

IM²AG - LICENCE 3 INFORMATIQUE :
MATHÉMATIQUES ET INFORMATIQUE

RAPPORT DE STAGE APPLICATIF

Caractérisation de filtres électroniques
simples à l'aide de PINNs (Physical
Informed Neural Networks) et étude du
compromis données / équations

Élèves :

Sahiya ABOUBACAR MOUSSA,
Abdel Kader EL HOUSSAMI,
Meryeme GHANNAM,
Kimia KHADEMLOU,
Maram MARSAOUI,
Ho Minh Khanh NGUYEN,
Hugo VIANA

Enseignant référent :

Marek BUCKI

Commanditaire :

Nathan CHANEZ,
STMicroelectronics

Table des matières

Introduction	3
1 Contexte	4
1.1 STMicroelectronics	4
1.1.1 Produits	4
1.1.2 Activité et clients	5
1.2 Environnement de projet	5
1.2.1 Les PINNs	5
1.2.2 Les filtres électroniques	7
2 Objectif	9
2.1 L'objectif principal	9
2.2 Contraintes et limites des méthodes traditionnelles	9
2.3 Pourquoi les PINNs ?	9
2.4 Optimiser la fonctionnalité de SPICE grâce aux PINNs	10
2.5 Livrables	10
3 Ressources du projet	11
3.1 Encadrants	11
3.2 Littérature scientifique	11
3.3 SPICE	11
3.4 Environnement de développement : Python	11
3.4.1 Bibliothèques spécifiques	11
3.5 Données	12
3.6 Machines	12
4 Mode de pilotage	13
5 Roadmap du projet	15
Glossaire	17
Annexe	18

Introduction

Ce rapport préliminaire a pour objectif de poser les bases et d'organiser la planification d'un projet visant à explorer l'utilisation des Physics-Informed Neural Networks (PINNs) pour la caractérisation de filtres électroniques simples. Ce projet se concentre sur la réduction de la dépendance aux données, en exploitant les lois physiques pour améliorer l'efficacité des prédictions. À travers une méthodologie agile et une roadmap claire, ce document détaille les étapes prévues, les outils à utiliser, et les objectifs à atteindre, garantissant une gestion structurée et efficace du projet avant son lancement.

1.1.2 Activité et clients

STMicroelectronics mène des activités diverses, allant de la recherche et développement à la conception, la fabrication et la commercialisation de semi-conducteurs. L'entreprise excelle dans le développement de solutions innovantes telles que les circuits intégrés spécifiques (ASICs), les capteurs MEMS, et les systèmes de gestion de puissance, tout en garantissant une production durable et de haute qualité. Elle collabore avec une clientèle variée, incluant des constructeurs automobiles (Tesla, Bosch), des fabricants d'électronique grand public (Apple, Samsung), des entreprises industrielles (Schneider Electric) et des acteurs des télécommunications (Huawei). Grâce à son expertise et à ses partenariats avec des startups, des universités et des grandes entreprises, STMicroelectronics répond aux besoins des secteurs technologiques les plus exigeants et consolide son leadership sur le marché mondial des semi-conducteurs.

1.2 Environnement de projet

1.2.1 Les PINNs

Notre projet se concentre sur l'expérimentation et la caractérisation des performances d'apprentissage de réseaux de neurones en exploitant les Physical Informed Neural Networks (PINNs). Ces réseaux intègrent des lois physiques, telles que les équations différentielles et les principes fondamentaux des circuits électriques, afin d'améliorer leur convergence et leur efficacité avec des données limitées.

Les réseaux de neurones sont des modèles informatiques inspirés du fonctionnement du cerveau humain. Ils sont constitués de couches de neurones interconnectés, où chaque neurone reçoit une entrée, effectue une transformation et transmet la sortie aux neurones suivants. Un réseau de neurones classique est divisé en trois parties : la couche d'entrée (input layer), une ou plusieurs couches cachées (hidden layers), et la couche de sortie (output layer). La couche d'entrée reçoit des données telles que des images, du texte ou des sons, qui sont ensuite traitées dans les couches cachées pour effectuer des tâches spécifiques comme la classification ou la prédiction. Lors de l'entraînement, le réseau ajuste ses "poids" et "biais" en utilisant des algorithmes d'optimisation comme la rétropropagation, afin de minimiser une fonction de perte qui mesure l'écart entre les résultats obtenus et les résultats attendus. Par exemple, ChatGPT est un grand réseau de neurones avec des milliards de paramètres.

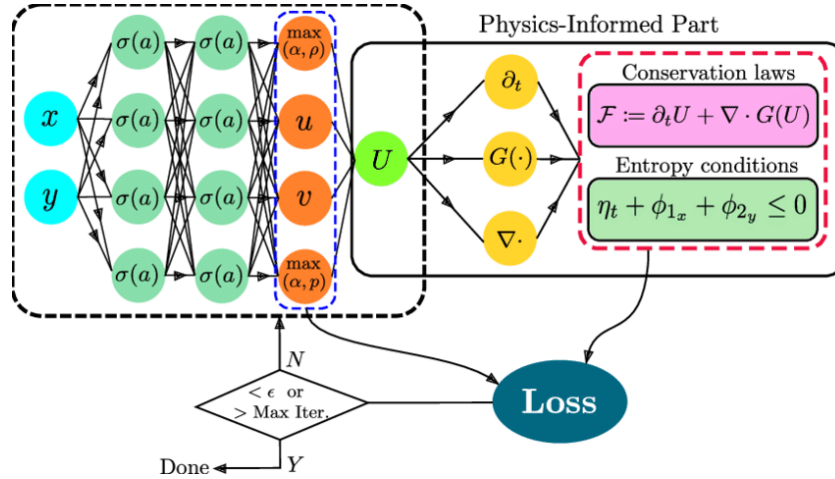


FIGURE 2 – Représentation schématique d'un PINNs pour les équations d'Euler

Les Physics-Informed Neural Networks (PINNs), quant à eux, sont une extension des réseaux de neurones classiques. Ils intègrent des lois physiques sous forme d'équations différentielles directement dans le processus d'apprentissage. Contrairement aux approches traditionnelles de Deep Learning, qui reposent principalement sur des données empiriques, les PINNs imposent des contraintes physiques en utilisant des termes de régularisation dans la fonction de perte. Cela permet d'améliorer la convergence du modèle et de résoudre des problèmes complexes, comme la simulation des fluides, la modélisation thermique, ou encore la dynamique des systèmes physiques, même lorsque les données disponibles sont limitées.

L'intégration des équations physiques dans les PINNs représente un pont entre deux modèles de modélisation : la modélisation basée sur les données (data-driven) et la modélisation basée sur les principes physiques (physics-driven). En exploitant la flexibilité des réseaux neuronaux pour approximer des solutions tout en respectant les lois physiques, les PINNs offrent une alternative puissante aux méthodes numériques traditionnelles, telles que la méthode des éléments finis. Cette approche hybride ouvre de nouvelles perspectives dans des domaines tels que la météorologie, l'ingénierie, ou encore la biologie, en permettant de résoudre des problèmes où les données sont rares ou difficiles à obtenir. Dans le cadre de cette approche, notre projet s'intéresse spécifiquement à l'application des PINNs à la caractérisation des filtres électroniques, qui jouent un rôle clé dans le traitement des signaux dans les circuits électriques.

1.2.2 Les filtres électroniques

Les filtres électriques admettent des composants électriques comme la résistance, l'inductance et le condensateur et parmi ces filtres on peut mentionner le filtre RC, RL ou RLC.

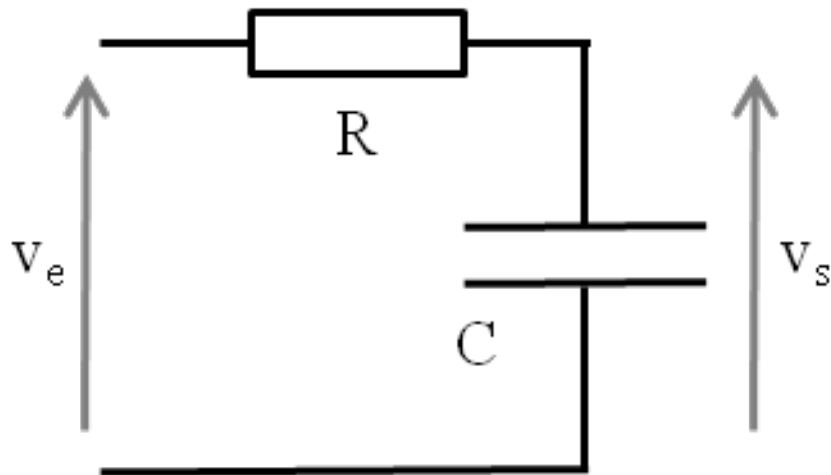


FIGURE 3 – Schéma d'un circuit RC

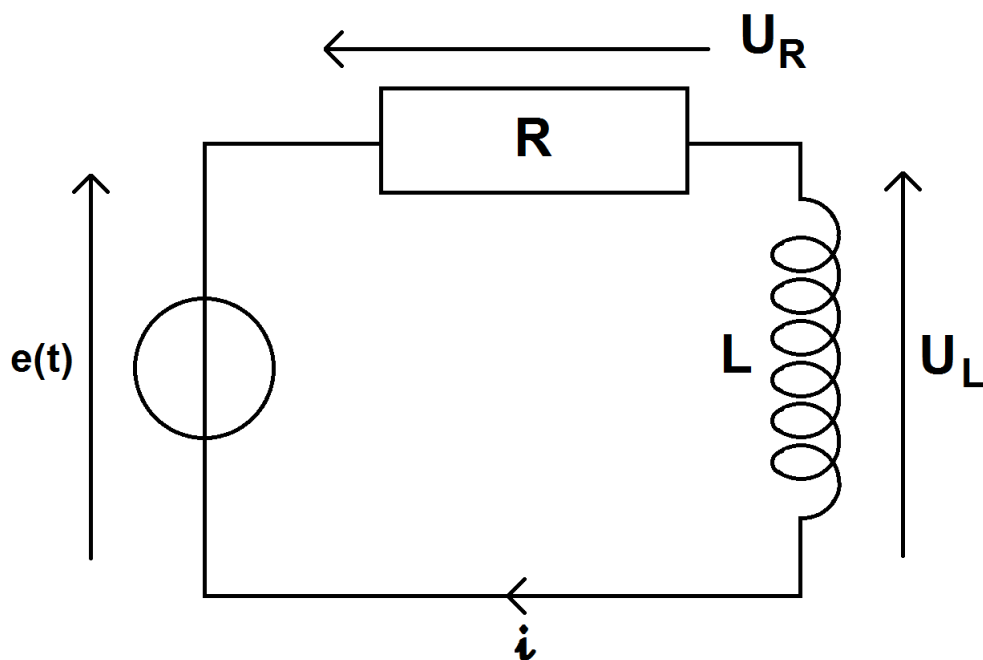


FIGURE 4 – Schéma d'un circuit RL

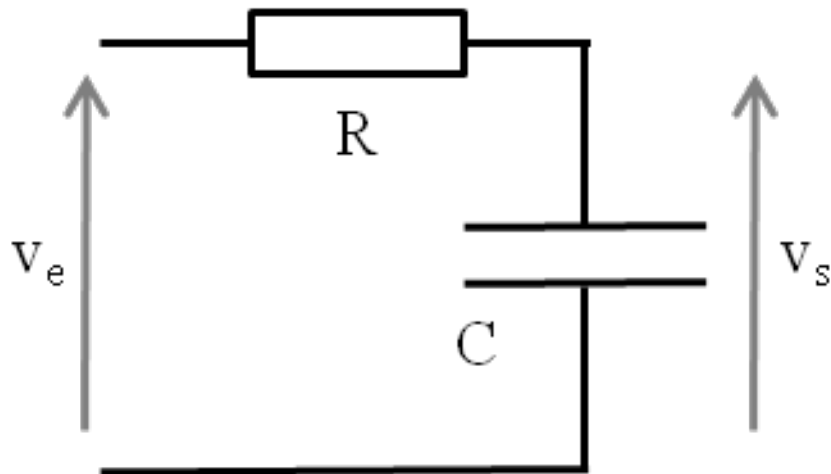


FIGURE 5 – Schéma d'un circuit RLC

Chaque circuit électrique peut se modéliser par des lois physiques telles que :

- Loi des mailles de Kirchhoff (KVL) : Elle impose que la somme des tensions dans une boucle est nulle.
- Loi des nœuds de Kirchhoff (KCL) : Elle stipule que la somme des courants entrant et sortant d'un nœud est nulle.
- Équations constitutives : Elles relient les variables des composants (résistances, condensateurs, inductances) avec des relations comme :
 - Loi d'Ohm : $V = R \cdot I$
 - Condensateur : $I = C \cdot \frac{dV}{dt}$
 - Inductance : $V = L \cdot \frac{dI}{dt}$

Ces relations donnent naissance à des équations différentielles qui décrivent le comportement dynamique des circuits en fonction du temps.

2 Objectif

2.1 L'objectif principal

L'objectif principal de ce projet est de remplacer les simulateurs traditionnels tels que SPICE par des réseaux de neurones capables de prédire efficacement les performances des circuits électriques (consommation, marge de gain, marge de phase, etc.). Les simulateurs actuels, bien que précis, souffrent de temps de calcul très longs et d'une complexité croissante lorsque le nombre de conducteurs dans les circuits modernes augmente.

2.2 Contraintes et limites des méthodes traditionnelles

Les méthodes traditionnelles comme SPICE présentent plusieurs contraintes majeures :

1. **Coût computationnel élevé** : Les simulations demandent beaucoup de temps et de ressources, en particulier pour les circuits complexes contenant un grand nombre de composants et de conducteurs. À mesure que la taille et la complexité des circuits augmentent, le temps de simulation augmente de manière exponentielle, rendant l'approche peu viable pour les grands systèmes.
2. **Nombre de simulations nécessaire** : Chaque variation de paramètre nécessite une nouvelle simulation, ce qui peut entraîner une multiplication du nombre de tâches et une lourdeur computationnelle.
3. **Divergence de la solution** : SPICE peut rencontrer des problèmes de convergence dans des circuits non linéaires complexes, en particulier lorsque les conditions initiales sont mal définies ou les équations différentielles présentent des discontinuités.
4. **Dépendance aux ressources matérielles** : L'exécution de simulations complexes nécessite des machines puissantes, ce qui augmente les coûts d'infrastructure.

Ces limitations rendent les simulateurs traditionnels inefficaces pour les besoins modernes d'optimisation rapide et précise des circuits électriques.

2.3 Pourquoi les PINNs ?

L'approche proposée combine des méthodes d'apprentissage automatique et des réseaux de neurones informés par la physique (PINNs) pour pallier ces limites. Les réseaux PINNs permettent d'intégrer directement des lois physiques (comme les équations différentielles ou les fonctions de transfert) dans les modèles d'apprentissage.

Cette combinaison est particulièrement adaptée pour :

- Réduire le nombre de simulations nécessaires pour optimiser les circuits. Améliorer l'efficacité des données en réduisant le besoin de grandes quantités d'exemples d'entraînement.
- Accélérer les prédictions des performances tout en maintenant une précision élevée. Tirer profit des connaissances physiques pour améliorer la convergence et l'efficacité des modèles.
- Optimiser les circuits électriques plus rapidement afin d'assister les ingénieurs dans leur travail sur des circuits modernes complexes.

En remplaçant les simulateurs traditionnels par des réseaux de neurones, ce projet vise à offrir aux ingénieurs un outil plus rapide, plus performant et mieux adapté aux circuits modernes complexes.

2.4 Optimiser la fonctionnalité de SPICE grâce aux PINNs

Les PINNs (Physics-Informed Neural Networks) peuvent améliorer SPICE en abordant ses limites :

1. **Simulation plus rapide** : Les PINNs réduisent le temps de calcul pour les circuits complexes en généralisant les solutions après entraînement.
2. **Meilleure convergence** : Ils résolvent plus facilement les circuits non linéaires en contournant les problèmes de divergence des méthodes classiques.
3. **Modèles avancés** : Les PINNs capturent mieux le comportement complexe des composants modernes.
4. **Analyse paramétrique efficace** : Une fois entraînés, ils prédisent rapidement le comportement du circuit avec différentes valeurs.
5. **Circuits multi-physiques** : Les PINNs gèrent simultanément les effets électriques, thermiques et mécaniques.

2.5 Livrables

1. **Création d'une base de données** : Génération de données d'entraînement et de validation via SPICE pour différents scénarios de circuits.
2. **Amélioration de l'efficacité des données** : Réduction du besoin de larges ensembles de données grâce à l'intégration de lois physiques dans les modèles.
3. **Capacité d'extrapolation** : Développement de réseaux capables de prédire avec précision au-delà des données fournies.
4. **Minimisation de la perte (loss)** : Atteinte d'une perte minimale pour garantir des prédictions précises et une convergence stable.
5. **Remplacement partiel des simulations SPICE** : Réseaux capables de prédire les performances des circuits plus rapidement que SPICE avec une précision comparable.

3 Ressources du projet

3.1 Encadrants

- Nathan Chanez : Ingénieur doctorant chez STMicroelectronics.
- Marek Bucki : CSO - Co-fondateur de TwInsight & Professeur à temps partiel à l'UGA

3.2 Littérature scientifique

Physics-informed Neural Networks : a simple tutorial with PyTorch, Theo Wolf¹

L'article propose une approche simple pour exploiter les réseaux de neurones en introduisant les contraintes physiques afin d'assurer l'apprentissage du réseau avec PyTorch.

L'article donne aussi l'exemple de la vitesse de refroidissement d'une tasse de café. Les résultats sont représentés sous forme d'une courbe. Sur cette courbe, on peut clairement visualiser les données fournies au réseau de neurone, tout comme la courbe qui représente les prédictions générées par l'IA.

3.3 SPICE

Spice, ou Simulation Program with Integrated Circuit Emphasis, est un logiciel libre de simulation de circuits électroniques. Il facilite la caractérisation des filtres électroniques qui sert principalement à filtrer le comportement temporel et fréquentiel en résolvant les équations différentielles qui gouvernent le circuit étudié.

Exemple de simulateur SPICE : NgSpice

Avantages :

- L'utilisation d'un simulateur permet la comparaison des résultats théoriques du filtre avec la version générée par le réseau de neurones.
- Spice génère des données spécifiques telles que tension , fréquence etc. et ces données peuvent servir à entraîner le réseau de neurones.

Limites : SPICE reste relativement coûteux en temps et en ressources.

3.4 Environnement de développement : Python

3.4.1 Bibliothèques spécifiques

- PySpice :

PySpice est une bibliothèque python qui permet d'interagir avec l'outil SPICE (Simulation Program with Integrated Circuit Emphasis) afin de modéliser, analyser et visualiser des circuits électroniques. PySpice permet d'abord de définir des composants électroniques, telles que les résistances, les inductances, les transistors et des générateurs. Ce qui nous permet d'éventuellement définir des circuits en entier.

1. Lien vers l'article

On peut également simuler le fonctionnement de ces circuits là sous plusieurs types de simulations, par exemple un circuit à courant continu ou à courant alternatif. Après la simulation du circuit, PySpice permet de visualiser les résultats obtenus. On peut même intégrer matplotlib afin de visualiser ces résultats en graphiques et des courbes qui serviront ensuite dans l'analyse du circuit. PySpice agit aussi en tant qu'une interface de simulateurs SPICE et rentre dans l'analyse des filtres électroniques.

— **PyTorch** :

Pytorch est une bibliothèque Python open source pour le Machine learning développée par Meta. Elle permet d'effectuer des calculs complexes et précis nécessaires pour le deep learning.

3.5 Données

L'une des ressources les plus importantes nécessaire pour un réseau de neurones, c'est la base de données qui sera utilisée dans l'entraînement de notre réseau. Dans le contexte de ce projet, on génère notre propre base de données. Pour cela, on réalise des simulations de circuits à travers l'outil SPICE. Les données récupérées à la fin seront celles fournies aux réseaux de neurones.

3.6 Machines

- **Mesures physiques** : Machines pour tester les filtres (oscilloscope, VNA, etc.) : Le but final étant de calculer précisément la perte, nous devons avoir accès aux valeurs exactes de la tension et de l'intensité dans le circuit. De cette manière, nous pourrions estimer le taux d'erreur entre l'approximation obtenue par le réseau de neurones et la valeur théorique que l'on doit calculer.
- **Traitement et calcul** : Ordinateur performant et logiciels d'analyse/simulation Les calculs seront exécutés par les réseaux de neurones. Ces dernières effectueront les calculs intermédiaires dans les neurones cachés. On obtient en sortie une approximation de la solution des équations différentielles qui gouvernent le circuit. On interprète ensuite les données obtenues à l'aide d'autres bibliothèques telles que matplotlib, qui nous aidera à visualiser ces données à partir de graphiques.

4 Mode de pilotage

Dans le cadre de notre projet, nous avons choisi d'appliquer la méthodologie Scrum pour une gestion agile et efficace. Le projet, d'une durée de deux mois, s'organise autour de sprints hebdomadaires, soutenus par des réunions structurées et un suivi sur l'outil Trello.

La méthode Scrum s'appuie sur les sprints qui sont des périodes de temps restreintes durant lesquelles une liste d'actions sélectionnées pour leur apport en valeur au projet sont réalisées. À la fin d'un sprint, une revue des actions est réalisée et le prochain sprint est planifié pour incrémenter sur le précédent en ajustant les erreurs commises.

Pour notre projet, nous ferons des sprints d'une semaine car le projet ne dure que deux mois, ce qui nécessite un suivi fréquent.

Nous ferons une réunion hebdomadaire d'une heure, durant ces réunions chaque membre du groupe fera une présentation de 5 minutes de son sprint passé où il présentera les actions qu'il a réalisées ou non et les problèmes rencontrés pour qu'on puisse ajuster pour les prochains sprints. Ensuite, on planifiera le sprint à venir et on assignera les différentes actions à chacun des membres.

Une action comprend trois éléments :

- Une description formulée avec un verbe d'action
- Une échéance
- Une ou plusieurs personnes responsables de l'action

Pour gérer les actions, nous utiliserons l'application Trello sur laquelle nous avons un tableau de bord avec cinq listes :

- Une pour le backlog, par défaut toutes les actions sont dans le backlog
- Une pour les actions à faire durant le sprint
- Une pour les actions en cours de réalisation
- Une pour les actions bloquées
- Une dernière pour les actions terminées

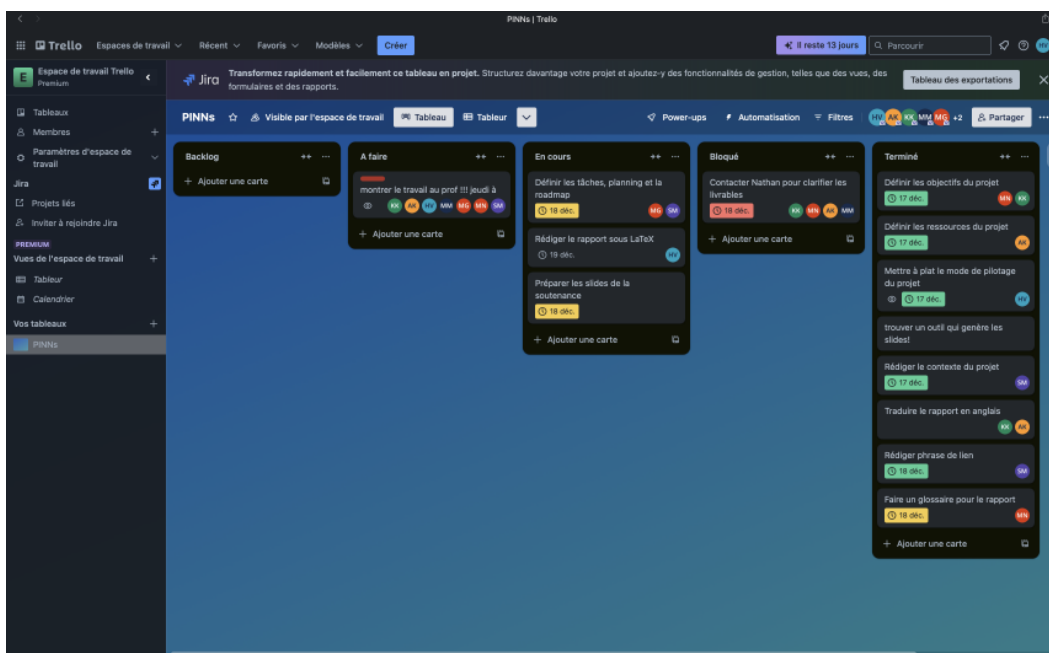


FIGURE 6 – Liste des actions sur Trello

Trello nous permet d'assigner une action à un ou plusieurs membres de l'équipe ainsi que d'ajouter une échéance. Pour un suivi efficace du projet, il est nécessaire que chaque action soit sur Trello et qu'elle ait les trois attributs définis précédemment. Chaque membre de l'équipe gère ses actions et les change de colonnes au fur et à mesure de l'avancement du projet.

En complément des réunions hebdomadaires, nous ferons des réunions quotidiennes de 10 minutes tous les matins à 9h. Cette réunion a pour objectif de faire le point sur ce que chacun a fait la veille et va faire dans la journée afin que chaque membre de l'équipe ait une vue d'ensemble de l'avancement du sprint. Chaque jour, un membre différent de l'équipe sera chargé d'animer la réunion quotidienne, son rôle est de cadrer la réunion pour éviter qu'elle ne dure trop longtemps et qu'elle ne se transforme en résolution de problèmes.

Cette organisation Scrum, centrée sur des sprints courts et une communication quotidienne, permettra une gestion fluide et réactive du projet, tout en favorisant la collaboration et l'amélioration continue.

5 Roadmap du projet

Pour assurer un déroulement structuré et efficace du projet, une planification détaillée a été réalisée. Nous avons divisé le travail en deux grandes phases, chacune avec des objectifs clairs et des échéances bien définies. Un diagramme de GANTT a été utilisé pour visualiser et organiser les grandes tâches ainsi que les jalons importants.

Phase 1

- Période : Du 16 au 20 Décembre 2024.
- Objectif principal : Préparer un rapport et une soutenance préliminaire.

Cette première phase vise à établir les bases du projet, clarifier les attentes et préparer une présentation préliminaire. Le diagramme de Gantt suivant illustre cette organisation.

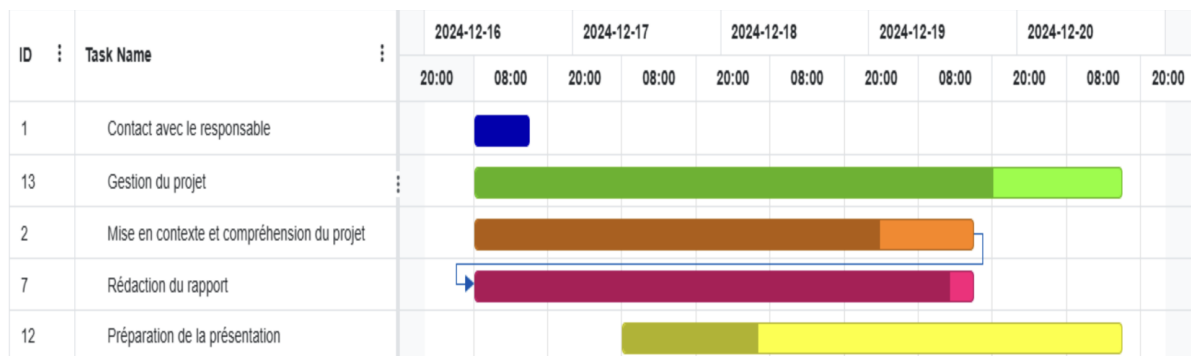


FIGURE 7 – Diagramme de Gantt de la phase 1

Phase 2

- Période : Du 14 avril au 9 juin 2025.
- Objectif principal : Finaliser le projet et atteindre les objectifs fixés.

La seconde phase, d'une durée de deux mois, nous permettra de se concentrer sur les tâches techniques et de finaliser le projet. Les tâches ont été planifiées de manière à équilibrer les efforts entre développement, tests et documentation.

Actions clés :

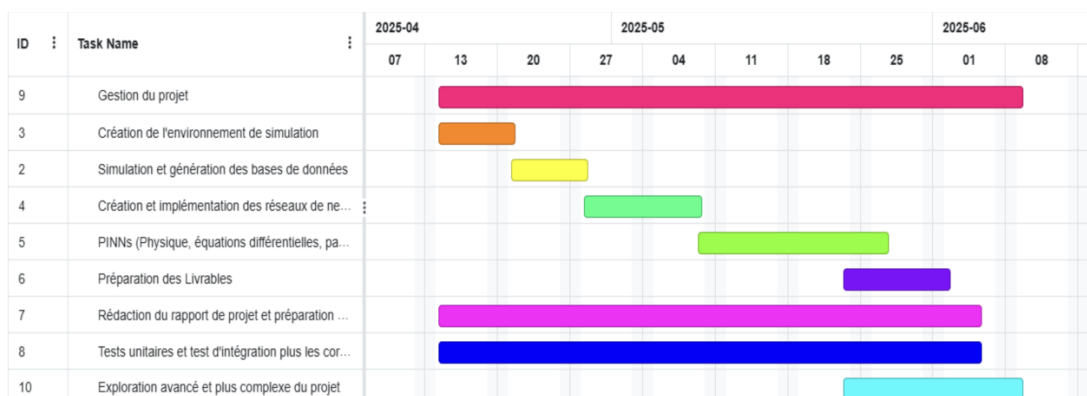


FIGURE 8 – Diagramme de Gantt de la phase 2

Alignement des actions

- **Gestion du projet** : s'étend sur toute la durée du projet comme cette tâche englobe le suivi et l'avancement global du projet, les ajustements et les validations.
- **Création de l'environnement de simulation** : se termine assez tôt dans le projet. Mais ça se peut que cette tâche nécessite une mise à jour ou une prolongation.

Glossaire

SGS : Società Generale Semiconduttori

MEMS : Micro Electronic Mechanical Systems

IoT : Internet of Things

RF : Radio Frequency

NFC : Near-Field Communication

ASIC : Application-Specific Integrated Circuit

PINNs : Physics-Informed Neural Networks

SPICE : Simulation Program with Integrated Circuit Emphasis

VNA : Vector Network Analyzer

Annexe

Sources

- Figure 1 : [st.com](https://www.st.com)
- Figure 2 : [researchgate.com](https://www.researchgate.com)
- Figure 3 : courselectronique.wordpress.com
- Figure 4 : methodephysique.fr
- Figure 5 : methodephysique.fr