

Simulation par HPC de dynamique causal de graphe quantique

Joseph TOUZET
encadré par Pablo ARRIGHI et Oguz KAYA
laboratoires LMF et LISN

Juin 2021

Abstract

La physique quantique permet d'étudier des superpositions d'objets, et de prédire l'évolution de ces superpositions. En plus d'étudier une superposition de particules, on peut aussi étudier une superposition de l'espace sur lequel se trouve ces particules.

Cet espace sera ici un graphe circulaire, et les dynamiques étudiées pourront créer et détruire des noeuds. Ce modèle peut modéliser une dilatation ou contraction de l'espace, qui serait quantifiée.

Nous chercherons à : avoir une implémentation efficace de ce modèle (parallélisme, calcul distribué, etc...) car la complexité de la simulation quantique est d'ordre exponentiel (car le nombre de graphes croît de manière exponentielle); concevoir une dynamique causant une croissance exponentielle de la taille moyenne des graphes, qui serait un modèle jouet de l'inflation pour un espace quantifié.

1 Introduction

Les graphes que nous étudions dans ce modèle jouet sont des cycles, où des particules peuvent se trouver ou non dans deux positions par noeud: le "port gauche" et le "port droit" qui dans l'étape de pas correspondent aux particules qui se déplacent respectivement vers la gauche et vers la droite.

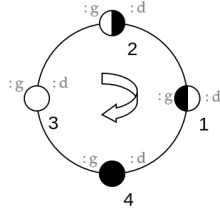


Figure 1: Exemple de graphe
 1: Une particule au port gauche
 2: Une particule au port droit
 3: Noeud sans particule
 4: Une particule sur chaque port

On peut ensuite appliquer à ces graphes des dynamique qui déterminent le futur état d'un noeud du graphe à partir de son état et de l'état de ses voisins. Une dynamique peut aussi créer ou détruire des noeuds et particules.

Il est intéressant de noter qu'une dynamique réversible peut causer une croissance du nombre de noeuds [4]. Nous nous posons donc la question de cette variation du nombre de noeuds si nous appliquons cette dynamique de manière analogue à une marche quantique [3]. Est-il alors possible d'observer une croissance qui ne suit pas la courbe moyenne classique ?

Pour répondre à ces questions il est important d'avoir une implémentation de ce modèle extrêmement efficace, dont la programmation est le sujet de mon stage.

2 Dynamique classique

Les dynamiques que nous étudierons peuvent être représentées par un ensemble de 2 motifs locaux qui sont échangés par la dynamique.

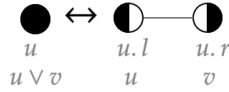


Figure 2: motifs échangés par SM

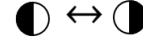


Figure 3: motifs échangés par la dynamique coin

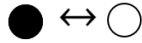


Figure 4: motifs échangés par EC



Figure 5: motifs échangés par M

motifs échangés par les différentes dynamiques

Les différentes dynamiques étudiées jusqu'ici sont:

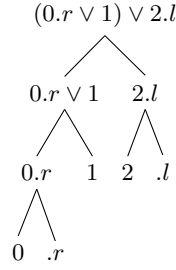
- La dynamique "erase create" (notée EC) permet la création de particules, mais ne "merge" ou "split" pas de noeuds.
- La dynamique "split merge" (notée SM) permet la création (par "split") et la destruction (par "merge") de noeuds, mais conserve le nombre de particules.
- L'étape de pas (notée M et qui sera toujours classique) fais simplement bouger les particules dans le port gauche d'un noeud vers le port gauche du noeud à leur gauche, et inversement pour les particules allant à droite. Cette dynamique peut être très simplement inversé en déplaçant les particules à droite vers la gauche et inversement.
- La dynamique coin est une dynamique qui permet une marche quantique en faisant changer de direction une particule.

2.1 Arithmétique de nom

Il est nécessaire de nommer les noeuds [2] pour conserver la réversibilité [1]. Cette condition nous donne aussi :

- $(u \vee v).l = u$
- $(u \vee v).r = v$
- $u.r \vee u.l = u$

En prenant comme noms de départ des entiers naturels, on peut représenter les noms de noeud comme des arbres, par exemple:



3 Dynamique quantique

3.1 Paramétrisation

Pour quantifier la dynamique, on rend l'échange de motif sujet à une opération unitaire paramétrée par 2 angles réels :

$$U = \begin{pmatrix} \cos\theta & \sin\theta e^{i\phi} \\ \sin\theta e^{-i\phi} & -\cos\theta \end{pmatrix}$$

En effet, en se plaçant dans la base "locale" (celle des deux motifs échangés par la dynamique), on décrit les amplitudes caractéristiques de la dynamique comme étant:

- $\sin\theta e^{i\phi}$ pour l'application dans le sens direct.
- $\cos\theta$ pour la non-application dans le sens direct.
- $\sin\theta e^{-i\phi}$ pour l'application dans le sens contraire.
- $\cos\theta$ pour la non-application dans le sens contraire.

On remarque que l'angle ϕ peut favoriser ou non les interférences destructives ou constructives en influençant la phase relative de graphes ayant eu un différent nombre d'opération.

L'angle θ influence quant à lui la probabilité d'application par rapport à la probabilité de non-application. On a donc la dynamique classique pour $\theta = \frac{\pi}{2}$, on a aucun changement d'une itération à l'autre pour $\theta = 0$, et des dynamique intermédiaire pour θ entre 0 et $\frac{\pi}{2}$.

4 Implémentation

L'implémentation que j'ai codé lors de ce stage est codé en C++, en reposant sur OpenMP pour le parallélisme et la bibliothèque **MPFR** pour permettre une plus grande précision de flottants dont nous avons besoin dans le calcul des probabilités.

4.1 Difficultés

Le nom des noeuds doit être représenté par un arbre binaire (cf section 2.1), ce qui est une structure de donnée irrégulière et peut donc facilement limiter la vitesses d'accès mémoire en causant de la fragmentation en fonction des implémentations et de leur flexibilité.

On a une croissance exponentielle du nombre de graphes puisque chaque graphe tend à avoir un nombre constant en moyenne de sous-graphes engendrés après une étape de dynamique quantique. Ce qui nous condamne à être limités en mémoire après un faible nombre d'itérations.

De plus chaque graphes est un relativement petit objet, et en créer et en détruire en permanence force un grand nombre d'allocations et de désallocations de petits objets, ce qui peut être très coûteux.

La dynamique quantique crée des graphes identiques qui doivent interférer (leurs amplitudes doivent être sommées) ce qui nous force à opérer un grand nombre de comparaisons d'objets irréguliers.

4.2 Techniques utilisé

Nous représentons les noms de noeuds de manière continue en mémoire en utilisant des vecteurs d'indices à gauche et à droite plutôt que des containers. Ce qui nous force à prévoir le nombre maximal de noeuds dans ces arbres avant de créer les graphes.

De plus, chaque propriété, relatives aux graphes individuels, aux noeuds ou aux arbre de noeud, est représentée sous la forme d'un seul vecteur contiguë (que l'on ne fait qu'agrandir si besoin, sans jamais le rétrécir, et en s'interdisant les insertion et les suppression). On se limite donc à une faible nombre constant (autour d'un vingtaine) de grand objet

contiguë, ce qui permet de meilleurs accès mémoire en évitant la fragmentation.

Pour limiter la croissance exponentielle, nous gardons les N graphes les plus probables après chaque itération. Nous prenons N le plus grand possible sans pour autant remplir l'entièreté de la mémoire disponible (N est de l'ordre de 10^7 pour une machine avec 128Go de mémoire et pour des graphes de 10 noeud de départ).

Pour pouvoir garder une représentation continue en mémoire de tout nos objets, et pour simplifier les interférences on commence par une itération symbolique où:

- On calcule la taille de chaque objet à la prochaine itération pour pouvoir assigner des espaces mémoire continus (sur les vecteurs décrits précédemment) pour chacun d'entre eux ce qui permettra leur accès en parallèle sans risque de ré-allocation.
- On calcule un hash de chaque graphe, pour ensuite effectuer l'interférence en les comparant plutôt que de comparer les graphes. On peut ensuite s'assurer de ne générer que des graphes distincts puisque l'interférence se fait à l'itération symbolique plutôt qu'à l'itération finale.

Ces techniques nous permettent d'avoir un fort gain de performance avec le nombre de coeurs (gain quasi-linéaire jusqu'à 8 coeur avant d'être limité par la bande passante de la mémoire, et arriver à un gain de 20x pour 32 coeurs).

5 Résultats

5.1 Test et validation

Le test le plus important d'une dynamique quantique est sa réversibilité: on veut s'assurer que après N étapes, on récupère bien l'état de départ avec une probabilité 1 en effectuant N étapes de la dynamique inverse.

Sachant que $EC \circ EC = I$ et $SM \circ SM = I$, et en notant M^{-1} tel que $M \circ M^{-1} = I$, on peut trouver simplement l'inverse de règles composé simple:

- $(EC \circ M)^{-1} = M^{-1} \circ EC$
- $(SM \circ M)^{-1} = M^{-1} \circ SM$

On peut ensuite opérer ce test d'objectivité pour des milliers de graphes aléatoires de tailles différentes, pour des valeurs de N différentes, et pour chacune de ces dynamiques. Permettant ainsi à la fois de vérifier la validité de chaque dynamique mais aussi la validité de l'implémentation de l'application d'une dynamique quantique.

5.2 Simulation quantiques

Nous calculons la taille moyenne et le nombre de particules moyen pour un certain nombre d'itérations de chaque règle, et la probabilité totale (pour mesurer l'impact de la suppression de graphes), le nombre de graphes et le ratio du nombre de graphes avant et après interférence.

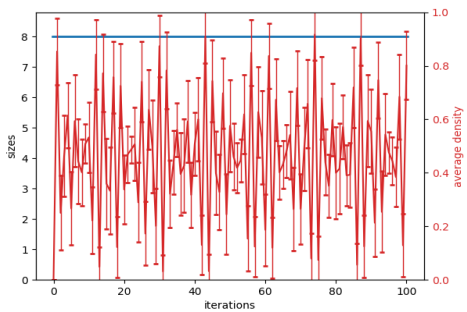


Figure 7: $EC \circ M$, $\theta = 0.375\pi$

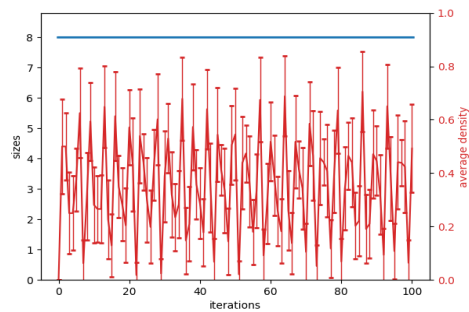


Figure 8: $EC \circ M$, $\theta = 0.25\pi$

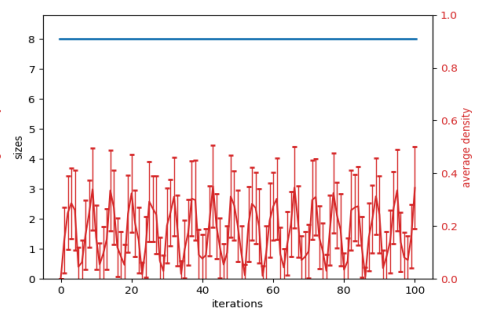


Figure 9: $EC \circ M$, $\theta = 0.125\pi$

Bleu foncé: probabilité total, Rouge: nombre de graphes (échelle log.), bleu clair: ratio du nombre de graphes avant/après interférence

On remarque que pour la dynamique $EC \circ M$ (figures 7 à 9, en partant d'un graphe vide) la densité de particules oscille avec une amplitude dépendante de θ (plus θ est grand plus la probabilité d'opération est grande, ce qui crée des particules à l'itération N et les détruit à l'itération $N-1$).

Pour la dynamique $SM \circ M$ (figures 11 à 13) la taille moyenne croît de manière classique (la densité décroît inversement au nombre de noeuds puisque le nombre de particules est constant) pour $\theta \neq 0.25\pi$ mais avec une croissance plus

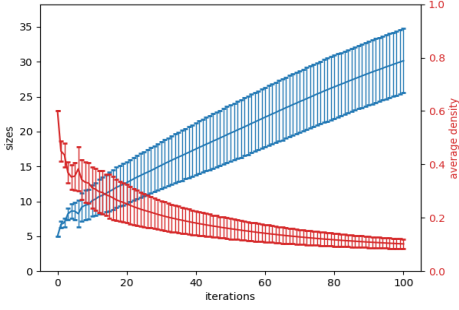


Figure 11: $SM \circ M$, $\theta = 0.375\pi$

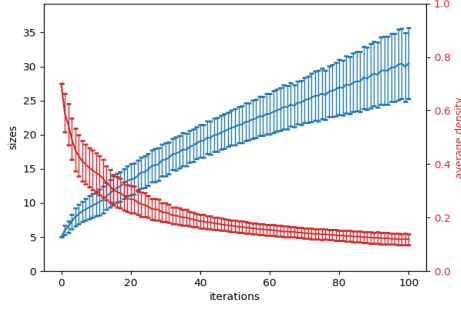


Figure 12: $SM \circ M$, $\theta = 0.25\pi$

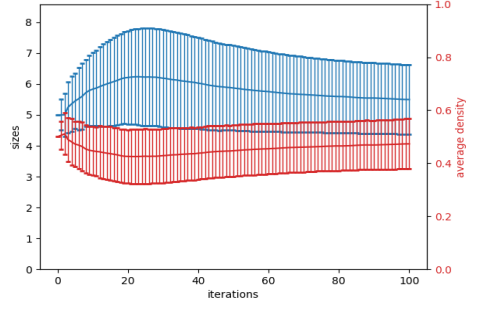


Figure 13: $SM \circ M$, $\theta = 0.125\pi$

Bleu: taille moyenne, Rouge: densité moyenne

faible pour θ plus faible.

Pour $\theta = 0.125\pi$ On a d'abord une croissance, puis une décroissance lorsque le nombre de graphes explose et que les graphes ayant beaucoup de "split merge" (et donc les plus grand) deviennent exponentiellement moins probables par rapport aux graphes restés de taille constante. Il n'est pas clair si ce résultat est uniquement dû à la suppression des graphes les moins probables ou non.

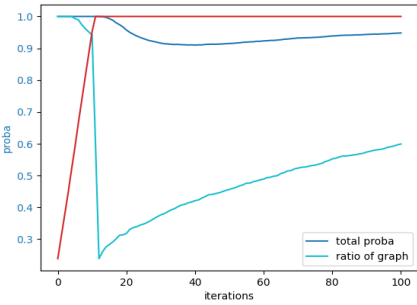


Figure 15: $SM \circ M$, $\theta = 0.375\pi$

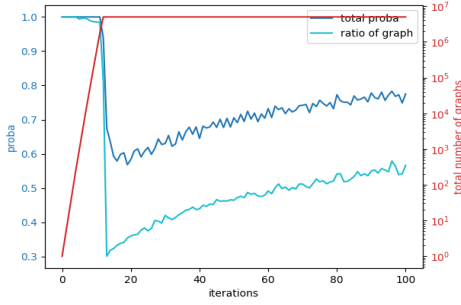


Figure 16: $SM \circ M$, $\theta = 0.25\pi$

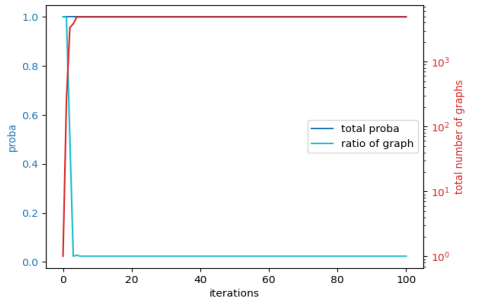


Figure 17: $EC \circ M$, $\theta = 0.25\pi$

Bleu foncé: probabilité total, Rouge: nombre de graphes (échelle log.), bleu clair: ratio du nombre de graphes avant/après interférence

Les instabilités pour la dynamique $SM \circ M$ et $\theta = 0.25\pi$ sont expliquées par la figure 16: on peut voir que l'on a une forte perte de probabilité totale, qui est due à une trop faible variation de probabilité des graphes (car $\theta = 0.25\pi$) ce qui amène à avoir une perte de probabilité quasi-égale à la proportion de graphes supprimés.

On remarque aussi que la croissance du nombre de graphes est très exactement exponentielle (droite en échelle log), et que la dynamique $EC \circ M$ (figure 17) a un comportement typiquement quantique puisque son ratio de graphes est proche de zéro (très grand nombre de graphes qui interagissent) car elle est limitée en nombre de graphes par 4^n avec n le nombre de noeuds.

6 Suite du stage

Maintenant que l'on a la capacité de faire une simulation quantique, les objectifs pour les 4 prochaines semaines de stage sont:

- Concevoir une dynamique qui cause une croissance exponentielle du nombre de noeuds.
- Observer des comportements typiquement quantiques (distinct du comportement probabiliste, et dépendant de l'angle ϕ).
- Observer les limites des approximations faites.

7 Conclusion

Les techniques de simulation et programmation, et les approximations utilisées jusqu'ici ont permis d'obtenir des résultats de simulation pour un problème qui, sans approximation est de complexité au moins exponentielle (avec le nombre de graphes). On peut maintenant espérer concevoir une dynamique causant une croissance exponentielle de la taille, bien que certains résultats préliminaires montrent que cette dynamique peut être plus complexe qu'il n'y paraît (non-stabilité de la densité avec la règle $EC \circ M$, et une observation récente que une dynamique $EC \circ M \circ SM \circ M$ ou $EC \circ SM \circ M$ a tendance à revenir à l'état de départ et donc à empêcher la croissance).

Ce stage est ma première expérience de programmation pour la recherche. C'est aussi la première fois que j'ai eu l'occasion d'utiliser des machines si puissantes, ce qui amène des considérations d'HPC que je n'ai jamais eues, et m'a permis de prendre en considération des aspects d'optimisation très enrichissants.

References

- [1] P. Arrighi, M. Christodoulou, and A. Durbec. Quantum superpositions of graphs, 2020.
- [2] P. Arrighi, N. Durbec, and A. Emmanuel. *Reversibility vs Local Creation/Destruction*, pages 51–66. Springer International Publishing, Cham, 2019.
- [3] B. Hasslacher and D. A. Meyer. Modeling dynamical geometry with lattice gas automata. 6 1998.
- [4] D. A. Meyer. From quantum cellular automata to quantum lattice gases. *Journal of Statistical Physics*, 85(5-6):551–574, Dec 1996.