

Simulation Quantique Neuronale Adaptative PHY08

Bossu Louis-Guillaume, Chen Ruikai, Doubez Natan, Zhang Le

Avril 2024

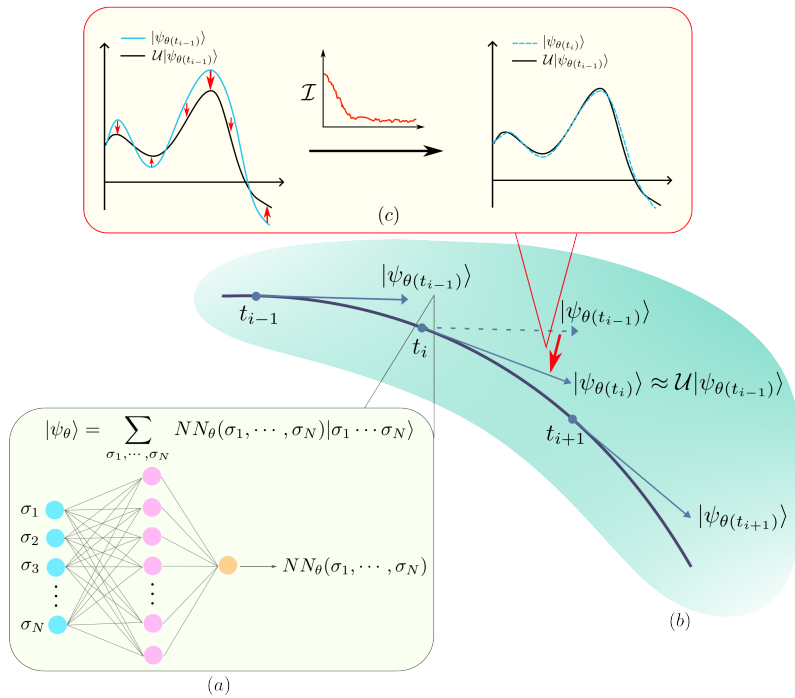


Figure 1: Schéma de divers aspects de notre PSC (a) Le rôle du réseau de neurones comme calculateur des coefficients d'un état quantique. (b) le calcul de l'évolution du système à des temps discrets. (c) la minimisation de l'écart entre l'état simulé et l'état réel

Contents

1	Introduction	3
1.1	Contexte général	3
1.2	Travaux connexes	4
1.3	Nos contributions	4
2	Préliminaire	6
2.1	Représentation théorique d'un état	7
2.2	Machine de Boltzmann Restreinte	7
2.3	Application du réseau de neurones dans la simulation quantique	8
3	Méthode symplectique implicite	10
3.1	Méthode du point milieu	10
3.2	Méthode symplectique d'ordre arbitraire	11
3.3	Méthode symplectique adaptative	14
3.4	Méthode symplectique pour les hamiltoniens dépendant du temps	17
4	Méthode du splitting de l'hamiltonien	20
4.1	Un nouvel ansatz: splitting de l'hamiltonien	20
4.1.1	Approximation théorique du splitting	20
4.1.2	Porte quantique d'opérateur diagonal	22
4.2	Méthode adaptée aux réseaux de neurones	23
5	Méthode de Monte-Carlo variationnelle	24
5.1	Echantillonnage: Metropolis-Hastings	25
5.2	Applications	26
5.2.1	Echantillonnage d'une observable	26
5.2.2	Echantillonnage d'un gradient	26
5.2.3	Propagation d'un système	27
5.2.4	Variable de contrôle	27
5.3	Etat variationnel Monte-Carlo	28
6	Étude sur l'architecture des RBM	29
6.1	Le pouvoir de représentation d'une RBM pour le modèle d'Ising	30
6.2	Le choix de dt optimal	30
6.3	Application : quench dynamique du modèle d'Ising	32
7	Conclusion	34

1 Introduction

1.1 Contexte général

La mécanique quantique est une théorie puissante qui permet théoriquement de prédire l'évolution de tout système. Cela permettrait de déterminer des propriétés de certaines molécules, entre autres leur stabilité ou leur comportement dans certains milieux, ce qui a des applications potentielles dans le secteur médical. Ou encore d'étudier des matériaux pour offrir des réponses rapides aux candidats supraconducteurs, ou même d'aider à construire des ordinateurs quantiques. Cependant si la théorie fonctionne, les demandes en calcul et en mémoire nécessaires pour appliquer naïvement l'équation de Schrödinger ne sont pas atteignables. Pour chaque particule en plus dans le système étudié, la dimension de l'espace des états augmente exponentiellement. Il n'y aurait ainsi pas assez d'atomes dans l'univers observable pour stocker l'état de 100 spins intriqués en inscrivant un bit d'information par atome. Les plus grands calculateurs modernes peinent à atteindre les 55 spins en force brute, sans technique de simplification.

Le calcul est donc une contrainte notable aux applications de la mécanique quantique.

Deux grandes approches

On distingue deux approches principales pour tenter de lever ce problème aujourd'hui.

La première est celle de l'informatique quantique. Avec une architecture informatique reflétant les propriétés quantiques des systèmes à l'étude, l'information quantique doit permettre de résoudre ces problèmes. Par exemple, la complexité du problème à n spins mentionné plus haut deviendrait linéaire, en associant en théorie un *qubit* à chaque spin.

La deuxième approche repose sur l'idée d'approximer la représentation des états quantiques. Au lieu d'une représentation exacte des vecteurs de l'espace de Hilbert, un mouvement récent propose d'opter pour une représentation par réseau de neurones d'un état quantique donné. Ces "approximateurs universels" doivent alors permettre de réduire le problème à une complexité polynomiale. Toutefois, cette solution a l'avantage de s'appuyer sur les techniques de l'informatique classique, qui contrairement aux ordinateurs quantiques existent déjà.

Nos travaux s'inscrivent dans ce second mouvement, utilisant les réseaux de neurones comme approximateurs universels, afin de représenter un état quantique en consommant une quantité polynomiale en le nombre de particules de mémoire et de temps.

1.2 Travaux connexes

Etat quantique neuronal Les états quantiques de réseau neuronal (NQS pour *Neural Quantum State* en anglais) sont une classe générale d'états quantiques variationnels paramétrés en termes de réseau neuronal artificiel. Il a été introduit pour la première fois à l'aide de machines de Boltzmann restreintes (RBM) [1]. Désormais, diverses architectures ont été proposées telles que le corrélateur RBM [3], les réseaux de neurones convolutionnels [4], les réseaux de neurones récurrents [5], etc. Les symétries du système peuvent également être implémentées dans le réseau de neurones, afin d'améliorer la performance [6].

Les applications classique des états quantiques par réseau neuronal comprennent la recherche de l'état fondamental d'un Hamiltonien, la tomographie d'état quantique [8], les simulations des circuits quantiques [9] et des systèmes ouverts [7] ou fermioniques [10], etc.

Evolution en temps réel Les NQS peuvent être aussi utilisés pour simuler l'évolution en temps réel de manière variationnelle grâce à une procédure connue comme Monte-Carlo variationnel dépendant du temps (t-VMC) [4]. Une autre approche, qui s'appelle *Dynamique Quantique Variationnelle projective* (p-VQD) [12], cherche d'appliquer une méthode implicite en résolvant un problème d'optimisation. Cependant, des inconvénients sont trouvés pour ces deux approches, et une amélioration de la complexité de l'échantillon de la deuxième approche est proposé dans [11], sous le nom du *Monte-Carlo variationnel dépendant du temps projective* (p-tVMC), qui cherche à minimiser directement l'infidélité entre $|\hat{\psi}\rangle$ et $U|\psi\rangle$.

1.3 Nos contributions

Le domaine des états quantiques neuronaux est suffisamment vaste pour offrir de nombreuses pistes de recherche inédites, surtout dans la dynamique de l'évolution. Notre objectif initial était de simuler l'évolution de spins disposés selon le modèle de Ising soumis à deux champs magnétiques. Nous utilisons les unités naturelles pour les grandeurs telles que le temps ou la magnétisation. Cet objectif nous a amenés à explorer plusieurs aspects fondamentaux du problème.

L'architecture du réseau de neurones. Le cœur du problème étant la quantité de mémoire nécessaire pour stocker l'ensemble des coordonnées d'un état quantique, nous avons utilisé Jax pour développer des réseaux de neurones qui compressent ces états. L'idée derrière cette compression est de voir un réseau de neurone comme une fonction qui prend en argument un vecteur de la base et renvoie la coordonnée de l'état qu'il représente selon ce vecteur. Il s'agit donc d'un outil qui recalcule les coordonnées au lieu de les stocker en mémoire. Formellement le réseau est la donnée d'une matrice de poids W et d'un vecteur de biais b , et les coordonnées correspondent aux produit scalaire avec les vecteurs de la base tensorielle calculés selon l'équation suivante:

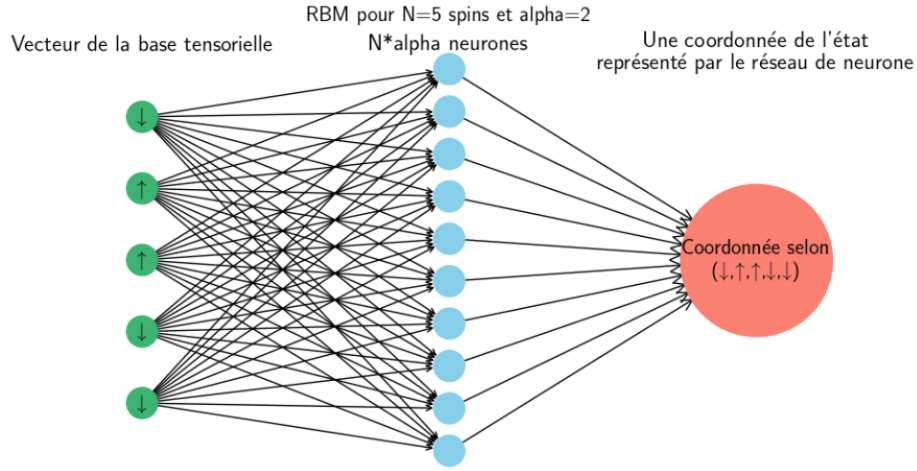


Figure 2: Représentation d'un réseau de neurone qui représente un état quantique d'un système de 5 spins. Dans notre programme, \uparrow est représenté par 1 et \downarrow par -1. Chaque flèche représente une multiplication par un paramètre du réseau puis une addition. La taille de la couche interne est égale au nombre de spins fois un paramètre entier alpha. Elle n'a pas de sens physique et sert uniquement à calculer les coordonnées de l'état quantique en utilisant une quantité plus faible de mémoire que si toutes les coordonnées étaient stockées.

$$\langle x|\psi\rangle = \prod_i \cosh \left[\sum_j W_{ij} x_j + b_i \right] \quad (1)$$

Nous avons évalué différentes architectures en analysant l'impact du nombre de couches, de la taille des couches, de la fonction de perte, et des méthodes d'initialisation des paramètres du réseau.

Le calcul des observables: l'échantillonnage. Une fois que l'état quantique est compressé, il est nécessaire de permettre le calcul d'observables sans calculer les coordonnées dans l'ensemble des états de la base. Pour réduire la complexité des calculs, nous avons choisi une méthode de Monte-Carlo par Chaîne de Markov en utilisant l'algorithme de Metropolis-Hastings. Cette technique nous permet d'estimer les observables à partir d'un sous-ensemble d'états. Nous avons amélioré l'efficacité de cette méthode en utilisant des variables de contrôle pour diminuer la variance des estimations.

L'équation d'évolution. Diagonaliser l'Hamiltonien étant coûteux, nous avons préféré des méthodes d'approximation. Initialement nous avons considéré la méthode d'Euler qui consiste à linéariser l'opérateur d'évolution $U_{dt} \sim I - idtH$.

Ainsi, à partir d'un état ψ_θ , il suffit de chercher un état le plus proche possible de $U_{dt}\psi_\theta$, c'est à dire qui minimise l'infidélité $\mathcal{I}(|\psi_{\hat{\theta}}\rangle, U_{dt}|\psi_\theta\rangle)$. Suite aux travaux de [11], nous avons privilégié une forme généralisée qui minimise l'infidélité $\mathcal{I}(V|\psi_{\hat{\theta}}\rangle, U|\psi_\theta\rangle)$. Cette approche nous permet d'approximer un nouvel état $|\psi_{\hat{\theta}}\rangle \approx V^{-1}U|\psi_\theta\rangle$. Lorsque $V = I$, notre méthode correspond à celle décrite dans [11] sous le nom de t-pVMC. En choisissant $V = I + \frac{idt\hat{H}}{2}$ et $U = I - \frac{idt\hat{H}}{2}$, elle s'aligne sur la méthode du point milieu mentionnée dans [4]. Nous avons poussé la généralisation plus loin en développant un schéma à s optimisations, capable d'atteindre un ordre de $2s$, dont les détails seront exposés dans la section 3. Cette méthode à ordre arbitraire pourrait significativement améliorer l'efficacité lors de la simulation de systèmes de grande taille. Dans nos expérimentations sur le "sudden quench" du modèle d'Ising, nous avons observé une amélioration de l'efficacité de la méthode du point milieu, multipliant par 3 à 5 fois les performances. Nous proposons également une adaptation de cette méthode pour les Hamiltoniens dépendants du temps, surmontant ainsi les défis posés par les simulations classiques de ces systèmes. Cette nouvelle version a également été utilisée pour simuler la dynamique de "slow quench" dans le modèle d'Ising. À notre connaissance, ces outils n'avaient jamais été utilisés explicitement dans ce contexte. Dans la pratique, nous avons réalisé nos dernières simulations avec $s = 2$ ce qui correspond à approximer l'opérateur d'évolution par:

$$U_{dt} = e^{idt\hat{H}} = (I + \frac{idt}{4}(1 + \frac{i}{\sqrt{3}})\hat{H})^{-1}(I + \frac{idt}{4}(1 - \frac{i}{\sqrt{3}})\hat{H})(I + \frac{idt}{2}\hat{H})^{-1}(I - \frac{idt}{2}\hat{H}) + o(dt^5)$$

La dynamique de l'évolution. Nous avons étudié l'influence de plusieurs paramètres sur la vitesse des simulations. Le pas de temps, le nombre d'étapes d'entraînement du réseau, le nombre de paramètres dans le réseau, et le nombre d'échantillons dans les méthodes de Monte-Carlo ont été optimisés pour équilibrer précision et temps d'exécution.

Nous avons également amélioré la structure de la fonction (l'ansatz) avec la méthode du splitting de l'hamiltonien : nous utilisons un nouvel ansatz qui combine le réseau de neurones et le calcul exact. Plus précisément, nous effectuons un calcul exact pour la partie diagonale de l'Hamiltonien dont on connaît déjà les valeurs propres, tandis que pour la partie hors-diagonale, le réseau de neurones performe comme précédemment.

2 Préliminaire

Nous considérons S un système contenant N particules de spin $1/2$, \hat{H} l'opérateur hamiltonien et \mathcal{H} l'espace hilbertien des états.

2.1 Représentation théorique d'un état

Les spins individuels dans notre système sont déjà identifiés comme ayant deux états propres, $|\uparrow\rangle$ et $|\downarrow\rangle$, constituant ainsi un espace \mathcal{H}_0 de dimension 2. Cette structure se généralise pour N particules par le produit tensoriel:

$$\mathcal{H} = \otimes_{i=1}^N \mathcal{H}_0$$

avec une base tensorielle e comprenant des vecteurs de la forme:

$$\otimes_{i=1}^N s_i, \quad s_i \in \{|\uparrow\rangle, |\downarrow\rangle\}.$$

Ainsi, la dimension de cet espace est 2^N . Pour tout état ψ dans \mathcal{H} , il est couramment représenté comme suit:

$$\psi = \sum_{x \in e} \lambda_x x.$$

Pour représenter cet état numériquement, il est nécessaire de stocker un vecteur de longueur 2^N . Cette taille croît exponentiellement avec le nombre de particules, rendant les calculs explicites sur des systèmes de grande échelle pratiquement infeasibles. Toutefois, bien que l'espace de Hilbert des systèmes quantiques soit vaste, les états physiquement significatifs durant une évolution sont souvent limités et peuvent être décrits avec moins de degrés de liberté, un fait qui est central à notre approche.

2.2 Machine de Boltzmann Restreinte

Nous présentons ici la machine de Boltzmann Restreinte (Restricted Boltzmann Machine, ou RBM), un type de réseau neuronal spécifique. La RBM se compose de deux couches : une couche visible qui reçoit en entrée un vecteur x de taille N , et une couche cachée de taille αN . La sortie de la machine, y , est donnée par la formule suivante :

$$y = G(W \cdot x + b), \quad W \in M_{N \times \alpha N}(\mathbb{R}), \quad b \in \mathbb{R}^{\alpha N}$$

où W et b sont les paramètres de la machine qui définissent une transformation affine, et G est une fonction d'activation non-linéaire définie a priori.

Un développement notable a été l'utilisation d'une RBM pour modéliser l'état d'un système quantique[1]. Dans cette application, la RBM reçoit en entrée N paramètres $x = (\sigma_1, \dots, \sigma_N)$, chacun pouvant prendre la valeur 1 ou -1 pour spin up ou down. La sortie est un nombre complexe qui correspond au produit scalaire de l'état quantique avec l'état donné en entrée $|\sigma_1, \dots, \sigma_N\rangle$. Une fois les paramètres de la machine ajustés, l'état quantique représenté s'exprime comme suit :

$$|\psi_\theta\rangle = \sum_{\sigma_1, \dots, \sigma_N} \exp(NN_\theta(\sigma_1, \dots, \sigma_N)) |\sigma_1 \dots \sigma_N\rangle.$$

Cette méthode permet de représenter un état quantique caractérisé par 2^N paramètres avec une RBM dotée de l'ordre de N^2 paramètres, réduisant ainsi la complexité de la représentation. Il reste à déterminer comment sélectionner de manière optimale les paramètres du réseau et à quantifier l'erreur induite.

2.3 Application du réseau de neurones dans la simulation quantique

Approximation d'un état quantique Le premier problème pour tester cette approche est d'essayer d'approximer un état donné $|\psi\rangle$ par notre machine. Pour cela, nous initialisons les paramètres de la RBM aléatoirement, et nous voulons définir une fonction de perte dans notre problème d'optimisation.

Nous considérons la fidélité :

$$\mathcal{F} = \frac{\langle \psi | \phi \rangle \langle \phi | \psi \rangle}{\langle \psi | \psi \rangle \langle \phi | \phi \rangle}.$$

Car elle a la propriété d'être invariante par changement de phase. Notons qu'elle parcourt l'intervalle $[0, 1]$ et elle atteint 1 lorsque les deux états sont égaux. Nous choisissons donc comme la fonction de perte l'infidélité:

$$\mathcal{I} = \text{Infidélité}(|\psi\rangle, |\phi\rangle) = 1 - \mathcal{F}(|\psi\rangle, |\phi\rangle)$$

Maintenant nous reformulons le problème d'approximation un état comme un problème variationnel :

Problème. Soit $|\psi_\theta\rangle$ un état variationnel représenté par une RBM et $|\psi\rangle$ un état cible. Approcher $|\psi\rangle$ par $|\psi_\theta\rangle$ correspond à trouver $\theta = (W, b)$ tel que $\mathcal{I}(|\psi\rangle, |\psi_\theta\rangle)$ est aussi petit que possible. Nous formulons ce problème comme

$$\theta = \arg \min_{\hat{\theta}} \mathcal{I}(|\psi\rangle, |\psi_{\hat{\theta}}\rangle)$$

Il faut remarquer que le calcul d'infidélité est assez coûteux si nous voulons la valeur exacte, une approche numérique de calcul va être introduit dans la section 5. Dans un premier temps, nous ignorons ce type de problème. Une fois que la fonction de perte est définie, nous pouvons entraîner notre modèle. Dans la plupart des cas, nous n'attendons à descendre l'infidélité à 0 car il y a moins de paramètres dans la machine que dans l'état réel. Cependant, nous espérons majorer borner l'erreur sous un seuil suffisamment faible pour que la simulation reste utile.

Le rôle de α est assez fondamental et sera la cible principale de nos premières études. Il détermine la taille de la couche cachée, donc le nombre de paramètres dans la machine. Il modifie à la fois le pouvoir de représentation de la machine et le coût du calcul. Notre première démarche était d'initialiser avec $\alpha = 1$ jusqu'à ce que l'erreur entre $|\psi\rangle$ et $|\psi_\theta\rangle$ devienne trop importante, c'est à dire qu'au cours de l'entraînement, notre réseau de neurone n'arrive jamais à simuler

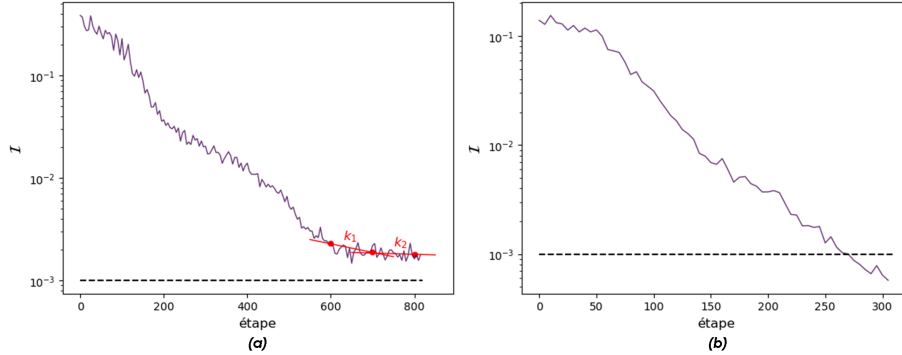


Figure 3: Conditions d'arrêt pour une infidélité cible de 10^{-3} (a) Cas de non-convergence : la moyenne glissante de l'erreur est calculée toutes les 100 étapes. Si la pente entre deux erreurs moyennes consécutives est suffisamment petite deux fois d'affilée : $|k_1|, |k_2| < 0.1$, un arrêt anticipé est effectué. (b) Cas de convergence : à chaque étape, si l'erreur lissée (moyenne glissante sur 20 périodes) est inférieure à la cible, après un temps de patience de 30 étapes, l'entraînement s'arrête.

fidèlement l'état réel. Dans ce cas nous l'arrêtons de façon anticipée. En pratique, nous nous intéressons plutôt à la moyenne de l'infidélité au cours de l'entraînement pour éviter les oscillations. Dans le cas de convergence où passons en dessous de l'erreur cible, un arrêt anticipé peut être effectué après une patience fixée (Figure 3(b)). Dans le cas où nous ne pouvons pas atteindre le cible, nous calculons la pente de l'infidélité pour quantifier la vitesse de l'amélioration de notre représentation. Si celle-ci devient trop basse, nous considérons que nous avons atteint la convergence et que nous n'atteindrons jamais la cible pour cette valeur de α (Figure 3(a)).

Le cas de non-convergence peut signifier que α n'est pas suffisamment grand pour le problème. Nous allons étudier l'architecture de RMB ainsi que son pouvoir de représentation dans la section 6. Une autre application directe est de trouver l'état fondamental d'un hamiltonien H en choisissant la fonction de perte comme $L(\theta) = \frac{\langle \psi_\theta | H | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle}$.

Evolution en temps réel Pour le premier temps, nous utilisons la méthode introduite dans [11] pour l'évolution en temps réel. Plus précisément, nous considérons le problème de trouver les paramètres d'un état variationnel $|\psi_\theta\rangle$ tels qu'il se rapproche de l'état $U|\psi_\theta\rangle$, où θ sont connus et U est une transformation arbitraire. Considérant une distance comme l'infidélité \mathcal{I} , cela peut être exprimé comme le problème d'optimisation suivant:

Problème. Soit $|\psi_\theta\rangle$ un état variationnel paramétré par une RBM qui représente l'état en t . Trouver l'état $|\psi_{\theta'}\rangle$ en t' pour un opérateur $U(t, t')$ donné correspond à trouver θ' tel que $\mathcal{I}(|\psi_{\theta'}\rangle, U|\psi_\theta\rangle)$ est aussi petit que possible. Nous formulons

ce problème comme

$$\theta' = \arg \min_{\hat{\theta}} \mathcal{I}(|\psi_{\hat{\theta}}\rangle, U|\psi_{\theta}\rangle) \quad (2)$$

En particulier, nous demandons de plus que U soit une matrice creuse pour l'efficacité de calcul dans le cas d'état variationnel Monte-Carlo, nous expliquerons plus de détails dans la section 5. Si l'hamiltonien d'un système n'a que des effets locaux, par exemple dans le modèle d'Ising,

$$\hat{H} = h \sum_i \sigma_i^x + J \sum_i \sigma_i^z \sigma_{i+1}^z$$

Alors $U(t, t + dt) := I - idt\hat{H}$ est un choix naturel. Nous remarquons aussi que, dans le problème d'optimisation de (2), nous pouvons initialiser $\hat{\theta}$ par θ car a priori, θ' est proche de θ si t' est proche de t .

3 Méthode symplectique implicite

Jusqu'à présent, nous avons discuté les capacités du réseau de neurones pour résoudre les problèmes de dynamique. Plus précisément, étant donné un état $|\psi_{\theta}\rangle$ paramétré par un réseau de neurones, nous pouvons obtenir un nouveau état paramétré $|\psi_{\hat{\theta}}\rangle \approx U|\psi_{\theta}\rangle$ en minimisant l'infidélité $\mathcal{I}(|\psi_{\hat{\theta}}\rangle, U|\psi_{\theta}\rangle)$ où U est une matrice creuse (dans le cas précédent, $U = I - idt\hat{H}$).

L'idée principal est que nous pouvons échantillonner les états $|\psi_{\hat{\theta}}\rangle$ et $|U\psi_{\theta}\rangle$ en utilisant un algorithme MCMC efficacement pour calculer leur infidélité. Une généralisation naturelle est d'échantillonner les états $V|\psi_{\hat{\theta}}\rangle$ et $U|\psi_{\theta}\rangle$ en même temps pour deux matrices creuses U et V . Dans ce cas, en minimisant l'infidélité $\mathcal{I}(V|\psi_{\hat{\theta}}\rangle, U|\psi_{\theta}\rangle)$, nous pouvons obtenir un état $|\psi_{\hat{\theta}}\rangle \approx V^{-1}U|\psi_{\theta}\rangle$.

Comme ce procédé permet d'utiliser des inversions de matrice, il ouvre la porte aux méthodes implicites réalisées par les réseaux neurones. Il est intéressant de noter que, pour ces réseaux, multiplier un état par une matrice ou par l'inverse de cette matrice génère des coûts de calcul similaires qui sont principalement attribuables aux opérations d'échantillonnage. Cela représente un avantage majeur des réseaux de neurones sur d'autres méthodes numériques.

Dans cette section, nous allons discuter les méthodes symplectique implicites adaptées pour les réseaux neurones. Nous allons commencer par la méthode de point milieu implicite, puis nous allons généraliser cette idée pour les méthodes symplectiques d'ordre arbitraire. Une méthode adaptives sera aussi discutée pour déterminer le pas de temps adaptive. Enfin, nous généralisons ces méthodes pour les Hamiltoniens dépendants du temps.

3.1 Méthode du point milieu

La méthode du point milieu est une méthode en une étape pour résoudre numériquement l'équation différentielle $y'(t) = f(t, y(t))$ avec $y(t_0) = y_0$. La

méthode consiste à calculer y_{n+1} à partir de y_n en utilisant la formule suivante:

$$y_{n+1} = y_n + dt f \left(t_n + \frac{dt}{2}, \frac{1}{2}(y_n + y_{n+1}) \right) \quad (3)$$

Dans le cadre de l'équation de Schrödinger, nous avons $f(t, y(t)) = -i\hat{H}y(t)$, donc la méthode du point milieu devient:

$$|\psi_{n+1}\rangle = |\psi_n\rangle - \frac{idt\hat{H}}{2}(|\psi_n\rangle + |\psi_{n+1}\rangle) \quad (4)$$

et donc

$$|\psi_{n+1}\rangle = \left(I + \frac{idt\hat{H}}{2} \right)^{-1} \left(I - \frac{idt\hat{H}}{2} \right) |\psi_n\rangle \quad (5)$$

A partir de l'équation (5), nous pouvons formuler un algorithme pour résoudre l'équation de Schrödinger paramétrée par un réseau de neurones. C'est une méthode d'ordre 2 qui effectue une optimisation à chaque pas de temps.

Algorithm 1 Midpoint method

Require: $|\psi_{\theta_0}\rangle, \hat{H}, dt, t_{\max}$ ▷ $|\psi_{\theta_0}\rangle$ is the initial paramized state
1: $t \leftarrow 0$
2: $\theta \leftarrow \theta_0$
3: **while** $t < t_{\max}$ **do**
4: $\theta \leftarrow \arg \min_{\tilde{\theta}} \mathcal{I}((I + idt\hat{H}/2)|\psi_{\tilde{\theta}}\rangle, (I - idt\hat{H}/2)|\psi_{\theta}\rangle)$ ▷ Optimization
5: $t \leftarrow t + dt$

La méthode du point milieu est une méthode symplectique implicite d'ordre 2. C'est une méthode très simple et efficace pour résoudre la dynamique. Cependant, pour un système plus large, nous avons besoin d'une méthode d'ordre plus élevé qui est donc potentiellement plus efficace. Nous allons discuter ces méthodes dans la section suivante.

3.2 Méthode symplectique d'ordre arbitraire

Inspiré par la méthode du point milieu, nous pouvons généraliser cette idée pour les méthodes symplectiques d'ordre arbitraire. Nous allons discuter la méthode U_s définie par:

$$|\psi_{n+1}\rangle = U_s |\psi_n\rangle, \quad U_s = A_s^{-1} B_s \cdots A_1^{-1} B_1$$

où A_i et B_i sont des matrices bien choisies. Au lieu de faire une optimisation comme dans la méthode du point milieu, nous faisons s optimisations pour chaque pas de temps. La réalisation par le réseau de neurones est donné dans Algorithme 5.

Algorithm 2 U_s method

Require: $|\psi_{\theta_0}\rangle, \hat{H}, dt, t_{\max}, A_i, B_i, i = 1, \dots, s$ $\triangleright U_s = A_s^{-1}B_s \cdots A_1^{-1}B_1$
1: $t \leftarrow 0$
2: **while** $t < t_{\max}$ **do**
3: **for** $i = 1$ to s **do**
4: $\theta_i \leftarrow \arg \min_{\tilde{\theta}} \mathcal{I}(A_i|\psi_{\tilde{\theta}}\rangle, B_i|\psi_{\theta_{i-1}}\rangle)$ $\triangleright |\psi_{\theta_i}\rangle \approx A_i^{-1}B_i|\psi_{\theta_{i-1}}\rangle$
5: $t \leftarrow t + dt$

s	a_1	a_2	a_3	a_4
1	1/2			
2	$\frac{1}{4} \left(1 + \frac{i}{\sqrt{3}}\right)$	$\frac{1}{4} \left(1 - \frac{i}{\sqrt{3}}\right)$		
3	0.2153	$0.1423 - 0.1358i$	$0.1423 + 0.1358i$	
4	$0.0916 - 0.1157i$	$0.0916 + 0.1157i$	$0.1584 - 0.0474i$	$0.1584 + 0.0474i$

Table 1: Valeurs de a_i pour différentes valeurs de s

Une manière de choisir A_i et B_i est

$$A_i = \left(I + a_i idt \hat{H}\right), \quad B_i = \left(I + b_i idt \hat{H}\right)$$

avec $a_i, b_i \in \mathbb{C}$ t.q.

$$e^{-idt \hat{H}} = \prod_{i=1}^s A_i^{-1} B_i + \mathcal{O}(dt^{2s+1})$$

Nous remarquons que dans ce cas, nous avons toujours $b_i = -a_i$, U_s devient une méthode symplectique d'ordre $2s$. Les valeurs de a_i pour s petit sont données dans le tableau 1. Pour les cas générales, voir Corollaire 3.1 dans la section 3.4. Une expérience numérique avec les calculs exactes pour le modèle d'Ising de $N = 9$ est donnée dans la figure 5. Nous voyons que la méthode U_s est généralement plus efficace et d'ordre $2s$. Cependant, le coût de calcul est aussi proportionnel à s . Nous faisons aussi une simulation avec les réseaux neurones en utilisant RBM de différents α . Ici nous prenons 10 spins de modèle d'Ising dans 1d, et nous considérons toujours le quench dynamique de $J = \infty \rightarrow 1$. Nous prenons le pas de temps $dt = 0.1$ et pour chaque pas de temps, nous initialisons le réseau de neurones avec les paramètres d'état précédent, et puis nous entraînons pour 1000α étapes avec l'optimiseur Adam. Les résultats sont montrés dans la figure 4.

Une question naturelle est de savoir comment choisir s et dt pour un système donné, particulièrement pour les systèmes très grands où l'évaluation de l'erreur n'est plus possible. Nous allons discuter une méthode adaptative pour déterminer dt dans la section suivante.

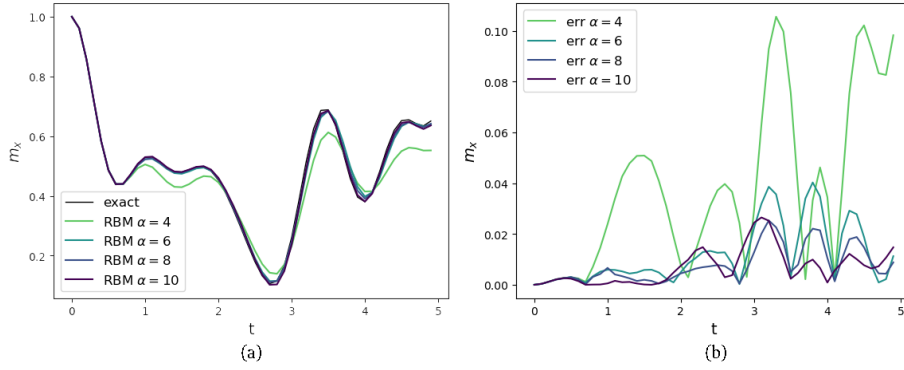


Figure 4: Simulation de la quench dynamique avec RBM en utilisant la méthode U_2 . a) L'évolution de $m_x = \frac{1}{N} \sum_i \langle \sigma_i^x \rangle$ b) L'évolution des erreurs.

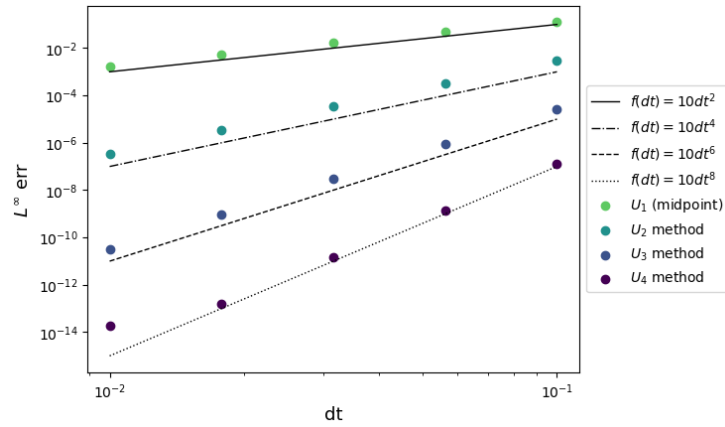


Figure 5: Erreur L^∞ en fonction de dt pour différentes valeurs de s . Ici nous considérons le quench dynamique $J = \infty \rightarrow J = 1$ de modèle d'Ising de $N = 9$. Pour $t_{\max} = 1$, nous traçons l'erreur $L^\infty([0, t_{\max}])$ entre la solution exacte et la solution numérique obtenue par les méthodes U_s .

3.3 Méthode symplectique adaptive

Une méthode adaptive est une méthode qui adapte le pas de temps dt en réaction à l'erreur estimée à chaque étape. Cette approche consiste à mesurer l'erreur à chaque pas en utilisant soit une méthode d'un ordre supérieur, soit en réduisant le pas de temps, puis à ajuster dt en conséquence. Plus précisément, pour les méthodes $U_s(dt)$, l'erreur locale est $C \cdot dt^{2s+1}$ où C est un scalaire qu'on suppose constant dans l'intervalle $[t, t + 2dt]$. Dans ce cas, nous avons:

$$\|U_s(dt)^2|\psi_t\rangle - |\psi_{t+2dt}\rangle\| = 2Cdt^{2s+1} + \mathcal{O}(dt^{2s+2})$$

$$\|U_s(2dt)|\psi_t\rangle - |\psi_{t+2dt}\rangle\| = C(2dt)^{2s+1} + \mathcal{O}(dt^{2s+2})$$

donc si nous notons $E_{\text{loc}} = \|U_s(dt)^2|\psi_t\rangle - |\psi_{t+2dt}\rangle\|$, nous avons

$$C(2^{2s+1} - 2)dt^{2s+1} + \mathcal{O}(dt^{2s+2}) \leq E_{\text{loc}} \leq C(2^{2s+1} + 2)dt^{2s+1} + \mathcal{O}(dt^{2s+2}).$$

Nous pouvons déduire une estimation de la constante C :

$$\frac{E_{\text{loc}}}{2^{2s+1} + 2} \leq Cdt^{2s+1} \leq \frac{E_{\text{loc}}}{2^{2s+1} - 2}$$

Si une erreur tolérée tol est fixée, nous voulons que l'erreur locale soit près de tol , donc pour le prochain pas de temps, nous changeons dt à qdt où q est donné par

$$2Cq^{2s+1}dt^{2s+1} \approx tol$$

Nous pouvons prendre q comme

$$q = \left(\frac{(2^{2s+1} - 1)tol}{2E_{\text{loc}}} \right)^{\frac{1}{2s+1}}.$$

Dans le pratique, nous ne souhaitons pas changer dt trop brusquement, donc nous restreignons q à un intervalle $[q_{\text{min}}, q_{\text{max}}]$, et nous multiplierons q par un facteur $\alpha < 1$ pour être sûr d'avoir une erreur locale plus petite que tol . Nous pouvons résumer la méthode adaptive par l'algorithme suivant.

Une expérience numérique est faite pour tester la méthode U_s : Ici nous prenons un modèle d'Ising de $N = 3 \times 3$ spins, et nous considérons la quench dynamique $J = \infty \rightarrow 1$, c'est à dire que

$$\hat{H} = \sum_i \sigma_i^x + \sum_{i,j \in E} \sigma_i^z \sigma_j^z, \quad \text{avec } E \text{ les arêtes du graphe,}$$

et $|\psi_0\rangle = \left(\frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle) \right)^{\otimes N}$. Une illustration du modèle est dans la figure 6. Ensuite, nous effectuons une méthode U_2 adaptive et prenons l'infidélité comme la mesure d'erreur, les résultats sont montrés dans la figure 7.

Algorithm 3 Adaptive U_s method

Require: $|\psi_0\rangle, U_s, dt_0, t_{\max}, \mathcal{I}, tol$ $\triangleright \mathcal{I}$ is a measure of error

```
1:  $t \leftarrow 0$ 
2:  $dt \leftarrow dt_0$ 
3:  $i \leftarrow 1$ 
4: while  $t < t_{\max}$  do
5:    $|\psi_i^1\rangle = U_s(2dt)|\psi_{i-1}\rangle$ 
6:    $|\psi_i^2\rangle = U_s(dt)^2|\psi_{i-1}\rangle$ 
7:    $E \leftarrow \mathcal{I}(|\psi_i^1\rangle, |\psi_i^2\rangle)$ 
8:    $q \leftarrow (tol * (2^{2s+1}) - 2) / 2E)^{1/(2s+1)}$ 
9:    $q \leftarrow \min(\max(q, 0.2), 2)$ 
10:  if  $q > 1$  then  $\triangleright$  If current error is less than tol
11:     $t \leftarrow t + 2 * dt$ 
12:     $dt \leftarrow 0.9 * q * dt$ 
13:     $|\psi_i\rangle = |\psi_i^2\rangle$   $\triangleright$  We take  $|\psi_i\rangle$  to be  $|\psi_i^2\rangle$ 
14:     $i \leftarrow i + 1$ 
15:  else  $\triangleright$  Otherwise, we update  $dt$  and repeat the process
16:     $dt \leftarrow 0.9 * q * dt$ 
```

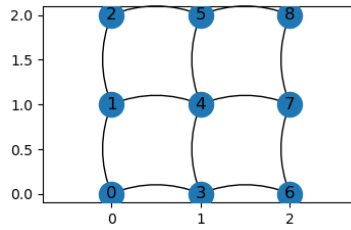


Figure 6: Illustration du modèle d'Ising 3×3 .

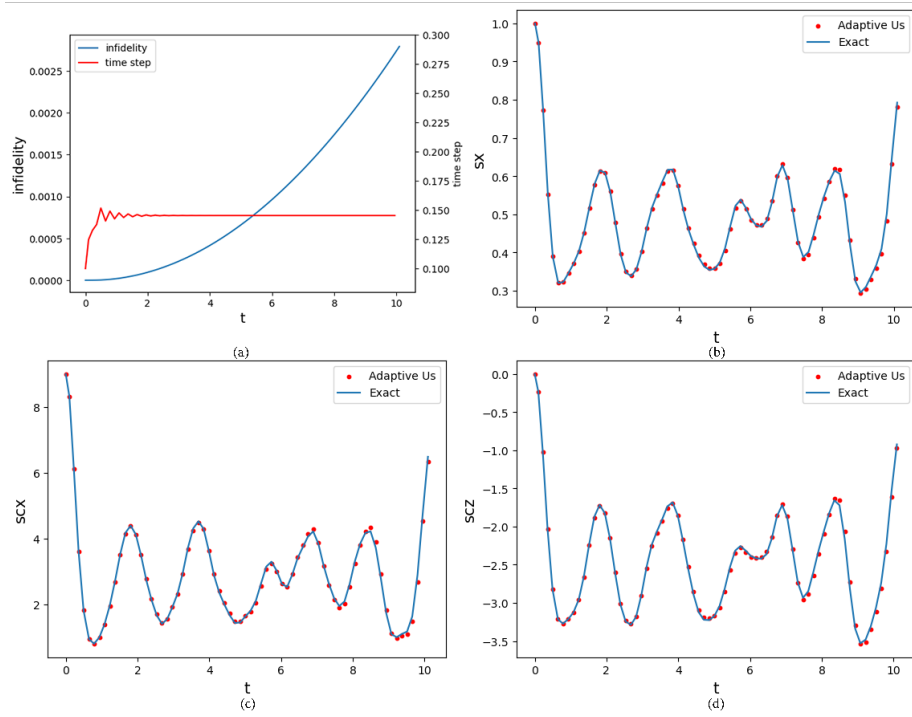


Figure 7: Méthode U_s adaptive. a) L'évolution de dt et de l'infidélité entre l'état obtenu par U_s adaptive et l'état exacte en fonction du temps. b) Mesure de $sx := \frac{1}{N} \sum_i \sigma_i^x$ pour l'état obtenu par U_s adaptive et pour l'état exacte. c) Mesure de $scx := \sum_i \sigma_i^x \sigma_{i+1}^x$. d) Mesure de $scz := \sum_i \sigma_i^z \sigma_{i+1}^z$

3.4 Méthode symplectique pour les hamiltoniens dépendant du temps

La simulation classique avec les états quantiques neuronaux pour l'Hamiltonien dépendant du temps est montré difficile dans le cas général. Dans cette section, nous allons généraliser les méthodes U_s pour l'Hamiltonien dépendant du temps qui a un même coût du calcul, ce qui résoud le problème de simulation pour un système plus général. Les théorèmes principaux pour les deux premiers ordres sont donnés comme les suivants:

Théorème 3.1 (Méthode du point milieu U_1 pour l'Hamiltonien dépendant du temps). *Supposons $H(t)$ est un hamiltonien dépendant du temps, un opérateur d'évolution d'ordre 2 est écrit comme*

$$U(t, t + dt) = \left(I + \frac{1}{2} idtH \left(t + \frac{dt}{2} \right) \right)^{-1} \left(I - \frac{1}{2} idtH \left(t + \frac{dt}{2} \right) \right)$$

Théorème 3.2 (Méthode U_2 pour l'Hamiltonien dépendant du temps). *Supposons $H(t)$ est un hamiltonien dépendant du temps, un opérateur d'évolution d'ordre 4 est écrit comme*

$$U(t, t + dt) = \left(I + a \cdot \tilde{H} \right)^{-1} \left(I + b \cdot \tilde{H} \right) \left(I + c \cdot \tilde{H} \right)^{-1} \left(I + d \cdot \tilde{H} \right)$$

avec

$$a, c = -\frac{1}{4} \pm \frac{\sqrt{3}}{12}i, \quad b = -a, \quad d = -c$$

et

$$\begin{aligned} \tilde{H} &= -i \int_t^{t+dt} H(s) ds + \frac{dt^3}{12} [H(t), H'(t)] + \frac{dt^4}{24} [H(t), H''(t)] \\ &= -\frac{idt}{6} \left(H(t) + 4H \left(t + \frac{dt}{2} \right) + H(t + dt) \right) + \frac{dt^3}{12} [H(t), H'(t)] \\ &\quad + \frac{dt^4}{24} [H(t), H''(t)] + \mathcal{O}(dt^5) \end{aligned}$$

Dans la suite, nous allons donner une démonstration avec plusieurs étapes:

Lemme 3.1. *Pour tout $s \in \mathbb{N}_+$, il existe $a_1, \dots, a_s \in \mathbb{C}$ t.q.*

$$\forall \lambda \in \mathbb{R}, \quad \exp(\lambda) = \prod_{i=1}^s (1 + a_i \lambda)(1 - a_i \lambda) + \mathcal{O}(\lambda^{2s+1})$$

En particulier, pour $s = 1$, nous avons $a_1 = -\frac{1}{2}$, pour $s = 2$, nous avons $a_1, a_2 = -\frac{1}{4} \pm \frac{\sqrt{3}}{12}i$.

Démonstration. Calcul direct dans \mathbb{R} . Nous montrons seulement les conclusions ici : si nous notons $s_{n,k}$ les polynômes symétriques élémentaires

$$s_{n,k}(X_1, \dots, X_n) = \sum_{1 \leq j_1 \leq \dots \leq j_k \leq n} X_{j_1} \cdots X_{j_k}$$

Alors a_1, \dots, a_s sont les solutions de

$$\sum_{k=0}^s \frac{s_{s,k}(a_1, \dots, a_s)}{(s+j-k)!} = 0, \quad j = 1, \dots, s.$$

C'est un système de s équations avec s inconnus. □

Corollaire 3.1. *Pour tout Ω dans $\mathbb{R}^{n \times n}$, nous avons*

$$\exp(\Omega) = \prod_{i=1}^s (1 + a_i \Omega)(1 - a_i \Omega) + \mathcal{O}(\|\Omega\|^{2s+1})$$

En particulier, si H ne dépend pas du temps, nous pouvons écrire l'opérateur d'évolution comme $\exp(-iHdt)$, et donc

$$\exp(-iHdt) = \prod_{i=1}^s (1 - a_i i dt H)(1 + a_i i dt H) + \mathcal{O}(\|\Omega\|^{2s+1}) = U_s(dt) + \mathcal{O}(\|\Omega\|^{2s+1})$$

Autrement dit, la méthode U_s dans la section 3.2 est d'ordre $2s$.

Lemme 3.2 (Magnus expansion). *Étant donné la matrice de coefficients $n \times n$ $A(t)$, la solution du problème de valeur initiale associé à l'équation différentielle ordinaire linéaire de la fonction vectorielle inconnue à n dimensions $Y(t)$*

$$Y'(t) = A(t)Y(t), \quad Y(0) = Y_0$$

peut être écrite comme

$$Y(t) = \exp(\Omega(t))Y_0, \quad \Omega(t) = \sum_{k \geq 1} \Omega_k(t)$$

où

$$\begin{aligned} \Omega_1(t) &= \int_0^t dt_1 A(t_1) \\ \Omega_2(t) &= \frac{1}{2} \int_0^t dt_1 \int_0^{t_1} dt_2 [A(t_1), A(t_2)] \\ &\dots \end{aligned}$$

Démonstration. Voir [13] pour une démonstration. □

Lemme 3.3. *Nous avons*

$$\Omega_2(t) = \frac{t^3}{12}[A(0), A'(0)] + \frac{t^4}{24}[A(0), A''(0)] + \mathcal{O}(t^5)$$

et

$$\Omega_k(t) = \mathcal{O}(t^5)$$

pour tout $k \geq 3$. Nous en déduisons que

$$\Omega(t) = \int_0^t dt_1 A(t_1) + \frac{t^3}{12}[A(0), A'(0)] + \frac{t^4}{24}[A(0), A''(0)] + \mathcal{O}(t^5)$$

Démonstration. Calculs avec *mathematica* et le package *NCAAlgebra*¹. □

Enfin nous donnons la démonstration du théorème 3.1, pour le théorème 3.2, c'est exactement la même preuve.

Démonstration du théorème 3.1. D'après Corollaire 3.1 and lemme 3.2, nous avons

$$U(t, t + dt) = \left(I - \frac{1}{2}\Omega(t) \right)^{-1} \left(I + \frac{1}{2}\Omega(t) \right) + \mathcal{O}(dt^3)$$

D'après lemme 3.3,

$$\Omega(t) = \int_0^t -iH(t_1)dt_1 + \mathcal{O}(t^3) = -idtH \left(t + \frac{dt}{2} \right) + \mathcal{O}(dt^3),$$

ce qui conduit au résultat. □

Pour les méthodes d'ordre plus haut, il suffit de refaire le processus du lemme 3.3 pour un ordre plus haut, mais le calcul est beaucoup plus compliqué. Une expérience numérique avec une simulation de "slow quench" du modèle d'Ising $N = 3$ est faite pour ces méthodes. Ici nous considérons un Halmitonian

$$\hat{H}(t) = \exp(-t) \sum_i \sigma_i^x + \sum_i \sigma_i^z \sigma_{i+1}^z,$$

et l'état initial $|\psi_0\rangle = \left(\frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle) \right)^{\otimes N}$. Nous évaluons ensuite l'erreur $L^\infty([0, t_{max}])$ pour $t_{max} = 1$ entre les états numérique obtenus par les méthodes U_s et les états exacts obtenus par la solution numérique (avec une précision 1e-10) de l'équation de Schrodinger. Les résultats sont montrés dans la figure 8.

¹<https://github.com/NCAAlgebra/NC>

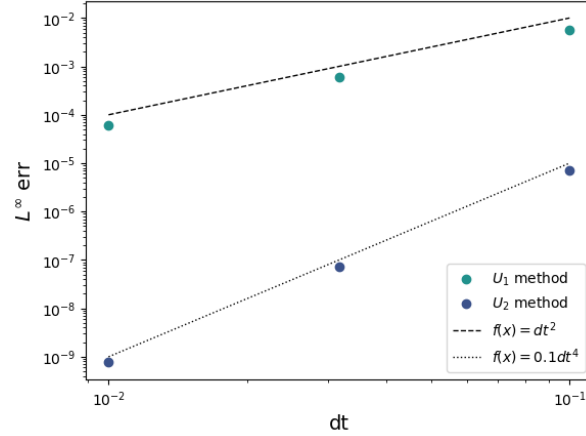


Figure 8: Erreur $L^\infty([0, 1])$ en fonction de dt pour différentes valeurs de s .

4 Méthode du splitting de l'hamiltonien

4.1 Un nouvel ansatz: splitting de l'hamiltonien

Dans le troisième chapitre, nous avons abordé les méthodes d'approximation applicables aux hamiltoniens généraux. L'hamiltonien que nous examinons plus précisément dans ce contexte s'écrit de la manière suivante :

$$H = h \sum_i \sigma_i^x + J \sum_{i,j} \sigma_i^z \sigma_j^z = H_x + H_{zz},$$

L'hamiltonien H_{zz} est diagonal lorsqu'il est exprimé dans la base formée par les produits tensoriels de $|\uparrow\rangle$ et $|\downarrow\rangle$. Il est évident que les opérateurs diagonaux sont plus simples à calculer, à l'instar d'une porte quantique. Par conséquent, notre algorithme peut être accéléré si nous parvenons à distinguer et à traiter séparément les effets induits par H_x et H_{zz} .

4.1.1 Approximation théorique du splitting

Définissons H comme la somme de deux parties non commutatives : $iH = A+B$. L'opérateur de transition correspondant est alors exprimé par : $U(t; 0) = \exp(t(A+B))$. La formule suivante est toujours vérifiée :

$$\exp\left(t(A+B) + \frac{1}{2}t^2[A, B]\right) = \exp(tA)\exp(tB) + O(t^3)$$

et immédiatement nous obtenons plus précisément

Théorème 4.1 (Verlet). *Soit A, B des opérateurs dans $\mathbb{R}^{n \times n}$. La formule d'ordre 2 suivante est vérifiée :*

$$\exp(t(A+B)) = \exp\left(\frac{tA}{2}\right) \exp(tB) \exp\left(\frac{tA}{2}\right) + O(t^3).$$

Dans la pratique une méthode d'ordre 2 est déjà utile, mais nous pouvons également obtenir des méthodes d'ordres plus haut. En général, un intégrateur symplectique se rapproche de l'opérateur d'évolution par un produit d'opérateurs de la forme

$$\exp[t(A+B)] = \prod_{i=1}^k \exp(c_i tA) \exp(d_i tB) + O(t^{k+1})$$

Avec c_i, d_i à déterminer. Nous pouvons nous attendre à ce qu'un opérateur de k produits ait l'ordre k . Et il existe des moyens généraux pour trouver le coefficient d'ordre supérieur, par exemple le travail de Yoshida [16]. Nous listons ici quelques méthodes d'ordre 3 ou 4:

Théorème 4.2. *Soit A, B des opérateur dans $\mathbb{R}^{n \times n}$, nous avons la formule d'ordre 3 suivante:*

$$\exp(t(A+B)) = \exp(a_1 tA) \exp(b_1 tB) \exp(a_2 tA) \exp(b_2 tB) \exp(a_3 tA) + O(t^4)$$

où

$$a_1, a_3 = \frac{1}{4} \left(1 \pm \frac{i}{\sqrt{3}}\right), \quad a_2 = \frac{1}{2}, \quad b_1, b_2 = \frac{1}{2} \left(1 \pm \frac{i}{\sqrt{3}}\right).$$

Voir [17] pour une démonstration.

Théorème 4.3 (Forest-Ruth). *Soit A, B des opérateur dans $\mathbb{R}^{n \times n}$. Nous avons la formule d'ordre 4 suivante:*

$$\begin{aligned} \exp(A+B) = \exp\left(\frac{\theta}{2}A\right) \exp(\theta B) \exp\left(\frac{1-\theta}{2}A\right) \exp((1-2\theta)B) \\ \exp\left(\frac{1-\theta}{2}A\right) \exp(\theta B) \exp\left(\frac{\theta}{2}A\right) \end{aligned}$$

où $\theta = 1/(2 - 2^{\frac{1}{3}})$.

Démonstration. Voir par exemple [18] pour une démonstration. □

Il y a aussi un moyen de construire une méthode d'ordre $2n+2$ à partir d'une méthode d'ordre $2n$: soit

$$S_{2n}(t) = \exp(t(A+B) + a_{2n+1}t^{2n+1} + O(t^{2n+3}))$$

une méthode symétrique obtenue d'ordre $2n$. On fait l'hypothèse que la méthode d'ordre $2n+2$ est de la forme

$$\begin{aligned} S_{2n+2}(t) = S_{2n}(b_n t) S_{2n}(c_n t) S_{2n}(b_n t) \\ = \exp((2b_n + c_n)t(A+B) + (2b_n^{2n+1} + c_n^{2n+1})a_{2n+1}t^{2n+1} + O(t^{2n+3})). \end{aligned}$$

Alors nécessairement $2b_n + c_n = 1$ et $2b_n^{2n+1} + c_n^{2n+1} = 0$ et nous avons la solution

$$b_n = \frac{1}{2 - 2^{1/(2n+1)}} \quad \text{et} \quad c_n = \frac{-2^{1/(2n+1)}}{2 - 2^{1/(2n+1)}}.$$

En partant de $S_2(t) = \exp(\frac{tA}{2}) \exp(tB) \exp(\frac{tA}{2})$ une méthode symétrique qui sépare A et B , par récurrence, nous pouvons calculer S_{2n} symétrique du splitting si nous voulons. Par exemple,

$$S_4(t) = \exp(x_1 t A) \exp(y_1 t B) \exp(x_2 t A) \exp(y_2 t B) \exp(x_3 t A) \exp(y_3 t B) \exp(x_4 t A),$$

où

$$x_1 = x_4 = \frac{1}{2(2 - 2^{1/3})}, x_2 = x_3 = \frac{1 - 2^{1/3}}{2(2 - 2^{1/3})}, y_1 = y_3 = \frac{1}{2 - 2^{1/3}}, y_2 = \frac{-2^{1/3}}{2 - 2^{1/3}}$$

Ce qui est exactement la formule du théorème 4.3.

4.1.2 Porte quantique d'opérateur diagonal

Après la séparation de l'Hamiltonian, nous pouvons traiter la partie diagonale et hors-diagonale indépendamment. Nous extrayons la partie diagonale $D = hH_{zz}$ et l'écrivons sous une forme générale $D = \text{Diag}(\lambda_x)_{x \in e}$ où e est la base que nous choisissons. L'opérateur d'évolution correspondant est donc $U(dt) = \exp(-idtD) = \text{Diag}(\exp(-it\lambda_x))_{x \in e}$.

Introduisons une nouvelle matrice J comme les paramètres venant de passer la porte diagonale, un nouvel ansatz s'écrit:

$$|\psi_{J,\theta}\rangle = \sum_{x \in e} \exp(x^T J x + N N_\theta(x)) |x\rangle$$

et

$$U(dt)|\psi_{J,\theta}\rangle = \sum_{x \in e} \exp(-it\lambda_x) \exp(x^T J x + N N_\theta(x)) |x\rangle.$$

Notons $x^T J' x = x^T J x - it\lambda_x$. Il suffit de connaître λ_x pour la mise à jour de la matrice J . Particulièrement, pour le modèle d'Ising qui est donné par un graph, nous pouvons déterminer $J' - J$ par la matrice d'adjoint de voisinage du graph et prendre x par la notion dans l'ensemble $\{1, -1\}^N$ dans le calcul. Plus précisément, si nous utilisons ± 1 pour les valeurs propres de qubit, nous avons

$$(\Delta J)_{jk} = \begin{cases} \frac{1}{2} & \text{si } (j, k) \in G \\ 0 & \text{sinon} \end{cases}$$

où $(j, k) \in G$ si et seulement s'il y a des interactions entre les qubits j et k . Dans ce cas, l'évaluation de la partie diagonale de l'Hamiltonian devient exacte, tandis que l'évaluation de la partie hors-diagonale devient plus facile pour le réseau neurone.

4.2 Méthode adaptée aux réseaux de neurones

Nous avons évoqué que pour un opérateur de la forme

$$V_k := \prod_{i=1}^k \exp(c_i t A) \exp(d_i t B) + \mathcal{O}(t^{k+1})$$

où A est diagonale (dans le modèle d'Ising, $A = H_{zz}$). Nous pouvons évaluer $\exp(c_i t A)$ exactement avec le nouvel ansatz. Dans le cas général, si nous souhaitons que V_k soit une méthode adaptée au réseau de neurones, il suffit de remplacer $\exp(d_i t B)$ par une méthode d'ordre au moins k , par exemple une méthode U_s décrite dans la section précédente avec $\frac{s}{2} \geq k$. Sous ce principe, nous pouvons proposer facilement une variante du théorème 4.1 qui est adaptée au réseau neurone :

Théorème 4.4 (Version NN du théorème 4.1). *La méthode*

$$V_2 = \exp\left(\frac{1}{2}h_{zz}\right) \left(I - \frac{1}{2}h_x\right)^{-1} \left(I + \frac{1}{2}h_x\right) \exp\left(\frac{1}{2}h_{zz}\right)$$

avec $h_{zz} := -idtH_{zz}$, $h_x := -idtH_x$ est d'ordre 2 qui prend 1 optimisation.

Si nous répétons le même processus pour le théorème 4.2, nous devons remplacer les termes concernant $\exp(d_i t B)$ par une méthode U_2 qui prend donc 4 optimisations en total. Cependant, nous soulignons que **si certaine propriété de symétrie est vérifiée pour la formule originale, nous pouvons remplacer directement les termes hors-diagonales par une approximation d'ordre 2**. Nous appellons ces méthodes comme **méthode symplectique padé**. Par exemple,

Théorème 4.5 (Version NN du théorème 4.2). *Une méthode V_3 d'ordre 3 qui prend 2 optimisations peut s'écrire comme*

$$V_3 = \exp(bh_{zz})(I - ah_x)^{-1}(I + ah_x) \exp\left(\frac{1}{2}h_{zz}\right) (I - ch_x)^{-1}(I + ch_x) \exp(dh_{zz})$$

avec $h_{zz} := -idtH_{zz}$, $h_x := -idtH_x$, et

$$a, c = \frac{1}{4} \pm \frac{\sqrt{3}}{12}i, \quad b = \frac{1}{3} - 2c^2, \quad d = \frac{1}{3} - 2a^2$$

Démonstration. Vérification avec *mathematica* et le package *NCAgebra*. □

Nous remarquons que V_s n'est plus une méthode symplectique car les coefficients sont complexes. Même idée marche pour la formule de Forest-Ruth :

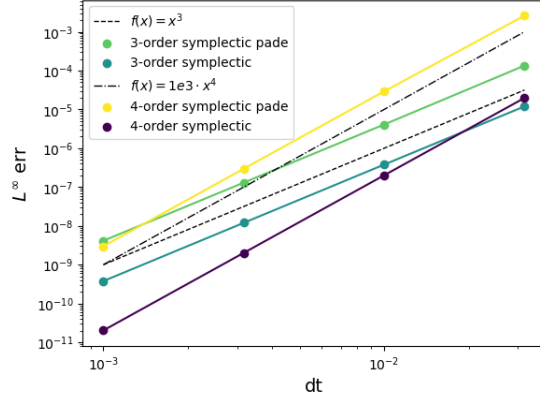


Figure 9: Méthode symplectique padé.

Théorème 4.6 (Version NN du théorème 4.3). *Une méthode V_3 d'ordre 4 qui prend 3 optimisations peut s'écrire comme*

$$V_4 = \exp\left(\frac{\theta}{2}h_{zz}\right) \left(I - \frac{\theta}{2}h_x\right)^{-1} \left(I + \frac{\theta}{2}h_x\right) \exp\left(\frac{1-\theta}{2}h_{zz}\right) \left(I - \frac{1-2\theta}{2}h_x\right)^{-1} \\ \left(I + \frac{1-2\theta}{2}h_x\right) \exp\left(\frac{1-\theta}{2}h_{zz}\right) \left(I - \frac{\theta}{2}h_x\right)^{-1} \left(I + \frac{\theta}{2}h_x\right) \exp\left(\frac{\theta}{2}h_{zz}\right)$$

avec $h_{zz} := -idtH_{zz}$, $h_x := -idtH_x$, et $\theta = \frac{1}{2-2^{1/3}}$.

Ces approches diminuent fortement le coût de calcul (nombre d'optimisation), en revanche, cela conduit un grand préfacteur. Nous voyons cet effet dans le figure 9. Ici nous considérons un modèle d'Ising de 3×3 (même pour U_s adaptive dans figure 7), et nous traçons l'erreur $L^\infty([0, 1])$ avec dt .

5 Méthode de Monte-Carlo variationnelle

Même si nous ne gardons que $O(N^2)$ paramètres en mémoire dans θ au lieu de stocker le vecteur $|\psi\rangle$ de dimension 2^N , nous ne pouvons pas encore calculer directement l'infidélité avec θ sans exprimer le vecteur $|\psi_\theta\rangle \in \mathbb{C}^{2^N}$, ce que nous voulions éviter. Plus généralement, le calcul des observables requerra des méthodes d'approximations.

Le méthode de Monte-Carlo par chaînes de Markov, (ou **MCMC** pour Markov chain Monte Carlo en anglais), est une méthode d'échantillonnage à partir de distributions de probabilité. Nous allons montrer que, en exprimant la formule intelligemment, nous pouvons approcher la valeur des observables par une méthode de Monte-Carlo avec une quantité polynomiale d'échantillons.

Dans cette partie, nous allons d'abord montrer comment l'infidélité peut être exprimée avec le formalisme des probabilités. Ensuite nous introduisons l'algorithme de Metropolis-Hastings et les autres applications dans la physique quantique.

5.1 Echantillonnage: Metropolis-Hastings

Prenons l'exemple de calcul pour l'infidélité. Considérons e une base orthonormale de \mathcal{H} . Nous avons

$$\begin{aligned} \text{Infidélité}(|\psi\rangle, |\phi\rangle) &= 1 - \frac{\langle\psi|\phi\rangle\langle\phi|\psi\rangle}{\langle\psi|\psi\rangle\langle\phi|\phi\rangle} \\ &= 1 - \sum_{x \in e} \sum_{y \in e} \frac{\langle\psi|x\rangle\langle x|\phi\rangle\langle\phi|y\rangle\langle y|\psi\rangle}{\langle\psi|\psi\rangle\langle\phi|\phi\rangle} \\ &= 1 - \sum_{x \in e} \frac{\langle\psi|x\rangle\langle x|\psi\rangle}{\langle\psi|\psi\rangle} \frac{\langle x|\phi\rangle}{\langle x|\psi\rangle} \sum_{y \in e} \frac{\langle\phi|y\rangle\langle y|\phi\rangle}{\langle\phi|\phi\rangle} \frac{\langle y|\psi\rangle}{\langle y|\phi\rangle}. \end{aligned}$$

Nous retrouvons ici les termes $\frac{\langle\psi|x\rangle\langle x|\psi\rangle}{\langle\psi|\psi\rangle}$ et $\frac{\langle\phi|y\rangle\langle y|\phi\rangle}{\langle\phi|\phi\rangle}$ qui donnent des probabilités d'observer les états x et y . Notons les \mathbb{P}_ψ et \mathbb{P}_ϕ et réécrivons l'infidélité comme une espérance:

$$\begin{aligned} \text{Infidélité}(|\psi\rangle, |\phi\rangle) &= \mathbb{E}_{x \sim \mathbb{P}_\psi, y \sim \mathbb{P}_\phi} \left[1 - \frac{\langle x|\phi\rangle}{\langle x|\psi\rangle} \frac{\langle y|\psi\rangle}{\langle y|\phi\rangle} \right] \\ &= 1 - \mathbb{E}_{x \sim \mathbb{P}_\psi} \left[\frac{\langle x|\phi\rangle}{\langle x|\psi\rangle} \right] \mathbb{E}_{y \sim \mathbb{P}_\phi} \left[\frac{\langle y|\psi\rangle}{\langle y|\phi\rangle} \right]. \end{aligned}$$

Dans nos modèles, les calculs de produits scalaires entre un état neuronal ψ et un état propre x s'effectuent en temps polynomial, donc nous avons ramené le problème au calcul d'une espérance, que nous pouvons approximer par échantillonnage avec une chaîne de Markov.

Théorème 5.1. *Soient $(X_n)_n$ une chaîne de Markov irréductible et récurrente positive sur E un espace dénombrable et P sa matrice de transition. Supposons π une mesure de probabilité invariante sur E . Alors pour toute fonction f intégrable sur (E, π) , nous avons presque sûrement*

$$\frac{1}{n} \sum_{i=1}^n f(X_i) \xrightarrow{n \rightarrow \infty} \mathbb{E}_\pi[f]$$

Pour satisfaire les conditions précédentes, étant déjà donné l'espace $E = e$ et $\pi = \mathbb{P}_\psi$, il suffit de construire la matrice de transition P en prenant $P(x, y) = \min\left(1, \frac{\pi(y)}{\pi(x)}\right)$.

Nous avons mis en pratique un algorithme Metropolis-Hastings avec une marche aléatoire de la façon suivante:

- Initialisation: choisir au hasard un état x_0 dans la base $e = \{0, 1\}^N$.
- Echantillonnage: pour chaque pas $n \in \mathbb{N}$, choisir uniformément un voisin de l'état x_n qu'on note y_n et calculer la probabilité d'acceptation:

$$p = \min \left(1, \frac{\mathbb{P}_\psi(y_n)}{\mathbb{P}_\psi(x_n)} \right) = \min \left(1, \frac{|\langle y_n | \psi \rangle|^2}{|\langle x_n | \psi \rangle|^2} \right).$$

Avec probabilité p , $x_{n+1} = y_n$, sinon $x_{n+1} = x_n$.

- Convergence: Après avoir généré N_{ech} échantillons, renvoyer la moyenne de la suite $\left(\frac{\langle x_n | \phi \rangle}{\langle x_n | \psi \rangle} \right)_n$ comme une approximation de $\mathbb{E}_{x \sim \mathbb{P}_\psi} \left[\frac{\langle x | \phi \rangle}{\langle x | \psi \rangle} \right]$.

Dans la pratique pour accélérer la convergence en loi, nous ne choisissons pas nécessairement un voisin direct, mais attribuons une certaine probabilité au fait de changer une coordonnée, deux coordonnées, et une probabilité faible au fait de modifier toutes les coordonnées simultanément.

5.2 Applications

Ces outils sont suffisants pour réduire la complexité de tous calculs qui nous intéresseront à l'ordre polynomial. Cependant, certaines grandeurs telles que la norme de l'état ne sont pas approximables avec cette méthode.

5.2.1 Echantillonnage d'une observable

Soit $|\psi_\theta\rangle$ un état paramétrisé, nous cherchons à calculer son énergie comme une fonction de θ , notée $F(\theta)$ par l'échantillonnage. Nous avons

$$\begin{aligned} F(\theta) &= \frac{\langle \psi_\theta | \hat{H}_0 | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} \\ &= \sum_{x \in e} \frac{\langle \psi_\theta | x \rangle \langle x | \hat{H}_0 | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} \\ &= \sum_{x \in e} \frac{\langle \psi_\theta | x \rangle \langle x | \psi_\theta \rangle \langle x | \hat{H}_0 | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle \langle x | \psi_\theta \rangle} \\ &= \mathbb{E}_{x \sim \mathbb{P}_\theta} \left[\frac{\langle x | \hat{H}_0 | \psi_\theta \rangle}{\langle x | \psi_\theta \rangle} \right] \end{aligned}$$

Ce calcul se généralise à n'importe quelle observable.

5.2.2 Echantillonnage d'un gradient

Pour estimer le gradient d'une fonction, il faut faire attention car modifier θ change a priori les échantillons. Nous pouvons heureusement nous en sortir par

une astuce de calcul. Soit $F(\theta) = \mathbb{E}_{x \sim \mathbb{P}_\theta}[f] = \sum_{x \in e} P(\theta, x) f(\theta, x)$, nous avons

$$\begin{aligned} dF(\theta) &= \frac{d}{d\theta} \sum_{x \in e} P(\theta, x) f(\theta, x) \\ &= \sum_{x \in e} \frac{\partial P}{\partial \theta}(\theta, x) f(\theta, x) + P(\theta, x) \frac{\partial f}{\partial \theta}(\theta, x) \\ &= \sum_{x \in e} P(\theta, x) \frac{\partial \log P}{\partial \theta}(\theta, x) f(\theta, x) + P(\theta, x) \frac{\partial f}{\partial \theta}(\theta, x) \\ &= \mathbb{E}_{x \sim \mathbb{P}_\theta} \left[\frac{\partial \log P}{\partial \theta}(\theta, x) f(\theta, x) + \frac{\partial f}{\partial \theta}(\theta, x) \right]. \end{aligned}$$

En particulier, si $F(\theta)$ est la fonction de perte dans un modèle, cette technique permet de calculer son gradient efficacement.

5.2.3 Propagation d'un système

Nous pouvons désormais étudier l'évolution d'un système pendant une période t . Comme nous ne pouvons pas calculer l'exponentielle de l'hamiltonien sans le diagonaliser, nous utilisons pour le moment un développement limité. Pour dt suffisamment petit,

$$\begin{aligned} |\psi(t + dt)\rangle &= e^{-i\hat{H}dt} |\psi(t)\rangle \\ &= (1 - idt\hat{H}) |\psi(t)\rangle \end{aligned}$$

Supposons que $|\psi(t)\rangle$ est décrit par θ_t et notons $\hat{U}(dt) = 1 - idt\hat{H}$. Nous voulons minimiser

$$G(\theta) = \text{Infidélité}(|\psi_\theta\rangle, \hat{U}|\psi(t)\rangle) = 1 - \mathbb{E}_{x \sim \mathbb{P}_\theta} \left[\frac{\langle x | \hat{U} | \psi(t) \rangle}{\langle x | \psi_\theta \rangle} \right] \mathbb{E}_{y \sim \mathbb{P}_t} \left[\frac{\langle y | \hat{U}^\dagger | \psi_\theta \rangle}{\langle y | \psi(t) \rangle} \right].$$

Nous remarquons que dans la pratique, l'Hamiltonien du système ne contient souvent que des interactions locales donc $\hat{U}(dt)$ est une matrice creuse. Le calcul de $\langle x | \hat{U}(dt) | \psi(t) \rangle$ ne pose donc pas de problème particulier.

5.2.4 Variable de contrôle

La méthode Monte-Carlo s'avère efficace pour approximer une espérance, et le théorème de la limite centrale nous informe que l'erreur suit une distribution normale $\hat{E}_n \sim \mathcal{N}(0, \sigma^2/n)$. La dépendance en n de la variance pousse à prendre un grand nombre d'échantillons, mais certaines méthodes telles que l'utilisation

d'une variable de contrôle permet de réduire la variance sans augmenter significativement le temps de calcul.

Notons

$$f(x, y) = \frac{\langle x|\hat{U}|\psi(t)\rangle \langle y|\hat{U}^\dagger|\psi_\theta\rangle}{\langle x|\psi_\theta\rangle \langle y|\psi(t)\rangle},$$

où \hat{U} est l'opérateur d'évolution qui vérifie $UU^\dagger = 1$. Alors $F = \mathbb{E}_{\mathbb{P}}[f]$ et

$$\begin{aligned} \mathbb{E}_{\mathbb{P}}[|f|^2] &= \sum_{x,y} \frac{\langle \psi_\theta|x\rangle \langle x|\psi_\theta\rangle \langle \psi(t)|y\rangle \langle y|\psi(t)\rangle}{\langle \psi_\theta|\psi_\theta\rangle \langle \psi(t)|\psi(t)\rangle} \frac{|\langle x|\hat{U}|\psi(t)\rangle|^2}{|\langle x|\psi_\theta\rangle|^2} \frac{|\langle y|\hat{U}^\dagger|\psi_\theta\rangle|^2}{|\langle y|\psi(t)\rangle|^2} \\ &= \sum_{x,y} \frac{|\langle y|\hat{U}^\dagger|\psi_\theta\rangle|^2}{\langle \psi_\theta|\psi_\theta\rangle} \frac{|\langle x|\hat{U}|\psi(t)\rangle|^2}{\langle \psi(t)|\psi(t)\rangle} \\ &= 1 \end{aligned}$$

Pour diminuer la variance, considérons $\tilde{f} = f - \mu(|f|^2 - 1)$. Alors $\mathbb{E}_{\mathbb{P}}[\tilde{f}] = F$ et $Var(\tilde{f}) = Var(f) - 2\mu Cov_{\mathbb{P}}(f, |f|^2) + \mu^2 Var_{\mathbb{P}}(|f|^2)$ qui atteint son minimum lorsque

$$\mu = \frac{Cov_{\mathbb{P}}(f, |f|^2)}{Var_{\mathbb{P}}(|f|^2)}.$$

Alors si prenons la limite $|\psi_\theta\rangle = \hat{U}|\psi(t)\rangle$ que nous attendons et écrivons $f(x, y) = 1 - \tau_{x,y}$,

$$\begin{aligned} Cov_{\mathbb{P}}(f, |f|^2) &= \mathbb{E}_{\mathbb{P}}[f|f|^2] - \mathbb{E}_{\mathbb{P}}[f]\mathbb{E}_{\mathbb{P}}[|f|^2] \\ &= \sum_{x,y} p_{x,y}(1 - \tau_{x,y})^3 - \sum_{x,y} p_{x,y}(1 - \tau_{x,y}) \sum_{x,y} p_{x,y}(1 - \tau_{x,y})^2 \\ &= 2 \sum_{x,y} p_{x,y} \tau_{x,y}^2 - 2 \left(\sum_{x,y} p_{x,y} \tau_{x,y} \right)^2 + O(\tau^3), \end{aligned}$$

$$\begin{aligned} Var_{\mathbb{P}}(|f|^2) &= \mathbb{E}_{\mathbb{P}}[|f|^4] - \mathbb{E}_{\mathbb{P}}[|f|^2]^2 \\ &= \sum_{x,y} p_{x,y}(1 - \tau_{x,y})^4 - \left(\sum_{x,y} p_{x,y}(1 - \tau_{x,y}^2) \right)^2 \\ &= 4 \sum_{x,y} p_{x,y} \tau_{x,y}^2 - 4 \left(\sum_{x,y} p_{x,y} \tau_{x,y} \right)^2 + O(\tau^3). \end{aligned}$$

Alors nous pouvons choisir $\mu = 1/2$ et calculer $\mathbb{E}_{\mathbb{P}}[1 - \tilde{f}]$ dans la pratique pour avoir une mieux estimation.

5.3 Etat variationnel Monte-Carlo

Maintenant nous pouvons définir un état variationnel dans notre simulation :

Définition. *Un état variationnel Monte-Carlo est une RBM qui représente la distribution de probabilité d'une fonction d'onde ainsi qu'un échantillonnage de cette distribution effectué par MCMC.*

La combinaison du réseau de neurones et de l'échantillonnage nous permet de calculer la fonction de perte efficacement au cours de l'entraînement. Un entraînement typique peut se décrire par le procédé suivant:

1. Calculer la fonction de perte (Infidélité dans 5.1 ou Hamiltonien dans 5.2.1) et son gradient (5.2.2) grâce à l'échantillonnage
2. Initialiser les paramètres de RBM
3. A chaque étape :
 - Échantillonner à partir de la distribution de probabilité déterminée par RBM et évaluer la valeur de la perte et de son gradient
 - Modifier les paramètres de RBM selon la valeur du gradient
 - Vérifier la condition d'arrêt
4. Après être bien arrivé vers la cible, augmenter le nombre d'échantillons pour évaluer plus précisément les grandeurs finales.

Il y a deux facteurs qui sont cruciaux dans l'état variationnel Monte-Carlo : le rapport α entre le nombre de neurones dans la couche cachée et dans la couche d'entrée, qui détermine le nombre de paramètres dans RBM, et le nombre d'échantillons. Un grand α représente une machine plus puissante qui atteint une perte plus faible, mais qui a aussi un coût de calcul plus important. Une taille grande d'échantillon permet d'évaluer les valeurs plus précisément donc conduit à moins de variation et une possibilité de descendre plus la perte. Nous étudions ces deux facteurs dans la section 6.1.

6 Étude sur l'architecture des RBM

Dans ce chapitre, nous évaluons initialement le pouvoir de représentation d'une machine de Boltzmann restreinte appliquée au modèle d'Ising. Lin et Pollmann, dans [2], ont exploré la capacité d'un réseau de neurones à modéliser un système d'Ising quantique non intégrable. Ils ont démontré que pour maintenir une précision quantique spécifique, le nombre de paramètres nécessaires doit croître de manière exponentielle avec le temps. Nous observons une augmentation similaire du nombre de paramètres requis pour une bonne représentation du système.

6.1 Le pouvoir de représentation d'une RBM pour le modèle d'Ising

. Pour tester le pouvoir de représentation d'une RBM, nous reprenons la série des états obtenue par la méthode U_s qui est une simulation du modèle d'Ising de $N = 10$ (Figure 4) comme notre cible. À chaque t_i , un état variationnel Monte-Carlo avec un certain α est généré aléatoirement avec une taille d'échantillon de 1008. Il est entraîné pour approcher l'état cible de t_i pour 6,000 étapes. Pour comparer, nous utilisons aussi un état complet paramétré par une RBM de même α (sans échantillonnage quand calculer la perte). La Figure 10 illustre les valeurs finales des observables. Nous observons qu'il n'y a pas de grande différence pour différent α avec les états complets. En revanche, avec les états Monte-Carlo, nous observons de meilleure performance pour grand α . C'est parce qu'une RBM avec un grand α a un meilleur pouvoir de représentation et un meilleur pouvoir de généralisation, quand l'évaluation de la perte n'est plus exacte, elle est donc plus stable. Si nous souhaitons bien approcher un état avec les états Monte-Carlo, une augmentation de α est nécessaire pour un certain seuil des erreurs.

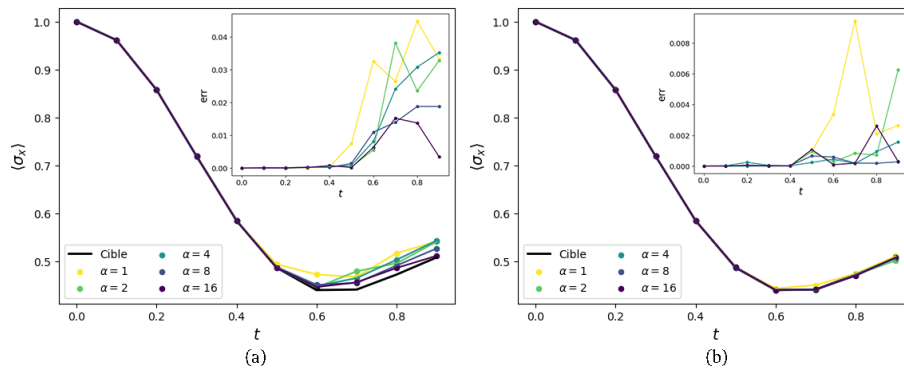


Figure 10: Le pouvoir de représentation de RBM pour différents α . (a) Evolution avec les états Monte-Carlo. (b) Evolution avec les états complets.

Comme le nombre d'échantillons est aussi un facteur crucial pour la performance de RBM, Figure 11 montre un exemple de l'impact de la taille d'échantillon. Nous observons qu'un grand nombre d'échantillons diminue les variances d'évaluation et donc améliore en même temps la précision des observables. Cependant, le coût de calcul est proportionnel à la taille de l'échantillon.

6.2 Le choix de dt optimal

Une fois les paramètres de représentations choisis, tels que le nombre de neurones et d'échantillons, la durée d'une simulation est proportionnelle au nombre de calculs d'infidélité. Ce qui est proportionnel au nombre d'étape d'entraînement pour chaque pas de temps fois le nombre d'optimisations. Nous nous sommes

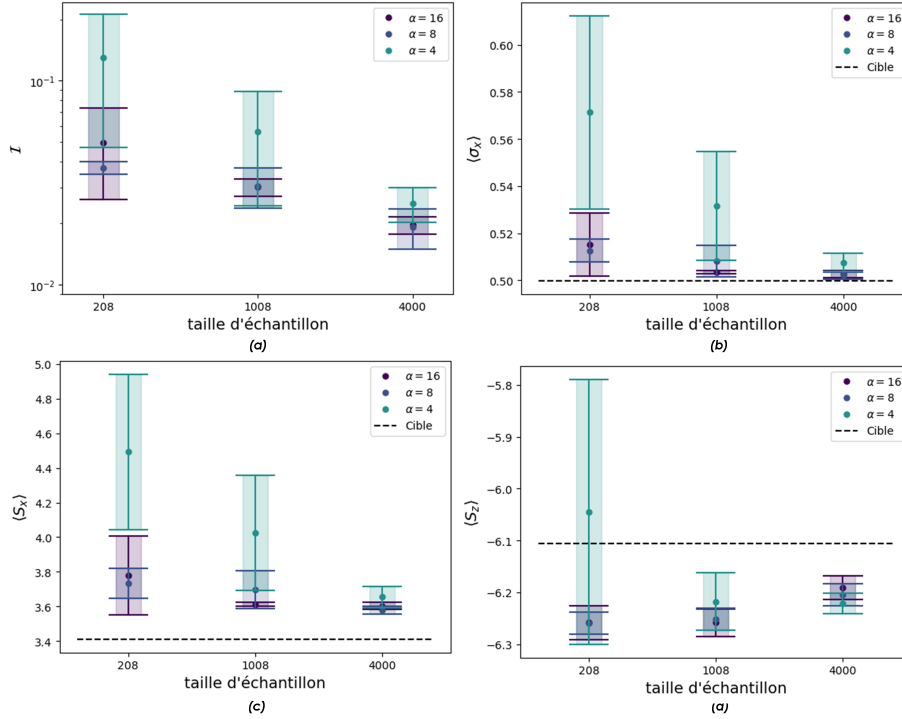


Figure 11: L'impact de la taille de l'échantillon. La cible est un état fortement intriqué. Nous cherchons à approcher cet état par un état variationnel Monte-Carlo aléatoirement initialisé. Huit expériences sont faites indépendamment. La moyenne et l'intervalle de confiance à 95% sont affichés dans chaque figure. (a) L'infidélité finale après 2,000 étapes. (b) L'évaluation de $\langle \sigma_x \rangle = \frac{1}{N} \sum_i \langle \sigma_x^i \rangle$ pour l'état final. (c) L'évaluation de $\langle S_x \rangle = \sum_i \langle \sigma_x^i \sigma_x^{i+1} \rangle$ pour l'état final. (d) L'évaluation de $\langle S_z \rangle = \sum_i \langle \sigma_z^i \sigma_z^{i+1} \rangle$ pour l'état final.

demandé comment optimiser le choix de dt pour améliorer la précision de la simulation avec un coût computationnel constant. Cette comparaison a été faite sans échantillonnage pour réduire la variance des données. Ce choix a cependant imposé une valeur de N assez faible, nous avons considéré 9 spins disposés en 3×3 . Pour des valeurs de N plus grandes, on devrait initialement atteindre des états plus simples car les conditions périodiques joueront un rôle moins important et donc enlèveront moins de régularité. Par conséquent un dt plus faible permettra une meilleure approximation théorique et le nombre d'étapes d'entraînement pourra être plus faible également sans trop perdre en précision. Cependant ce phénomène est transitoire, et asymptotiquement la nature de l'état de changera plus. Cependant le vecteur à compresser augmentera en taille, ce qui requièrera plus d'étape d'entraînement pour l'approximer, une quantité proportionnelle à la racine carrée du nombre de spins. On s'attend donc à ce

À coût T constant, effet de dt sur l'erreur de l'approximation.
 $N = 3 \times 3$, $h=1$, $0 < t \leq 2$, méthode U_2 entraîné pendant $T \cdot dt/4$ étapes.

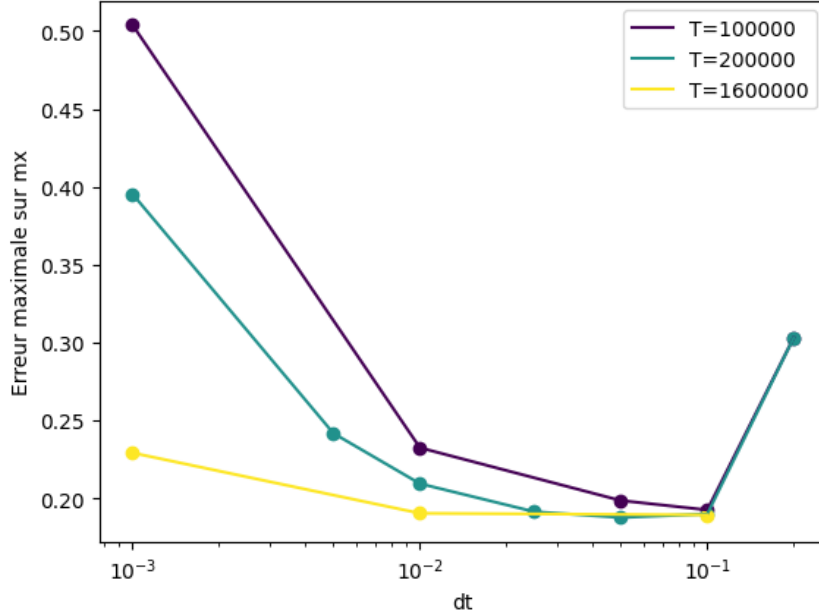


Figure 12: L'impact de dt sur la précision de la simulation en un nombre d'étapes d'optimisation constant, pour différents coûts. On remarque que quand dt devient assez grand, les courbes se rejoignent car les entraînements arrivent tous à leur valeur asymptotique optimale. La valeur optimale de dt est ici comprise entre 10^{-2} et 10^{-1}

qu'asymptotiquement, les valeurs optimales de dt soient plus élevées.

6.3 Application : quench dynamique du modèle d'Ising

Dans cette section, nous appliquons la RBM ansatz pour le modèle d'Ising de dimension 2 avec la méthode U_2 décrite dans le théorème 3.2. Nous considérons 4×4 spins, où la solution de la diagonale exacte n'est plus possible. Dans cet ordre de grandeur, les états complets représentés par une RBM de $\alpha = 8$ demandent presque 35GB de RAM dans un GPU, donc il est préférable d'utiliser les états Monte-Carlo pour un calcul plus efficace. Son hamiltonien est donné par

$$\hat{H} = h \sum_i \sigma_i^x + J \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z$$

Un tel système subit une transition de phase du régime ferromagnétique au régime paramagnétique lorsque $h = 3.04438J$. Dans ce qui suit, nous définissons le couplage constant $J = 1$ et $h = 3.04438$ [15]. Nous considérons ensuite une

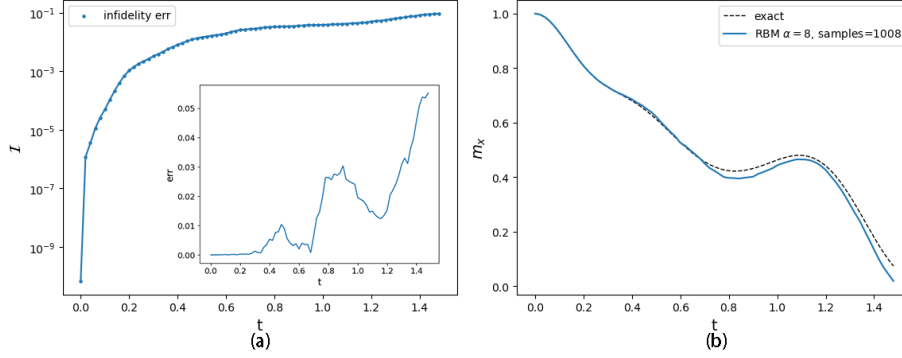


Figure 13: Simulation de quench dynamique du modèle d'Ising. a) L'évolution des erreurs (infidélité) entre l'état exact et l'état neuronal. La vignette illustre l'erreur de l'observable $\langle m_x \rangle$. b) L'évolution de $\langle m_x \rangle$.

dynamique où le champ magnétique diminue exponentiellement avec le temps, i.e.

$$\hat{H}(t) = \exp(-t)h \sum_i \sigma_i^x + J \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z, \quad |\psi_0\rangle = \left(\frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle) \right)^{\otimes N}$$

Nous prenons les états Monte-Carlo représentés par une RBM avec $\alpha = 8$ et 1,008 d'échantillons. Pour chaque pas de temps $dt = 0.02$, nous entraînons le réseau de neurones pendant 3 000 étapes sur un seul GPU, ce qui prend environ 1,2s pour chaque étape. La figure 13 illustre les résultats obtenus. Ici nous approchons l'évolution exacte en mettant $dt = 0.005$ dans la méthode U_2 et en utilisant l'itération résiduelle minimale généralisée (GMRES) [14] pour résoudre $Vx = b$. Quand $N = 16$ cette approche est encore possible, tandis que pour le réseau de neurones nous pouvons aller beaucoup plus loin.

7 Conclusion

Nous avons exploré l'application des réseaux de neurones, spécifiquement des machines de Boltzmann restreintes, pour simuler des états quantiques dans des systèmes complexes. Nous avons montré que ces réseaux peuvent effectivement représenter des états quantiques avec une complexité réduite par rapport aux méthodes traditionnelles. Nous avons développé des méthodes d'évolution temporelle basées sur le Monte-Carlo variationnel et des intégrateurs symplectiques, qui améliorent la précision des simulations tout en limitant les coûts computationnels.

Nos résultats confirment le potentiel des réseaux de neurones pour la simulation quantique et suggèrent que la généralisations des méthodes implicites pour l'évolution du système permet d'améliorer la précision de la simulation sans subir de coût trop important. L'ensemble de nos méthode nous a permis de réaliser une simulation de l'évolution de 16 spins avec des erreurs très faibles sur les observables.

References

- [1] Giuseppe Carleo, Matthias Troyer, Solving the quantum many-body problem with artificial neural networks. *Science* 355,602-606(2017). DOI:10.1126/science.aag2302
- [2] Lin, Sheng-Hsuan, and Frank Pollmann. "Scaling of Neural-Network Quantum States for Time Evolution." *physica status solidi (b)* 259.5 (2022): 2100172.
- [3] Valenti, A., Greplova, E., Lindner, N. H., Huber, S. D. (2022). Correlation-enhanced neural networks as interpretable variational quantum states. *Physical Review Research*, 4(1), L012010.
- [4] Gutiérrez, Irene López, and Christian B. Mendl. "Real time evolution with neural-network quantum states." *Quantum* 6 (2022): 627.
- [5] Vicentini, Filippo, Riccardo Rossi, and Giuseppe Carleo. "Positive-definite parametrization of mixed quantum states with deep neural networks." arXiv preprint arXiv:2206.13488 (2022).
- [6] Reh, Moritz, Markus Schmitt, and Martin Gärttner. "Optimizing Design Choices for Neural Quantum States. 2023. doi: 10.48550." arXiv preprint arXiv:2301.06788.
- [7] Vicentini, Filippo, et al. "Variational neural-network ansatz for steady states in open quantum systems." *Physical review letters* 122.25 (2019): 250503.
- [8] Koutný, D., Motka, L., Hradil, Z., Řeháček, J., and Sánchez-Soto, L. L. (2022). Neural-network quantum state tomography. *Physical Review A*, 106(1), 012409.
- [9] Jónsson, Bjarni, Bela Bauer, and Giuseppe Carleo. "Neural-network states for the classical simulation of quantum computing." arXiv preprint arXiv:1808.05232 (2018).
- [10] Choo, Kenny, Antonio Mezzacapo, and Giuseppe Carleo. "Fermionic neural-network states for ab-initio electronic structure." *Nature communications* 11.1 (2020): 2368.
- [11] Sinibaldi, A., Giuliani, C., Carleo, G., and Vicentini, F. Unbiasing time-dependent Variational Monte Carlo by projected quantum evolution (2023). arXiv preprint arXiv:2305.14294.
- [12] Barison, Stefano, Filippo Vicentini, and Giuseppe Carleo. "An efficient quantum algorithm for the time evolution of parameterized circuits." *Quantum* 5 (2021): 512.
- [13] Blanes, S., Casas, F., Oteo, J. A., and Ros, J. (2009). The Magnus expansion and some of its applications. *Physics reports*, 470(5-6), 151-238.

- [14] Saad, Youcef, and Martin H. Schultz. "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems." *SIAM Journal on scientific and statistical computing* 7.3 (1986): 856-869.
- [15] Blöte, Henk WJ, and Youjin Deng. "Cluster Monte Carlo simulation of the transverse Ising model." *Physical Review E* 66.6 (2002): 066110.
- [16] Yoshida, Haruo. "Construction of higher order symplectic integrators." *Physics letters A* 150.5-7 (1990): 262-268.
- [17] Chambers, J. E. "Symplectic integrators with complex time steps." *The Astronomical Journal* 126.2 (2003): 1119.
- [18] García-Ripoll, Juan José. "Time evolution of matrix product states." *New Journal of Physics* 8.12 (2006): 305.