

Article de vulgarisation

Au cours des vingt dernières années, la réalisation d'un ordinateur quantique a été un des points focaux de la recherche en physique. Un tel outil, suggéré apte à réaliser des opérations sur une échelle inaccessible aux ordinateurs classiques, permettrait aux chercheurs de résoudre certaines questions qui sont encore hors d'atteinte avec les moyens technologiques actuels. Les ordinateurs quantiques seraient possiblement capables de traiter des problèmes d'une grande complexité ou qui demandent des temps de calculs déraisonnables pour un ordinateur classique. Bien qu'aucune machine dite quantique ait montré une amélioration significative dans son temps de computation comparé aux algorithmes classiques utilisés, des compagnies telles que D-Wave Systems imaginent un futur dans lequel ces ordinateurs pourront aider à la conception de nouveaux médicaments, à la manufacture d'automobiles qui peuvent se conduire elles-mêmes et à l'application de la méthode d'optimisation Monte-Carlo, utilisée dans la quasi-totalité des secteurs privés et publics. Présentement, les poussées scientifiques sont divisées en deux catégories principales : la réalisation d'un ordinateur quantique physique, et le développement de codes et d'algorithmes pour ce type d'ordinateur. L'équipe de David Poulin se concentre sur ce deuxième volet, soit l'informatique quantique.

Mais quelles sont les différences entre un ordinateur classique et un ordinateur quantique? La différence principale réside dans leur manière de stocker l'information : un ordinateur classique utilise des bits, tandis que son homologue quantique programme avec des qubits. Tandis qu'un bit classique existe d'une manière binaire, soit 1 ou 0, un qubit peut être dans les deux états *simultanément*, ce qui offre une plus grande puissance de calcul. Ce phénomène se nomme *superposition*, et est intrinsèque à la mécanique quantique. Or, cet état est précaire : si le qubit est observé, mesuré ou interagit de quelque manière avec son environnement, la superposition est détruite et l'opération devient inutile. Cette différence peut être illustrée en imaginant deux trains. Le premier train, qui est "classique", embarque une certaine quantité de passagers à la gare, puis fait quelques arrêts lors desquels certains montent à bord et d'autres descendent. À la fin du rail, tous les passagers descendent. Si le conducteur le désire, il pourrait savoir combien de personnes ont embarqué et débarqué du train à chaque



arrêt, mais si le même scénario était reproduit avec un train "quantique", il ne pourrait savoir que le nombre initial et le nombre final de passagers, et essayer d'observer ce qui se passe lors d'un arrêt intermédiaire dérailerait le train ! Une autre caractéristique de la mécanique quantique est la notion d'enchevêtrement, qui se produit quand l'état de plusieurs particules ne peut être décrit particule par particule, seulement pour l'ensemble du groupe. Cela a ses avantages : le plus de particules sont enchevêtrées dans un système, le plus robuste celui-ci sera. Le véritable problème émerge quand un qubit devient enchevêtré avec une particule qui fait partie de l'environnement externe. Cette occurrence se produit en la présence de stimuli externe, telle la chaleur ou un champ électromagnétique, et amène le qubit dans un état excité, i.e. une erreur.

Considérant qu'un ordinateur quantique ne peut pas savoir l'état de chaque particule dans son environnement pour des raisons évidentes, il est impossible de connaître l'état d'un qubit enchevêtré avec une telle particule. Donc, cet état doit être tracé, ce qui mène à une perte d'information due aux phénomènes expliqués précédemment. Le but de la recherche en informatique quantique est de trouver des méthodes de correction d'erreurs qui minimisent ou éliminent cette perte, ainsi que de trouver des systèmes qui réduisent la fréquence de ces erreurs. Bien que de tels problèmes apparaissent également en informatique classique, les méthodes de correction sont plus faciles à implémenter et le type d'erreur est moins varié. Réussir à préserver l'information quantique est un des premiers pas vers la création d'un ordinateur quantique réaliste. Le code utilisé dans cet article est le code torique, qui a pour but de prolonger le temps de vie de l'information quantique en rendant le code plus tolérant à l'apparition d'erreur, c'est-à-dire que la création d'une erreur dans le code ne mènerait pas nécessairement à la destruction d'information.

Dans le code torique, les qubit sont arrangés en grille sur un tore en deux dimensions. Quand une excitation apparaît, elle génère deux syndromes qui peuvent se déplacer sans coût d'énergie sur le tore (Figure 1a). Ces syndromes peuvent se déplacer jusqu'à temps qu'ils se retrouvent, formant une boucle (Figure 1b). Si cette boucle peut être diminuée jusqu'à sa disparition, aucune erreur ne s'est produite, mais si la boucle entoure le tore dans un des deux axes, une erreur logique est apparue et une partie de l'information est perdue de manière



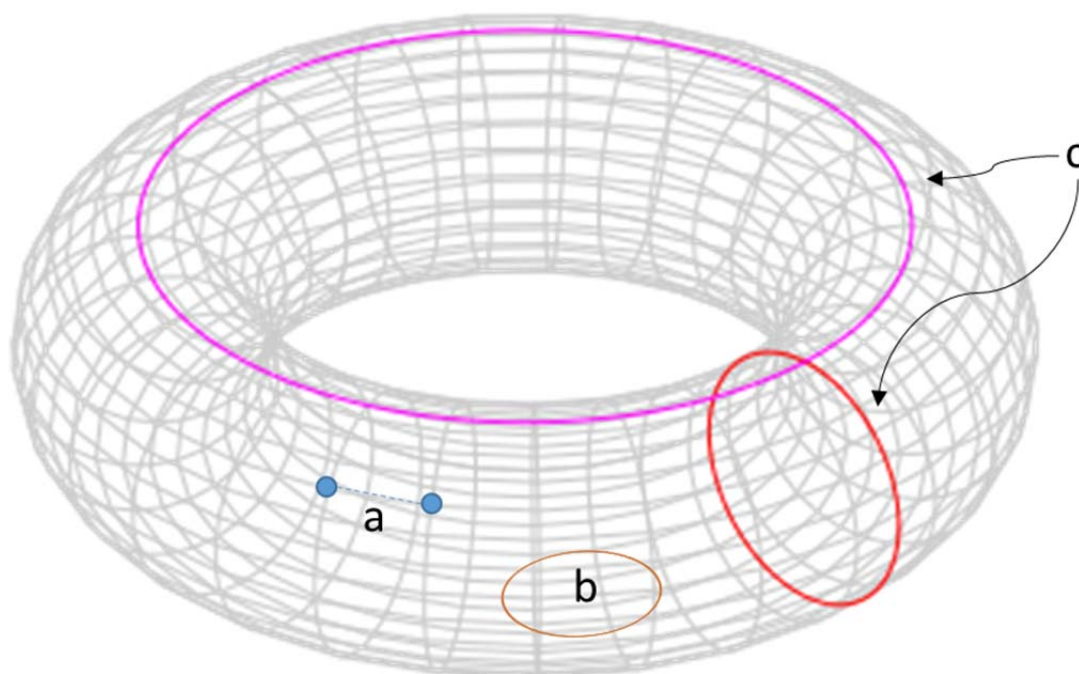


Figure 1

permanente (Figure 1c). Le code torique fonctionne en mesurant la position de ces syndromes, et en utilisant un algorithme qui les regroupe en paires de façon à minimiser la distance totale entre eux. Si une paire de syndromes ou une chaîne de paires fait le tour du tore, une erreur logique est produite. Un modèle proposé par Hamma et al. (2009) élimine supposément le besoin d'un code de correction d'erreur, substituant une force de gravitation engendrée par un champ bosonique. Selon leur article, les syndromes vont être attirés par cette force et vont éventuellement s'effondrer en un point, conservant l'information pour une durée proportionnelle à la taille du tore, si la température de l'environnement est en-dessous d'un certain point, c'est-à-dire si la densité des excitations sur le tore est assez faible ; une fois cette densité surpassée, une augmentation de la taille diminuerait le temps de cohérence du système. Cette méthode permettrait de créer des systèmes qui pourraient prolonger le temps de cohérence de l'information pour une durée limitée seulement par leur taille, rendant ainsi une mémoire quantique macroscopiquement significative atteignable.

Le but du projet sur lequel j'ai travaillé lors de mon stage est de vérifier les affirmations faites par cet article. Premièrement, j'ai défini correctement le potentiel de gravitation émis par

une particule de masse unitaire sur le tore. L'échelle de distance et de masse sur le tore étant définies par la taille et la masse d'un syndrome, j'ai pu simplifier plusieurs facteurs de proportion triviaux dans mes équations en redéfinissant l'unité de masse. Le calcul du potentiel, et subséquemment de la force de gravitation, était particulièrement difficile dû à la topologie avec laquelle nous travaillons. Étant donné que le tore est périodique par définition, la force de gravitation s'applique multiple fois sur une même particule avec des orientations opposées. Donc, utilisant des techniques de sommation de séries développées par Eisenstein, j'ai réussi à trouver une formule analytique pour le potentiel et la force. Cependant, appliquer cette formule dans le code informatique simulant un code torique est futile, car calculer la force exacte de gravitation sur chaque particule prend un temps immense comparativement à la troncation d'une somme, et n'augmente pas la précision suffisamment pour compenser l'augmentation du temps de calcul. En comparant plusieurs méthodes de troncation, nous en avons trouvé deux comparables à la méthode analytique, et nous avons utilisé celle qui était plus précise.



La prochaine étape était de confirmer que le code de simulation marchait correctement, ce qui a été accompli en générant une certaine densité d'excitations (la densité étant définie par nombre de syndromes/aire totale du système) initiale, laissant ensuite le système évoluer par soi-même. Les particules sont créées possédant une vitesse initiale et s'attirent mutuellement jusqu'à temps qu'elles soient à une distance unitaire l'une de l'autre, s'annulant quand cette condition est remplie. La simulation évolue temporellement par incrémentation de 0.01 unité temporelle, ce qui préserve la trajectoire des particules et l'énergie totale du système. Cette simulation était répétée 2000 fois par permutation des paramètres, qui incluaient la densité de création, le diamètre du tore et la friction. Pour une certaine friction donnée, nous observons un graphique du nombre de simulation dans lesquelles une erreur logique avait été retrouvée en fonction de la densité et de la taille (Figure 2). Sous le seuil de densité discuté précédemment, le nombre d'erreur moyen devrait être moins élevé pour un



tore de grande taille comparativement à un tore de taille inférieure, et l'inverse serait observé en haut du seuil. Nous avons donc réussi à observer que le seuil attendu était autour de 1% à 3%.

L'étape suivante dans le projet était de répéter les simulations, mais cette fois en ajoutant la quatrième dimension, c'est-à-dire en faisant évoluer le système lors de la création initiale. Nous avons permis aux syndromes de se déplacer pendant une certaine période pendant la création, puis nous arrêtons la création d'excitations et mesurons les mêmes facteurs qu'auparavant, en plus de la variation dans le nombre de particules dans le système. Nous avons remarqué que la densité de saturation du système était d'environ 230% à 250% de la densité initiale (Figure 3), mais que le seuil de saturation restait le même. L'idée nous est

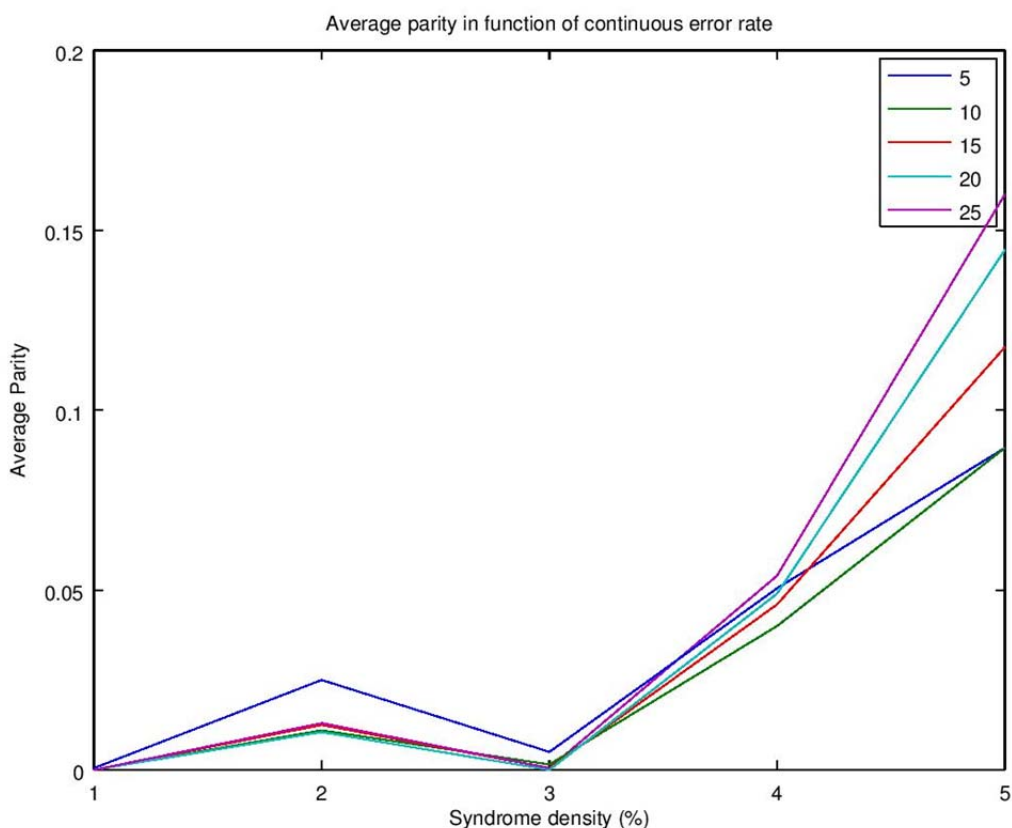


Figure 2

ensuite venue de laisser la simulation continuer, sans arrêter la création, mais d'utiliser un programme indépendant qui mesurerait la présence d'erreur logique à un intervalle de temps spécifique, puis d'enregistrer le temps de cohérence directement. Ces simulations ont posées deux problèmes particulier : la boucle « while » utilisée ne coupait pas la simulation correctement, et le temps de simulation est très élevé, prenant en moyenne 18 heures pour une seule itération du code de simulation. Bien que nous ayons pu modifier le code avec un autre type de boucle, le temps de simulation est encore un problème considérable, car nous n'avons pas une quantité suffisante de simulations pour éliminer le bruit dans nos résultats.

Les résultats obtenus par ces dernières simulations ont fait apparaître une nouvelle complication : le seuil a disparu des graphiques. Les courbes se comportent maintenant comme si elles étaient au-delà du seuil, même à une densité de 0.01%, ce qui s'approche des limites possibles des simulations numériques (Figure 3). De plus, ces résultats sont très incertains, avec des erreurs variant entre 40% et 105%, donc même si les courbes étaient mieux définies, nous ne pourrions pas distinguer les temps de décohérence moyens les uns des autres.



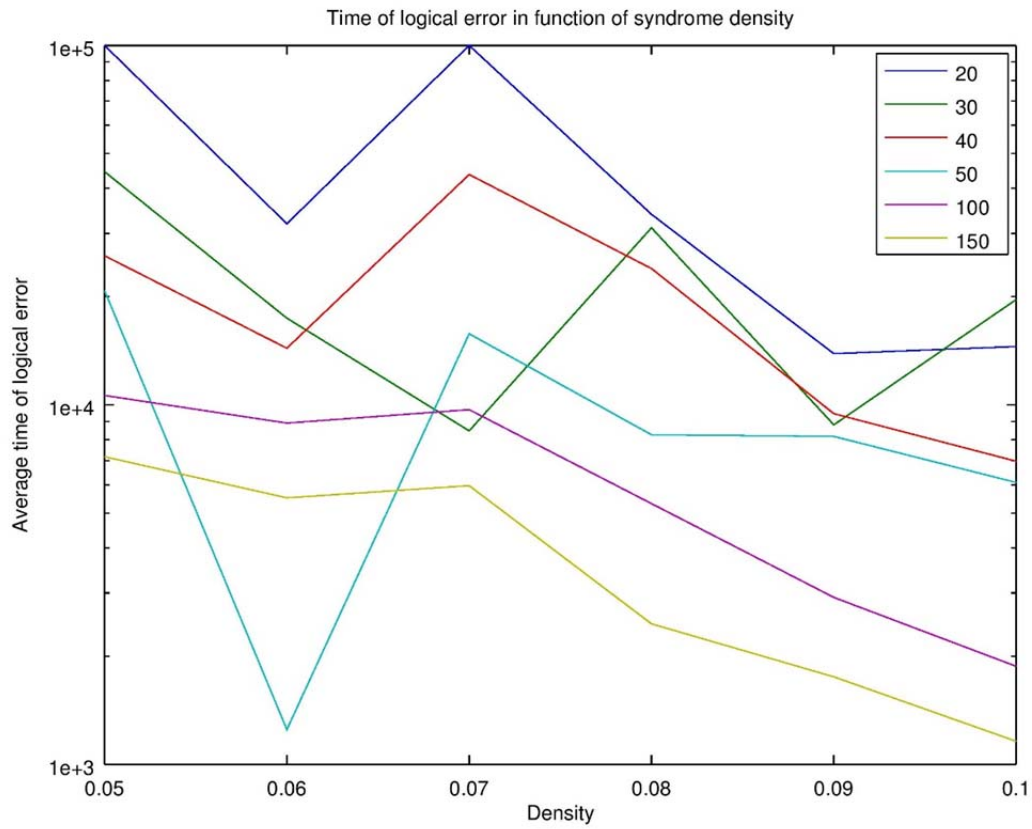


Figure 3

Donc, bien que les simulations initiales aient bien marché, les derniers résultats que

nous avons retrouvés montrent clairement qu'il existe soit une erreur dans le code de simulation, ou certains facteurs dans le système physique que nous ne prenons pas en considération. Nous simulons présentement certains systèmes qui ont pour but de comparer les deux différentes méthodes de décodage utilisées lors des simulations, et si les résultats ne sont pas fructueux nous allons retourner au concept initial afin de déterminer la meilleure approche à suivre.

