

# Simulation par IA du procédé LPBF de fabrication additive et optimisation du laser

Martins Hugo, Milan William

Stage de 3ème année de BUT Informatique effectué à l'I2M de Bordeaux

June 10, 2025

# Sommaire

- 1 Projet, contexte et objectif
- 2 IA trajectoire
- 3 Analyse des données
- 4 Thermique
- 5 Boucle d'apprentissage
- 6 Conclusion

# Présentation de l'I2M

L'I2M est un laboratoire de l'Université de Bordeaux dont la thématique porte sur les sciences mécaniques. On notera aussi les choses suivantes :

- Le laboratoire est divisé en 6 domaines différents, nous avons intégré celui intitulé MPI (Matériaux, procédés et interactions).
- Il est proche de l'IUT, situé à environ 2.5km. On pouvait donc facilement y aller avec divers moyens de transports.

## Contexte du projet : Fabrication additive

La fabrication additive, ou impression 3D, est la création de pièces par ajouts successifs de couches de matière. C'est présent dans de nombreux domaines, telle la médecine avec la création de prothèses, ou le secteur aéronautique avec la création de pièces de fusées.



# Contexte du projet : Problème

Actuellement, la fabrication additive a des limites :

- Les trajectoires et vitesses du laser sont prédéfinies, ou sont tout du moins impossibles à changer en temps réel.
- Les pièces ont parfois des défauts de porosité si certaines couches ont trop été chauffées, dû à l'aspect thermique du laser.

On voudrait pouvoir :

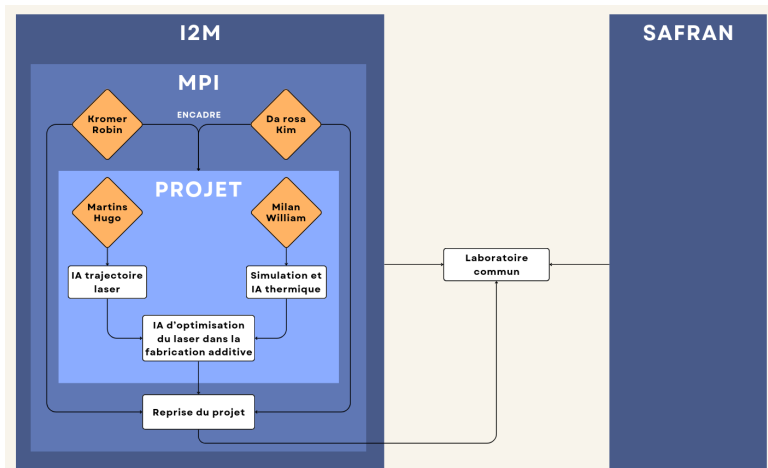
- Faire des simulations de fabrications de pièce et pouvoir discerner les meilleurs choix à l'aide d'IA.
- Ajuster la trajectoire et vitesse du laser de manière automatisée et en temps réel, afin de créer des pièces sans défauts.

## Contexte du projet : Objectif

Pour répondre au mieux à ce problème, l'objectif s'est concrétisé par :

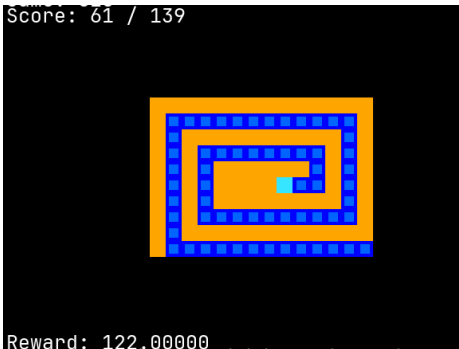
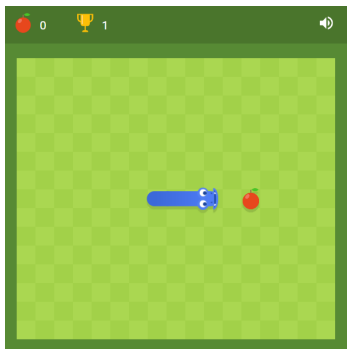
- Développer une IA basé sur l'Apprentissage par Renforcement pour imiter et simuler l'impression laser simplement.
- Développer un programme simulant l'aspect thermique de LPBF et une IA permettant de régler la puissance du laser.
- Améliorer l'IA sur la trajectoire du laser avec la répartition de chaleur pour éviter les problèmes et avoir une IA plus proche de la réalité.

# Intégration dans le projet



## Bases du travail : environnement Snake

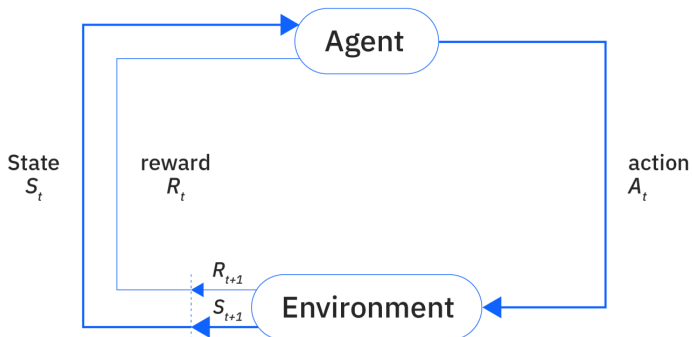
Pour représenter le laser et les trajectoires à faire, une implémentation similaire au jeu **Snake** a été choisie. Le laser est le serpent, les formes sont les fruits. Ceux-ci apparaissent simultanément pour faire la forme, et le but est de réussir à tous les parcourir.





# Bases du travail : Apprentissage par renforcement

L'Apprentissage par Renforcement est une technique d'apprentissage automatique, dans laquelle l'IA interagit avec l'environnement. En connaissant l'état de l'environnement et en obtenant des récompenses en fonctions des actions, l'IA est capable de faire des choix optimaux.



# Bases du travail : État et récompense

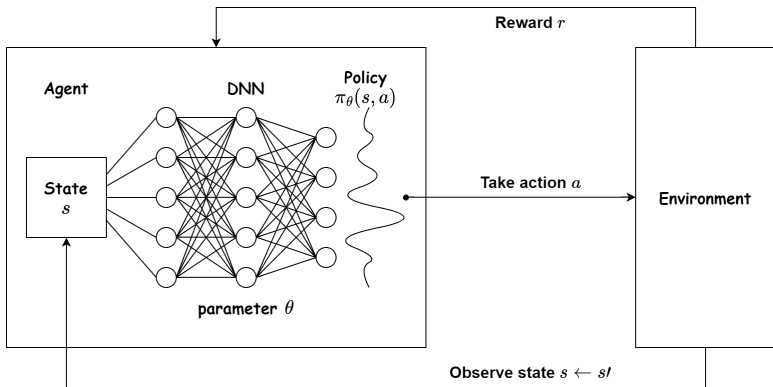
La fonction de récompense et l'état sont les principes majeurs en apprentissage par renforcement. Ici, c'est assez intuitif :

- 1 La récompense est ce que l'on obtient à la fin d'une action. Si le serpent mange une pomme, l'IA obtient une récompense positive; si il ne mange rien, la récompense est nulle.
- 2 L'état quant à lui correspond aux informations que l'IA possède. Cela peut être la direction du serpent, sa vitesse, la direction vers le fruit, etc.

Il faudra adapter ces rewards au fur et à mesure que l'environnement évolue, comme par exemple choisir une reward pour inciter l'IA à accélérer.

## Bases du travail : Algorithme DQN

L'algorithme DQN est un algorithme d'apprentissage par renforcement profond basé sur le Q-learning. C'est comme son évolution, adapté à des problèmes plus complexes.



# Objectif de l'IA trajectoire laser

Le but à atteindre est la création d'une IA capable d'adapter la vitesse et trajectoire du laser par rapport aux formes à imprimer. Pour au mieux simuler l'impression laser, le serpent devra pouvoir :

- Accélérer et ralentir
- Tourner à différents angles
- Simuler un dérapage en déviant de sa trajectoire si il va trop vite et tourne brusquement

On adaptera donc les récompenses, états et actions afin que le serpent puisse faire cela et apprenne correctement.

# Rewards pour atteindre l'objectif

Afin que l'IA puisse comprendre ces concepts, j'ai ajouter 2 nouvelles rewards.

- 1 Reward de vitesse : Plus le serpent va vite, plus les récompenses qu'il gagne sont importantes.
- 2 Reward de vitesse de sécurité : Si le serpent respecte la vitesse de sécurité, il obtient une reward. Dans le cas contraire, on lui donne une pénalité.

# Benchmark

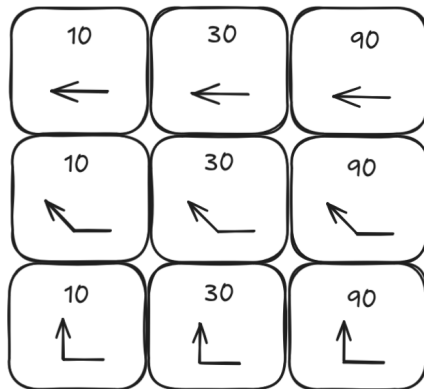
Dans le but d'améliorer l'IA et de pouvoir expliquer son comportement, j'ai réalisé un benchmark afin de comparer les performances de l'IA sur un test standardisé. Dans notre cas :

- Les métriques de performance seront notamment le score et la vitesse.
- La comparaison portera sur les divers paramètres de l'IA.

# Benchmark

Les comparaisons se feront sur 9 cas de tests différents.

Concrètement, il s'agit de 3 directions et de 3 distances différentes, comme on peut le voir ci-dessous :



## Du benchmark au dataset

Pour pouvoir étudier les résultats du benchmark, on générera des logs. Le logging est comme la gestion d'un journal de bord; dans notre cas, on retrouvera les paramètres de l'IA, les métriques de performances (scores) et les tests (parcours). Exemple :

```
IA numéro 1; paramètres food 1.0 - collision 10.0 [...]  
; parcours 1; game 200; avg_score 27.93; avg_speed [...]
```

Ces logs nous permettent d'enregistrer toutes ces données mais pas de les comprendre, il faudra les analyser.



# Méthodes clés: Analyse des données

La visualisation de données sert à comprendre et analyser des motifs dans les données via des graphiques. C'est la transformation de données pures en informations intelligibles.

Country	Region	Sub-region	Least developed countries (LDC)	Africa sub-regions	Africa region	Total	Male	Female
Afghanistan	SA		LDC			65	74	56
Albania	ECA	EECA				99	99	100
Algeria	MENA			Northern Africa	All	97	98	97
Andorra	ECA	WE						
Angola	SSA	ESA	LDC	Southern Africa	All			
Anquilla	LAC							
Antigua and Barbuda	LAC							
Argentina	LAC					100	99	100
Armenia	ECA	EECA						
Australia	EAP							
Austria	ECA	WE						
Azerbaijan	ECA	EECA						
Bahamas	LAC							
Bahrain	MENA					100	100	99
Bangladesh	SA		LDC			93	92	95
Barbados	LAC							
Belarus	ECA	EECA				100	100	100
Belgium	ECA	WE						
Belize	LAC							
Benin	SSA	WCA	LDC	Western Africa	All	61	70	52
Bhutan	SA		LDC					
Bolivia (Plurinational State of)	LAC							
Bosnia and Herzegovina	ECA	EECA						
Botswana	SSA	ESA		Southern Africa	All			
Brazil	LAC					99	99	99

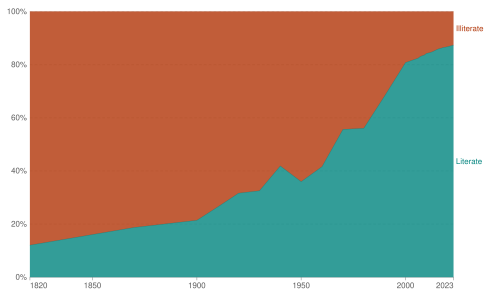
# Méthodes clé : Analyse des données

La visualisation de données sert à comprendre et analyser des motifs dans les données via des graphiques. C'est la transformation de données brutes en informations intelligibles.

## Literate and illiterate world population

The share of adults aged 15 and older who can read and write a short, simple statement on their everyday life.

Our World  
in Data



Data source: World Bank (2024); van Zanden, J. et al. (2014)

OurWorldinData.org/illiteracy | CC BY

# Méthodes clé : Analyse des données

Quand on étudie des données, il faut s'assurer que celles-ci soient correctes et proprement traités, donc il faut obligatoirement :

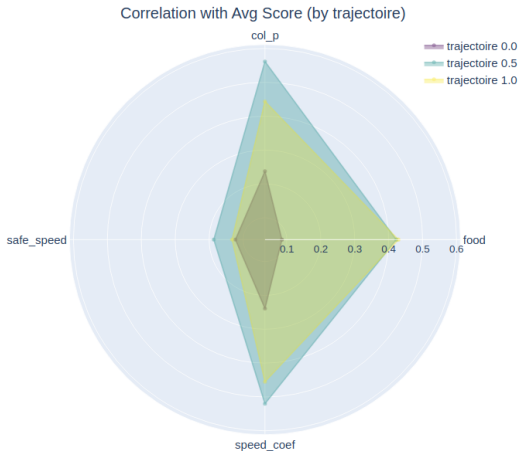
- 1 Nettoyer les données, c'est à dire gérer les données manquantes, convertir les valeurs en numériques (nombres), vérifier leur cohérence.

Potentiellement, il faudra aussi :

- 2 Gérer les données aberrantes si ce sont bien des erreurs ou si elles sont non nécessaires.
- 3 Standardiser ou normaliser les données si le graphique le demande.

# Étude des résultats : Radar des corrélations

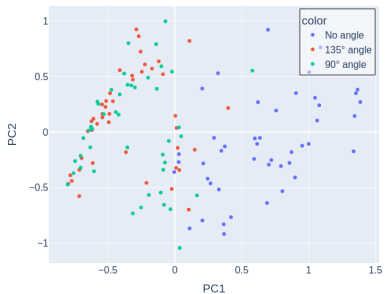
Pour commencer, j'ai cherché à connaître les relations entre résultats et paramètres :



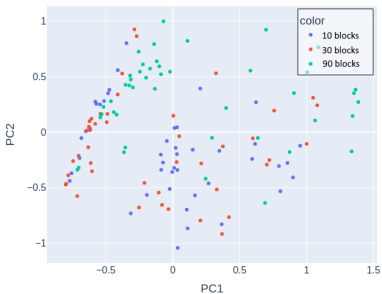
# Étude des résultats : PCA

Ayant remarqué que pour les trajectoires 1 et 2, l'importance des paramètres était similaire, j'ai voulu voir si ces similarités se retrouvaient avec des techniques de réduction de dimensionnalité.

PCA grouped by trajectoire



PCA grouped by distance



# Étude des résultats : Random Forest et Random Sampling

Les résultats sont similaires selon la trajectoire donc on cherchera à optimiser les paramètres du modèle pour les 2 types de trajectoire.

- Pour chaque type de trajectoire, on entraînerait un random forest model.
- Pour chaque random forest, on effectuerait un random sampling afin de chercher les meilleures combinaisons de paramètres.

On utiliserait donc les random forest comme des Surrogate Model de l'IA trajectoire.

# Résultats de l'analyse de données

L'analyse de données des logs du benchmark nous a permis de comprendre que les trajectoires étaient les aspects les plus compliqués pour l'IA.

- Tout d'abord, on sait que les paramètres sont similairement important pour les trajectoires 1 et 2.
- On aurait souhaité pouvoir déterminer plus précisément leurs valeurs avec la Random Forest, mais malheureusement les résultats n'étaient pas assez fiables pour être utilisables.

# Pourquoi s'intéresser à la thermique?

Température joue un rôle central : elle détermine la qualité de la pièce.

Défauts (porosité / manque de fusion) → mauvaise maîtrise de la puissance

Objectifs :

- comprendre ce qu'il se passe physiquement + mieux maîtriser le procédé
- développer un module thermique capable de simuler la température



## Découpage du travail

- Point de départ (existant, technos)
- Modèle thermique analytique de Eagar-Tsai
- Implémentation numérique sous Python (refactorisation, d'optimisation, tests (benchmarks))
- Intégration avec la boucle principale

# Point de départ

## Code existant au début du stage :

- Résultats peu réalistes physiquement
- Implémentation peu claire et peu efficace
- Fonctionnalités manquantes ou défectueuses

## Pourquoi Python ?

- Développement rapide, code lisible
- Écosystème riche : NumPy, SciPy, Matplotlib
- Idéal pour IA

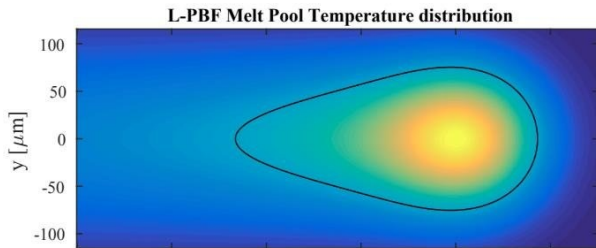
## Structure du programme :

- **Modèle thermique** : résolution des équations de diffusion
- **Code principal** : application du modèle aux trajectoires laser
- **IA** : régulation dynamique de la puissance (abordée plus tard)

# Modèle physique : pourquoi et lequel ?

## Modèle utilisé : Eagar-Tsai

- Basé sur le modèle de Rosenthal.
- Prend en compte la forme réelle du faisceau laser (plus intense au centre).
- Considère les effets de surface dans la dissipation thermique.



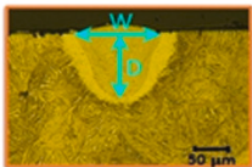
# Le bain de fusion : un indicateur clé

## Définition du meltpool (bain de fusion)

- Zone localement fondue sous l'action du laser.
- Formée lorsque la température dépasse le point de fusion de la poudre métallique.
- Se solidifie rapidement après passage du laser.

## Pourquoi c'est important ?

- Trop petit : fusion incomplète, risques de défauts.
- Trop grand : surchauffe, déformations.
- Sert d'indicateur de qualité thermique dans la simulation.



# Le modèle thermique

- Tous les calculs de température passent par ce modèle.
- Code initial fonctionnel mais problématique :
  - Présence de code mort.
  - Fonctions dupliquées.
  - Variables obsolètes.
  - Sections commentées sans explication.
- Ces défauts nuisent à la compréhension et à la réutilisation.

# Refactorisation orientée objet

- Introduction d'une **classe abstraite** regroupant méthodes et attributs communs.
- La méthode `solve()` reste à implémenter individuellement, car dépendante de la position de la source.
- Respect des principes de polymorphisme et factorisation en POO.
- Résultat : réduction significative des duplications et meilleure maintenabilité.

# Principe de la vectorisation

**Sans vectorisation :**

$$\begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} + \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix} = \begin{pmatrix} 1+3 \\ 2+4 \\ 0+5 \end{pmatrix} = \begin{pmatrix} 4 \\ 6 \\ 5 \end{pmatrix}$$

**Avec vectorisation :**

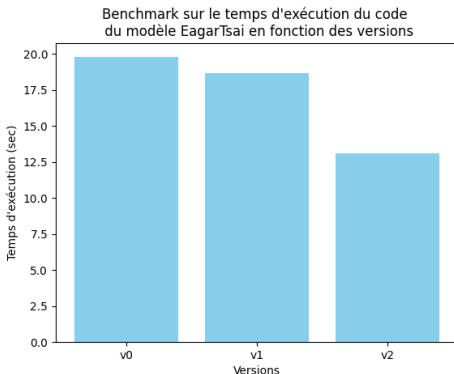
$$(1 \ 2 \ 0) + (3 \ 4 \ 5) = (4 \ 6 \ 5)$$

⇒ Opérations faites directement élément par élément, en une seule instruction. Possible en python grâce à numpy.

# Test de performances

Comparaison des performances entre les différentes versions du code : (centaine de simulations linéaires)

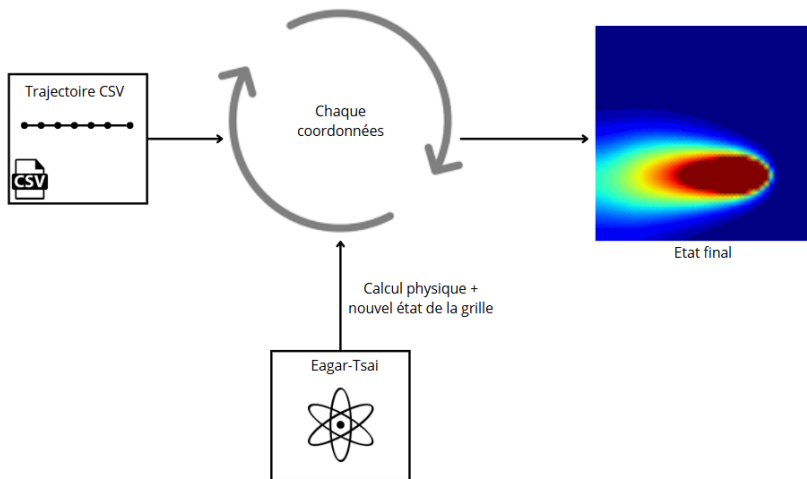
- v0 : version initiale
- v1 : version refactorisée
- v2 : version optimisée avec vectorisation





# Fonctionnement du programme

Fonctionnement général du programme de simulation thermique :



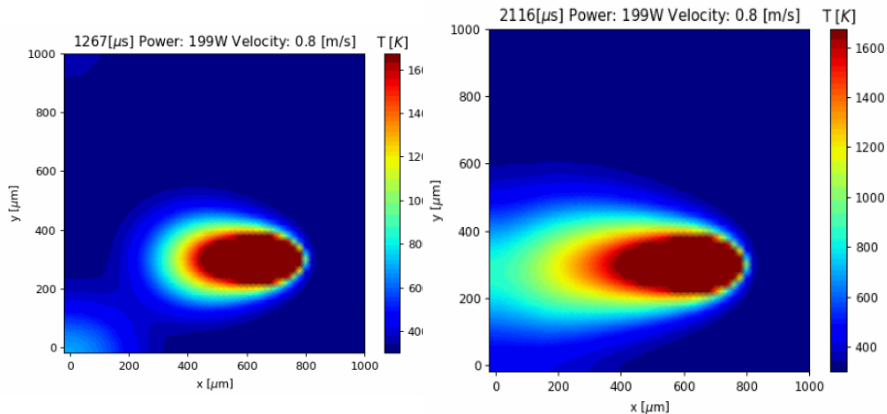
# Validation du comportement physique

Objectif : vérifier la cohérence physique pendant les simulations quelle que soit la trajectoire.

Problèmes rencontrés :

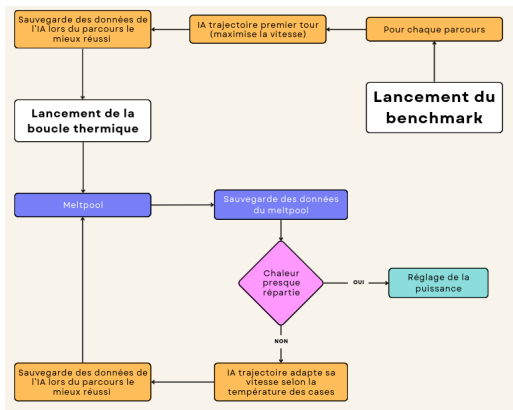
- Problème d'unité avec la trajectoire : fonction de normalisation
- Absence de calcul de vitesse et d'angle
- Erreur de positionnement initial

# Résultats



# Aller plus loin avec la thermique

En combinant mon travail sur la trajectoire avec celui de William sur la partie thermique, nous pourrions avoir une IA réaliste qui prend en compte les aspects réels de la fabrication additive.



## Comment l'IA s'améliore

Comme vu dans le schéma précédent, on remarque que nos programmes interagissent. En effet, on peut le résumer comme:

- l'IA trajectoire fait le parcours, et retient la partie et la trajectoire qui a donnée les meilleurs résultats.
- Le programme de simulation thermique réalise la simulation concernant la partie et trajectoire retenue par l'IA précédente, et retient à son tour les résultats de la simulation thermique.

Une fois qu'on a les résultats thermiques, on entraînera de nouveau l'IA trajectoire en ajoutant les informations afin de s'adapter par rapport aux données thermiques, et ainsi de suite.

# Intégration

Intégration au code principal s'est fait de manière assez fluide

→ Adapter le code à la structure globale

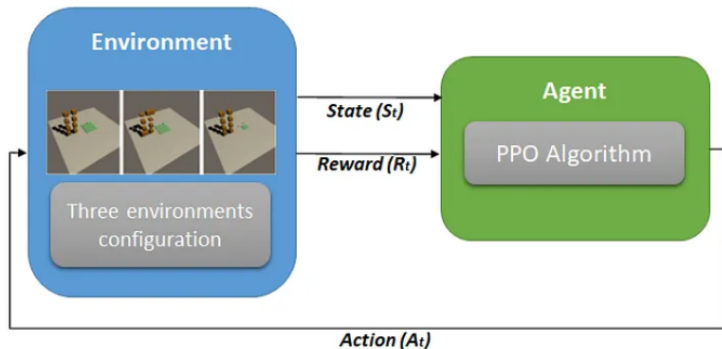
Restructuration en 2 parties :

- Classe dédiée à l'application thermique pour juste un chauffage à 150W
- Classe qui reprend la structure d'un environnement compatible avec Gym pour l'IA thermique

# Proximal Policy Optimization (PPO)

Contrairement à l'IA sur la trajectoire avec DQN, j'ai travaillé sur une IA avec PPO. IA dite policy-based, l'agent apprend directement une politique.

PPO repose sur une logique d'essai-erreurs : maximise sa fonction de récompense au fur et à mesure

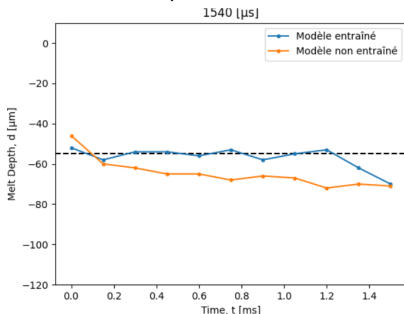


## Calcul de la récompense

Pour la récompense, il faut prendre en compte plusieurs éléments :

- Profondeur de bain de fusion : on cherche à chauffer chaque points à environ  $55\mu\text{m}$
- Stabilité de la puissance : Pénalité si changement brutal de puissance entre 2 coordonnées

$$R_{totale} = R_{depth} - \alpha * R_{Puissance}$$

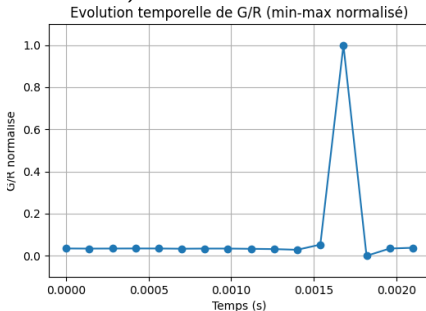




## Pour aller plus loin avec la thermique

On pourrait ajouter dans la récompense une nouvelle métrique : le ratio  $G/R$

- $G$  : Gradient thermique (vitesse de variation de la température dans la matière)
- $R$  : Taux de refroidissement (vitesse à laquelle le matériau revient à une température fixe).



# Conclusion

Compétences mobilisées :

- Compétence 1 - Réaliser un développement d'application :  
Refactoriser un code existant avec POO
- Compétence 2 - Optimiser des applications informatiques :  
Vectorisation avec Numpy + développement d'une IA PPO
- Compétence 6 - Travailler en équipe : Travail d'équipe pour la phase d'intégration

# Conclusion

Matrice SWOT :

- Forces : Acquis du BUT (programmation orientée objet) pour refactoriser du code.
- Faiblesses : Progrès sur le plan humain mais pas suffisants
- Opportunités : Découvrir des notions que je n'avais vues qu'en théorie (IA RL)
- Menaces : Aspects scientifiques de l'environnement.

# Conclusion

- L'IA principale est capable de prendre divers aspects du procédé LPBF, suivant les trajectoires à effectuer en maximisant la vitesse, désormais tout en cherchant à s'adapter en fonction de la température.
- On a aussi pu développer une IA qui a pour but d'optimiser la puissance du laser afin de faire les derniers ajustements pour obtenir une bonne impression.

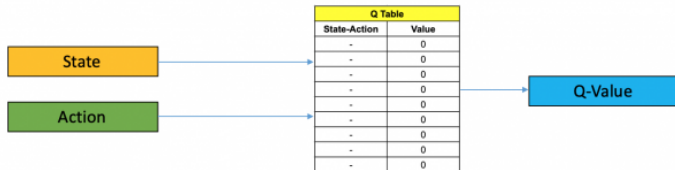
# Possibles améliorations

- Améliorer l'environnement avec des valeurs plus précises, comme plus de vitesses à disposition pour avoir un meilleur champ d'actions, par exemple avec des nombres flottants.
- Améliorer le modèle directement comme passer de DQN à DDQN par exemple.
- Améliorer la boucle d'apprentissage en utilisant dès le départ la thermique et en ajustant d'une meilleure manière l'apprentissage.

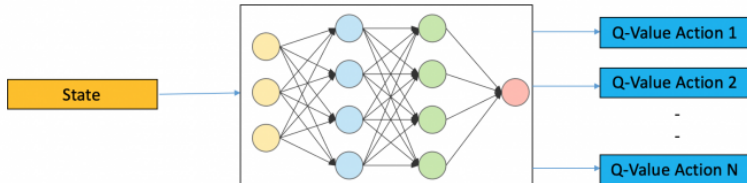
# Merci pour votre écoute

N'hésitez pas à nous poser des questions si vous en avez !

# Deep Q-learning vs Q-learning



Q Learning



Deep Q Learning

## Deep Q-learning vs Q-learning

Avec une grille de 100x100 cases, soit 10000 cases au total, sachant que chaque case a 3 états possibles, et en utilisant les règles de base (3 actions différentes). Si on veut ces informations dans l'état, cela donne :

- Avec Q-learning :  $3^{10000} * 3$  lignes pour notre Q-table, un résultat trop grand pour certaines calculatrices.
- Avec Deep Q-learning : couche d'entrée \* couche intermédiaire + couche intermédiaire \* couche de sortie. Avec une couche intermédiaire de 250 neurones, cela donne  $10000 * 250 + 250 * 3$ , ce qui est égal à 2 500 750.