAPH-NON-ISOMORPHISM.
- Classes à nombre de tour polynomial: ABPP, IP. Théorème de Shamir: ABPP=IP=PSPACE.
- (s'il y a le temps) problèmes d'approximation. Les seuils d'approximation de NODE COVER, TSP, KNAPSACK, MAXSAT. Le théorème d'Arora-Safra: NP=PCP(O (logn), O (1)) (sans démonstration). Equivalence du théorème d'Arora-Safra et de l'inapproximabilité de MAX3SAT.

Pré-requis

On s'attend à ce que les étudiants aient une certaine familiarité avec la notion de Machine de Turing, la classe P (temps polynomial déterministe), la classe NP (temps polynomial non-déterministe), les notions de réductions en temps polynomial, le théorème de Cook (SAT est NP-complet) même si toutes ses notions seront rapidement revues au début du cours.

(Cours Info : 6 ECTS)

Pour plus de renseignements sur ce cours MPRI 1-17 (6 ECTS),
consulter sa page sur le site du MPRI :
https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-1-17

Fondements sur la modélisation des réseaux

(François Baccelli, Jean Mairesse)

Le but de ce cours est double :

* proposer des modèles mathématiques pertinents pour les réseaux de communications;
* donner les bases théoriques permettant de mener a bien l'analyse de la dynamique de ces modèles.

Le cours est structuré en thèmes, pouvant être plus ou moins développés suivant les années :

* Réseaux de files d'attente et modélisation markovienne (réseaux à commutation de paquets, réseaux à commutation de circuits).
* Dynamique des systèmes à événements discrets temporisés (semi-anneau max plus, inf convolutions, fonctions topicales, réseaux de Petri temporisés, modèles d'empilements de pièces, etc.).
* Contrôle de flux dans les réseaux de communication (TCP, contrôle de flux et de congestion, régulation, network calculus, ordonnancement etc.).
* Graphes aléatoires (à la Erdos-Renyi, géométriques) et modèles de percolation.

Pré-requis

Une familiarité avec les probabilités discrètes et les chaînes de Markov, est préférable.

Pour plus de renseignements sur ce cours MPRI 2-17-1 (3 ECTS),
consulter sa page : https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-2-17-1

Génie logiciel et cloud computing

(Joannès Vermorel)

Ce cours présente les concepts fondamentaux du génie logiciel, avec un intérêt pour les systèmes complexes / distribués, notamment dans le cadre du "cloud computing".
Le cours est associé à un projet de développement logiciel. Chaque séance inclut un cours magistral suivi d'un bilan collectif sur l'avancement du projet.

Pré-requis: Ce cours ne forme pas à la programmation. On attend des élèves qu'ils soient déjà familiers avec un ou plusieurs langages de programmation.
Sans être indispensable, la participation préalable aux cours "Algorithmique et programmation" et "Système digital: de l'algorithme au circuit" lors du premier semestre est un plus.
Le génie logiciel est l'étude de l'activité de production de logiciels en tant qu'activité économique, où les ressources matérielles/humaines (ainsi que les délais) sont limitées.
Les avancées de la dernière décennie dans ce domaine ont permis des gains de productivité très importants.

On s'attachera à comprendre comment des pratiques associées à des avancées technologiques influencent (en bien ou en mal) la productivité dans le domaine logiciel.

Les systèmes informatiques distribués, notamment le "cloud computing", seront intégrés au cours comme objet d'étude, mais aussi comme projet en équipe par les élèves.
La motivation de ce choix est double: l'évolution du matériel informatique tend aujourd'hui vers le "tout-distribué"; par ailleurs les systèmes distribués sont redoutablement difficiles à développer et à debugger.

Notes de cours : http://www.vermorel.com/softeng.html

Bibliographie:

- AntiPatterns by William J. Brown, Raphael C. Malveau, Hays W. "Skip" McCormick, Thomas J. Mowbray
- Joel on Software: And on Diverse and Occasionally Related Matters That Will Prove of Interest to Software Developers, Designers, and Managers, and to Those Who, Whether by Good Fortune or Ill Luck, Work with Them in Some Capacity by Joel Spolsky
- Design Patterns: Elements of Reusable Object-Oriented Software by Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides

(Cours Info : 6 ECTS)

L'Informatique scientifique par la pratique

(David Naccache)

Ce cours aborde les disciplines scientifiques liées au traitement automatique de l'information à travers la microprogrammation et la mise en oeuvre optimisée.
L'objectif pédagogique du cours est double :

1. Initier les étudiants aux technologies de conception proches de la machine: matériel et microprogramme.
   Initiation puis maîtrise de nouveaux outils et langages assembleur.
2. Ce faisant familiariser les étudiants avec certains algorithmes communs intervenant en informatique scientifique : compression, ramasse-miettes, hachage géométrique, correction d'erreurs, arithmétique entière, arithmétique en virgule flottante, solution par retraits itérés (backtracking), inférence de type, FFT et déconvolution. Elargissement, par la pratique, de la "culture informatique scientifique" de l'étudiant.

Les étudiants affronteront les contraintes rencontrées lors de programmation de ces algorithmes (e.g. taille de code, complexités mémoire et temps etc) ainsi que les différentes techniques permettant d'adapter les algorithmes à des architectures (e.g. mise en uvre en bit-slice, calcul partagé, évaluation paresseuse etc). A titre de projet les étudiants pourraient mettre en uvre de manière optimisée des algorithmes vus lors ou du cours ou lors d'autres cours dispensés dans le cadre de la formation interuniversitaire en informatique de l'ENS (traitement des images, optimisation de circuits électroniques, cryptographie, compilation etc).

1. Le microprocesseur 68HC05
   - Présentation des outils de programmation en assembleur 68HC05. Architecture, Ports, registres, mémoires, ALU.
   - TD : conception d'une librairie de calcul sur les nombres flottants en assembleur. Changement de la saturation des points d'une photographie du portique de l'ENS en utilisant une multiplication à virgule flottante arrondie.

2. La compression
   - Introduction : Théorie de Shannon. Compression entropique. Compression par les méthodes RLE, LZW et Huffman.
   - TD : mise en uvre de RLE, LZW et Huffman sur le 68HC05. Compression de la photographie du portique.

3. La correction d'erreurs
   - Introduction : Contrôle des erreurs par codage algébrique. Codes de Viterbi. Turbo codes.

- TD : codeur / décodeur pour les codes de Hamming et de Reed-Solomon sur le 68HC05. Protection par code correcteur de l'image du portique. Bruitage de l'image à différents SNRs, affichage et correction.

4. Arithmétique sur les grands nombres : multiplication et réduction modulaire
   - Introduction : Multiplication multi-précision et algorithmes de réduction de Montgomery et de Barrett. Preuve des deux algorithmes et leur analyse. Techniques de multiplication rapide (Solovay-Strassen) et multiplication à faible nombre de changements d'état.
   - TD : Multiplication multi-précision et réduction modulaire Montgomery et Barrett sur le 68HC05. Codage de fonctions de calcul RNS (Residue Number System).

5. La cryptographie par la pratique.
   - Introduction : Signature et chiffrement, présentation et/ou rappel de RSA.
   - TD codage d'un chiffrement RSA et d'une vérification de signature à l'aide de la librairie de calcul sur les grands nombres codée lors du TD précédent. Vérification d'une signature numérique créée sur Mathématica sur l'image du portique.

6. Ramasse-miettes par la pratique.
   - Introduction au ramassage de miettes (garbage collection)
   - TD : Ramasse-miettes par la méthode mark & sweep. Découpage de l'image du portique en morceaux et assemblage des morceaux en un chemin de moindres mouvements à l'aide du programme de mark & sweep.

7. La machine de Minsky
   - Architecture.
   - TD : codage d'un simulateur de machine de Minsky et mise en uvre d'un programme très simple sur ce simulateur.

8. La technique des retraits itérés
   - Présentation de la technique des retraits itérés.
   - TD : Résolution du problème de lasermaze par la technique des retraits itérés. Détection du plus long chemin de reflets dans un extrait de l'image du portique par la méthode des retraits itérés.

9. La transformée de Hough et le hachage géométrique.
   - Le problème de la détection de droites, présentation des algorithmes.
   - TD : Codage en assembleur de la transformée de Hough. Extraction d'une droite de l'image du portique.

10. FFT et déconvolution. Régularisation de Tikhonov
   - TD : Codage en assembleur d'une routine de convolution et de déconvolution. Convolution de l'image du portique avec un filtre gaussien et déconvolution. Essais avec différents SNRs.

Séance finale :
   - Exposition des résultats de tous les TDs du module (avec des posters et démonstrations) à l'intention des élèves et chercheurs internes et externes.

(Cours Info : 6 ECTS)

Initiation à la cryptologie

(Jacques Stern, Damien Vergnaud)

Ce cours sert à la fois d'initiation à la cryptologie et de préparation au cours de niveau 2. Il s'adresse aux étudiants ayant un goût pour l'algorithmique, à la fois dans ses aspects mathématiques et dans ses aspects pratiques. Le but de ce cours est d'enseigner la problématique de la cryptologie, et les principaux outils utilisés par la cryptologie pour proposer des solutions aux problèmes de sécurité.

Le cours est découpé en six parties, de 4 heures chacune. Ce sont :

- Introduction à la cryptographie
  - Permutations, substitutions, cryptanalyse (types d'attaques).
  - Intégrité, confidentialité, authenticité. One Time Pad.
- Cryptographie symétrique
  - Chiffrement par flot.
  - Chiffrement par bloc.
  - Modes d'opération (CBC, ECB, CTR).
  - Exemples: DES, AES, RC4, A5/1.
  - Hachage, MAC.
- Compléments d'algorithmique
  - Algorithmique des entiers.
  - Arithmétique modulaire.
  - Corps finis.
- Cryptographie asymétrique
  - RSA, Diffie-Hellman, El Gamal.
  - Multiplicativité de RSA (Hastad, attaques multiplicatives).
  - One-way Functions, trappes.
  - Générateurs pseudo aléatoires.
  - Signatures RSA, El Gamal.
- Protocoles
  - Introduction aux preuves à divulgation nulle de connaissance (ZK).
  - Identification, signatures (FS, Schnorr).
- Applications
  - PKI, IPSEC.
  - Canal sécurisé : SSL.

Pré-requis : On aura besoin des notions de classes de complexité, de machine de Turing, de problèmes NP. Un minimum de connaissance en algèbre et en probabilité sera aussi requis. Enfin les outils algorithmiques de base doivent être maîtrisés. Les élèves doivent aussi connaître le langage C ou Python pour certains TDs.

(Cours Info : 6 ECTS)
Cours MPRI 1-13
https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-1-13

Initiation à la modélisation et à la simulation numérique

(Erwan Faou - David Lannes)

Ce cours a pour objet de présenter certaines équations différentielles issues de la physique et d'expliquer comment l'analyse mathématique guide la mise en oeuvre de méthodes numériques permettant la résolution de ces équations à l'aide de calculs par ordinateurs.

Prérequis : Bases de calcul différentiel, intégration, analyse fonctionnelle et curiosité scientifique.

Programme :

Rappels sur les Equations Différentielles Ordinaires et leur approximation numérique (schémas d'Euler, méthodes de splitting, schémas d'ordres élevés, notions de stabilité et consistance, introduction au temps long).

Equations Hamiltoniennes: schémas symplectiques, préservation numérique de l'énergie. Introduction aux systèmes hautement oscillants et aux systèmes de dimension infinie.

Méthodes spectrales et pseudo-spectrales. Rôle pour le calcul de solutions d'équations pseudo-différentielles non locales (méthode de la FFT). Malédiction de la dimension (curse of dimensionality) et sparse grids.

Equations non linéaires dispersives (exemples de KdV et NLS). Structures Hamiltonienne. Ondes solitaires. Analyse de schémas de splitting.

Phénomènes de diffusion, méthodes de différences finies (analyse de stabilité, schémas explicite et implicite, problèmes de convection-diffusion...). Comparaison avec les méthodes probabilistes (si le temps le permet).

Systèmes hyperboliques avec exemple des équations de Saint-Venant (volumes finis, schémas équilibrés et méthode de méthode de reconstruction hydrostatique).

(Cours Maths : 12 ECTS)
Voir les mises à jour de ce cours sur :
http://www.math.ens.fr/enseignement/fiche_cours.html?cours=58

Initiation à la programmation pour non-informaticiens (deuxième semestre)

(Damien Vergnaud)

Ce cours est ouvert aux élèves de toutes les disciplines, littéraires comme scientifiques. Aucune connaissance préalable en programmation n'est requise. Le cours n'est pas orienté à priori vers une application particulière. Il s'adaptera aux besoins des élèves. Il sera utile au non informaticien qui aura un jour à programmer rapidement une simulation, mais aussi à toute personne souhaitant comprendre comment sont faits les programmes informatiques.

- Python en ligne de commande (la calculatrice, les variables, les types, …)
- Programmation (scripts, conditions, boucles, fonctions)
- Calcul scientifique en Python (Scipy/Numpy/Pylab)
- Calcul efficace : programmation vectorielle (comme Matlab)

Pour plus de renseignements sur ce cours, consulter : http://www.di.ens.fr/~vergnaud/initPython.html
(Cours Info : 3 ECTS)

Intégration et probabilités

(Zhan SHI)

1. Intégration
   - Espaces Mesurés
   - Intégration par rapport à une mesure
   - Construction de mesures
   - Espaces L^p
   - Mesures produits
   - Mesures signées
   - Formule de changement de variables

2. Probabilités
   - Fondements de la théorie des probabilités
   - Indépendance
   - Convergence de variables aléatoires

Voir les mises à jour de ce cours sur :
http://www.math.ens.fr/enseignement/fiche_cours.html?cours=42
(Cours Maths : 12 ECTS)

Interprétation Abstraite : Application à la Vérification et à l'Analyse Statique

(Patrick Cousot, Radhia Cousot)

L'analyse statique de programmes consiste à vérifier statiquement (sans les exécuter) des propriétés dynamiques (à l'exécution) des programmes.

Les classes de propriétés à vérifier sont très diverses comme la sûreté (par exemple, absence d'erreurs à l'exécution), la vivacité (par exemple, garantie de réponse à un signal), la sécurité (par exemple, confidentialité d'informations traitées par un programme), etc.

La grande difficulté pour démontrer automatiquement ces propriétés dynamiques est de trouver les arguments inductifs pour faire la preuve (par exemple, par induction sur le nombre de pas de calcul). Diverses solutions sont possibles : demander à l'utilisateur (méthodes déductives), utiliser un modèle finitaire (vérification exhaustive) ou calculer l'argument inductif par approximation de la sémantique du programme (en utilisant les techniques d'approximation de point fixe de l'interprétation abstraite).

Le cours explore cette dernière technique, en rappelle rapidement les bases, afin d'explorer un certain nombre d'abstractions infinitaires qui permettent de traiter un grand nombre d'applications à systèmes d'états infinis, qu'elles soient émergentes, classiques ou industrialisées

Plan du cours (à titre indicatif) :

- Introduction à l'interprétation abstraite ;
- Domaines abstraits numériques ;
- Domaines abstraits symboliques ;
- Combinaison et raffinement de domaines abstraits ;
- Conception d'un analyseur statique par interprétation abstraite ;
- Analyse statique de programmes séquentiels, procéduraux, récursifs, modularité ;
- Domaines abstraits probabilistes ;
- Analyse statique de programmes parallèles asynchrones ;
- Analyse statique de programmes distribués ;
- Analyse statique de code mobile ;

- Vérification par abstraction paramétrée de prédicats ;
- Vérifications statiques sur des logiciels critiques embarqués temps-réel ;
- Sujets pratiques et théoriques ouverts en analyse statique par interprétation abstraite, perspectives.

Pour plus de renseignements sur ce cours MPRI 2-6 (6 ECTS), sur sa page sur le site du MPRI : https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-2-6

Introduction à la vision artificielle

(Jean Ponce)

Ce cours présente les principes et les fondations techniques de la vision artificielle, un domaine scientifique dont le but est de doter les ordinateurs de la capacité d'interpréter le contenu des images numériques (photographies et vidéos).
Le cours comprend des exercices de programmation en Matlab/Scilab.

Plan :
1. Formation des images : Modèles des appareils de prise de vue, de la lumière et de la couleur.
2. Traitement d'image local : Filtres, détection de contours, caractéristiques visuelles, texture.
3. Groupes de pixels : Méthodes de "clustering", régression, et segmentation.
4. Plusieurs images : Géométrie multi images, stéréo, analyse du mouvement.
5. Analyse de scène : Détection et reconnaissance de visages, sacs de caractéristiques visuelles pour la reconnaissance de catégories d'objets.

Bibliographie :
D.A. Forsyth et J. Ponce, "Computer Vision: A Modern Approach", Prentice-Hall, 2002.
(Cours Info : 4 ECTS)

Langages de programmation et compilation

(Jean-Christophe Filliâtre)

Ce cours présente les principaux concepts des langages de programmation au travers de l'étude de leur compilation, c'est-à-dire de leur traduction vers le langage machine. Les TDs ont pour objectif de programmer certaines des notions vues en cours. L'évaluation comprend un projet consistant en la réalisation d'un petit compilateur.

- Mise à niveau Caml
- Principes de la compilation / Architecture MIPS
- Syntaxe abstraite / Sémantique / Interprète
- Analyse sémantique
  - Typage monomorphe
  - Polymorphisme / algorithme W
  - Propriétés d'un système de types
- Analyse lexicale et syntaxique
- Compilation des langages impératifs
  - Tableaux d'activation
  - Mode de passage des paramètres
- Compilation des langages fonctionnels
  - Fonctions comme valeurs de première classe
  - Compilation du filtrage

- Compilation des langages à objets
  - Typage statique et typage dynamique
  - Représentation des objets et exécution
- Glaneurs de cellules (GC)
- Production de code efficace
  - Langages intermédiaires RTL, ERTL, LTL
  - Allocation de registres

(Cours Info : 6 ECTS)
Pour plus de renseignements sur ce cours, consulter : http://www.lri.fr/~filliatr/ens/compil/

Langages formels, calculabilité et complexité

(Eugene Asarin)

1: Langages réguliers, leurs propriétés et leur caractérisation par automates, expressions régulières, formules logiques, monoïdes. Langages sans étoile.

Premières notions sur les langages de mots infinis.

2: Grammaires et hiérarchie de Chomski. Langages hors contexte, leurs propriétés, leur caractérisation par automates à pile.

3: Calculabilité (fonctions récursives et Machines de Turing). Problèmes décidables, indécidables, semi-décidables.

4: Complexité en temps et espace. Bornes de complexité. Classes de complexité (NP, Pspace) et problèmes complets.

Page du cours 2011-2012 de M Asarin sur : http://www.liafa.jussieu.fr/~asarin/ENS/lf.html

Livre support du cours : http://www.liafa.jussieu.fr/~carton/Lfcc/

Page du TD de Anne Bouillard : http://www.di.ens.fr/~bouillar/enseignement.html

Ancienne page du cours : http://www.liafa.jussieu.fr/~carton/Enseignement/Complexite/ENS/

(Cours Info : 6 ECTS)

Logique

(Martin Hils)

1. Théorie naïve des ensembles
- Théorème de Cantor-Bernstein
- Ordinaux et cardinaux
- Les différentes formes de l'axiome du choix

2. Théorie des modèles
- Langages, structures, formules, théories, modèles
- Théorèmes de complétude et de compacité
- Théorèmes de Löwenheim-Skolem
- Critères d'élimination des quantificateurs
- Une application à l'algèbre : Théorème d'Ax sur les fonctions polynomiales injectives.

3. Récursivité, indécidabilité, incomplétude
- Fonctions récursives
- Arithmétique de Peano, indécidabilité de l'arithmétique
- Théorèmes d'incomplétude de Gödel.

4. Retour à la théorie des ensembles
- Les axiomes de Zermelo-Frankel
- Modèles de la théorie des ensembles et hypothèse du continu

Voir les mises à jour de ce cours sur :
http://www.math.ens.fr/enseignement/fiche_cours.html?cours=44
(Cours Maths : 12 ECTS)

Logique et informatique

(Jean Goubault-Larrecq)

Ce cours explore les bases du lambda-calcul, un outil inventé par le logicien Alonzo Church dans les années 1930 et qui est aujourd'hui fondamental tant en sémantique des langages de programmation (informatique) qu'en théorie de la preuve (logique).

1. Aspects informatiques:
- Lambda-calcul, langages fonctionnels, sémantique opérationnelle (réduction).
- Expressivité. Combinateurs de point fixe et récursion.
- Terminaison, développements finis, confluence et réductions parallèles.
- Stratégies de réduction: par nom, par nécessité. Standardisation.
- Modèles du lambda-calcul, P-omega.
- Calculs à substitutions explicites, machines. Géométrie de l'interaction

2. Aspects logiques :
- Lambda-calcul simplement typé;
- Correspondance de Curry-Howard entre ce dernier et les preuves en logique minimale propositionnelle;
- Extension à la logique classique, captures de continuations et gestion d'exceptions;
- Arithmétique, le système T;
- Lambda-calcul typé du second ordre : le système F de Girard-Reynolds, correspondance avec la logique intuitionniste d'ordre deux;

- Propriétés de normalisation forte, élimination des détours.

Bibliographie :

- Henk Barendregt. The lambda-calculus, its syntax and semantics. North-Holland, 1984.
- Jean-Louis Krivine. Lambda-calcul, types et modèles. Masson, 1992.
- Jean-Yves Girard, Yves Lafont & Paul Taylor. Proofs and Types. Cambridge University Press 1989.

(Cours Info : 6 ECTS)
Notes de cours :
http://www.lsv.ens-cachan.fr/~goubault/Lambda/loginfoindex.html
Ce cours fortement recommandé a lieu à l'ENS de Cachan

Méthodes mathématiques pour les neurosciences

(Olivier Faugeras)

Nous présentons dans ce cours quelques problèmes importants de modélisation en neurosciences. Ces problèmes nécessitent pour les aborder des outils mathématiques issus de l'analyse fonctionnelle, de la théorie des systèmes dynamiques et du calcul stochastique. Les prérequis sont une bonne connaissance du calcul différentiel et du calcul des probabilités dans le cadre de la théorie de la mesure. Sans trahir la rigueur mathématique, le cours s'efforcera de mettre en valeur l'applicabilité aux neurosciences des concepts présentés.

- Modèles mésoscopiques de certaines structures corticales
  - Structure anatomique du cortex visuel (aire V1)
  - Architecture fonctionnelle de V1
  - Modèles de champs neuronaux
- Introduction à la théorie des bifurcations
  - Dimension 1 (noeud-selle, transcritique, fourche)
  - Dimension 2 (Hopf)
  - Variété centrale
  - Formes normales
  - Bifurcations équivariantes
- Introduction aux systèmes dynamiques
  - Orbites et portraits de phase
  - Variétés invariantes
  - Equivalence de systèmes dynamiques
  - Classification topologique des équilibres
  - Stabilité structurelle
  - Variété centrale en dimension finie
- Applications : sensibilité à l'orientation des contours visuels, formation de structures corticales et hallucinations visuelles
  - Le ``ring model'' de perception des orientations des contours visuels
  - Mécanisme de Turing pour la formation des structures corticales
  - Modèle de Bressloff-Cowan-Golubitsky pour les hallucinations visuelles
- Modèles de neurones
  - Le modèle de Hodgkin-Huxley sans espace
  - Modèles simplifiés
  - Modèles de synapses
  - Modèles spatiaux
- Le rôle du bruit
  - Mouvement Brownien

- Équations différentielles stochastiques
- Application aux neurones

– Modèles de champ moyen
- La théorie de Sompolinsky-Ben Arous-Guionnet des verres de spin
- Application aux modèles de neurones à taux de décharge
- La théorie de McKean-Tanaka-Sznitman de particules en interaction
- Application aux modèles de neurones à potentiels d'action
- Application aux masses neurales

La page web du cours se trouve(ra) à l'adresse :
http://www-sop.inria.fr/members/Olivier.Faugeras/MVA/MMN11
L'ancienne se trouve à l'adresse :
http://www-sop.inria.fr/members/Olivier.Faugeras/MVA/MMN10

Bibliographie:

– Wulfram Gerstner et W. Kistler, Spiking neuron models, Cambridge University Press, 2002.
– Yuri A. Kuznetsov, Elements of applied bifurcation theory.
– Eugene Izhikevich, Dynamical systems in neuroscience: the geometry of excitability and bursting, MIT Press, 2006.
– Jean-Pierre Françoise, Oscillations en biologie, Springer, 2000.
– Lawrence C. Evans, An introduction to stochastic differential equations, http://math.berkeley.edu/~evans/SDE.course.pdf
– Jean-François Le Gall, Mouvement brownien et calcul stochastique, http://www.dma.ens.fr/~legall/DEA96.pdf
– G. Bard Ermentrout et D. H. Terman, Mathematical Foundations of Neuroscience
– Sylvie Benzoni, Cours de M1 sur les EDOs, http://math.univ-lyon1.fr/~benzoni/

(Cours Info, MVA : 4 ECTS et Cours Master Mathématiques et Applications UPMC)

Planification de mouvement en robotique et en animation graphique : du continu au combinatoire via la commandabilité des sytèmes

(Jean-Paul Laumond)

La planification de mouvement s'intéresse au calcul automatique de chemins sans collision pour un système mécanique (robot mobile, bras manipulateur, personnage animé...) évoluant dans un environnement encombré d'obstacles. Les méthodes consistent à explorer l'espace des configurations du système : une configuration regroupe l'ensemble des paramètres permettant de localiser le système dans son environnement. Aux obstacles de l'environnement correspondent des domaines à éviter dans l'espace des configurations. La planification de mouvement pour le système mécanique se trouve ainsi ramenée au problème de la planification de mouvement d'un point dans une variété non simplement connexe.

– Introduction : le mouvement en Robotique, en CAO et en Animation Graphique.

– Modélisation du problème de la planification de mouvement
  - L'espace des configurations par l'exemple : trois méthodes de résolution du problème de déplacement d'un polygone en translation.
  - L'apport de la géométrie algébrique réelle : le problème est décidable.
  - L'apport de la topologie algébrique pour le mouvement au contact.

- Les grandes méthodes de résolution
  - Décomposition cellulaire. Exemple du mouvement coordonné de deux disques.
  - Rétraction. Exemple du problème de la manipulation d'objets.
- Les systèmes non holonomes
  - Eléments de géométrie différentielle. Commandabilité. Degré de non holonomie.
  - Chemins optimaux pour robots mobiles de type voiture. La planification de mouvement par algorithme d'approximation de chemins holonomes. Complexité.
  - Systèmes chaînés et commandes sinusoïdales. Systèmes plats et approche géométrique.
  - Un cas d'étude complet : planification et exécution de mouvements pour un chariot à remorque.
- Les nouvelles méthodes par recherche aléatoire.

Bibliographie :

- J.T. Schwarzt, M. Sharir and J. Hopcroft (Eds), Planning, Geometry and Complexity of Robot Motion, Ablex Series in Artificial Intelligence, Ablex Publishing, 1987.
- J.C. Latombe, Robot Motion Planning Kluwer Academic Publishers, 1991.
- J.P. Laumond (Ed), Robot Motion Planning and Control, Lectures Notes in Control and Information Science, 229, Springer Verlag, 1998 (épuisé mais disponible gratuitement sur http://www.laas.fr/~jpl).

(Cours Info et MPRI 1-19 : 6 ECTS)

Processus Aléatoires

(Josselin Garnier)

(Cours Maths : 12 ECTS)
Voir descriptif de ce cours sur : http://www.math.ens.fr/enseignement/fiche_cours.html?cours=56

Protocoles cryptographiques: preuves formelles et calculatoires

(Hubert Comon-Lundh & David Pointcheval)

Les protocoles cryptographiques sont des programmes distribués qui visent à sécuriser des communications et transactions en utilisant des primitives cryptographiques. La conception des protocoles cryptographiques est difficile: de nombreuses erreurs ont été découvertes dans des protocoles après leur publication. Il est donc particulièrement important de pouvoir obtenir des preuves que ces protocoles sont sûrs.

Deux modèles des protocoles ont été considérés: le modèle symbolique et le modèle calculatoire. Nous présenterons ces deux modèles, les techniques de preuves associées, et des résultats qui font le lien entre eux. Nous considérerons aussi leur mise en oeuvre, en montrant un outil de preuve automatique pour chaque modèle, et en les appliquant à la vérification de programmes qui implémentent des protocoles cryptographiques.

Ce cours sera l'occasion d'adapter et d'utiliser des outils formels, comme les calculs de processus, la sémantique, le typage et la logique, au cas particulier de l'étude des protocoles cryptographiques.

Bibliographie et plus de renseignements sur ce cours MPRI 2-30(6 ECTS),sur sa page sur le site du MPRI : https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-2-30

Reconnaissance d'objets et vision artificielle

(Ivan Laptev, Jean Ponce, Josef Sivic, Cordelia Schmid)

La reconnaissance automatique des objets --et de manière plus générale, l'interprétation de la scène-- figurant dans une photographie ou une vidéo est le plus grand défi de la vision artificielle. Ce cours présente les modèles d'images, d'objets, et de scènes, ainsi que les méthodes et algorithmes utilisés aujourd'hui pour affronter ce défi.

Plan du cours :

- Caractéristiques visuelles : points d'intérêt, régions affines, invariants, descripteurs Sift.

- Détection d'objets et de classes spécifiques : alignement 2D et 3D, méthodes de votes, détection de visages et Adaboost.

- Classification d'images : sacs de caractéristiques visuelles et machines à vecteurs de support, grilles et pyramides, réseaux convolutionnels.

- Détection de catégories d'objets : constellations de caractéristiques visuelles, assemblages de fragments, méthodes de fenêtre glissantes, apprentissage faiblement supervisé de modèles.

- Aller plus loin : analyse de scène, analyse des activités dans les vidéos.

Bibliographie :

- D.A. Forsyth and J. Ponce, ``Computer Vision: A Modern Approach",Prentice-Hall, 2003.
- J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, ``Toward Category-Level Object Recognition", Lecture Notes in Computer Science 4170, Springer-Verlag, 2007.

(Cours Info et MVA : 4 ECTS)

Réseaux de Communication

(Ana Busic)

Ce cours est une introduction aux réseaux de communication qui consiste en :

1. Un ensemble de lectures ;
2. Un cours sur les bases de la simulation à événements discrets ;
3. La réalisation d'un simulateur à événements discrets et l'étude d'un réseau sur la base de ce simulateur.

I. Lectures. Les lectures sont centrées sur les thèmes suivants:

• Accès multiple dans les réseaux locaux (Livre: Multiple Access Protocols, R. Rom et M. Sidi, Springer, 1990).

– Réseaux fixes: Aloha, Ethernet, Protocoles en arbres, TDMA, CSMA ;

– Réseaux sans fils: 802.11, Aloha spatial ;

– Stabilité et instabilité ; Maximisation des débits.

• Contrôle de congestion (Livre: An Engineering Approach to Computer Networking, Addison-Wesley, 1997.)

– TCP et ses variantes (Reno, Tahoe, Vegas) ;

– Représentation par des problèmes d'optimisation ; allocations max-min fair, proportional fair.

• Routage dans les réseaux IP (Livre: An Engineering Approach to Computer Networking, Addison-Wesley, 1997.)

– Routage IP et algorithme de Dijkstra ;

– BGP ;

– Routage dans les réseaux ad hoc.

II. Cours d'introduction à la simulation des réseaux. Le cours abordera les questions suivantes

• Génération de variables aléatoires ;

• Schémas de Matthes, tables d'événements et simulation à événements discrets ;

• Intervalles de confiance ;

• Simulation parfaite.

III. Projets. Une liste de projets de simulation sera proposée. Voici quelques exemples: instabilité des réseaux multiclasse ; simulation parfaite de réseaux CSMA ; interaction de flots TCP ; routage opportuniste dans les réseaux ad hoc ; diffusion épidémique dans les réseaux pair à pair ; réseaux mobiles aléatoires.

Prérequis. Avoir suivi un premier cours de probabilités.

(Cours Info : 6 ECTS)

Statistique

(Gérard Biau)

Objectifs : Ce cours vise à donner aux étudiants les bases fondamentales du raisonnement et de la modélisation statistique. L'accent est particulièrement mis dans cet enseignement sur l'utilisation pratique des nouveaux objets rencontrés.

Prérequis : Une bonne connaissance du calcul des probabilités et de l'algèbre linéaire.

Thèmes abordés :
- Rappels de probabilités, estimation ponctuelle, estimation par intervalles, tests.
- Estimateurs par maximum de vraisemblance.
- Modèle linéaire : estimation, intervalles de confiance et tests.
- Modèles exponentiels, exhaustivité.

(Cours Maths : 12 ECTS)

Voir les mises à jour de ce cours sur :
http://www.math.ens.fr/enseignement/fiche_cours.html?cours=57

Structures et Algorithmes Aléatoires

(Anne Bouillard, Pierre Brémaud)

Objectif : Ce cours vise à donner aux étudiants les bases de probabilités qui sont utilisées dans divers domaines de l'informatique (algorithmique, algorithmes stochastiques, réseaux de communication,...)

Plan : ce cours est divisé en deux parties :

Probabilités discrètes et applications
- Variables aléatoires, indépendance, conditionnement
- Méthode probabiliste
- Graphes aléatoires

Modèles markoviens
- Chaînes de Markov, comportement asymptotique
- Simulation Monte Carlo et simulation parfaite
- Champs de Gibbs

Pour chaque thème abordé, des exemples d'application dans divers domaines de l'informatique seront présentés.
Intervenants :
Cours : Anne Bouillard et Pierre Brémaud
TD : Anne Bouillard
Pour plus de renseignements sur ce cours, consulter : http://www.di.ens.fr/~bouillar/SAA/index.html
(maj descriptif : juillet 2012)
(Cours Info : 6 ECTS)

Système digital : de l'algorithme au circuit

(Jean Vuillemin)

Le ***cours théorique*** présente la composante ***matérielle*** du monde informatique. Des principes de conception et de réalisation des ***circuits***, à diverses applications du calcul numérique haute performance : en physique, électronique, algèbre et télécommunication. Chaque application va de l'algorithme (logiciel) au circuit (matériel) : mêmes opérations, autres performances.
La ***partie pratique*** du cours est un projet, à réaliser par groupes : chaque groupe doit entièrement concevoir un ***microprocesseur***, et le réaliser au moyen de portes logiques élémentaires ; il faut ensuite simuler les portes en fonctionnement, et programmer le microprocesseur pour en faire une *montre numérique*, simulée en temps-réel.

1. ***Circuit digital synchrone*** : Circuit combinatoire et portes logiques. Registre et circuit digital synchrone. Réalisation d'un circuit de *montre numérique*. Complexité et synthèse BDD de circuits.
2. ***Nombres binaires*** : des bits aux entiers 2-adiques ; algèbre et anneau de Boole ; hyper-cube et ensembles d'entiers. Arithmétique 2-adique et circuits en séries. Opérations logiques et ensemblistes sur les entiers : algèbre binaire.
3. ***Circuits électroniques*** : des portes aux transistors ; de la logique à son dessin sur silicium. Schémas électriques et dessin au micron d'un additionneur série. Mémoires ROM et RAM. Technologies de fabrication : ASIC, FPGA. Lois de Moore.
4. ***Arithmétique sur silicium*** : additionneurs et multiplicateurs, en série et en parallèle ; profondeur minimale. Compromis optimaux entre surface et temps. Unité arithmétique et logique. Division par les poids faibles ; racine 2-adique.
5. ***Machines universelles*** : machine de Turing sur silicium. Microprocesseur programmable à la Church. Nombre réel calculable, et limites du calcul automatique. Arithmétiques en ligne :

réels vs. 2-adiques. Logique programmable FPGA et systèmes dynamiquement reconfigurables.

6. ***Physique numérique*** : algorithme (transformée de Hough rapide) et réalisation d'un circuit d'identification de lignes droites dans des images digitales haute fréquences du détecteur ATLAS du LHC ; principe et réalisation d'un circuit pour connaitre les flux thermiques d'un microprocesseur en marche, par résolution numérique massivement parallèle de l'*équation de la chaleur*.
7. ***Télécommunications*** : introduction à la théorie de Shannon, source et canal ; entropie des données, algorithme de Huffman, compression LZW. Contrôle des erreurs ; entropie du bruit ; code de Hamming ; code de Viterbi.
8. ***Audio et vidéo*** : convertisseurs A/D et D/A ; compromis vitesse/résolution. Saisie, codage et transmission des images ; compression sans perte visible à l'œil : images fixes JPEG et séquences vidéo MPEG. Codage MP3 et transmission du son.

Pour plus de renseignements sur ce cours, consulter :
http://www.di.ens.fr/~jv/HomePage/teaching.html
(Cours Info : 6 ECTS)

Systèmes et réseaux

(Marc Pouzet)

Le cours de systèmes présente les concepts fondamentaux des systèmes d'exploitation, leur utilisation et leur mise en œuvre dans un système UNIX.
Ce cours abordera, en autres, les points suivants :

- système de fichiers;
- gestion des processus;
- mémoire virtuelle;
- communication et synchronisation entre processus concurrents (mémoire partagée, signaux, sémaphores, sockets);
- ordonnancement préemptif et non-préemptif; OS temps réel;
- modèles de concurrence de haut niveau.

Les notions introduites seront illustrées par l'écriture d'applications en C ou en Ocaml, en utilisant l'interface POSIX.

La page du cours : http://www.di.ens.fr/~pouzet/cours/systeme/
(Cours Info : 6 ECTS)

Systèmes synchrones

(Marc Pouzet, Jean Vuillemin)

Les langages synchrones ont été créés pour programmer les systèmes réactifs embarqués à la fois très complexes et très sûrs. Ils ont connu, depuis, un succès industriel majeur dans la programmation de systèmes critiques: avions, trains, automobiles, centrales électriques, etc. Le système de commande de vol des Airbus, par exemple, est développé avec l'outil SCADE issu du langage synchrone Lustre.

Ces langages ont évolué sans cesse depuis pour traiter des applications et domaines nouveaux: calcul vidéo intensif (TVHD); grandes simulations (réseaux électriques, réseaux de capteurs); systèmes mixtes continu/discrets (environnement physique, interface analogique/discret en électronique).

Ils sont fondés sur un modèle original dit du parallélisme synchrone qui combine parallélisme et déterminisme. Le programme est décrit dans un langage parallèle de haut niveau mais pour lequel le

compilateur garantit des propriété de sûreté fortes: sémantique déterministe, absence de blocage (deadlock), génération de code séquentiel s'exécutant en temps et mémoire bornés, etc. En somme, les langages synchrones permettent de programmer dans un formalisme de haut niveau, le code final embarqué étant produit directement par le compilateur.

Le cours donne une introduction au modèle synchrone et aux principaux langages qui en sont issus. Il présente leurs fondements sémantiques et logiques, les techniques de compilation vers du logiciel et des circuits, leur vérification formelle (par model-checking) et certains travaux de recherche récents. Nous montrerons dans ce cours les liens étroits entre la théorie des langages synchrones et la théorie des langages fonctionnels typés.

Pour plus de renseignements sur ce cours MPRI 2-23-1 (3 ECTS), consulter sa page :
https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-2-23-1

Techniques en Cryptographie et Cryptanalyse

(Michel Abdalla, Vadim Lyubashevsky)

L'objectif de ce cours est d'amener les élèves à la frontière de la recherche actuelle en cryptographie à clé publique et sur le calcul sur des données chiffrées.

Ce cours est divisé en deux parties: le chiffrement fonctionnel et le chiffrement complètement homomorphe.

Dans la première partie, nous considérerons la notion de chiffrement fonctionnel qui est une généralisation de la notion classique de chiffrement à clé publique. Dans ces systèmes, les clés de déchiffrement ne permettent à un utilisateur que de calculer certaines fonctions particulières des données chiffrées. Nous examinerons notamment plusieurs cas particuliers de chiffrement fonctionnel, tels que le chiffrement à basé d'identité (identity-based encryption), le chiffrement indexable (searchable encryption) et le chiffrement à base d'attribut (attribute-based encryption).

Dans la deuxième partie de ce cours, nous étudierons la notion de chiffrement complètement homomorphe, qui permet des calculs arbitraires sur les données chiffrées. Pour atteindre cet objectif, nous examinerons plusieurs problèmes calculatoires sur les réseaux et les systèmes de chiffrement à base de réseau qui sont utilisés dans les constructions de schémas de chiffrement complètement homomorphe.

Pour plus de renseignements sur ce cours MPRI 2-12-1 (3 ECTS),
consulter sa page sur le site du MPRI :
https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-2-12-1

Théorie de l'information et codage

(Marc Lelarge)

- Notions de base :
Entropie, information mutuelle, suites typiques, inégalité de Fano.

- Compression de données :
Codage de source, inégalité de Kraft, codages de Huffman, Ziv-Lempel, théorie de la distorsion.

- Capacité d'un canal :
Théorème de Shannon.

- Codes correcteurs d'erreur :
codes linéaires, codes cycliques, codes de Hamming, BCH, Reed-Solomon.

Bibliographie :
- R.J. McEliece, The Theory of Information and Coding, 1982.
- T. Cover, J. Thomas, Elements of Information Theory, Wiley, 1991.
- C. Shannon, A Mathematical Theory of Communication, 1948.
- http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf

Pour plus de renseignements sur : http://www.di.ens.fr/~lelarge/info.html
(Cours Info : 6 ECTS)

Topologie et Calcul Différentiel

(Patrick Bernard)

Topologie générale : Définitions, topologie produit, topologie quotient, exemples de topologies (topologie de Schwartz,...). Espaces topologiques connexes, compacts, espaces vectoriels topologiques, limites et valeurs d'adhérence, semi-continuité, théorème de Tietze-Urysohn. Théorie de Baire, théorème de Tychonoff. Théorèmes du point fixe, théorème d'Ascoli, théorème de Stone-Weierstrass, théorème de Hahn-Banach et Banach-Steinhaus.

2) Calcul différentiel banachique : théorèmes d'inversion locale, des fonctions implicites et du rang constant.

3) Espaces de Hilbert : convexité, dualité, bases hilbertiennes, exemple des séries de Fourier, théorème spectral.

4) Équations différentielles ordinaires : théorème de Cauchy-Lipschitz, flots de champs de vecteurs, linéarisation.

Voir les mises à jour de ce cours sur :
http://www.math.ens.fr/enseignement/fiche_cours.html?cours=43#
(Cours Maths : 12 ECTS)

Traitement du Signal

(Stéphane Mallat)

Ce cours présente les bases du traitement du signal digital, et des applications aux traitements des sons et de l'image. Chaque cours sera suivi d'une séance
de travaux dirigés pouvant inclure des simulations informatiques. Les notions suivantes seront introduites:

- Intégrale de Fourier et transformée de Fourier discrète.
- Filtrage et théorème d'échantillonnage pour la conversion analogique/digitale.
- Modélisation de signaux par processus stationnaires et applications au débruitage par filtrage de Wiener.
- Analyse temps-fréquence et traitement des sons.
- Théorie de l'information, entropie et codage par transformée pour la compression de signaux et d'images.
- Introduction aux traitements non-linéaires.

(Cours Info : 6 ECTS)