# Intégration de l’intelligence artificielle dans le cours 025747 – Algèbre et trigonométrie

Dans ces unités d’apprentissage, l'intelligence artificielle est plutôt simulée et expliquée à travers des concepts mathématiques de base. Il ne s'agit pas d'un déploiement ou d'une utilisation active d'IA, mais plutôt d'une introduction à certains principes utilisés dans les systèmes d'IA.

On aborde les concepts d'IA de manière pédagogique, mais sans déployer un système d'IA complet ou autonome.

## Unité d’apprentissage : Équations algébriques (025756)

### Module : Systèmes d’équations linéaires

• Niveau d’intégration IA : Modérée   
• Objectif pédagogique : Comprendre qu’un système de deux équations représente deux contraintes à satisfaire simultanément, comme dans la prise de décision automatisée.  
• Intérêt IA : Un système expert ou un agent intelligent peut rechercher une solution optimale qui respecte plusieurs contraintes, modélisées ici par deux équations linéaires.  
• Exemple : Un robot mobile doit atteindre un point précis tout en minimisant un coût. Les contraintes sont traduites par deux équations ; la solution devient une décision automatisée.  
• Activité : Étude de cas simulée (type IA symbolique), résolution avec GeoGebra ou Python (SymPy).  
• Outils proposés : GeoGebra, Python avec la bibliothèque SymPy.

## Unité d’apprentissage : Équations algébriques (025756)

### Module : Vecteurs

• Niveau d’intégration IA : Modérée   
• Objectif pédagogique : Montrer que les vecteurs sont utilisés pour représenter des objets (visages, images, textes) dans les systèmes IA.  
• Intérêt IA : Le produit scalaire de deux vecteurs permet d’évaluer la similarité qui est un concept de base en reconnaissance faciale ou dans les moteurs de recommandation.  
• Exemple 1 : Transformer une petite image en matrice (valeurs RGB) pour illustrer la structure numérique d’un fichier visuel.  
• Exemple 2 : Comparer deux vecteurs à l’aide du produit scalaire pour simuler une reconnaissance faciale simplifiée.  
• Exemple 3 : Utiliser playground.tensorflow.org pour observer comment un modèle IA apprend à classer en ajustant ses poids.  
• Activité : Atelier de manipulation vectorielle, simulation de similarité, interprétation visuelle avec outils interactifs.  
• Outils proposés : Python (NumPy), Excel, TensorFlow Playground.

# Intégration de l’intelligence artificielle dans le cours 025749 – Architecture de l’ordinateur

Dans ce cours, l’intelligence artificielle est abordée à travers l’analyse des capacités matérielles nécessaires à son fonctionnement. L’objectif n’est pas d’exécuter de l’IA, mais de comprendre l’infrastructure physique qui la rend possible : processeurs, mémoire, bus, architectures spécialisées (GPU Graphics Processing Unit, NPU Neural Processing Unit).

L'intégration de l'IA dans ce contexte se fait en expliquant comment différentes plateformes matérielles (Windows, Apple) sont utilisées pour entraîner ou exécuter des modèles d'IA, mais sans qu'une IA active ne soit directement impliquée dans les exercices.

## Unité d’apprentissage : Architecture d’un ordinateur

### Module : Composant d’un ordinateur

• Niveau d’intégration IA : Modérée

• Objectif pédagogique : Comprendre le rôle de chaque type de processeur dans les tâches liées à l’intelligence artificielle.

• Intérêt IA : L’IA moderne repose sur des traitements parallèles massifs (GPU) et des unités spécialisées (NPU) que les architectures classiques ne supportent pas toujours efficacement.

Les GPUs sont utilisés pour entraîner des **modèles d'IA** en raison de leur capacité à effectuer un grand nombre de calculs en parallèle, ce qui est essentiel dans le processus de formation des réseaux neuronaux.

Un NPU est conçu spécifiquement pour accélérer les **calculs liés aux réseaux neuronaux**. Contrairement aux GPU, qui peuvent être utilisés pour une large gamme de calculs parallèles, le NPU est spécialement optimisé pour les **tâches d'intelligence artificielle**, comme l'entraînement et l'inférence des modèles de machine learning et deep learning.

* Activités proposées :

- Comparaison de matériel IA : Étude de cas comparant l’efficacité de plateformes Windows (Intel/AMD + GPU Nvidia) et Apple Silicon (M1–M3 avec NPU).  
 - Tableau comparatif : Résumer les différences entre CPU, GPU et NPU dans un tableau (critères : rôle, performance, consommation, coût).  
 - Choix d’un matériel : Simulation d’un achat d’ordinateur pour IA. Les étudiants doivent choisir la meilleure configuration selon l’usage : entraînement (training) ou inférence (utilisation).

**Exemple concret**

- Windows avec GPU Nvidia → mieux adapté pour l'entraînement de modèles IA grâce à CUDA et Tensor Cores.  
 - Apple Silicon → mieux optimisé pour l'inférence locale avec CoreML et les NPU intégrés (Siri, reconnaissance faciale, etc.).

# Intégration de l’intelligence artificielle dans le cours 025742 – Introduction à la programmation

Dans cette unité d’apprentissage, l’intelligence artificielle est simulée par des systèmes de règles simples codés à l’aide de la logique conditionnelle. On n’utilise pas de modèle d’IA préentraîné, mais on introduit les principes fondamentaux de raisonnement automatisé, qui sont à la base des systèmes experts. Il s’agit donc d’une initiation pédagogique à l’IA symbolique, qui permet aux étudiants de comprendre les liens entre programmation modulaire et logique d’un système intelligent.

## Unité d’apprentissage : Programmation modulaire

* Module proposé: Notions de programmation modulaire : Décomposition du problème en sous-problèmes de complexité plus simple
* Niveau d’intégration IA : Modéré
* Objectif pédagogique : Apprendre à structurer un programme en modules tout en imitant le fonctionnement d’un système intelligent basé sur des règles et des modules. Monter que L’IA peut fonctionner avec des règles simples, sans modèle complexe et que la modularité du code est essentielle pour créer des systèmes experts évolutifs.
* Intérêt IA : Un système expert repose sur des fonctions logiques et des règles conditionnelles. On montre ici que la programmation modulaire constitue une base pour structurer un raisonnement automatisé.
* Exemple : Construction d’un assistant santé simple qui pose des questions à l’utilisateur et recommande une action (repos, médecin, etc.) selon les symptômes décrits en utilisant la structure conditionnelle if/else.
* Activité : Projet de mini-chatbot médical en console C# avec fonctions LireSymptomes(), Diagnostiquer(), AfficherResultat().

- Utiliser la logique conditionnelle (if/else/switch) pour simuler une prise de décision.  
- Structurer le raisonnement avec des fonctions, comme dans un système expert.  
- Concevoir un système qui réagit intelligemment à une situation donnée (simulation d’un raisonnement).

* Outils proposés : pas d’outils IA

# Intégration de l’intelligence artificielle dans le cours 025748 – Introduction à la gestion d’un système d’exploitation

Dans cette unité d’apprentissage, l’intelligence artificielle est utilisée de manière directe, soit en interrogeant une IA distante via une API, soit en exécutant un modèle localement. Cela permet aux étudiants de comprendre comment un système Linux peut devenir une interface vers ou un hôte pour des technologies d’IA.

## Unité d’apprentissage : Commandes intermédiaires (025759)

### Module proposé: Interrogation d’une API IA depuis le terminal

• Niveau d’intégration IA : Direct

• Objectif pédagogique : Comprendre comment un système Linux peut interagir avec un service d’intelligence artificielle via la ligne de commande curl.

• Intérêt IA : L’intelligence artificielle n’est pas exécutée sur l’ordinateur local, mais sur un serveur distant (par exemple chez OpenAI). Le terminal Linux devient alors une passerelle pour utiliser cette IA.

* Rôle de curl : curl n’est pas une IA, mais un outil en ligne de commande qui permet d’envoyer une requête (texte, question, commande…) à un service web comme une API d’IA et de recevoir une réponse. C’est donc le moyen par lequel on utilise l’IA, mais ce n’est pas l’IA elle-même.

**Exemple concret d’utilisation de curl**

Un script Bash utilise curl pour envoyer une question à l’API OpenAI. L’IA génère une réponse sur le serveur distant. curl affiche ou enregistre cette réponse localement.

* Activité proposée :

Écrire un script Bash qui :  
- lit une question dans un fichier texte,  
- l’envoie à l’API OpenAI avec curl,  
- et sauvegarde la réponse dans un autre fichier.

• Outils proposés : curl, Bash, API OpenAI (clé API nécessaire)

### Module Proposé : Exécution locale d’un modèle IA léger

• Niveau d’intégration IA : Direct

• Objectif pédagogique : Découvrir qu’un modèle d’IA peut être installé et exécuté localement, sans connexion Internet.

• Intérêt IA : Cela permet d’explorer une IA réellement hébergée sur le poste de travail, avec un traitement 100 % local.

• Outil utilisé : llama.cpp, Large Language Model Meta AI) est une **famille de modèles d’intelligence artificielle** développée par **Meta (anciennement Facebook)**. Il s'agit de modèles de **traitement du langage naturel** comparables à ceux d’OpenAI (comme GPT), capables de répondre à des questions, résumer des textes, traduire, générer du code ou du contenu etc., sans connexion Internet.

* Activité proposée:

- Installer llama.cpp sur Linux  
- Lancer une session locale  
- Poser une question au modèle et analyser la réponse générée

• Outils proposés : llama.cpp, terminal Linux, ressources système locales.

**Intégration réelle de l’intelligence artificielle dans le cours 030832 – Réseau pour programmeur**

Dans un contexte avancé, il est possible d’utiliser de vraies technologies d’IA pour détecter des anomalies réseau à partir de modèles d’apprentissage (comme dans les outils SIEM, ou avec Python + Scikit-learn). Cependant, cela dépasse largement le cadre de ce cours d’introduction. L’objectif ici est d’initier les étudiants à une logique de prise de décision automatique, sans apprentissage automatique réel.

## Unité d’apprentissage : Configuration de serveurs

### Module : Analyse des erreurs de connexion et résolution

* Niveau d’intégration IA : modérée (simulation de l’IA)
* Objectif pédagogique : Comprendre comment une application ou un script peut simuler une prise de décision autonome pour optimiser ou corriger une configuration réseau.
* Intérêt IA : Préparer les étudiants à des usages réels de l’IA dans des domaines réseaux : détection d’anomalies, surveillance intelligente, etc.
* Exemple : Assistant de diagnostic réseau (simulé); L’étudiant conçoit un script ou une application console qui :  
   - lit les paramètres d’une configuration (IP, DNS, port),  
   - détecte des erreurs courantes (ping échoué, port fermé),  
   - propose une recommandation en fonction des erreurs détectées

Ce système n’utilise pas de vraie IA, mais simule un raisonnement automatique basé sur des règles conditionnelles (if/else).

* Activité : Création d’un assistant de diagnostic réseau (simulation IA symbolique)
* Outils proposés : pas d’outils IA

**Bash**, **PowerShell**, et **Python**

**Intégration réelle de l’intelligence artificielle dans le cours 030832 – Réseau pour programmeur**

Il est possible d’intégrer de véritables technologies d’intelligence artificielle pour détecter automatiquement des anomalies de connexion réseau. Cette approche fait appel à des algorithmes d’apprentissage automatique, qui apprennent à partir de données historiques à distinguer un comportement normal d’un comportement anormal.

**Unité d’apprentissage : Configuration de serveurs**

**Module proposé: Détection intelligente des anomalies réseau**

* Niveau d’intégration IA : Élevée (utilisation réelle de modèles d’apprentissage automatique)
* Objectif pédagogique : Mettre en œuvre un système capable de détecter automatiquement des comportements anormaux dans les connexions réseau à l’aide de l’IA.
* Intérêt IA : Illustrer un usage concret de l’IA dans les infrastructures réseau. Préparer les étudiants à l’analyse de données, à la cybersécurité et aux systèmes intelligents.

**Activité pédagogique :** Détection d’anomalies réseau à l’aide d’un modèle IA

Les étudiants :

* Collectent un petit jeu de données réseau: consiste à lire les journaux de connexion réseau (fichiers logs) et extraire des caractéristiques (adresse IP, heure, protocole, durée, etc.)
* Préparent les données (nettoyage, transformation et organisation des données réseau en Dataframe avec la bibliothèque Pandas)
* Entraînent un modèle d’IA comme KMeans et SVM pour détecter les anomalies en utilisant la bibliothèque Scikit-learn
* Visualisent les données pour afficher la fréquence des erreurs et les anomalies détectées avec la bibliothèque Matplotlib
* Évaluent la précision de la détection sur de nouvelles connexions

**Outils proposés**

* **Langage :** Python
* **Bibliothèques :** Scikit-learn, Pandas, Matplotlib
* **Environnement :** Jupyter Notebook