Groupe EMY

Groupe EMY

1^{er} juin 2022

1 Introduction

Le but de ce document est d'aider à la mise en œuvre d'améliorations de performances. Ce travail se décline en trois parties : l'**analyse** (de la complexité temporelle C.T. et spatiale C.S.), les **propositions** (idées pour rendre les algorithmes plus efficaces), et l'**implémentation** (qui ne se trouve pas dans ce document mais directement dans des fichiers Python).

2 Analyse

On note une *opération* comme on le ferait avec des variables (avec $\cdot$ et $+$), mais en remplaçant la variable par l'ensemble auquel elle appartient : par exemple $\mathcal{M}_{n,p}(\mathbb{Z}) \cdot \mathcal{M}_{p,r}(\mathbb{Z})$ désigne la multiplication de deux matrices. La *complexité temporelle* d'une opération $\mathcal{O}$ est notée $T(\mathcal{O})$, et sa *complexité spatiale* est notée $S(\mathcal{O})$. Le signe ⚠ signale opérations qui sont jugées sous-optimales.

2.1 Calcul formel

2.1.1 C.T. d'instanciation

L'opération d'instanciation de E est notée $\tilde{E}$, celle d'égalité $= E$, la fonction `sous` est notée $\downarrow$ et `sur`(E) est notée $\uparrow E$.

- $T(\tilde{\mathbb{N}}) = T(\tilde{\mathbb{Z}}) = O(1)$
- $T(\tilde{0}) = T(\tilde{1}) = O(0)$
- $T(\tilde{\mathbb{Q}}(a, b)) = O(\log_2(\min(|a|, |b|)))$
- $T(\tilde{\mathbb{P}}(x, p)) = T(\tilde{\mathbb{Q}}(x)) + T(\tilde{\mathbb{Q}}(p))$
- $T(\tilde{\mathbb{C}}(x, y)) = T(\downarrow x) + T(\downarrow y)$
- $T(\texttt{Matrice}(p, q)) = O(pq)$ (zéros) ⚠
- $T(E_1 + E_2) = O(T(E_1 \uparrow E_2) + T(E_2 \uparrow E_1))$ ⚠
- $T(E_1 \cdot E_2) = O(T(E_1 \uparrow E_2) + T(E_2 \uparrow E_1))$ ⚠

2.1.2 C.T. d'opérations algébriques

Pour les opérations sur les matrices :

- $T(\mathcal{M}_{p,q}(E_1) + \mathcal{M}_{p,q}(E_2)) = O(pq \cdot T(E_1 + E_2))$

- $T(\mathcal{M}_{p,q}(E_1) \cdot \mathcal{M}_{q,r}(E_2)) = O(pqr \cdot T(E_1 \cdot E_2))$ ⚠
- $T(\mathcal{M}_{a,b}(E_1) \otimes \mathcal{M}_{c,d}(E_2)) = O(abcd \cdot T(E_1 \cdot E_2))$ ⚠
- $T(\mathcal{M}_{p,q}(E)^{\otimes n}) = O(n \log_2 n \cdot T(E \cdot E))$

2.2 Circuit quantique

Dans cette partie, on considère que les opérations sur les scalaires, appartenant au corps $\mathbb{K}$, se font en $O(1)$. On considère un circuit quantique C de n qubits, non nécessairement états propres au cours du calcul, et on dispose d'un circuit de m étapes, c'est-à-dire que chaque qubit passe à travers m portes. On suppose que l'état initial est propre et que chaque étape est composée de n portes prenant un qubit en entrée chacune.

$$\begin{aligned}
T(C) &= T(\texttt{création qudit}) + m \cdot T(\texttt{cr. étape}) + m \cdot T(\texttt{passage étape}) \\
&= O(2^n) + O(m2^{n^2+n}) + O(m4^n) \quad \text{(cf. ci-dessous)} \\
&= O(m2^{n^2+n})
\end{aligned}$$

2.2.1 C.T. de la création du qudit

$$\begin{aligned}
T(\texttt{cr. qudit}) &= \sum_{i=1}^{n} T(\texttt{cr. qubit}) + T(\mathcal{M}_{2^i,1}(\mathbb{K}) \otimes \mathcal{M}_{2,1}(\mathbb{K})) \\
&= \sum_{i=1}^{n} O(1) + O(2^i \cdot 1 \cdot 2 \cdot 1) \\
&= O((4 \cdot 2^n - 3) \\
&= O(2^n)
\end{aligned}$$

2.2.2 C.T. de la création des étapes

$$\begin{aligned}
T(\texttt{cr. étape}) &= \sum_{i=1}^{n} T(\mathcal{M}_{(2^n)^i}(\mathbb{K}) \otimes \mathcal{M}_{2^n}(\mathbb{K})) \\
&= \sum_{i=1}^{n} O(2^{n(i+1)}) \\
&= O\left(2^n \sum_{i=1}^{n} (2^n)^i\right) \\
&= O\left(2^n \left(2^{n^2} - 1\right)\left(1 + \frac{1}{2^n - 1}\right)\right) \\
&= O(2^{n^2+n}) \quad (?)
\end{aligned}$$

2.2.3 C.T du passage des étapes

On travaille d'abord sans chaînage : le qudit d'état passe successivement dans chaque porte :

$$\begin{aligned} m \cdot T(\texttt{pass. étape}) &= m \cdot T(\mathcal{M}_{2^n}(\mathbb{K}) \cdot \mathcal{M}_{2^n,1}(\mathbb{K})) \\ &= m \cdot O(2^n \cdot 2^n \cdot 1) \\ &= O(m4^n) \end{aligned}$$

Comparons avec un travail en chaînage total. On suppose que toutes les portes sont chaînables, et on les multiplie entre elles :

$$\begin{aligned} m \cdot T(\texttt{pass. étape}) &= T(\texttt{chaînage}) + T(\texttt{pass. qudit}) \\ &= m \cdot T(\mathcal{M}_{2^n}(\mathbb{K}) \cdot \mathcal{M}_{2^n}(\mathbb{K})) + T(\mathcal{M}_{2^n}(\mathbb{K}) \cdot \mathcal{M}_{2^n,1}(\mathbb{K})) \\ &= m \cdot O((2^n)^3) + O(4^n) \\ &= O(m8^n) \end{aligned}$$

On observe une complexité bien plus importante que pour un calcul sans chaînage : on limitera donc le chaînage à de petits circuits.

3 Propositions

3.1 Calcul formel

Pour les matrices :

- Le constructeur `Matrice` se comporte comme `Matrice.tableau` (qui est supprimé), et on implémente `Matrice.zeros`
- Multiplication de matrices avec 0 et 1
- Matrice tensorielle identité
- Matrices avec entiers de Gauss $\mathbb{P} \times \mathcal{M}(\mathbb{Z}[i]))$
- Strassen (pour de grandes matrices)

Pour le typage :

- Simplification dans la fonction `Nombre.sur` (pour les cas où on retourne `None`)
- Supprimer le type `Zero`

3.2 Portes et qudits

- Formule générale pour $H^{\otimes n}$