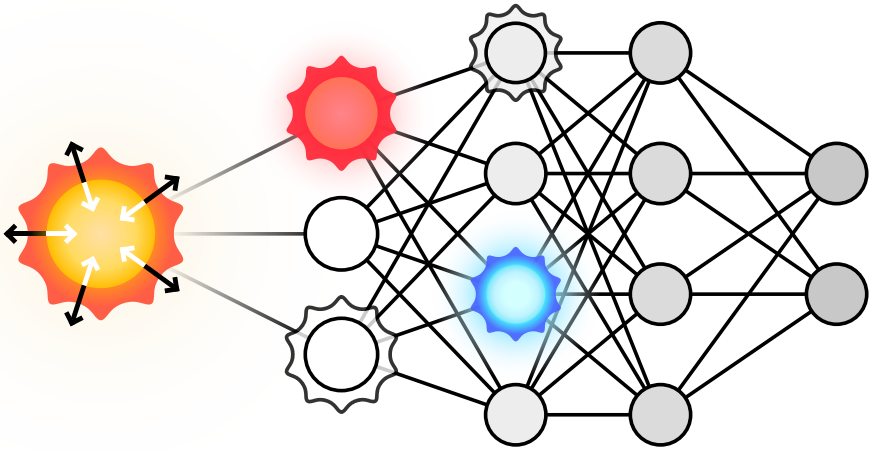
Modélisation de la Structure Interne des Étoiles en utilisant la Technique des Réseaux Neuronaux

Clément Perrochaud

Rapport de stage du projet tuteuré
supervisé par M. Hubert Baty
en coopération avec Stanislas Christophe-Valle

Université de Strasbourg – Faculté de Physique & Ingénierie
Licence III de Physique – Magistère I de Physique Fondamentale



Université de Strasbourg

Observatoire astronomique de Strasbourg | ObAs

Résumé

Ce rapport se décompose en deux parties très distinctes l'une de l'autre. Effectivement, ce stage portait autant sur le sujet physique qu'est la structure interne des étoiles, que sur la méthode que l'on compte utiliser afin d'intégrer les équations que nous serions susceptibles de rencontrer.

Dans un premier temps, nous nous penchons sur les réseaux de neurones. Particulièrement d'actualité, les réseaux de neurones sont représentés par une succession de couches de neurones connectées deux à deux. Lorsque la première couche reçoit un signal d'entrée sous la forme d'un vecteur, celui-ci est propagé en avant et est modifié de façon non linéaire au travers de chacune des couches. Le Théorème d'Approximation Universelle dit qu'un réseau de neurones peut théoriquement approcher toute fonction continue. La méthode des PINNs (pour *Physics-Informed Neural Networks*), utilise ces réseaux neuronaux dans le but d'intégrer des équations différentielles. Cette méthode, particulièrement novatrice, aborde l'intégration d'une manière radicalement différente des méthodes traditionnelles comme Runge-Kutta, permettant d'esquiver des problèmes de valeurs indéfinies et offrant une solution totalement continue.

Dans un second temps, nous étudierons la structure interne des étoiles. Nous nous limiterons à un modèle en particulier, celui décrit par l'équation de Lane-Emden (voir ci-dessous), qui admet que le fluide – plasma – constituant les étoiles se comporte comme un polytrope définit par un indice polytropique n, et néglige les effets de rotation et de champ magnétique. L'hypothèse polytropique est également synonyme de modèle stellaire en une seule couche. Nous utiliserons donc les outils que nous aurons mis en place dans la première partie pour résoudre l'équation de Lane-Emden pour différentes valeurs de n. Finalement, nous pourrons regarder le cas du Soleil et essayer de trouver sa valeur de $n_\odot$. En conclusion, il n'y a pas de $n_\odot$ exacte car il semble varier en fonction de la profondeur de l'étoile voir même distinguer en réalité plusieurs couches (on peut deviner notamment les couches radiatives et convectives).

$$\frac{1}{\xi^2}\frac{\mathrm{d}}{\mathrm{d}\xi}(\xi^2\frac{\mathrm{d}\theta}{\mathrm{d}\xi}) = -\theta^n$$

Table des matières

On pourra rencontrer les notations de définitions suivantes :

1. $X := ... \Leftrightarrow$ définition d'un objet X ;
2. $X :\propto ... \Leftrightarrow X$ définit proportionnel à «...», équivalent à $X := k \cdot ...$ avec $k \in \mathbb{R}$;
3. $X :\propto ..., \ ..., \ ... \Leftrightarrow$ définition comme combinaison linéaire, $X := k_1 \cdot ... + k_2 \cdot ... + k_3 \cdot ...$

INTRODUCTION

Ce rapport de stage de projet tuteuré se divise en deux parties bien distinctes, une première partie où nous verrons le fonctionnement des réseaux de neurones et de la méthode des PINNs, une seconde partie où nous mettrons en œuvre ces méthodes dans l'étude de notre sujet physique : la structure interne des étoiles et l'équation de Lane-Emden.

Historiquement, l'étude de la structure stellaire n'apparaît pas avant le XIX$^{\text{ème}}$ siècle avec l'avènement de la spectroscopie et l'étude de la composition des étoiles par des astronomes tels que Williamina Fleming et Annie Jump Cannon. Avant cela, on ne faisait guère plus qu'observer le soleil et cartographier les étoiles. À la fin du siècle, l'astrophysicien Jonathan Lane est le premier à proposer un modèle de structure des étoiles, notamment en élaborant la fameuse équation de Lane-Emden. Le *Emden* dans le nom de l'équation fait référence à l'astrophysicien Robert Emden qui travailla également sur ces travaux en parallèle, obtenant des résultats similaires à ceux de Jonathan Lane. Néanmoins, il n'est pas le seul et il faudrait alors citer également Lord Kelvin et Albert Ritter ou encore Ralph H. Fowler et Subrahmanyan Chandrasekhar plus tard.

Un réseau de neurones est un système pouvant représenter une fonction multivariée réelle. Comme son nom le laisse entendre, ce système reprend une conception similaire aux réseaux de neurones cérébraux, dits *naturels* – en opposition aux réseaux de neurones *artificiels* que nous étudierons ici. Comme une grande partie des inventions inspirées de la nature, les réseaux de neurones semblent aujourd'hui marquer le seuil d'une nouvelle révolution technologique, ici numérique. De plus en plus, nous les retrouvons dans notre quotidien et ils induisent un intérêt si fort chez les chercheurs et concepteurs du monde entier que les progrès dans ce domaine sont à l'heure actuelle extrêmement rapides. L'efficacité étonnante des réseaux artificiels de neurones dans une multitude de domaines est encore mal comprise, à l'instar des réseaux de neurones naturels pourrait-on dire. Évidemment, nous verrons que leur champ d'action s'étend également aux sciences physiques et c'est pourquoi ce stage y est en partie consacré.

1 Les Réseaux de Neurones

Les réseaux de neurones sont ici des outils dont on souhaite se servir pour résoudre les équations différentielles que nous verrons en partie 2. Il existe plusieurs modules Python fournissant des réseaux de neurones accompagnés de toutes les méthodes nécessaires comme PyTorch, néanmoins, dans le but de s'approprier ces outils et comprendre leur fonctionnement, nous nous proposons dans cette partie de construire également notre propre programme de réseaux de neurones. Finalement, nous pourrons comparer l'efficacité des réseaux de neurones face à d'autres méthodes d'intégration plus *traditionnelles* comme celle de Runge-Kutta. Les implémentations Python des différentes parties qui suivent vous sont disponibles en Annexe **A.1**.

1.1 Architecture d'un réseau de neurones

Les réseaux de neurones artificiels sont donc construits sur une structure assez simple constituée de couches de neurones [1]. Chaque neurone d'une certaine couche est relié à ceux des couches précédentes, renvoyant une combinaison affine (passée au travers d'une fonction dite d'activation que nous verrons plus bas) des valeurs des précédents neurones. En entrée, la première couche est définie comme les arguments donnés à la fonction globale que représente le réseau. En sortie, la dernière couche renvoie le résultat de cette fonction.

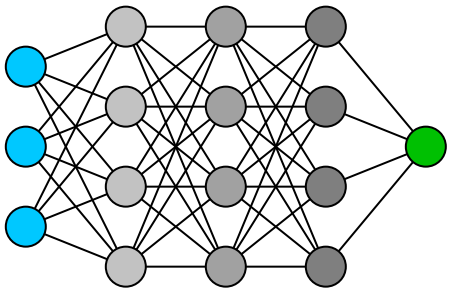


Fig. 1 : Exemple de la structure d'un réseau de neurones de la forme $(3, \overbrace{4, 4, 4}^{\text{couches cachées}}, 1)$
i.e. 3 valeurs d'entrée, 1 de sortie et 3 couches cachées de 4 neurones chacune.

Dans un premier temps, étudions le fonctionnement d'un réseau de neurones pour tenter de construire le notre. On note n_i le nombre de neurones dans la couche i [2]. Prenons le j-ième neurone de la couche i, connecté aux n_{i-1} neurones de la couche précédente. Nous pourrions simplement définir cette valeur, pour tout i non nul, de la sorte :

$$a_{ij} := \sum_{k=0}^{n_{i-1}} w_{ijk} \cdot a_{(i-1)k} + b_{ij}$$

où les w_{ijk} sont les coefficients de pondérations et où b_{ij} est le coefficient de biais.

1. Nous ne travaillerons que sur des réseaux de neurones dits à propagation avant.
2. On peut interpréter la couche n°0 comme les valeurs d'entrée. De plus, on définit l comme le nombre de couches, ainsi la couche n°l sera celle de sortie du réseau.

Plus simplement, on peut interpréter cette relation d'un point de vu matriciel :

$$a_i = w_i \cdot a_{i-1} + b_i$$

où a_i devient le vecteur renvoyé par la i-ème couche, et où w_i et b_j sont respectivement la matrice de pondération et le vecteur biais de la couche i.

Néanmoins si l'on définissait les a_{ij} (ou a_i) de la sorte on peut facilement se rendre compte que le résultat final ne serait qu'une transformation affine des valeurs d'entrée. C'est pourquoi nous devons également introduire une fonction non linéaire dite d'activation $\phi : \mathbb{R} \to \mathbb{R}$ et redéfinir chaque couche comme suit :

$$a_{ij} := \phi\Big(\sum_{k=0}^{n_{i-1}} w_{ijk} \cdot a_{(i-1)k} + b_{ij}\Big) \qquad \text{ou} \qquad a_i = \phi(\, w_i \cdot a_{i-1} + b_i\,) \tag{1}$$

Généralement, la fonction d'activation ϕ est utilisée pour séparer plusieurs régions de $\mathbb{R}$. Le but ici serait de distinguer une tendance *éteinte* d'une tendance *activée*. Voici quelques exemples communs de fonctions d'activation :

$$\begin{aligned}
&\text{(sigmoïde)} && \phi_0(x) = \frac{1}{1+e^{-x}} \\
&\text{(hyperbolique)} && \phi_1(x) = \tanh(x) \\
&\text{(ReLU)} && \phi_3(x) = \begin{cases} x \text{ si } x > 0 \\ 0 \text{ si } x \leq 0 \end{cases} \\
&\text{(logarithmique)} && \phi_3(x) = \operatorname{sgn}(x) \cdot \ln(1+|x|)
\end{aligned} \tag{2}$$

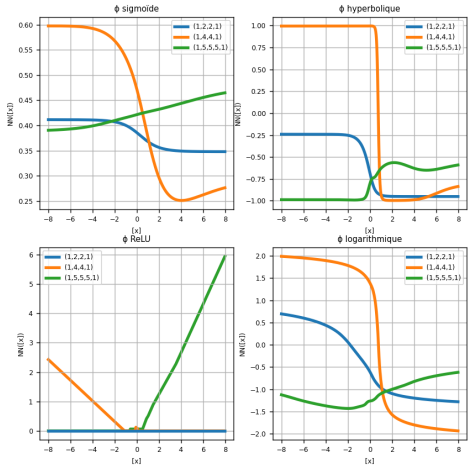


FIG. 2 : Graphes $\mathfrak{N}_\Theta(x)$ de différentes formes (1,...,1) pour les différentes fonction d'activations définies ci-contre.

Plusieurs caractéristiques peuvent ressortir de ces fonctions d'activation. Les fonctions sigmoïde et hyperbolique sont toutes deux bornées et offrent une transition douce entre leurs bornes. À noter qu'afin de prévenir le fait que le résultat final soit lui-même borné, on peut simplement ne pas appliquer la fonction d'activation sur la dernière couche. La fonction ReLU présente un point anguleux et transforme le réseau comme une succession de segments rectilignes. Elle permettra d'approcher une courbe par une succession de segments continus. La fonction logarithmique ressemble aux deux premières mais n'est elle pas bornée. Il existe évidemment bien d'autres fonctions d'activations, mais ce sont là les plus utilisées.

Ainsi, nous comprenons que les réseaux de neurones sont définis par un grand nombre de coefficients qui interviennent dans chacun des neurones. Aussi performants pourraient-ils être, c'est sur le choix méticuleux de chaque coefficient que repose leur mécanisme et leur précision, à la manière des termes d'un développement limité. Leur finesse réside dans le grand nombre de neurones et de couches utilisées. Ainsi, il n'est absolument pas pertinent d'essayer de calculer à la main ces coefficients, il nous faut donc trouver des méthodes pour déduire quels seraient les coefficients optimaux, c'est le *Deep Learning*[3].

3. On parle de Deep Leanrning, ou d'apprentissage profond, pour des réseaux à plus d'une couche cachée, d'où l'adjectif « profond ».

1.2 Les descentes de gradient

1.2.1 Principe

On peut associer à une fonction un ensemble $\Theta \in \mathbb{R}^{\mathbf{N}}$ constitué des $\mathbf{N}$ coefficients impliqués dans le calcul de cette dernière. Pour un réseau de neurones, Θ sera par exemple l'ensemble de ses coefficients w_{ijk} et b_{ij}. Le but est de trouver la valeur optimale de Θ, et ici, nous souhaitons que notre réseau de neurones donne les résultats les plus *précis* possible. Ainsi, nous devons introduire une fonction de perte J que nous appellerons la Loss, mesurant le degré d'erreur de notre réseau pour un Θ donné. L'expression de J dépendra donc du problème que doit résoudre le réseau de neurone.

Supposons par exemple un réseau de neurones de la forme représenté par $\mathfrak{N}_\Theta$ [4] devant approcher un ensemble de n valeurs *expérimentales* $E = \{(x_i, y_i), \forall i \in [\![0, n[\![\}$. Nous pouvons définir la Loss comme la moyenne des écarts quadratiques :

$$J_0(\Theta) = \frac{1}{n} \sum_{(x,y)\in E} ||\mathfrak{N}_\Theta(x) - y||^2 \geq 0 \tag{3}$$

Nous devons donc maintenant apprendre à trouver le minimum de cette fonction. Pour ce faire, nous pouvons utiliser un algorithme de descente de gradient. Ce procédé consiste à suivre l'opposé du gradient de $J(\Theta)$ jusqu'à trouver un minimum global (ou au moins local). Nous en venons donc à créer une suite $(\Theta_n)_{n\in\mathbb{N}}$ convergent [5] vers ce minimum. L'implémentation la plus simple de cet algorithme est exprimée comme :

$$\Theta_{n+1} = \Theta_n - \gamma \cdot \nabla J(\Theta_n) \tag{4}$$

où $\gamma > 0$ est appelé taux d'apprentissage et doit être suffisamment petit.

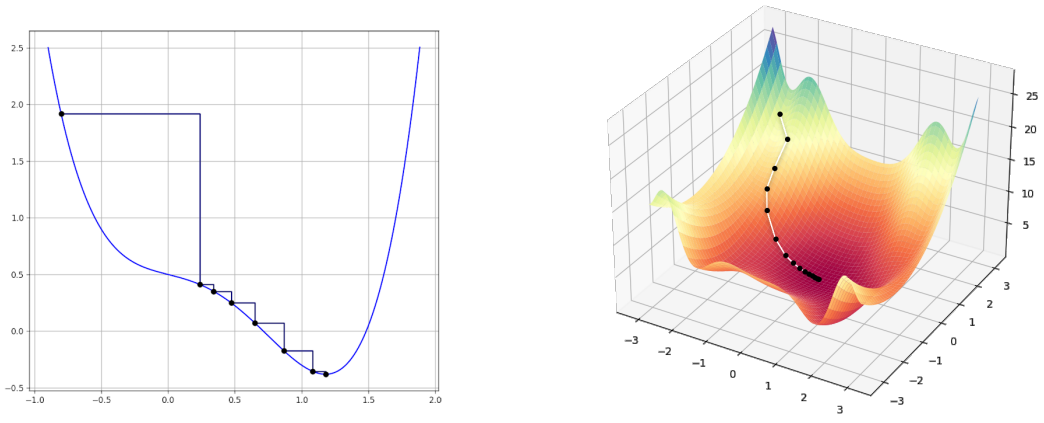


FIG. 3 : Exemples visuels simples de descentes de gradient pour $\mathbf{N} = 1$ à gauche, et $\mathbf{N} = 2$ à droite, représentant $J(\Theta)$ et l'évolution de la suite (Θ_i) au cours de la recherche.

4. Pour un réseau de neurones, $\mathfrak{N}_\Theta : a_0 \mapsto a_l$. Au delà des réseaux de neurones, on pourrait simplement la définir comme l'expression de n'importe quel modèle de fonction associée à un Θ de la forme recherchée.

5. En fonction des algorithmes utilisés, nous pourrons constater une convergence *globale* de cette suite sans pour autant que localement $J(\Theta_n + 1) < J(\Theta_n)\ \forall n$, on pourra même simplement sauvegarder la meilleure valeur Θ.

1.2.2 Algorithmes du premier ordre plus avancés

Très vite, cet algorithme basique peut montrer des faiblesses. Premièrement, les pentes douces ralentissent fortement la recherche. Deuxièmement, la présence de minima locaux peut représenter un risque de blocage. Pour remédier à cela, de nombreuses alternatives existent, en voici quelques exemples [6].

(momentum)
$$\begin{aligned} v_n &= \beta \cdot v_{n-1} + (1-\beta) \cdot \nabla J(\Theta_n) \\ \Theta_{n+1} &= \Theta_n - \gamma \cdot v_n \end{aligned} \tag{5}$$

(AdaGrad)
$$\begin{aligned} g_n &= g_{n-1} + (\nabla J(\Theta_n)^2 \geq 0 \\ \Theta_{n+1} &= \Theta_n - \frac{\gamma}{\sqrt{g_n} + \epsilon} \cdot \nabla J(\Theta_n) \end{aligned} \tag{6}$$

(RMSprop)
$$\begin{aligned} v_n &= \beta \cdot v_{n-1} + (1-\beta) \cdot (\nabla J(\Theta_n))^2 \geq 0 \\ \Theta_{n+1} &= \Theta_n - \frac{\gamma}{\sqrt{v_n} + \epsilon} \cdot \nabla J(\Theta_n) \end{aligned} \tag{7}$$

(Adam)
$$\begin{aligned} m_n &= \beta_1 \cdot v_{n-1} + (1-\beta_1) \cdot \nabla J(\Theta_n) \\ v_n &= \beta_2 \cdot v_{n-1} + (1-\beta_2) \cdot (\nabla J(\Theta_n))^2 \geq 0 \\ \hat{m_n} &= \frac{m_n}{1-(\beta_1)^n} \quad \text{et} \quad \hat{v_n} = \frac{v_n}{1-(\beta_2)^n} \\ \Theta_{n+1} &= \Theta_n - \gamma \cdot \frac{\hat{m}_n}{\sqrt{\hat{v_n}} + \epsilon} \end{aligned} \tag{8}$$

Le premier algorithme ajoute un *momentum* v (*quantité de mouvement* en anglais, sous-entendant une inertie) à la recherche de minimum. On choisira généralement une valeur du paramètre $\beta \in [0, 1[$ proche de 1 ($\beta = 0$ correspondant à l'algorithme basique **(4)**). Cet ajout permet d'accélérer la recherche autant que de protéger contre les minima locaux.

Adagrad (2011), pour *Adaptative Gradient*, propose de diminuer le taux d'apprentissage au fur et à mesure de la convergence via la série des gradients g. ($\epsilon > 0$ est un terme de régulation définit au plus proche de 0)

RMSprop (2012), pour *Root Mean Square Propagation*, combine les idées des deux précédents algorithmes pour ajuster le taux d'apprentissage de façon non monotone, contrairement à Adagrad, et évite une convergence trop précoce.

Adam (2014), pour *Adaptative Moment Estimation*, est probablement l'un des algorithmes de descente de gradient les plus connus. Il reprend à nouveau les idées de ses prédécesseurs et utilise un momentum sur le gradient en plus de l'ajustement sur le taux d'apprentissage.

6. On précise que toutes les opérations sont ici effectuées élément par élément.

1.2.3 Mise en pratique

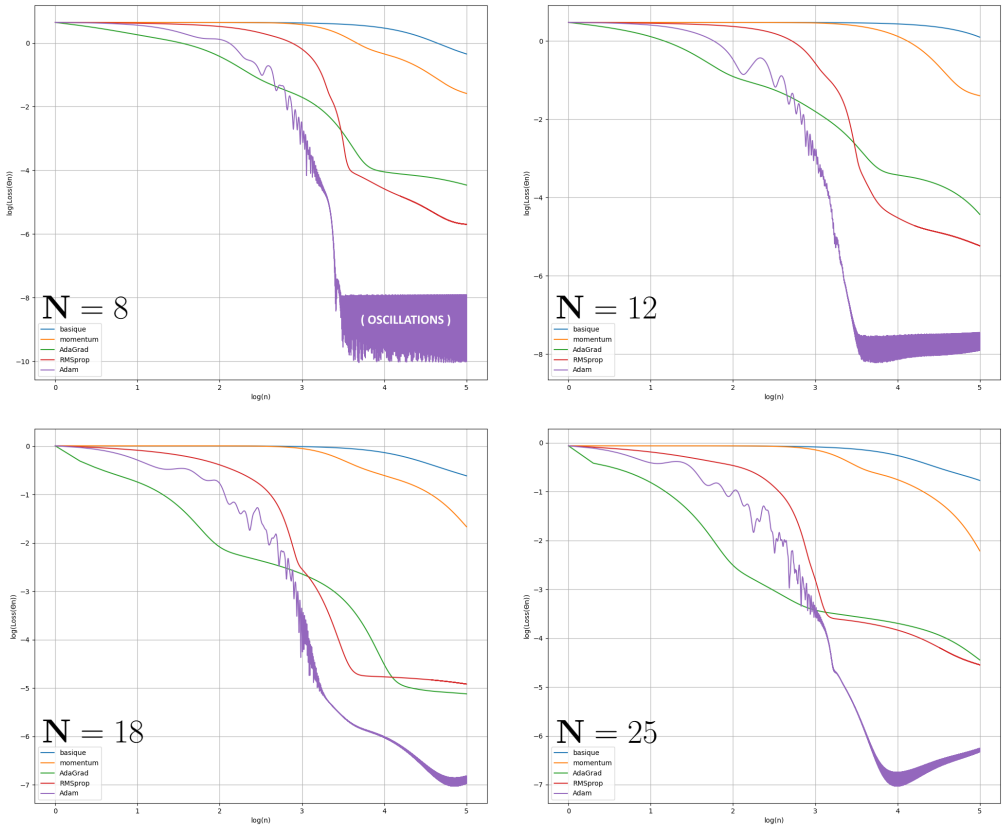


FIG. 4 : Recherche de minimum des algorithmes de descente de gradient précédemment définis, pour différentes valeurs de **N**.
Sont ici représentées les valeurs de $J_0(\Theta_n)$ en fonction du nombre d'itérations n.

Non spécifique aux réseaux de neurones, cet exemple recherche les coefficients d'un polynôme connaissant un ensemble de points et la Loss J_0 décrite en **(3)**, avec les paramètres donnés ci-contre [7]. Vous pouvez retrouver l'implémentation Python de ces algorithmes de descente de gradient de premier ordre en Annexe **A.1**.

basique	$\gamma = 10^{-5}$
momentum	$\gamma = 10^{-4}$ $\beta = 0.999$
AdaGrad	$\gamma = 0.1$
RMSprop	$\gamma = 10^{-3}$ $\beta = 0.97$
Adam	$\gamma = 10^{-3}$ $\beta_1 = 0.99$ $\beta_2 = 0.98$

et $\mathrm{d}x = 10^{-5}$, $\epsilon = 10^{-5}$

Les graphes de la **FIG.4** montrent l'évolution de la Loss seulement en fonction du nombre d'itérations [8]. Il serait légitime de se dire qu'un algorithme tel que Adam aurait des itérations plus lentes qu'un algorithme basique, et donc que ce graphique serait biaisé. Même s'il y a une différence de vitesse d'exécution, elle est étonnamment faible entre chacun des algorithmes (voir Annexe **A.1.1**). C'est donc bien l'algorithme Adam qui est le plus performant ici, et en général.

7. On pourrait chercher les meilleurs paramètres pour chaque algorithmes et pour chaque cas, mais pour la suite, on utilisera ces paramètres. De plus, au vu de la supériorité de l'algorithme Adam, on utilisera celui-ci quasiment exclusivement par la suite.

8. D'autres abscisses, telles qu'en temps de calcul, sont disponible dans le fichier correspondant à l'Annexe **A.1**.

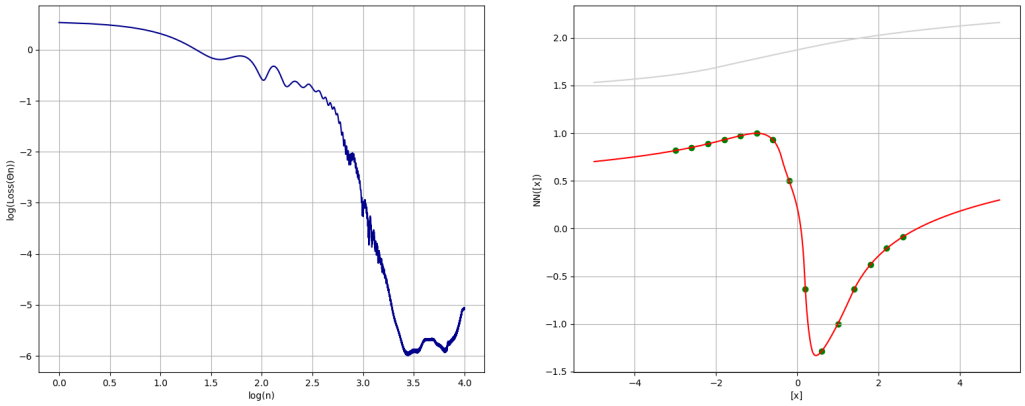
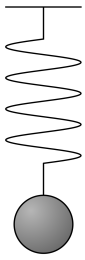


FIG. 5 : Exemple d'approche d'une courbe décrite par 15 points par un réseau de neurones de forme (1,3,3,1), de fonction d'activation logarithmique (ϕ_3) et optimisé par l'algorithme Adam. À droite : la forme finale du réseau de neurones en rouge, sa forme initiale (aléatoire donc) en gris, et les points expérimentaux en vert. À gauche : l'évolution de la Loss $J_0(\Theta_n)$ **(3)** au cours de la recherche.

On peut se poser la question de la complexité d'une telle descente de gradient. Effectivement, à chaque étape, une grande quantité d'appels du réseau de neurones va être effectuée. Chaque appel à une complexité proportionnel au nombre de coefficients $\mathbf{N} = \sum_{i=0}^{l} (n_{i-1} + 1) \cdot n_i$, et ce pour plusieurs dizaines voir centaines de milliers d'itérations, comme vous verrez par la suite. Il sera donc important d'essayer d'optimiser l'algorithme. On peut penser à de la vectorisation, du multi-processing, à de la dérivation analytique ou à écrire dans des langages plus rapides comme le C++. Néanmoins, nous ne développerons pas ce sujet davantage dans ce rapport.

1.3 Physics-Informed Neural Networks

Essayons de modéliser un oscillateur harmotique amorti. On suppose une masse m soumise à un frottement fluide, fixée à un ressort de masse négligeable et de raideur κ, sans gravité. On impose des conditions initiales x_0 et v_0. Son mouvement est ainsi décrit par l'équation différentielle suivante :



$$\mathrm{D}x := m\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + \gamma\frac{\mathrm{d}x}{\mathrm{d}t} + \kappa x = 0$$

1.3.1 Méthode directe

Essayons d'injecter nos conditions initiales dans un réseau de neurones $(1, 20, 20, 1)$:

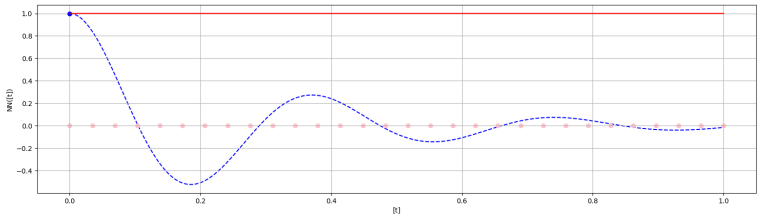


FIG. 6 : $\mathfrak{N}_\Theta(t)$ en rouge après entraînement et solution analytique en bleue (à approcher). Les (Le) points bleus représentent les points expérimentaux (ici x_0).

Évidemment, notre réseau de neurones tel quel n'aurait aucune raison de s'approcher de la solution. Pour qu'il y arrive, il faut trouver un moyen de lui fournir l'équation différentielle. Ce n'est pas le réseau lui-même qu'il faut redéfinir mais la Loss. Le but de la Loss est de fournir un degré d'erreur ≥ 0, on peut interpréter la valeur de $\mathrm{D}x = m\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + \gamma\frac{\mathrm{d}x}{\mathrm{d}t} + \kappa x$ de notre réseau comme un écart à un temps donné, valant 0 dans la meilleure configuration. On définit donc une Loss *physique* comme cet écart quadratique moyen sur un ensemble de n points de collocation C :

$$J_{\Phi 0}(\Theta) :\propto \underbrace{\frac{1}{n}\sum_{t\in C} |\,\mathrm{D}\,\mathfrak{N}_\Theta(t)\,|^2}_{\text{Loss sur l'équation différentielle}} \quad , \quad \underbrace{|\mathfrak{N}_\Theta(t=0) - x_0|^2 \quad , \quad |\frac{\mathrm{d}\mathfrak{N}_\Theta}{\mathrm{d}t}(t=0) - v_0|^2}_{\text{Loss sur les conditions initiales } J_0 \text{ et } J_0'} \geq 0 \tag{9}$$

En utilisant cette nouvelle Loss, nous formons la méthode des PINNs (Physics-Informed Neural Networks). Évidemment chaque itération d'optimisation sera plus lente car le calcul de la Loss J_Φ est plus lourd que J_0. Néanmoins, une fois optimisé, le réseau est techniquement très puissant car représente une solution approximative continue de l'équation différentielle fournie.

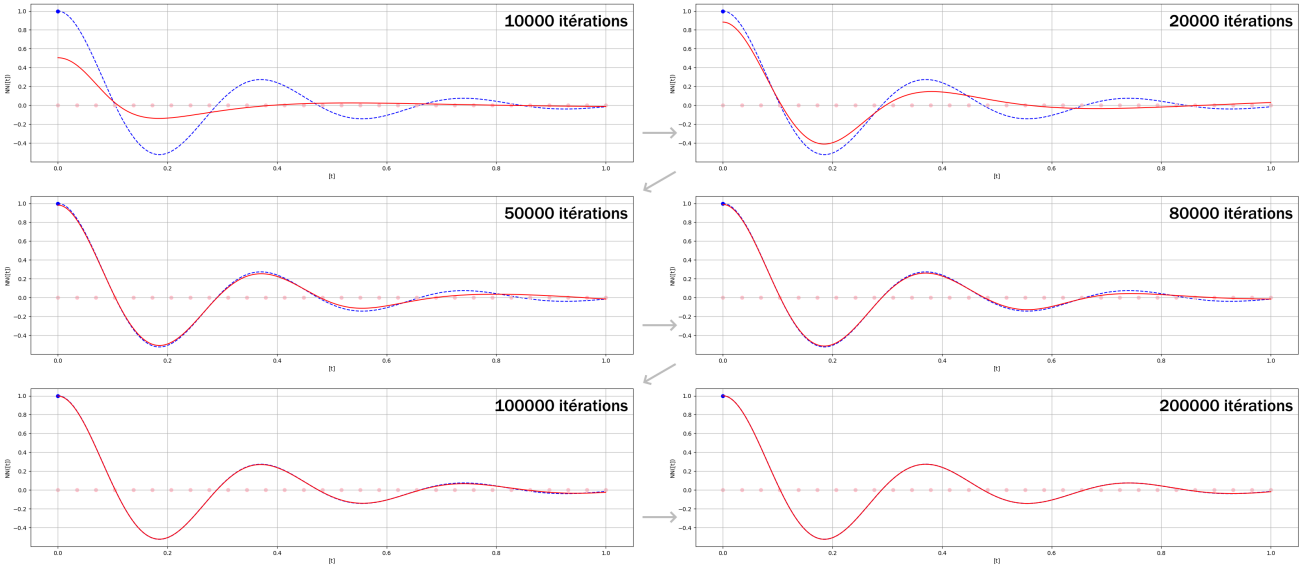


FIG. 7 : Mêmes graphes que la **FIG.6**, cette fois avec la Loss physique J_Φ, pour différentes profondeurs de recherche. Les points de collocation sont visibles en rose.

À noter le réseau de la **FIG.7** présente une répartition uniforme des points de collocation. Il est tout à fait possible d'imaginer augmenter leur densité là où l'on prévoit de plus fortes variations pour encore améliorer la précision de l'intégration, ou encore les distribuer aléatoirement les points de collocation.

1.3.2 Méthode inverse

Il est également tout à fait possible de faire la méthode inverse et, à partir de points expérimentaux, remonter aux paramètres de l'équation différentielle. On pourrait en quelque sorte *ajouter* m, γ et/ou κ à l'ensemble Θ comme nouveaux paramètres afin que le réseau approche leurs valeurs réelles. Contrairement aux autres coefficients Θ_i qui interviennent dans la propagation avant du réseau – son appel, ces nouveaux paramètres seraient utilisés dans le calcul de la Loss, ce qui revient à peu près au même pour l'optimiseur. Cette Loss reprend la forme de J_Φ **(9)** :

$$J_{\Phi 1}(\Theta) :\propto \frac{1}{card(C)} \sum_{t \in C} |\, \mathrm{D}\, \mathfrak{N}_\Theta(t) \,|^2 \,, \quad \frac{1}{card(E)} \sum_{(t,x) \in E} |\, \mathfrak{N}_\Theta(t) - x \,|^2 \;\geq 0 \tag{10}$$

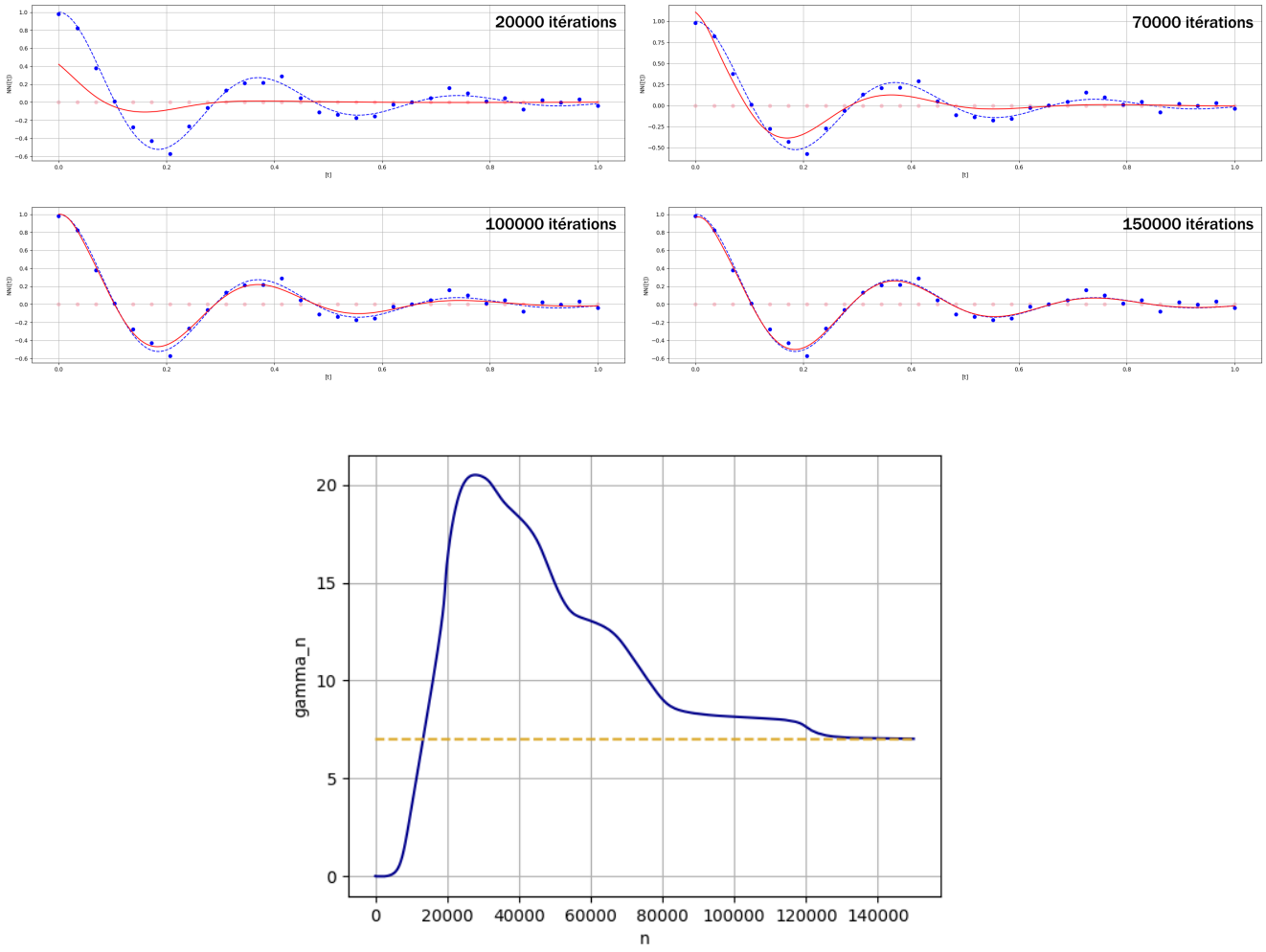


FIG. 8 : Méthode inverse, on fournit au réseau 30 points expérimentaux *bruités* et l'on essaye de retrouver la valeur de γ. Les points expérimentaux ont été générés avec un valeur $\gamma_{\text{réel}} = 7\,\text{kg.s}^{-1}$. Le graphe du bas représente l'évolution de la recherche de la valeur γ.

2 Structure Interne des Étoiles

Une étoile est un corps massif autogravitant principalement constitué d'éléments légers – hydrogène en majorité et hélium en moindre mesure – à l'état de plasma, et suffisamment massif pour que des réactions de fusion nucléaire aient lieu en son sein. Pour la majeur partie de la vie d'une étoile, ces réactions thermonucléaire transforme l'hydrogène en hélium.

2.1 Polytrope et équation de Lane-Emden

Le modèle que nous allons explorer est celui décrit par l'équation de Lane-Emden. Il suppose que le plasma de l'étoile se comporte comme un polytrope, c'est-à-dire qu'il serait décrit par une équation d'état ne faisant intervenir que la masse volumique ρ et la pression P, tel que :

$$P = K\rho^{\Gamma} \tag{1}$$

où $\Gamma := 1 + \frac{1}{n}$ et n est appelé l'indice polytropique. Cette hypothèse peut paraître a première vue étonnante lorsque l'on sait qu'une loi polytropique peut également être définit comme une transformation réversible d'un gaz parfait. Néanmoins, il se trouve que la loi polytropique offre un modèle bon et simple lors que l'on suppose quelques autres hypothèses. Nous supposerons donc également la symétrie sphérique de l'étoile, c'est-à-dire que l'on négligera les effets de sa rotation, et l'équilibre entre la pression interne et l'attraction gravitationnelle, donc un équilibre hydrostatique. Nous ne prendrons également pas en compte le champ magnétique généré par l'étoile, et supposerons un modèle en une seule zone, en opposition aux modèles impliquant des zones radiatives et convectives.

Le champ gravitationnel en un point $\vec{r}$ est :

$$\vec{g}(\vec{r}) = \iiint_{V_E} G\frac{\vec{r}-\vec{r}\,'}{||\vec{r}-\vec{r}\,'||^3}\rho(\vec{r}\,')\mathrm{d}^3\vec{r}\,' \overset{\text{Gauss}}{=} -G\frac{M(r)}{r^2}\hat{r} \tag{2}$$

où $M(r)$ est la masse contenue sous la sphère de rayon r,

$$M(r) = \iiint_{\mathcal{B}(O,r)} \rho(\vec{r}\,')\mathrm{d}^3\vec{r}\,' = 4\pi\int_0^r r\,'^{\,2}\rho(r\,')\mathrm{d}r\,' \tag{3}$$

V_E le volume de l'étoile et G la constante de gravitation universelle.

La conservation de la masse impose également :

$$\mathrm{d}M(r) = 4\pi r^2\rho(r)\mathrm{d}r \tag{4}$$

Finalement, l'hydrostatique est décrite par :

$$\vec{\nabla}P = \Sigma\vec{F} = \vec{g}\rho(r) \overset{(2)}{=} -G\frac{M(r)\rho(r)}{r^2}\hat{r} \tag{5}$$

cas particulier de la mécanique des fluides décrite par l'équation de Navier-Stokes où $\vec{v} = \vec{0}$:

$$\rho\frac{\mathrm{D}\vec{v}}{\mathrm{D}t} := \rho\cdot(\frac{\partial\vec{v}}{\partial t} + (\vec{v}\cdot\vec{\nabla})\vec{v}) = \Sigma\vec{F} - \vec{\nabla}P + \eta\vec{\nabla}^2\vec{v}$$

Ainsi,

$$M(r) \overset{(5)}{=} -\frac{r^2}{G\rho(r)}\frac{\mathrm{d}P}{\mathrm{d}r} \Rightarrow G\frac{\mathrm{d}M}{\mathrm{d}r} = -\frac{\mathrm{d}}{\mathrm{d}r}(\frac{r^2}{\rho(r)}\frac{\mathrm{d}P}{\mathrm{d}r}) \overset{(3)}{=} 4\pi G r^2 \rho(r) \tag{6}$$

On change alors de variables en posant θ et ξ adimensionnées telles que :

$$\rho := \rho_c\,\theta^n \quad \text{et} \quad r := \alpha\,\xi \tag{7}$$

où ρ_c est la masse volumique au centre de l'étoile, n est l'indice polytropique et α une constante.

$$\begin{aligned}
&\textbf{(1)} \overset{(7)}{\Leftrightarrow} P = K\rho_c^{\Gamma}\Theta^{n\Gamma} = K\rho_c^{\Gamma}\Theta^{n+1} \\
&\textbf{(6)} \overset{(7)}{\Leftrightarrow} \frac{1}{\alpha}\frac{\mathrm{d}}{\mathrm{d}r}(\frac{\alpha^2\xi^2}{\rho_c\theta^n}\frac{K\rho_c^{\Gamma-1}(n+1)}{\alpha}\theta^n\frac{\mathrm{d}\theta}{\mathrm{d}\xi}) = -4\pi G\alpha^2\xi^2\rho_c\theta^n \\
&\quad\Leftrightarrow \frac{1}{\xi^2}\frac{\mathrm{d}}{\mathrm{d}r}(\xi^2\frac{\mathrm{d}\theta}{\mathrm{d}\xi}) = -\frac{4\pi G\rho_c}{K\rho_c^{\Gamma-1}(n+1)}\alpha^2\theta^n
\end{aligned} \tag{8}$$

On pose alors :

$$\alpha := \sqrt{\frac{(n+1)K}{4\pi G}\rho_c^{\frac{1}{n}-1}} \tag{9}$$

et on obtient la fameuse équation de Lane-Emden :

$$\boxed{\frac{1}{\xi^2}\frac{\mathrm{d}}{\mathrm{d}\xi}(\xi^2\frac{\mathrm{d}\theta}{\mathrm{d}\xi}) = -\theta^n} \tag{10}$$

On définit les conditions aux centre comme :

$$\theta(\xi = 0) := 1 \quad \text{et} \quad \frac{\mathrm{d}\theta}{\mathrm{d}\xi}(\xi = 0) := 0 \tag{11}$$

où la première repose sur la définition de θ sachant $\rho_c := \rho(r = 0)$ **(7)** ; la seconde est argumentée en Annexe **A.2.2**

On pourra également poser l'équation de Lane-Emden sous une forme linéaire :

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}\xi^2} + \frac{2}{\xi}\frac{\mathrm{d}\theta}{\mathrm{d}\xi} + \theta^n = 0 \tag{12}$$

On définit $\xi_R := R/\alpha$ la première valeur de ξ où θ s'annule. R sera donc définit comme le rayon de l'étoile, car c'est dépassé cette limite ρ pourrait admettre des valeurs négatives.

2.1.1 Résolutions analytiques de l'équation de Lane-Emden

L'équation de Lane-Emden n'est pas toujours résoluble analytiquement. Néanmoins, il existe quelques valeurs de n pour lesquelles c'est le cas :

n	θ	ξ_R
0	$1-\frac{\xi^2}{6}$	$\sqrt{6}$
1	$\frac{\sin\xi}{\xi}$	π
5	$(1+\frac{\xi^2}{3})^{-\frac{1}{2}}$	∞

Table. 1 : Résolutions analytiques pour n=1, 2 et 5.

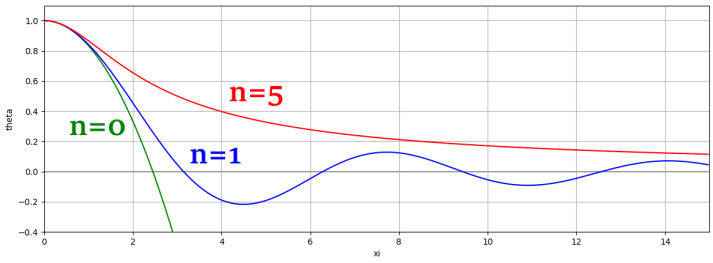


Fig. 9 : Graphes représentant l'allure exacte des $\theta(\xi)$ décrites ci-contre.

Les démonstrations des résolution présentées dans la **Table.1** pour $n = 0$ et $n = 1$ sont disponibles en Annexe **A.2.1**. La solution à $n = 5$ fut quant à elle trouvée en 1882 par Arthur Schuster.

2.1.2 Résolutions numériques de l'équation de Lane-Emden

Il est temps d'utiliser nos réseaux de neurones. Nous prenons donc un réseau $(1, 20, 20, 1)$ que nous utilisons comme PINN avec 50 points de collocation pour s'assurer de la meilleure intégration. Dans un premier temps, essayons avec $n = 0$, 1 et 5, et comparons nos résultats avec les résolutions analytiques précédemment évoquées :

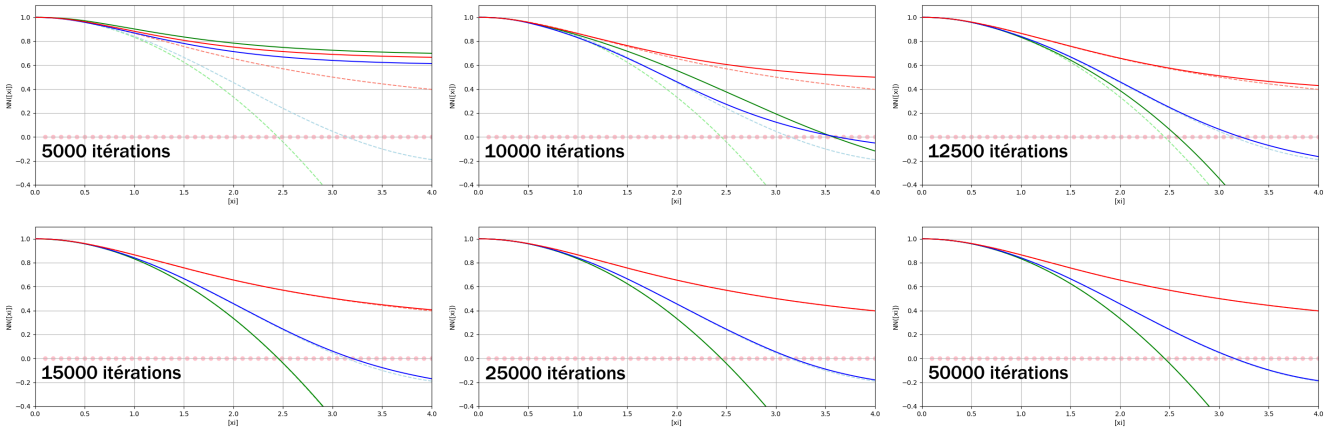


Fig. 10 : Graphes représentant la recherche du PINN pour chaque n (même code couleur de la **Fig.9**) avec en pointillés les résolutions analytiques.

Nous obtenons effectivement des résultats très proches de nos résolutions analytiques. On peut considérer que notre algorithme fonctionne donc suffisamment bien pour lui accorder une certaine confiance quant aux intégrations à n autre que 0, 1 ou 5.

2.1.3 Traduction vers des grandeurs plus *physiques*

Nous avons définit plus haut ξ_R tel que $R = \alpha\xi_R$ et où $\theta(\xi_R) = 0\,;\, \forall\xi \in [0, \xi_R[\,:\, \theta(\xi) \geq 0$. De cette valeur, on peut remonter au rayon normalisé :

$$\frac{r}{R} = \frac{\xi}{\xi_R} \tag{13}$$

On peut également retrouver l'expression de la masse de l'étoile M avec les grandeurs θ et ξ :

$$\begin{aligned} M = M(R) &\overset{(3)}{=} 4\pi\alpha^3\rho_c \int_0^{\xi_R} \xi^2\theta(\xi)^n \mathrm{d}\xi \\ &\overset{(10)}{=} 4\pi\alpha^3\rho_c \cdot [-\xi^2\frac{\mathrm{d}\theta}{\mathrm{d}\xi}]_0^{\xi_R} = -4\pi\alpha^3\rho_c\xi_R^2\frac{\mathrm{d}\theta}{\mathrm{d}\xi}(\xi_R) \\ &= -4\pi\rho_c\frac{R^3}{\xi_R}\frac{\mathrm{d}\theta}{\mathrm{d}\xi}(\xi_R) \end{aligned} \tag{14}$$

Cette expression permet également de remonter à la valeur de ρ_c connaissant la masse M et le rayon R de l'étoile, et la résolution de $\theta(\xi)$.

En partant de nos résolutions – analytiques ou numériques – pour $n = 0$, ½, 1, 2, 3 et 4 (nous ne traiterons évidemment pas $n = 5$ car alors $\xi_R \longrightarrow \infty$) on peut tracer les courbes de ρ/ρ_c **(7)** en fonction de r/R **(13)** :

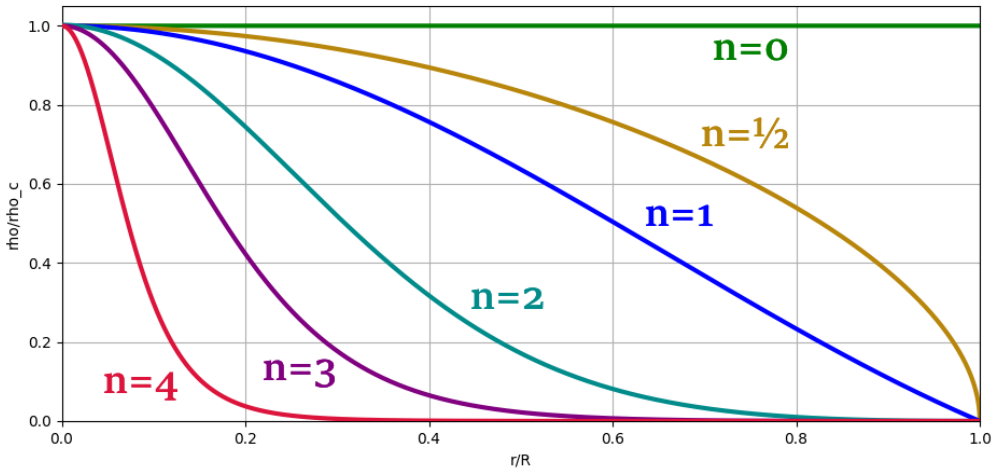


FIG. 11 : Graphes de ρ/ρ_c en fonction de r/R, pour les différentes valeurs de n.

Le cas à $n = 0$ correspond à une densité constante, soit une étoile totalement incompressible. Aux valeurs de n aux autour de 1.5 on peut associer des étoiles entièrement connectives – soit petites et froides – et pour des n à valeurs proches de 3 peuvent correspondre des étoiles radiatives (toujours dans l'hypothèse d'étoile à une unique couche).

2.1.4 Cas du Soleil

Le Soleil est une étoile de type naine jaune de la séquence principale. Au vu de ses propriétés, il est décrit comme une étoile d'indice polytropique $n_\odot \approx 3$. Comme décrit précédemment, une telle étoile est supposée radiative.

On peut alors se proposer d'utiliser la méthode inverse des PINNs décrite dans la première partie afin de retrouver cette valeur $n_\odot$. On pourra prendre comme données expérimentales la Base Solaire Standard, une base de donnée sur la structure du Soleil donnant notamment ρ en fonction de $r/R_\odot$.

Néanmoins, le problème est plus compliqué qu'il n'en a l'air à cause des différents changements de variables opérés. Effectivement, on cherche à résoudre $\theta(\xi) = (\rho/\rho_c)^n(\xi)$ mais les données expérimentales sont au mieux $\rho/\rho_c(\xi/\xi_R)$, où l'on obtenait précédemment ξ_R après avoir résolu $\theta(\xi)$ pour un n donné. Si l'on essaye à chaque étape de l'approche par notre PINN, de calculer ξ_R à partir de notre θ *non résolu*[9], le modèle diverge et ne s'approche pas de la solution. Nous devons donc pouvoir obtenir ξ_R sans passer par θ. On peut se ramener à la définition $R = \alpha\xi_R$ avec α exprimé en **(9)**. La Base Solaire Standard nous donner également les valeurs de pression $P(r/R_\odot)$ afin de remonter à K pour un n donné avec **(1)** [10].

$$\xi_{R_\odot}(n) = \frac{R_\odot}{\alpha(n)} = R_\odot \sqrt{\frac{4\pi G}{(n+1) < \frac{P}{\rho^{1+1/n}} > \rho_c^{1/n-1}}} \tag{15}$$

Cependant, lorsque la réalité s'écarte du modèle de Lane-Emden ici utilisé, cette méthode peut s'avérer peu concluante. Une autre solution serait plus simplement de réaliser une sorte de descente de gradient sur les n. On pose un n initial que l'on décale au fur et a mesure des intégrations successives de $\theta(\xi)$ en fonction de la Loss expérimentale. Cette méthode moins subtile troque sa rapidité de calcul contre une plus grande simplicité.

On peut voir après mise en pratique sur la **Fig.12** que $n_\odot$ semble être optimale autour d'une valeur de 3.52. À cette valeur $n_\odot$ correspond un écart de l'ordre de 10^{-4}, ce qui n'est pas aussi faible que ce que l'on pourrait s'attendre. Cette valeur traduit donc bien un écart entre le modèle et la réalité. On peut effectivement voir sur la **Fig.13** qu'aucune valeur de n de colle parfaitement aux valeurs expérimentales sur tout le long de $r/R_\odot$.

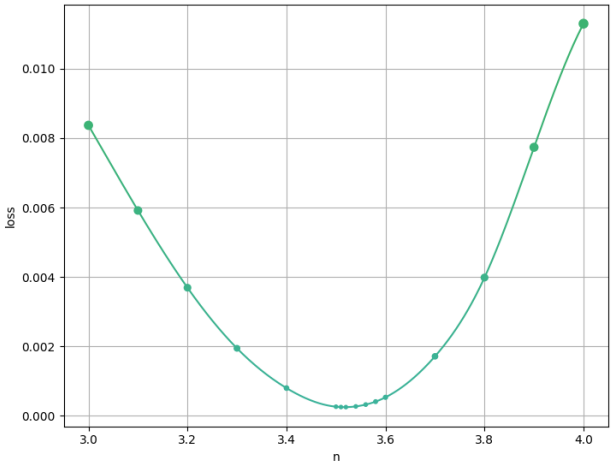


Fig. 12 : Évolution de la Loss expérimental (écart quadratique moyen) en fonction de n.

9. On entend pas là la fonction que représente notre réseau de neurones $a_0 \mapsto a_l$.
10. On pourra ici faire une moyenne $K = < P/\rho^\Gamma >$ sur $r/R \in [0, 1]$.

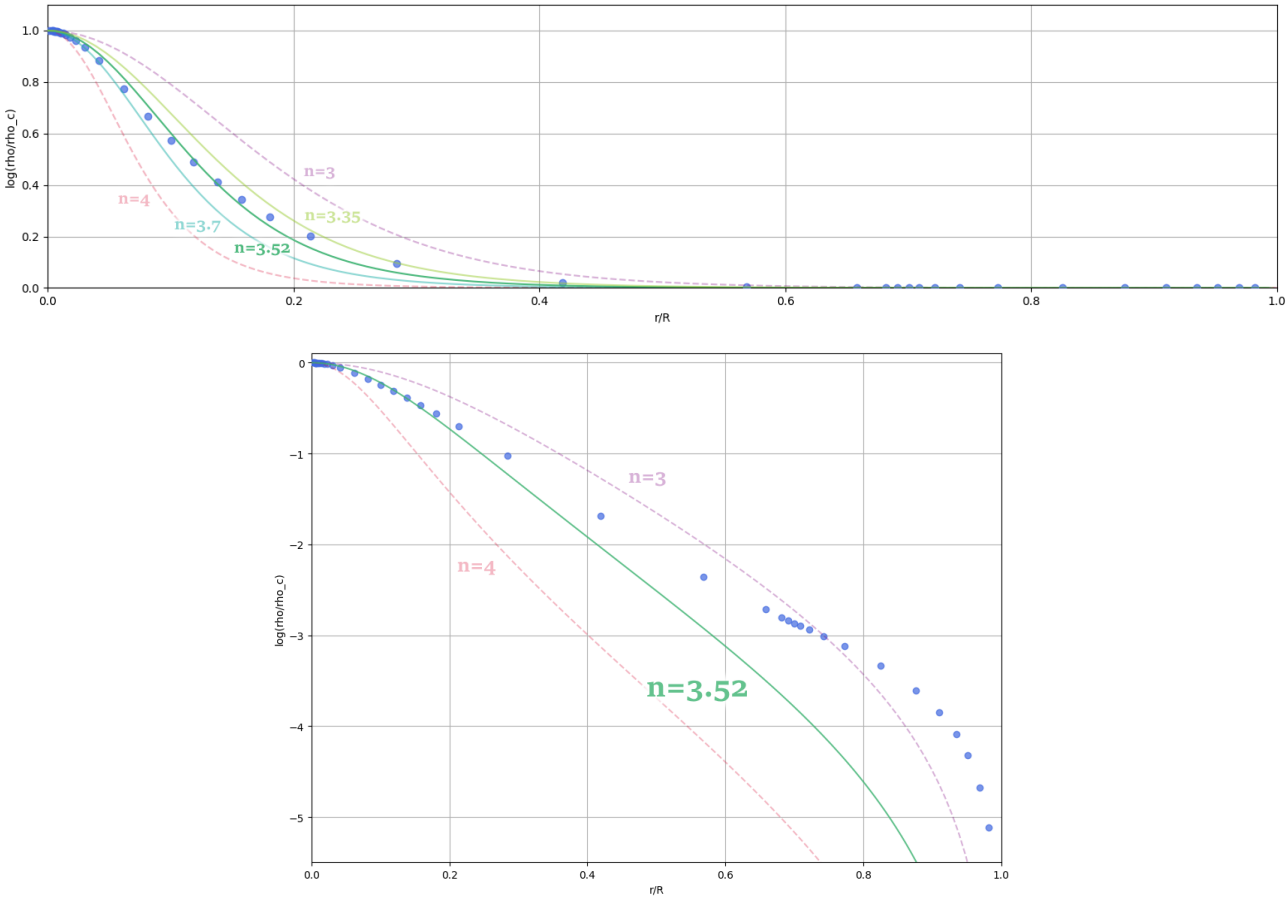


FIG. 13 : Graphe de ρ/ρ_c en fonction de $r/R_\odot$ pour différentes valeurs de n (dont $n = 3.52$).
En bleu, les données expérimentales.
En haut : représentation linéaire des ordonnées ; En bas : logarithmique.

On voit donc ici la limite flagrante du modèle de LANE-EMDEN. Il semblerait que n ne soit *pas constant* le long de $r/R_\odot$. Cette interprétation concorde avec un modèle en couche. Effectivement, on sait que le Soleil est consitué en son sein de trois couches : son noyau – où intervient la fusion nucléaire à densité très élevée, une zone radiative et enfin une zone convective. Comme on a pu rapidement le dire en **2.1.3**, on peut associer à la zone convective un n plus faible, et plus élevé à la zone radiative.

Si l'on analyse l'allure des données expérimentales, on voit une transition assez flagrante d'une zone radiative à une zone convective autour de $r/R_\odot = 0.7$ [11] où l'on passe d'un n entre 3.3 et 3.6 à un n de l'ordre de 2.9 à 2.5. Finalement, à des faibles valeurs de $r/R_\odot$, il semble que n augmente encore légèrement plus, ce qui pourrait correspondre au noyau du Soleil. Néanmoins, du aux réactions thermonucléaires, on modélisera différemment le noyau, notamment via l'équation plus générale d'EMDEN-CHANDRASEKHAR :

$$\frac{1}{\xi^2}\frac{\mathrm{d}}{\mathrm{d}\xi}\left(\xi^2\frac{\mathrm{d}\psi}{\mathrm{d}\xi}\right) = e^{-\psi} \quad \text{avec} \quad \rho := \rho_c\, e^{-\psi}$$

11. Ce n'est d'ailleurs pas étonnant que la base de donnée concentre une densité de relevés plus importante à ces niveaux de profondeur.

CONCLUSION

Dans un premier temps, nous avons pu comprendre au mieux les mécanismes et les subtilités des réseaux de neurones. Les avantages de ces derniers étant la polyvalence – on peut associer la résolution d'équations physiques avec une analyse de données expérimentales entraînant la possibilité de remonter aux grandeurs mises en jeu (notamment avec la méthode inverse des PINNs vue en **1.3.2**), mais aussi la continuité de la résolution obtenue et l'adaptation du système à des équations faisant intervenir des valeurs non définies, notamment aux conditions initiales. Néanmoins, on ne peut pas passer dans l'ombre quelques inconvénients des réseaux neuronaux dans ce contexte. Premièrement, l'outil est tout de même plus compliqué à mettre en place, il demande une programmation plus poussée et nécessite d'avoir accès au programme, et les temps de calculs sont tout de même bien plus longs que pour une intégration plus conventionnelle. Il arrive également que la méthode inverse des PINNs soit entravée pas des particularités comme un changement de variables similaire à celui sur lequel nous nous somme heurtés en **2.1.4**.

Finalement, on peut dire que cette méthode est juste une façon alternative d'aborder un problème d'intégration, à prendre en considération autant qu'une autre. Nous ne somme pas non plus à l'abri que la révolution numérique que nous traversons fasse émerger de nouvelles méthodes utilisant des réseaux de neurones.

Du côté de notre sujet d'astrophysique, nous pouvons avancer que le modèle de la structure interne des étoiles décrit par l'équation de Lane-Emden se définit très bien dans l'idée d'élégance physique. Effectivement, l'équation de Lane-Emden semble offrir un modèle très proche de la réalité au vu de sa simplicité. Elle arrive à distinguer plusieurs types d'étoiles et de mécanismes en jeu au sein de ces dernière par un simple argument n. Pour ce qui est des données expérimentales concernant le Soleil, on retrouve bien un indice polytropique $n_\odot$ de l'ordre de 3 à 3.5 au niveau de la couche radiative comme prévu.

Nous aurions sûrement pu développer l'idée des couches au sein des étoiles – caractérisées par différentes valeurs de n – et ajouter au modèles les effets de rotation et de magnétisme (ce stage n'était malheureusement par assez long pour développer autant ces considérations plus avancées, en plus de l'importance ici apportée aux outils numériques).

A Annexes

A.1 Implémentations Python (Annexes pour la partie 1)

Tous les programmes ci-dessous sont disponibles en version test [12]
à l'adresse GitHub suivante :
`https://github.com/ClementPerrochaud/projet_tuteure_S6`

A.1.1 Algorithmes de descente de gradient

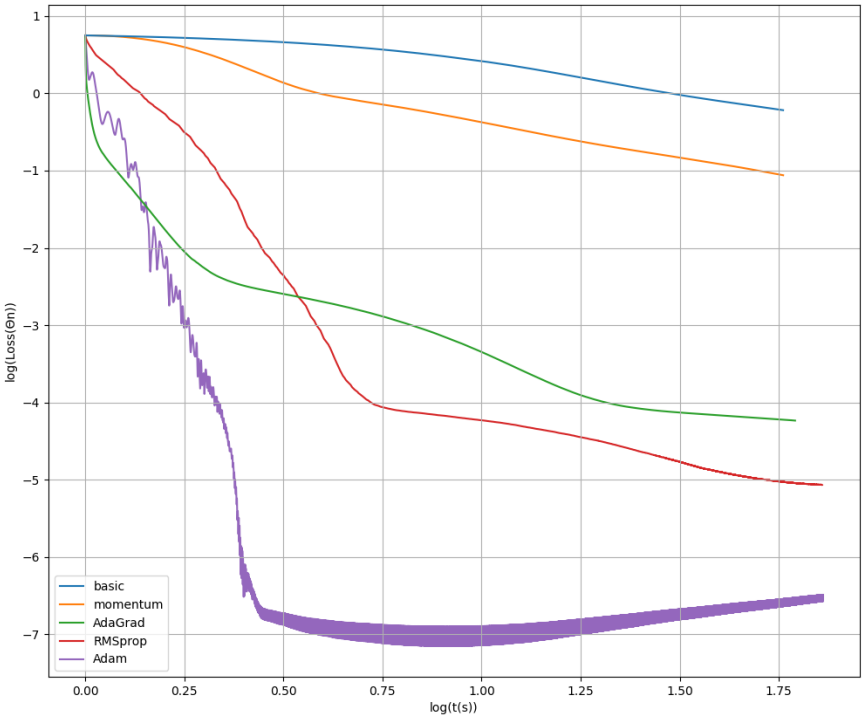


FIG. 14 : Descente de gradient similaire à la **FIG.4** pour $\mathbf{N} = 20$, ici en fonction du temps de calcul. On voit que l'algorithme Adam n'est pas beaucoup plus lent que les autres.

12. Entendre par *version test*, ces même programmes en version plus complète, comportant notamment une partie exécutable (`if __name__ == "__main__": ...`)

A.1.2 Intégration par Runge-Kutta 4

L'équation de Lane-Emden est :

$$\frac{1}{\xi^2}\frac{\mathrm{d}}{\mathrm{d}\xi}(\xi^2\frac{\mathrm{d}\theta}{\mathrm{d}\xi}) = -\theta^n \quad \Leftrightarrow \quad \frac{\mathrm{d}^2\theta}{\mathrm{d}\xi^2} = -\frac{2}{\xi}\frac{\mathrm{d}\theta}{\mathrm{d}\xi} - \theta^n := f(\xi, \theta, \frac{\mathrm{d}\theta}{\mathrm{d}\xi}) = 0$$

connaissant :

$$\theta(\xi = 0) := 1 \quad \text{et} \quad \frac{\mathrm{d}\theta}{\mathrm{d}\xi}(\xi = 0) := 0$$

alors :

$$\begin{aligned}
k_1 &:= f(\xi,\ \theta,\ \frac{\mathrm{d}\theta}{\mathrm{d}\xi}) \\
k_2 &:= f(\xi + \frac{\mathrm{d}\xi}{2},\ \theta + \frac{\mathrm{d}\xi}{2}\frac{\mathrm{d}\theta}{\mathrm{d}\xi},\ \frac{\mathrm{d}\theta}{\mathrm{d}\xi} + \frac{\mathrm{d}\xi}{2}k_1) \\
k_3 &:= f(\xi + \frac{\mathrm{d}\xi}{2},\ \theta + \frac{\mathrm{d}\xi}{2}\frac{\mathrm{d}\theta}{\mathrm{d}\xi} + \frac{\mathrm{d}\xi^2}{4}k_1,\ \frac{\mathrm{d}\theta}{\mathrm{d}\xi} + \frac{\mathrm{d}\xi}{2}k_2) \\
k_4 &:= f(\xi + \mathrm{d}\xi,\ \theta + \mathrm{d}\xi\frac{\mathrm{d}\theta}{\mathrm{d}\xi} + \frac{\mathrm{d}\xi^2}{4}k_2,\ \frac{\mathrm{d}\theta}{\mathrm{d}\xi} + \mathrm{d}\xi k_3)
\end{aligned}$$

enfin :

$$\begin{aligned}
\theta(\xi + \mathrm{d}\xi) &= \theta(\xi) + \mathrm{d}\xi\frac{\mathrm{d}\theta}{\mathrm{d}\xi} + \frac{\mathrm{d}\xi^2}{6}(k_1 + k_2 + k_3) \\
\frac{\mathrm{d}\theta}{\mathrm{d}\xi}(\xi + \mathrm{d}\xi) &= \frac{\mathrm{d}\theta}{\mathrm{d}\xi}(\xi) + \frac{\mathrm{d}\xi}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}$$

Les deux principaux avantages d'utiliser les PINNs plutôt que cette méthode Runge-Kutta sont :

1. ici, le résultat $\theta(\xi)$ est connu de façon discret (même si l'on peut toujours interpoler entre chacun des points) alors que les PINNS offrent une solution directement continue (en supposant que l'on à accès au réseau de neurone).
2. ici, les conditions initiales sont décrites à $\xi = 0$ or l'équation différentielle fait intervenir un $1/\xi$ (il faudrait alors contourner ce problème en posant notre condition initiale à un valeur très faible ϵ après zéro) là où avec les PINNs il suffit de ne pas poser de point de collocation en zéro.

A.2 Compléments physiques (Annexes pour la partie 2)

A.2.1 Résolutions analytiques de l'équation de Lane-Emden pour $n = 0$ et $n = 1$:

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}\xi^2} + \frac{2}{\xi}\frac{\mathrm{d}\theta}{\mathrm{d}\xi} = -1$$

$$\frac{\mathrm{d}^2\theta_h}{\mathrm{d}\xi^2} = -\frac{2}{\xi}\frac{\mathrm{d}\theta_h}{\mathrm{d}\xi} \quad \Rightarrow \quad \frac{\mathrm{d}\theta_h}{\mathrm{d}\xi} = -\frac{\lambda}{\xi^2} \quad \Rightarrow \quad \theta_h = \frac{\lambda}{\xi} + \mu$$
$$\text{et} \quad \theta_p = -\frac{\xi^2}{6} \quad \Rightarrow \quad \theta = \theta_h + \theta_p = \frac{\lambda}{\xi} + \mu - \frac{\xi^2}{6} \ , \ (\lambda, \mu) \in \mathbb{R}^2$$
$$\theta(0) := 1, \ \frac{\mathrm{d}\theta}{\mathrm{d}\xi}(0) := 0 \quad \Rightarrow \quad \lambda = 0, \mu = 1 \quad \Rightarrow \quad \boxed{\theta = 1 - \frac{\xi^2}{6}}$$

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}\xi^2} + \frac{2}{\xi}\frac{\mathrm{d}\theta}{\mathrm{d}\xi} + \theta = 0$$

Changement de variable : $\theta := \exp(-\int \frac{\mathrm{d}\xi}{\xi})\nu = \frac{\nu}{\xi}$
$$\Rightarrow \quad (\frac{2}{\xi^3}\nu - \frac{2}{\xi^2}\frac{\mathrm{d}\nu}{\mathrm{d}\xi} + \frac{1}{\xi}\frac{\mathrm{d}^2\nu}{\mathrm{d}\xi^2}) + \frac{2}{\xi}(-\frac{1}{\xi^2}\nu + \frac{1}{\xi}\frac{\mathrm{d}\nu}{\mathrm{d}\xi}) + \frac{\nu}{\xi} \ = \ \frac{1}{\xi}\frac{\mathrm{d}^2\nu}{\mathrm{d}\xi^2} + \frac{\nu}{\xi} = 0$$
$$\Leftrightarrow \quad \frac{1}{\xi}\frac{\mathrm{d}^2\nu}{\mathrm{d}\xi^2} = -\frac{\nu}{\xi} \quad \Leftrightarrow \quad \frac{\mathrm{d}^2\nu}{\mathrm{d}\xi^2} = -\nu \quad \Leftrightarrow \quad \nu = \lambda\cos(\xi) + \mu\sin(\xi) \ , \ (\lambda, \mu) \in \mathbb{R}^2$$
$$\Rightarrow \quad \theta = \frac{\lambda\cos(\xi) + \mu\sin(\xi)}{\xi}$$
$$\theta(0) := 1, \ \frac{\mathrm{d}\theta}{\mathrm{d}\xi}(0) := 0 \quad \Rightarrow \quad \lambda = 0, \mu = 1 \quad \Rightarrow \quad \boxed{\theta = \frac{\sin\xi}{\xi}}$$

A.2.2 Argument sur la condition au centre :

$$\frac{\mathrm{d}P}{\mathrm{d}r} \overset{(5)}{=} -G\frac{M(r)\rho(r)}{r^2} \underset{r\to 0}{\longrightarrow} 0 \quad \text{car} \quad M(r) \overset{(3)}{\propto} r^3 \text{ et } \rho \text{ borné}$$
$$\text{et} \quad P = K\rho^\Gamma \ \Rightarrow \ \frac{\mathrm{d}P}{\mathrm{d}r} = \Gamma K\rho^{\Gamma-1}\frac{\mathrm{d}\rho}{\mathrm{d}r} \quad \Rightarrow \quad \frac{\mathrm{d}\rho}{\mathrm{d}r} \underset{r\to 0}{\longrightarrow} 0$$
$$r := \alpha\xi \ \Rightarrow \ \frac{\mathrm{d}\theta}{\mathrm{d}\xi} = \alpha\frac{\mathrm{d}\theta}{\mathrm{d}r} \quad \text{et} \quad \rho := \rho_c\theta^n \ \Rightarrow \ \frac{\mathrm{d}\rho}{\mathrm{d}r} = n\rho_c\theta^{n-1}\frac{\mathrm{d}\theta}{\mathrm{d}r}$$
$$\Rightarrow \ n\rho_c\theta^{n-1}\frac{\mathrm{d}\theta}{\mathrm{d}\xi} = \alpha\frac{\mathrm{d}\rho}{\mathrm{d}r} \underset{r\to 0}{\longrightarrow} 0 \quad \Rightarrow \quad \boxed{\frac{\mathrm{d}\theta}{\mathrm{d}\xi} \underset{\xi\to 0}{\longrightarrow} 0}$$

Références

[1] Daniel Etiemble & Fabrice Auzanneau, *Introduction aux réseaux de neurones* H3730 V1, issue de *Technologies de l'information*, 10 mars 2023.

[2] *Wikipedia : Stochastic Gradient Descent*, 16 Avril 2024, https://en.wikipedia.org/wiki/Stochastic_gradient_descent#cite_ref-38

[3] Ben Moseley, *GitHub : Harmonic oscillator PINN*, 28 Août 2021, https://github.com/benmoseley/harmonic-oscillator-pinn/blob/main/Harmonic%20oscillator%20PINN.ipynb

[4] Subrahmanyan Chandrasekhar, *An Introduction to the Study of Stellar Structure*, 1958.

[5] John Bahcall, *Standard Solar Model (BS2005-OP)*, https://www.sns.ias.edu/~jnb/SNdata/solarmodels.html