



CentraleSupélec

CATALOGUE DE COURS

Diplôme d'Ingénieur Informatique

Deuxième année

Campus de Metz de CentraleSupélec

dernière mise à jour : 25 janvier 2026

Semestre 7

ISP-INF-S07-05		Intelligence artificielle S07	8 ECTS
3MD1530	2.5	Modèles statistique 1	29.0 h
3MD1540	3	Apprentissage automatique 1	35.5 h
3MD4040	2.5	Apprentissage profond	28.5 h

ISP-INF-S07-06		Informatique 1 S07	7 ECTS
SPM-INF-012	2	HPC sur un nœud de calcul	22.5 h
3MD4130	2	Modèles de calcul du Big Data	21.0 h
SPM-INF-011	3	Génie logiciel	36.0 h

ISP-INF-S07-27		Informatique 2 S07	7 ECTS
SPM-INF-014	2	Traitement des langages	21.0 h
SPM-INF-013	2	Recherche opérationnelle	19.5 h
SPM-MAT-004	2	Optimisation	25.0 h
SPM-MAT-007	1	Théorie de l'information	11.0 h

ISP-INF-S07-23		HEP S07	4 ECTS
SPM-HEP-014	2	Les entreprises et leurs transformations	26.5 h
SPM-HEP-013	P/F	Séminaires	10.0 h
SPM-HEP-019	1	Gestion des ressources humaines	18.0 h
SportS07	P/F	Éducation Physique et Sportive S07	21.0 h

ISP-INF-S07-17		Langues S07	4 ECTS
LV1S07	1	Langues Vivantes et Culture 1	21.0 h
LV2S07	1	Langues Vivantes et Culture 2	21.0 h

Semestre 8

ISP-INF-S08-07		Développement d'applications S08	10 ECTS
SPM-INF-015	3	Architectures applicatives	41.0 h
SPM-PRJ-003	7	Projet de développement logiciel	84.0 h

ISP-INF-S08-08		Informatique S08	9 ECTS
SPM-INF-004	2.5	Introduction à la Programmation et l'Algorithmique Quantique	35.5 h
SPM-INF-016	2.5	Paradigmes de programmation	28.5 h
SPM-INF-017	2.5	Robotique autonome	41.0 h
SPM-PRJ-004	1.5	Projet Robotique et IA	21.0 h

ISP-INF-S08-24		HEP S08	7 ECTS
SPM-HEP-011	1	Ingénieur, environnement et société	14.0 h
SPM-HEP-018	1	Controverse	18.0 h
SPM-HEP-017	2	Ingénierie système	21.5 h
SportS08	P/F	Éducation Physique et Sportive S08	21.0 h
SPM-STA-002	4	Stage Ingénieur	0.0 h

ISP-INF-S08-18		Langues S08	4 ECTS
LV1S08	1	Langues Vivantes et Culture 1	21.0 h
LV2S08	1	Langues Vivantes et Culture 2	21.0 h

Responsable de cours : Frédéric Pennerath

Total : 29.0 h

CM : 16.5 h, **TD :** 4.5 h, **TP :** 6.0 h

3MD1530

[retour](#)

Description : Les cours "Modélisation statistique" ModStat1 et ModStat2 traitent de la modélisation de systèmes pour lesquels les sorties sont suffisamment incertaines pour qu'elles nécessitent d'être modélisées par des variables aléatoires. Le cours débute avec des rappels en statistique et l'introduction de modèles élémentaires (e.g. naive Bayes, régression linéaire, etc) pour aller progressivement vers des modèles plus complexes. Si les cours présentent les modèles et méthodes élémentaires les plus utiles dans ce contexte de modélisation, ils ne se veulent pas être un catalogue exhaustif. L'objectif est davantage de présenter au sein d'une théorie cohérente les concepts et outils théoriques communs à l'ensemble de ces modèles et méthodes et de montrer comment à partir d'hypothèses de modélisation propres à chaque problème concret, ces concepts sont assemblés logiquement avant d'aboutir à une méthode opérationnelle. D'un point de vue pratique l'enjeu de ce cours est non seulement de donner aux étudiants les moyens de comprendre et d'utiliser à bon escient les implémentations existantes de modèles mais aussi de concevoir ses propres implémentations pour prendre en compte les particularités d'un problème donné. Le cours s'attache à rattacher la théorie à la pratique : sont d'abord identifiées en cours les hypothèses associées à une classe de problèmes donnée, s'ensuit un travail théorique de modélisation, qui conduit à définir un modèle et ses algorithmes d'estimation. Ces résultats sont alors appliqués à un cas d'étude en TD avant de faire l'objet en TP d'un travail d'implémentation (en Python) et d'évaluation sur des données. Le cours ModStat1 introduira les outils de bases de la modélisation statistique quand le cours ModStat2 se concentrera sur les modèles à variables cachées.

Prérequis : - Connaissances de base en théorie des probabilités, statistique et apprentissage machine (machine learning) - Niveau débutant en programmation Python / Numpy

Acquis d'apprentissage : - Être capable de choisir un modèle/ une méthode statistique adaptée au problème considéré et de la mettre en œuvre de façon appropriée. - Être capable de comprendre les concepts théoriques sous-jacents à une méthode d'inférence statistique présentée dans un article scientifique. - Être capable d'implémenter un modèle / une méthode statistique dans un langage tel que Python. - Être capable d'adapter un modèle / une méthode pour tenir compte des spécificités du problème traité.

Modalités d'évaluation : Examen écrit de 2h sans documents, rattrapable.

Compétences évaluées :

- Modélisation
- Recherche et Développement

CM :

1. Modèles statistiques (1.5 h)
2. Estimation (1.5 h)
3. Estimation bayésienne (1.5 h)
4. Réseaux bayésiens 1 (1.5 h)
5. Naive Bayes (1.5 h)
6. Réseaux bayésiens 2 (1.5 h)
7. Causalité (1.5 h)
8. Modèles gaussiens (1.5 h)
9. Modèles linéaires (1.5 h)
10. Famille exponentielle (1.5 h)
11. GLM (1.5 h)

TD :

1. Estimation bayésienne (1.5 h)
2. Modélisation causale (1.5 h)
3. Régression (1.5 h)

TP :

1. Modélisation et modèles gaussiens (3.0 h)
2. Régression (3.0 h)

Responsable de cours : Hervé Frezza-Buet**Total :** 35.5 h**CM :** 13.5 h, **TP :** 20.0 h

3MD1540

[*retour*](#)

Description : Ce cours donne le cadre général de l'apprentissage automatique, permettant de situer les différentes approches du domaine. Sont abordés les notions de prétraitement des données, une introduction à la théorie de l'apprentissage statistique (risques, sur-apprentissage, proxys convexes, régularisation), la différence entre approches fréquentistes et bayésiennes, les paradigmes d'apprentissage supervisé, non-supervisé, semi-supervisé, et par renforcement. Quelques approches sont détaillées (méthodes à Noyaux, SVM, Boosting, Bagging, Arbres de décision...).

Acquis d'apprentissage : À l'issue de ce cours, les élèves sauront reconnaître dans le paysage des nombreuses méthodes disponibles sur l'étagère les différentes classes d'algorithmes. Ils auront de plus les notions statistiques qui leur permettront un usage raisonné de ces méthodes, évitant ainsi une approche boîte noire avec essai des paramètres à l'aveugle.

Moyens : Les cours et travaux de pratiques sont donnés par Hervé Frezza-Buet, Arthur Hoarau, Jérémy Fix. Les cours présentent les aspects théoriques, des preuves mathématiques, mais sont également illustrés par des démonstrations des algorithmes. Les travaux pratiques s'effectueront en python, en utilisant sickit-learn, en binômes.

Modalités d'évaluation : Examen écrit de 2h, rattrapable.

Compétences évaluées :

- Recherche et Développement
- Développement

CM :

1. Datasets and learning (1.5 h)
2. Frequentist, Bayesian, evaluation (1.5 h)
3. Risks (1.5 h)
4. C-SVC, Lagrange formulation (1.5 h)
5. Kernels, numerical resolution (1.5 h)
6. SVMs for regression, unsupervised learning, nu-versions of SVMs. (1.5 h)
7. Arbres de décision (1.5 h)
8. Bagging (1.5 h)
9. Boosting (1.5 h)

TP :

1. Data Science en Python (3.0 h)
2. Arbres de décision (3.0 h)
3. Bagging (3.0 h)
4. Forêts aléatoires (3.0 h)
5. TP+ 1/2 (4.0 h)
6. TP+ 2/2 (4.0 h)

Responsable de cours : Jérémy Fix**Total :** 28.5 h**CM :** 13.5 h, **TP :** 15.0 h

3MD4040

[retour](#)

Description : L'apprentissage profond est une technologie en plein essor ces dernières années, notamment grâce à l'utilisation des GPUs (Graphical Processing Units), la disponibilité de grandes masses de données mais aussi la compréhension d'éléments théoriques permettant de mieux définir des architectures de réseaux de neurones plus facilement entraînaibles. Dans ce cours, les étudiants seront introduits aux bases des réseaux de neurones et également aux différents éléments architecturaux qui permettent de concevoir un réseau de neurones en fonction du problème de prédiction considéré. Le cours est décomposé en modules dans lesquels on aborde les questions des algorithmes d'optimisation, leur initialisation, les techniques de régularisation, les architectures complètement connectées, les réseaux convolutifs, les réseaux récurrents, les techniques d'introspection. Des travaux pratiques sur GPUs sont associés aux cours.

Contenu : Les sujets abordés seront les suivants :

- Introduction historiques aux réseaux de neurones, classifieur/régresseur linéaire (1.5 HPE) - Graphe de calcul et descente de gradient, Réseaux complètement connectés, RBFs, Auto-encodeurs, Méthodes d'optimisations, initialisation, régularisation (3 HPE) - Réseaux convolutifs : architectures (1.5 HPE) - Réseaux convolutifs : classification, détection d'objet, segmentation sémantique (1.5 HPE) - Réseaux récurrents : architectures et entraînement (1.5 HPE) - Réseaux récurrents : applications (1.5 HPE) - Les approches par transformers (1.5 HPE) - Modèles génératifs : modèles auto-regressifs, VAE, GANs et diffusion (1.5 HPE)

Les TP porteront sur :

- Introduction à pytorch par la classification avec des réseaux linéaires, complètement connectés et convolutifs (3 HPE) - Réseaux convolutifs pour la segmentation sémantique (3 HPE) - Réseaux récurrents pour la conversion séquence à séquence (3 HPE) - Réseaux génératifs adversariaux (3 HPE) - Apprentissage auto-supervisé (3 HPE)

Prérequis : On supposera que les étudiants auront des connaissances en algèbre linéaire, en optimisation, en vision par ordinateur et en programmation python. Les étudiants doivent disposer d'une certaine aisance dans un environnement Linux.

Acquis d'apprentissage : Être capable d'implémenter et de déployer un algorithme de deep learning (prise en main des frameworks de deep learning et déploiement du calcul du GPU) Être capable de choisir l'architecture de réseau de neurones adaptée au problème de prédiction considéré Savoir diagnostiquer l'apprentissage d'un réseau de neurones (qu'apprend le réseau de neurone ? comment l'apprend t'il ? apprend t'il ? est-il capable de généraliser ?)

Méthodes pédagogiques : Le cours est construit autour d'un cours magistral pendant lequel les notions théoriques et expérimentales sont introduites et illustrées au travers d'exemple. Des TPs sont régulièrement répartis le long du cours pour pouvoir mettre en œuvre les notions vues en cours.

Moyens : Nous utiliserons le framework python Pytorch. Les étudiants pourront travailler en binôme et disposeront des GPUs du Data Center d'Enseignement du campus de Metz pour faire les TPs. Une page dédiée sera créée sur eduno. Des forums de discussion seront ouverts pour permettre aux étudiants de poser leurs questions sur le cours et d'interagir entre eux et avec l'équipe enseignante.

Modalités d'évaluation : L'évaluation reposera sur deux éléments : des évaluations sur des questionnaires papier en début de chaque TP et la participation, en équipe, à un challenge créé pour l'occasion. Les élèves devront également réaliser une présentation filmée décrivant leur approche et leurs résultats.

Compétences évaluées :

- Modélisation
- Recherche et Développement
- Management

Ressources externes :

- Site du cours
 - Sujets des TPs
-

CM :

1. Introduction et réseaux linéaires (1.5 h)
2. Graphe de calcul, descente de gradients et réseaux feedforward (1.5 h)
3. Optimisation, Initialisation, Régularisation (1.5 h)
4. Eléments d'architectures des réseaux convolutifs (1.5 h)
5. Applications des réseaux convolutifs à la détection d'objets et la segmentation sémantique (1.5 h)
6. Les approches par transformers (1.5 h)
7. Eléments d'architecture des réseaux récurrents (1.5 h)
8. Architectures et applications des réseaux récurrents (1.5 h)
9. Modèles génératifs : modèles auto-regressifs, VAE, GANs et diffusion (1.5 h)

TP :

1. Introduction à PyTorch par la classification d'images (3.0 h)
2. Réseaux convolutifs pour la segmentation sémantique (3.0 h)
3. Application des RNNs pour la retranscription de la parole (3.0 h)
4. Modèles génératifs (3.0 h)
5. Apprentissage auto-supervisé (3.0 h)

Responsable de cours : Stéphane Vialle

Total : 22.5 h

CM : 9.0 h, **TD :** 3.0 h, **TP :** 9.0 h

SPM-INF-012

[retour](#)

Description : Calcul parallèle sur un ordinateur multicœurs par multithreading, vectorisation et optimisation sérielle. Ce cours présente l'architecture générique des CPU modernes (multi-cœurs, unités SIMD, hiérarchie de mémoires caches), puis la gestion efficace des accès mémoire pour exploiter efficacement les caches, puis la vectorisation des boucles pour profiter des unités SIMD, et enfin le multithreading avec OpenMP pour exploiter tous les cœurs de la machine. Des aspects de programmation et d'algorithmique parallèle constituent l'essentiel du cours, accompagnés de concepts de mesure et d'analyse de performance.

Acquis d'apprentissage : A l'issue de ce cours, les élèves sauront concevoir et implanter un code de calcul multithreadé vectorisable au sein de chaque thread, et réalisant un maximum d'accès aux données en mémoire cache, puis le compiler avec les options de compilation les plus adaptées, afin de tirer un maximum de performances d'un processeur multi-cœurs moderne.

Modalités d'évaluation : Examen écrit 1h30 (rattrapable) et compte-rendus de TP (non rattrapables).

Compétences évaluées :

- Développement
- Modélisation

CM :

1. Cours d'architecture des CPU multi-cœurs et des caches (1.5 h)
2. Cours de compilation optimisée et de vectorisation (1.5 h)
3. Cours de mesure et d'analyse de performances (1.5 h)
4. Cours d'OpenMP et de bibliothèques BLAS (3.0 h)
5. Cours d'algorithmique parallèle (1.5 h)

TD :

1. TD d'optimisation sérielle et de vectorisation (1.5 h)
2. TD de programmation OpenMP (1.5 h)

TP :

1. TP d'optimisation sérielle et de vectorisation (3.0 h)
2. TP de programmation OpenMP (3.0 h)
3. TP d'algorithmique et de programmation parallèle optimisée (3.0 h)

Responsable de cours : Stéphane Vialle

Total : 21.0 h

CM : 10.5 h, **TD :** 1.5 h, **TP :** 9.0 h

3MD4130

[retour](#)

Description : Ce cours a pour objectif d'apprendre aux élèves à développer des applications performantes d'analyse de données en environnement Spark sur des plates-formes distribuées (clusters et Clouds). Des mécanismes de systèmes de fichiers distribués comme HDFS seront étudiés, ainsi que le modèle de programmation et l'algorithme du "map-reduce étendu" de Spark au dessus des Spark "RDD", puis des modèles de programmation de plus haut niveau au dessus de Spark "Data Frames", et enfin des modèles de programmation sur Cloud. Des critères et métriques de passage à l'échelle seront également étudiés. Tout au long du cours des mises en oeuvre auront lieu sur des clusters et dans un Cloud, et les solutions développées seront évaluées par les performances obtenues sur les cas-tests, et par leur aptitude à passer à l'échelle.

Contenu : Emergence des technologies Big Data : motivations, besoins industriels, principaux acteurs. Pile logicielle d'Hadoop, architecture et fonctionnement de son système de fichier distribué (HDFS) Architecture et mécanisme de déploiement de calculs distribués de Spark Modèle de programmation par "RDD" et algorithmique du "map-reduce" étendu de Spark Modèle de programmation de Spark par "Data Frames" appliqué à l'analyse de graphes (GraphX) Architecture et environnement d'analyse de données sur Cloud Expérimentations et mesures de performances Critères et métriques de performances

Acquis d'apprentissage : A l'issue de ce cours, les élèves seront en mesure :

AA1 : de concevoir et d'implanter des algorithmes "map-reduce étendu" en Spark, efficaces sur des plates-formes distribuées, et passant à l'échelle. AA2 : d'analyser les capacités de "passage à l'échelle" d'une application AA3 : d'utiliser un cluster ou un cloud pour réaliser des analyses de données distribuées à large échelle. AA4 : de présenter synthétiquement une solution d'analyse de données conçue en "map-reduce"

Méthodes pédagogiques : Ce cours enchaîne 3 parties relatives à des modèles de calculs du "Big Data" : la première sur clusters de PC, la seconde dans les Cloud, et la troisième qui évalue des solutions de "passage à l'échelle".

Plan du cours en 4 parties :

Partie 1 : Architecture et développement en Spark RDD sur HDFS et clusters de PC.

Partie 2 : Critères et métriques pour la performance et le passage à l'échelle.

Partie 3 : Calcul et analyse de données large échelle sur Cloud.

Partie 4 : Développement en Spark Data Frames sur HDFS et clusters de PC.

Moyens : Equipe enseignante : Stéphane Vialle et Gianluca Quercini (CentraleSupélec), Wilfried Kirschenmann (ANEO) Plateforme de développement et d'exécution : clusters de calcul du Data Center d'Enseignement (DCE) du campus de Metz de CentraleSupélec accès à un Cloud professionnel Environnements de développement : Spark + HDFS sur le DCE autre environnement sur le Cloud

Modalités d'évaluation : Evaluation à partir des TP :

Les comptes rendus des TP seront notés (le contenu et le nombre de pages des comptes rendus sont contraints, afin de forcer les étudiants à un effort de synthèse et de clarté) En cas d'absence non justifiée à un TP la note de 0 sera appliquée, en cas d'absence justifiée le TP n'interviendra pas dans la note finale. L'examen de rattrapage sera un examen oral, qui constituera 100

Compétences évaluées :

- Développement
- Système

CM :

1. Emergence du Big Data et technologie HDFS d'Hadoop (1.5 h)
2. Technologie Spark-RDD et programmation Map-Reduce (1.5 h)
3. Optimisation et déploiement de codes Map-Reduce (1.5 h)
4. Métriques de passage à l'échelle et architecture des Data Lakes (1.5 h)
5. Spark Data-Frames et Spark SQL (1.5 h)
6. Problématiques du cloud (1.5 h)
7. Environnement de développement sur cloud (1.5 h)

TD :

1. Algorithmique Map-Reduce en Spark (1.5 h)

TP :

1. Map-Reduce sur cluster Spark-HDFS : programmation et performances (3.0 h)
2. Analyse de données en Spark Data-Frame et Spark SQL sur cluster Spark-HDFS (3.0 h)
3. Développement et déploiement d'une analyse de données extensible sur Cloud (3.0 h)

Description : Ce cours aborde les méthodes et les outils pour prendre en charge les différentes phases du cycle de vie d'un logiciel. Il se décompose en 3 parties : (1) Modélisation et architecture logicielle : UML et patrons de conception architecturaux, (2) Qualité logicielle : critères, différents types de tests et mesures et stratégies de tests, et (3) gestion de projet de développement logiciel : cycles de vie du logiciel, méthodes agiles, DevOps

Acquis d'apprentissage : A l'issue de ce cours, les élèves sauront s'appuyer sur une notation et des solutions standards pour concevoir et communiquer l'architecture de leurs logiciels, et ils sauront mettre en oeuvre un processus d'industrialisation pour les développer.

Modalités d'évaluation : Examen écrit 3h, rattrapable

Compétences évaluées :

- Modélisation
- Développement
- Système

CM :

1. modélisation et diagrammes UML (3.0 h)
2. patrons de conception (3.0 h)
3. qualité logicielle (3.0 h)
4. projet logiciel (1.5 h)
5. méthodes agiles (1.5 h)

TD :

1. modélisation (1.5 h)
2. système à refactorer (1.5 h)

TP :

1. UML (3.0 h)
2. refactoring par application de patrons (3.0 h)
3. tests unitaires et plus (4.0 h)
4. intégration continue (4.0 h)
5. scrum (4.0 h)

Responsable de cours : Benoît Valiron

Total : 21.0 h

CM : 7.5 h, **TD :** 4.5 h, **TP :** 9.0 h

SPM-INF-014

[retour](#)

Description : L'objectif est de présenter les notions de théorie des langages et de les mettre en pratique pour que les élèves assimilent bien et voient l'intérêt de ces notions pour traiter des problèmes concrets. On abordera les langages réguliers, les grammaires et le fonctionnement d'un compilateur.

Acquis d'apprentissage : À l'issue de ce cours, les élèves auront compris la pertinence de la théorie des langages en informatique.

Modalités d'évaluation : Évaluation à partir des participations aux manipulations (TD/TP) et des résultats rendus

Compétences évaluées :

— Certification

CM :

1. Expressions régulières, langage reconnu par un automate (3.0 h)
2. Grammaires (1.5 h)
3. Descente récursive, parseur LL (1.5 h)
4. Fonctionnement d'un compilateur (1.5 h)

TD :

1. Grammaires (1.5 h)
2. Descente récursive, parseur LL (1.5 h)
3. Compilation (1.5 h)

TP :

1. Reconnaissance de tokens dans un document (3.0 h)
2. Evaluation d'expressions arithmétiques (3.0 h)
3. Mise en oeuvre du compilateur sur machine (3.0 h)

Description : Le cours est une introduction aux problèmes d'optimisation combinatoire dans le cadre appliqué de la recherche opérationnelle. Cette dernière recouvre les méthodes d'optimisation formelles utiles aux organisations (entreprises, etc) pour prendre des décisions rationnelles, que ce soit en logistique, en stratégie (plans d'investissement, etc), en économie, etc. Ce cours et celui d'optimisation sont deux cours indissociables au sens où ils se répartissent la présentation des différentes techniques d'optimisation, selon la nature des problèmes que ces techniques résolvent : quand le cours d'optimisation se concentre sur les méthodes d'analyse numérique optimisant une fonction continue définie dans un espace euclidien, RECHOP aborde les problèmes d'optimisation combinatoire dans un espace discret dont la résolution est davantage algorithmique. De ce fait, ce cours fait une large place aux algorithmes et à leur complexité et peut à ce titre vu comme le prolongement du cours de 1ère année "Structures de données et algorithmes" (SDA). Le cours commencera par aborder les problèmes combinatoires dans le cadre très général de la programmation par contraintes. On se concentrera ensuite sur des instances de problèmes pour lesquels une résolution en temps polynomial existe, comme la programmation dynamique ou les problèmes de couplages de graphes. On développera également des algorithmes d'approximation efficaces pour le problème difficile de la programmation linéaire entière après que la programmation linéaire a été traitée en cours d'optimisation, faisant ainsi le lien entre les deux cours.

Acquis d'apprentissage : A l'issue de ce cours, les élèves auront une culture générale sur le type de problèmes rencontrés en recherche opérationnelle. Ils sauront formaliser un besoin en recherche opérationnelle et reconnaître la nature précise du problème combinatoire sous-jacent. Le cas échéant, ils sauront associer à ce problème combinatoire une technique de résolution adaptée.

Modalités d'évaluation : Examen écrit 1h30, rattrapable

Compétences évaluées :

- Modélisation
- Recherche et Développement

CM :

1. Programmation par contraintes (1.5 h)
2. Problèmes d'optimisation temporelle. Programmation dynamique (1.5 h)
3. Problème d'affectations : couplage de graphes et mariages stables (1.5 h)
4. Programmation linéaire entière (1.5 h)

TP :

1. Programmation par contraintes (3.0 h)
2. Programmation dynamique (3.0 h)
3. Problèmes de couplage (3.0 h)
4. Programmation linéaire entière (3.0 h)

OPTIMISATION

Responsable de cours : Michel Barret

Total : 25.0 h

CM : 10.5 h, **TD :** 4.5 h, **TP :** 9.0 h

SPM-MAT-004

[retour](#)

Description : Dans ce cours les élèves devront acquérir et maîtriser divers aspects fondamentaux de l'optimisation continue. Les notions suivantes seront abordées et mises en oeuvre pratiquement : formulation des problèmes d'optimisation, conditions d'existence de minimiseurs globaux et locaux, convexité, dualité, multiplicateurs de Lagrange, méthodes du premier ordre, programmation linéaire. L'utilisation de la programmation différentiable sera présentée en travaux pratiques. Les méthodes stochastiques 'gradient-free', comme CMAES et PSO, seront également abordées.

Acquis d'apprentissage : À l'issue de ce cours, les élèves maîtriseront les concepts fondamentaux de l'optimisation continue (conditions d'existence de minimiseurs globaux et locaux, convexité, dualité, multiplicateurs de Lagrange, méthodes du premier ordre, programmation linéaire, méthodes stochastiques).

Modalités d'évaluation : Examen écrit 1h, rattrapable

Compétences évaluées :

- Modélisation
- Recherche et Développement

CM :

1. Bases de l'optimisation 1/2 (1.5 h)
2. Bases de l'optimisation 2/2 (1.5 h)
3. Convexité, quelques algorithmes itératifs (1.5 h)
4. Dualité (1.5 h)
5. Programmation linéaire (1.5 h)
6. Méthode des multiplicateurs de Lagrange (1.5 h)
7. Méthodes stochastiques gradient-free (1.5 h)

TD :

1. Dualité (1.5 h)
2. Programmation linéaire (1.5 h)
3. Méthode des multiplicateurs de Lagrange (1.5 h)

TP :

1. Bases de l'optimisation (3.0 h)
2. Convexité, quelques algorithmes itératifs (3.0 h)
3. Méthodes stochastiques gradient-free (3.0 h)

Description : Dans ce cours les élèves devront acquérir et maîtriser les bases de la théorie de l'information. Les notions suivantes seront abordées : problème de codage source, entropie d'un échantillon et d'une distribution de probabilité, entropie conjointe et conditionnelle, information mutuelle et divergence de Kullback-Leibler, et principe de maximum d'entropie. L'utilisation d'arbre à préfixe pour déterminer des codages uniquement décodable est abordée par l'exemple du codage optimal de Huffman.

Acquis d'apprentissage : À l'issue de ce cours, les élèves seront capables de décrire et expliquer les concepts fondamentaux du codage de sources ainsi que les problématiques associées, de présenter une intuition des implications des concepts du cours pour le problème du codage source, de manipuler et analyser l'entropie d'un point de vue mathématique, de définir la divergence de Kullback-Leibler et l'articuler avec les notions d'entropie et d'information mutuelle et de mettre en œuvre le principe du maximum d'entropie

Modalités d'évaluation : Examen écrit 2h, rattrapable

CM :

1. Codage source de canaux discrets (1.5 h)
2. Théorie de l'information 1/2 (1.5 h)
3. Théorie de l'information 2/2 (1.5 h)

TD :

1. Entropie et divergence KL (1.5 h)

TP :

1. Codage de Huffman (3.0 h)

Responsable de cours : Francis Dorveaux

Total : 26.5 h

CM : 5.5 h, **TD :** 12.0 h, **TP :** 8.5 h

SPM-HEP-014

[retour](#)

Description : L'objectif de ce cours est d'explorer la diversité des entreprises et leur intégration à un tissu socio-économique et environnemental global, ainsi que les dynamiques qu'elles engendrent. L'approche pédagogique retenue consiste à combiner des apports théoriques en cours magistral et un travail d'enquête de terrain en groupe permettant de former les futurs entrepreneurs en les confrontant à la réalité et au développement d'une entreprise.

Acquis d'apprentissage : À l'issue de ce cours, les élèves auront acquis une vision globale de l'écosystème entrepreneurial et des forces qui l'animent.

Modalités d'évaluation : QCM (30 min), restitutions individuelles

Compétences évaluées :

— Business Intelligence

CM :

1. Introduction (1.0 h)
2. Panorama des entreprises (2.0 h)
3. Synthèse et débat (2.5 h)

TD :

1. TD 1 : Les théories modernes de l'entreprise + explications des attendus des TD 2 à 6 (1.5 h)
2. TD 2 : Les processus "colonne vertébrale" de l'organisation (1.5 h)
3. TD 3 : La dynamique externe (1.5 h)
4. TD 4 : Transformation des entreprises (3.0 h)
5. TD 5 : Business Development 1 (à préciser) (1.5 h)
6. TD 6 : Business Development 2 (à préciser) (3.0 h)

TP :

1. 1ère visite en entreprise (4.5 h)
2. 2ème visite en entreprise (4.0 h)

SÉMINAIRES

Responsable de cours : Damien Rontani, Hervé Frezza-Buet

Total : 10.0 h

CM : 10.0 h

SPM-HEP-013

[retour](#)

Description : Ce cours est en ensemble de séminaires d'ouverture.

Acquis d'apprentissage : À l'issue de ces conférences, les élèves auront élargi leur regard sur les enjeux environnementaux, économiques, sociétaux, éthiques, en fonction de l'expertise des intervenants.

CM :

1. tdb (10.0 h)

Responsable de cours : Damien Rontani, Hervé Frezza-Buet

Total : 18.0 h

TD : 18.0 h

SPM-HEP-019

[retour](#)

Description : Ce cours a un double objectif : d'une part équiper les étudiants de connaissances sur leurs droits et obligations face aux pratiques RH, notamment lors du recrutement, et d'autre part permettre aux ingénieurs devenant managers de collaborer efficacement avec la direction et les services RH.

Acquis d'apprentissage : À l'issue de ce cours, les élèves auront acquis un socle de connaissances en droit du travail et en fonctions RH qui leur sera utile à la fois en tant qu'employés et également dans leurs missions d'encadrement.

Compétences évaluées :

— Management

TD :

1. tbd (18.0 h)

ÉDUCATION PHYSIQUE ET SPORTIVE S07

Responsable de cours : Hervé Frezza-Buet

Total : 21.0 h

TD : 21.0 h

SportS07

[retour](#)

Description : Au-delà du développement des compétences motrices (physiques, techniques et tactiques) les enseignements d'éducation physique et sportive ont pour objectifs de permettre aux étudiants de développer leurs compétences personnelles de connaissance et de contrôle de soi mais aussi leurs compétences relationnelles (collaboration au sein de l'équipe, écoute, communication, animation, ...).

TD :

1. Cours de sport (21.0 h)

Responsable de cours : Elisabeth Leuba

Total : 21.0 h

TD : 21.0 h

LV1S07

[retour](#)

Description : La Langue Vivante 1 sera généralement l'anglais. Répartis par niveau, les étudiants travailleront non seulement les 4 compétences langagières mais aborderont aussi des sujets variés qu'ils approfondiront suivant leur niveau. Les sujets traités peuvent être d'ordre civilisationnel, sociétal, professionnel, etc. Les effectifs des groupes constitueront un environnement propice à une participation active et une progression conséquente dans la langue. Différentes méthodes pédagogiques seront utilisées : travail en groupe, exposés, exercices spécifiques, recherche, débats, etc.

Acquis d'apprentissage : À l'issue de ce cours, l'élève aura progressé pour communiquer dans un environnement universitaire, professionnel ou personnel internationalisé

Compétences évaluées :

- Recherche et Développement
- Conseil
- Business Intelligence
- Management

TD :

1. Cours (21.0 h)

Responsable de cours : Beate Mansanti

Total : 21.0 h

TD : 21.0 h

LV2S07

[retour](#)

Description : En Langue Vivante 2, une offre de plusieurs langues sera proposée aux étudiants, en poursuite d'étude ou en débutant. Répartis par niveau, les étudiants travailleront non seulement les 4 compétences langagières mais aborderont aussi des sujets variés qu'ils approfondiront suivant leur niveau. Les sujets traités peuvent être d'ordre civilisationnel, sociétal, professionnel, etc. Les effectifs des groupes constitueront un environnement propice à une participation active et une progression conséquente dans la langue. Différentes méthodes pédagogiques seront utilisées : travail en groupe, exposés, exercices spécifiques, recherche, débats, etc.

Acquis d'apprentissage : À l'issue de ce cours, l'élève aura progressé pour communiquer dans un environnement universitaire, professionnel ou personnel internationalisé

Compétences évaluées :

- Recherche et Développement
- Conseil
- Business Intelligence
- Management

TD :

1. Cours (21.0 h)

Responsable de cours : Virginie Galtier

Total : 41.0 h

CM : 12.0 h, **TD :** 1.5 h, **TP :** 25.5 h

SPM-INF-015

[retour](#)

Description : Ce cours dresse un panorama des architectures applicatives (historique, principes, bénéfices et limites, cadres d'utilisation...). Deux architectures seront plus particulièrement approfondies : l'architecture REST (avec OpenAPI) et une architecture basée sur les MOM (Kafka). Les élèves seront sensibilisés aux problématiques de tolérance aux pannes et passage à l'échelle, et découvriront une solution de déploiement (Kubernetes).

Acquis d'apprentissage : A l'issue de ce cours les élèves sauront analyser une proposition architecturale et sauront mettre en oeuvre et déployer deux architectures classiques dans un contexte simple.

Modalités d'évaluation : Examen écrit 2h, rattrapable

Compétences évaluées :

- Développement
- Système

CM :

1. panorama des architectures applicatives (3.0 h)
2. architecture REST, OpenAPI (1.5 h)
3. architecture basée sur les MOM (Kafka) (1.5 h)
4. tolérance aux pannes, passage à l'échelle et déploiement (K8s) (3.0 h)
5. cloud (3.0 h)

TD :

1. études de cas architecture (1.5 h)

TP :

1. client REST (3.0 h)
2. serveur REST (3.0 h)
3. tests avec SoapUI (3.0 h)
4. GraphQL (1.5 h)
5. tuto Kafka (4.5 h)
6. Kubernetes (4.5 h)
7. cloud 1/2 (3.0 h)
8. cloud 2/2 (3.0 h)

PROJET DE DÉVELOPPEMENT LOGICIEL

Responsable de cours : Hervé Frezza-Buet

Total : 84.0 h

Projet : 84.0 h

SPM-PRJ-003

[*retour*](#)

Description : Il s'agit dans ce module d'appliquer les méthodes de génie logiciel pour développer un projet informatique d'ampleur. Ce projet pourra faire l'objet de collaborations industrielles, et de travail mis en commun avec d'autres spécialités. L'idée du module est d'avoir l'expérience de la conception et la réalisation d'un projet en situation la plus proche possible d'un environnement professionnel.

Acquis d'apprentissage : À l'issue de ce cours, les élèves auront mis en œuvre une méthodologie professionnelle de réalisation d'un développement informatique conséquent en équipe.

Modalités d'évaluation : Rapport et soutenance

Compétences évaluées :

- Modélisation
- Recherche et Développement
- Développement
- Management

INTRODUCTION À LA PROGRAMMATION ET L'ALGORITHMIQUE QUANTIQUE

Responsable de cours : Damien Rontani, Stéphane Vialle

Total : 35.5 h

CM : 16.5 h, **TD :** 4.5 h, **TP :** 9.0 h, **Projet :** 5.5 h

SPM-INF-004

retour

Description : Ce cours de Programmation et d'Algorithmique Quantique offre une plongée exhaustive dans les fondamentaux et les applications pratiques des accélérateurs quantiques. Les étudiants exploreront les architectures quantiques contemporaines, comprenant les principes des architectures analogiques et digitales, ainsi que les innovations telles que les architectures hybrides CPU-QPU, NISQ et LSQ. Le cours couvrira le formalisme des qubits et de la programmation quantique digitale, mettant en évidence l'importance de la superposition et de l'intrication pour les calculs quantiques. Les principes et méthodes de mesure des résultats seront également abordés. Une introduction aux circuits quantiques, incluant les portes de base et les premiers circuits, permettra aux étudiants de comprendre les bases pratiques de la programmation quantique. En utilisant des outils tels que QFT, Grover, QPE et QMC, les étudiants exploreront les circuits classiques et leurs applications, tout en examinant les limitations sur les architectures NISQ. Enfin, les étudiants se plongeront dans les circuits et algorithmes variationnels, notamment les circuits QAOA et Vxx, et étudieront les modèles de temps d'exécution et de performance pour les QPU ainsi que pour les boucles CPU-QPU, en se familiarisant avec les ordres de grandeur des temps d'exécution actuels.

Bibliographie :

- Ref. [1] : R. Hundt, Quantum Computing for Programmers, Cambridge University Press (2022)
- Ref. [2] : P. Kaye, R. Laflamme, M. Mosca, An Introduction to Quantum Computing, Oxford University Press (2006)

Acquis d'apprentissage : A l'issue de ce cours, les élèves sauront concevoir et implanter un algorithme à base de portes quantiques, utiliser des bibliothèques de routines quantiques, concevoir et implanter un code hybride CPU+QPU, analyser ses résultats, mesurer ses performances et sa sensibilité au bruit en fonction de la taille des données et de l'utilisation du QPU.

Modalités d'évaluation : Evaluation du mini-projet

Compétences évaluées :

- Modélisation
- Développement

CM :

1. Cours d'architectures quantiques (1.5 h)
2. Cours de formalisme pour l'informatique quantique digitale (3.0 h)
3. Cours d'introduction aux portes et circuits quantiques (3.0 h)
4. Cours de présentation des circuits quantiques classiques (4.5 h)
5. Cours de modèles de temps d'exécution et de performance (1.5 h)
6. Cours de présentation des circuits quantiques variationnels (3.0 h)

TD :

1. TD de formalisme et d'analyse de circuits quantiques (1.5 h)
2. TD de conception d'algorithmes quantiques sur QPU (1.5 h)
3. TD de conception d'algorithmes variationnels sur CPU+QPU (1.5 h)

TP :

1. TP de mise en oeuvre de circuits quantiques en qiskit sur simulateur et machines quantiques (3.0 h)
2. TP de conception et mise en oeuvre d'algorithmes quantiques à partir de circuits connus (3.0 h)
3. TP de conception d'une méthode d'optimisation par algorithme variationnel sur CPU+QPU (3.0 h)

Responsable de cours : Hervé Frezza-Buet

Total : 28.5 h

CM : 6.0 h, **TP :** 22.5 h

SPM-INF-016

[*retour*](#)

Description : Ce cours aborde différents paradigmes de programmation par l'introduction de différents langages très 'marqués' par ces paradigmes. On pourra s'appuyer sur des langages fonctionnels comme Haskell et Ocaml pour les approches fonctionnelles (notion d'inférence de type et d'évaluation paresseuse), Prolog pour la programmation logique, LISP pour la construction par le calcul d'expressions exécutables...

Acquis d'apprentissage : À l'issue de ce cours, les élèves auront pu penser la programmation autrement que par les méthodes classiques. Il sauront ainsi reconnaître ces paradigmes quand ils se présentent implicitement et mobiliser des résolutions idoines (par exemple, reconnaître dans la conceptions de templates récursifs en C++ une approche fonctionnelle apprise en caml, reconnaître dans les makefiles un paradigme de programmation logique, etc...). L'idée est également de pouvoir proposer des solutions élégantes à certains problèmes, sans être dépendants de ce qu'induit la structure du langage utilisé (par exemple mettre en œuvre une approche paresseuse en C++).

Modalités d'évaluation : Evaluation à partir des participations aux manipulations (TP) et des résultats rendus

CM :

1. Cours-1 (1.5 h)
2. Cours-2 (1.5 h)
3. Cours-3 (1.5 h)
4. Cours-4 (1.5 h)

TP :

1. TP-1 (4.5 h)
2. TP-2 (4.5 h)
3. TP-3 (4.5 h)
4. TP-4 (4.5 h)
5. TP-5 (4.5 h)

Description : Ce cours présentera le domaine de la robotique autonome (conduite de véhicules, robot d'exploration et d'inspection, etc...) en montrant comment cette problématique intègre des technologies très diverses (localisation (SLAM), nuages de points, planification, reconnaissance de formes) et comment cette intégration se réalise au niveau système (illustrations avec ROS). Les travaux de laboratoire associés au cours seront réalisés sur les robots de la plateforme robotique du campus de Metz et en simulation. Ces travaux seront l'occasion d'intégrer différentes techniques d'apprentissage automatique et de traitement du signal sur des robots se déplaçant dans leur environnement. Le cours et les mises en application pratiques permettront de découvrir ces techniques dans des cas réels.

Contenu : ...

Prérequis : Il est nécessaire de maîtriser la programmation Python et de disposer de bonnes connaissances en probabilités. Les travaux pratiques nécessitent également un minimum d'être familier avec Linux.

Acquis d'apprentissage : À l'issue de ce cours, les élèves auront été familiarisé avec ROS et avec les concepts clés de la robotique autonome. Ils auront expérimenté la difficulté de coupler les traitements d'information avec un système évoluant dans un environnement réel et auront eu l'expérience de la mise en œuvre de solutions, guidés par la méthodologie induite par ROS.

Méthodes pédagogiques : Les cours permettent d'introduire les concepts et les algorithmes essentiels pour la robotique autonome (13.5 HPE), complétés par des exercices en TD (3 HPE) et largement accompagnés par des travaux pratiques en simulation et sur plateformes réelles (22.5 HPE).

Moyens : Les travaux dirigés, constitués d'exercices, permettront d'utiliser les concepts vus en cours. Les travaux pratiques permettront de programmer et expérimenter les algorithmes vus en cours en simulation et sur des plateformes robotiques réelles en utilisant le framework ROS2.

Modalités d'évaluation : Examen écrit 2h, rattrapable

CM :

1. Intro (1.5 h)
2. Introduction à ROS (1.5 h)
3. Rappels de probabilités (1.5 h)
4. Estimation d'état (1.5 h)
5. Localisation (1.5 h)
6. Carto + SLAM (1.5 h)
7. Planif (1.5 h)
8. Navigation (1.5 h)
9. Architecture et interaction (1.5 h)

TD :

1. Filtres de Kalman (1.5 h)
2. Localisation (1.5 h)

TP :

1. ROS et simulation (3.0 h)

2. Navigation et robots réels (3.0 h)
3. Filtre de Kalman et estimation d'état (1.5 h)
4. Localisation (3.0 h)
5. Carto + SLAM (3.0 h)
6. Path planning (3.0 h)
7. Path following (3.0 h)
8. Integration (3.0 h)

PROJET ROBOTIQUE ET IA

Responsable de cours : Hervé Frezza-Buet

Total : 21.0 h

Projet : 21.0 h

SPM-PRJ-004

[*retour*](#)

Description : L'objectif de ce projet est d'intégrer des algorithmes d'intelligence artificielle sur des plateformes robotiques, pour se confronter à la réalité des approches d'intelligence artificielle situées. Il s'agira de faire un synthèse des compétences acquises dans les cours du semestre 7 et dans le cours de robotique. De plus, on insistera sur la rigueur expérimentale (mise en oeuvre, compte-rendus), afin de préparer les élèves à rendre compte d'expérimentations de recherche.

Acquis d'apprentissage : À l'issue de ce cours, les élèves auront pratiqué l'IA en dehors de la manipulation de datasets, mais sur des systèmes dynamiques en interaction.

Modalités d'évaluation : Évaluation à partir du code déposé sur le git et du suivi au fil de l'eau par les encadrants

Compétences évaluées :

- Modélisation
- Recherche et Développement
- Développement
- Management

Responsable de cours : Julien Colin

Total : 14.0 h

TD : 14.0 h

SPM-HEP-011

[retour](#)

TD :

1. tbd (14.0 h)

CONTROVERSE

Responsable de cours : Hervé Frezza-Buet, Damien Rontani

Total : 18.0 h

TD : 18.0 h

SPM-HEP-018

[retour](#)

Description : L'objectif de ce cours est de contribuer à la formation citoyenne et critique des étudiants, en leur permettant de développer une capacité d'analyse des débats publics et de dialoguer de manière constructive face à des opinions divergentes.

Acquis d'apprentissage : À l'issue de ce cours les élèves sauront reconnaître les dimensions d'une controverse et seront familiers avec quelques techniques permettant d'avancer vers sa résolution.

TD :

1. tbd (18.0 h)

Description : Ce cours propose une introduction à l'ingénierie des systèmes complexes, en mettant l'accent sur les approches modernes de modélisation et de simulation. Il s'appuie sur la démarche MBSE (Model-Based Systems Engineering) pour structurer la conception des systèmes tout au long de leur cycle de vie. Les étudiants découvrent le langage SysML à travers un atelier permettant de formaliser certaines exigences, fonctions et structures. Le cours explore également des paradigmes complémentaires : la modélisation multi-agents (avec NetLogo) pour représenter des systèmes à comportements émergents, et la modélisation de systèmes cyber-physiques (avec OpenModelica) pour simuler des composants physiques et leur interaction avec des systèmes de contrôle. Une introduction à la norme FMI permet de comprendre les enjeux de co-simulation entre modèles hétérogènes. En complément, une présentation de l'AFIS éclaire le rôle de l'ingénieur système dans l'industrie française, et un témoignage professionnel permet d'illustrer les usages concrets des jumeaux numériques.

Acquis d'apprentissage : À l'issue de ce cours, les élèves seront capables de formuler les questions pertinentes pour la conception d'un système, en particulier sur les aspects de modélisation, et d'en esquisser des solutions possibles.

Modalités d'évaluation : QCM et étude de cas (1h30)

Compétences évaluées :

— Modélisation

CM :

1. Introduction à l'ingénierie système (définitions, historique, motivations, concepts système) (2.0 h)
2. MBSE (définitions, évolution, cycles de vie, interfaces, processus, exigences, V&V (2.0 h)
3. Introduction à SysML (1.0 h)
4. modélisation et simulation à base d'agents (1.0 h)
5. modélisation de systèmes cyber-physiques et co-simulation (1.0 h)
6. AFIS (2.0 h)
7. Jumeaux Numérique (2.0 h)

TP :

1. familiarisation avec un logiciel de modélisation SysML, mise en évidence des liens entre diagrammes (3.0 h)
2. modélisation et simulation à base d'agents (3.0 h)
3. modélisation de systèmes cyber-physiques et co-simulation (3.0 h)

ÉDUCATION PHYSIQUE ET SPORTIVE S08

Responsable de cours : Hervé Frezza-Buet

Total : 21.0 h

TD : 21.0 h

SportS08

[retour](#)

Description : Au-delà du développement des compétences motrices (physiques, techniques et tactiques) les enseignements d'éducation physique et sportive ont pour objectifs de permettre aux étudiants de développer leurs compétences personnelles de connaissance et de contrôle de soi mais aussi leurs compétences relationnelles (collaboration au sein de l'équipe, écoute, communication, animation, ...).

TD :

1. Cours de sport (21.0 h)

Responsable de cours : Hervé Frezza-Buet, Damien Rontani

Description : Stage en entreprise, entre la deuxième et la troisième année, d’une durée minimum de 12 semaines.

Ce stage a pour objectif de mettre les étudiants dans une situation d’ingénieur-assistant afin qu’ils approfondissent leurs connaissances et compétences professionnelles. L’étudiant doit être intégré à une équipe et placé sous la responsabilité d’un ingénieur. Il doit participer aux activités de l’équipe à un niveau d’initiative et d’autonomie cohérente avec son niveau de formation.

Lors de son immersion, l’étudiant doit à la fois répondre aux enjeux techniques de leur travail et prendre le recul nécessaire sur les dimensions méthodologiques, éthiques et organisationnelles de son travail.

Modalités d’évaluation : Evaluation PASS/FAIL sur la base de la remise d’un rapport de stage. L’évaluation prendra en compte ces deux dimensions et les capacités d’intégration, de travail en équipe et de communication de l’étudiant.

Compétences évaluées :

- Modélisation
- Recherche et Développement
- Développement
- Certification
- Système
- Conseil
- Business Intelligence
- Management

Responsable de cours : Elisabeth Leuba

Total : 21.0 h

TD : 21.0 h

LV1S08

[retour](#)

Description : La Langue Vivante 1 sera généralement l'anglais. Répartis par niveau, les étudiants travailleront non seulement les 4 compétences langagières mais aborderont aussi des sujets variés qu'ils approfondiront suivant leur niveau. Les sujets traités peuvent être d'ordre civilisationnel, sociétal, professionnel, etc. Les effectifs des groupes constitueront un environnement propice à une participation active et une progression conséquente dans la langue. Différentes méthodes pédagogiques seront utilisées : travail en groupe, exposés, exercices spécifiques, recherche, débats, etc.

Acquis d'apprentissage : À l'issue de ce cours, l'élève aura progressé pour communiquer dans un environnement universitaire, professionnel ou personnel internationalisé

Compétences évaluées :

- Recherche et Développement
- Conseil
- Business Intelligence
- Management

TD :

1. Cours (21.0 h)

Responsable de cours : Beate Mansanti

Total : 21.0 h

TD : 21.0 h

LV2S08

[retour](#)

Description : En Langue Vivante 2, une offre de plusieurs langues sera proposée aux étudiants, en poursuite d'étude ou en débutant. Répartis par niveau, les étudiants travailleront non seulement les 4 compétences langagières mais aborderont aussi des sujets variés qu'ils approfondiront suivant leur niveau. Les sujets traités peuvent être d'ordre civilisationnel, sociétal, professionnel, etc. Les effectifs des groupes constitueront un environnement propice à une participation active et une progression conséquente dans la langue. Différentes méthodes pédagogiques seront utilisées : travail en groupe, exposés, exercices spécifiques, recherche, débats, etc.

Acquis d'apprentissage : À l'issue de ce cours, l'élève aura progressé pour communiquer dans un environnement universitaire, professionnel ou personnel internationalisé

Compétences évaluées :

- Recherche et Développement
- Conseil
- Business Intelligence
- Management

TD :

1. Cours (21.0 h)