

# Outils numériques pour l'ingénieur (DATA610\_MIMC)



## En bref

- > **Langues d'enseignement:** Français
- > **Méthodes d'enseignement:** En présence
- > **Forme d'enseignement :** Projet personnel et professionnel
- > **Ouvert aux étudiants en échange:** Oui

## Présentation

### Description

Ce cours a pour objectif de rendre les élèves ingénieurs autonomes dans le choix et l'utilisation d'outils numériques pour résoudre des problèmes concrets d'ingénierie. Il met l'accent sur le développement de compétences pratiques en programmation scientifique, en analyse de données et en modélisation numérique. Le langage Python, libre et largement utilisé dans le monde de la recherche et de l'industrie, constitue l'outil principal du cours. À travers des études de cas et des projets, les étudiants apprennent à structurer un code efficace, à exploiter des bibliothèques spécialisées et à intégrer des méthodes numériques dans une démarche d'ingénierie.

### Objectifs

Ce cours a pour objectif de former des ingénieurs capables de mobiliser des outils numériques avancés pour analyser, modéliser et résoudre des problèmes concrets en ingénierie, en particulier dans le domaine de la **mécatronique**, où l'intégration de la mécanique, de l'électronique et de l'informatique est essentielle.

À l'issue de la formation, les étudiants seront en mesure de :

- **Utiliser le langage Python et ses bibliothèques scientifiques** pour développer des solutions logicielles robustes, reproductibles et adaptées aux problématiques mécatroniques.
- **Traiter, visualiser et exploiter des données issues de capteurs**, d'images ou de simulations, dans des contextes tels que le contrôle de systèmes, la vision embarquée ou la surveillance d'état.
- **Modéliser le comportement dynamique de systèmes mécatroniques** à l'aide d'équations différentielles ordinaires (ODE), en simuler le comportement et en interpréter les résultats pour l'analyse ou la commande.
- **Appliquer des techniques d'optimisation** dans des cas d'usage liés à la conception de structures, à l'ajustement de paramètres ou à l'amélioration des performances de systèmes complexes.
- **Explorer et mettre en œuvre des méthodes de machine learning** pour le traitement de données mécatroniques, par exemple pour la reconnaissance d'images, la détection d'anomalies ou l'estimation de paramètres en ligne.
- **Travailler en groupe sur un projet intégrateur**, mobilisant des compétences en programmation, modélisation et traitement numérique dans le cadre d'une problématique d'ingénierie mécatronique (système embarqué, chaîne de mesure, robotique, etc.).

Ce cours vise ainsi à développer chez les étudiants une **approche systémique et numérique** des systèmes mécatroniques, en les préparant à des environnements industriels où la simulation, l'analyse de données et l'automatisation jouent un rôle central.

---

## Heures d'enseignement

CM	Cours Magistral	7,5h
TD	Travaux Dirigés	7,5h
TP	Travaux Pratiques	20h

---

## Pré-requis obligatoires

Aucun.

---

## Plan du cours

Le cours est structuré en six modules thématiques, conçus pour développer progressivement les compétences des élèves ingénieurs en programmation scientifique, traitement numérique et modélisation appliquée :

### Bloc 1 – Traitement d'images :

Ce module introduit les notions fondamentales des images matricielles (raster) et les principales techniques de traitement d'images. Les étudiants mettent en œuvre des opérations telles que le filtrage, la détection de contours ou l'analyse morphologique à l'aide des bibliothèques *SciPy*, *Scikit-Image* et *OpenCV*.

### Bloc 2 – Traitement de données :

Les élèves découvrent les outils de manipulation, de structuration et d'analyse de données avec la bibliothèque *Pandas*. Ce bloc leur permet de traiter efficacement des jeux de données issus de capteurs, d'expériences ou de fichiers externes, en vue d'analyses statistiques ou de visualisation.

### Bloc 3 – Équations différentielles ordinaires (ODE) :

Ce module aborde la modélisation de phénomènes dynamiques à l'aide d'équations différentielles ordinaires. Les étudiants apprennent à formuler des modèles ODE, à les résoudre numériquement (par exemple avec les méthodes de Runge-Kutta) et à interpréter les résultats dans un contexte physique ou technique.

#### **Bloc 4 – Optimisation :**

Ce bloc explore les techniques d'optimisation, essentielles en ingénierie pour l'ajustement de modèles, la calibration de paramètres ou la conception de systèmes. Les étudiants apprennent à poser un problème d'optimisation et à utiliser les outils numériques pour le résoudre efficacement.

#### **Bloc 5 – Apprentissage automatique :**

Introduction progressive aux méthodes de machine learning, en commençant par les algorithmes supervisés classiques (comme les *k-plus proches voisins*) jusqu'aux premières notions de réseaux de neurones. Les bibliothèques *Scikit-Learn* et *PyTorch* sont utilisées pour expérimenter sur des données réelles.

#### **Bloc 6 – Projet de groupe :**

Les étudiants forment des groupes pour concevoir et réaliser un projet intégrant les compétences développées au fil du cours. Le projet repose sur la résolution d'un problème d'ingénierie nécessitant l'analyse et le traitement de données numériques. Il donne lieu à une évaluation basée sur la qualité du travail, la rigueur de la démarche et la restitution finale.

---

## Compétences visées

À l'issue du cours, les étudiants auront acquis les compétences suivantes :

#### **Compétences techniques et scientifiques:**

- **Maîtriser l'environnement Python scientifique** et ses bibliothèques (NumPy, SciPy, Pandas, Matplotlib, etc.) pour le développement d'outils d'analyse numérique.
- **Concevoir et mettre en œuvre des chaînes de traitement de données** issues de capteurs, d'images ou de simulations numériques.
- **Résoudre numériquement des équations différentielles ordinaires (ODE)** pour modéliser le comportement dynamique de systèmes physiques ou mécatroniques.
- **Mettre en œuvre des techniques d'optimisation numérique**, notamment pour la calibration de modèles, l'ajustement de courbes ou la conception assistée.
- **Utiliser des outils de machine learning supervisé** (Scikit-Learn, PyTorch) dans des contextes liés à la mécatronique : classification, régression, détection de motifs ou d'anomalies.
- **Exploiter des bibliothèques spécialisées en traitement d'image** (OpenCV, Scikit-Image) pour des applications telles que la vision industrielle ou la détection de défauts.

#### **Compétences appliquées à la mécatronique:**

- **Intégrer des outils numériques dans une approche multidisciplinaire**, combinant mécanique, électronique et informatique pour l'analyse ou la commande de systèmes mécatroniques.
- **Analyser et interpréter des données expérimentales ou simulées** issues de systèmes embarqués ou de bancs de test.
- **Développer des scripts ou des prototypes numériques** pour soutenir la conception, la validation ou le diagnostic de systèmes mécatroniques.

#### **Compétences transversales:**

- **Structurer un raisonnement algorithmique clair**, réutilisable et documenté, dans le cadre de projets d'ingénierie.
- **Travailler en équipe dans le cadre d'un projet numérique**, de la définition du besoin à la restitution technique.
- **Communiquer les résultats de manière rigoureuse**, en s'appuyant sur des représentations graphiques pertinentes et des outils de présentation scientifique (Jupyter Notebooks, rapports, etc.).

---

## Bibliographie

L'ensemble des supports du cours sont disponibles sur notre site: <http://https://symmehub.github.io/positron/intro.html>

**Libellé court** : DATA610\_MIMC

**Nature** : MODULE

## Infos pratiques

---

### Lieux

› Le Bourget-du-Lac (73)