TECHNICAL REPORT: AMO41

Quantum-Computer Algorithms

Author(s): Samuel Aliprandi **Date:** August 30, 2023

1 Summary

This report describes how to reproduce a quantum simulator for a Gaussian wave packet, as in the paper [1] , using the library Quantum++ of C++.

2 Updates

The total internship period consists of 12 weeks.

week 11-12: correct the gate implementation for the potentials and get the final version of the code for five different situations.

week 8-9-10: wrong output, comparison between FFT (from scipy) and QFT and analytical calculation. Correct the significance of the qubits, since it is inverted compared to the paper one.

week 6-7: set the circuit equal to the null circuit at every loop iteration.

week 4-5: gate implementations. For switching in the momentum space, it is necessary to apply the CQFT (centered QFT), i.e. we need to apply an X-gate on the most significant qubit before and after the QFT (and the TFQ).

week 1-2-3: papers reading and library testing, in particular test and check of the QFT implementation.

3 Objective of the work reported

New implementation of a quantum circuit for simulating the time evolution of a Gaussian wave packet under different potentials using the library Quantum++ of C++.
A simulation of a quantum system by means of a quantum circuit makes possible to represent an exponentially large number of lattice sites in a small set of qubits, encoding the amplitudes of the wave function in the different basis states. Thus, we can represent such a huge Hilbert space due to their entanglement, which is hard for classical computers, particularly if more particles are added.

4 Present approach

In order to make the code efficient for a classical computer it was used the library Quantum++ of C++, specifically the implementation was made by the circuit option instead of the matrix representation option. In such a way it is easier to implement the different gates and apply them in the circuit.

4.1 Analysis of the problem

The problem consists of the time evolution of an initial state given by the following Gaussian:

$$\psi(x,0) = \frac{1}{\sqrt{\sqrt{\pi}\sigma}} \exp\left\{ -\frac{(x-x_0)^2}{2\sigma^2} + \frac{i}{\hbar}\left[p_0\,(x-x_0)\right] \right\}. \tag{1}$$

As it is explained in the paper [1], we discretize the wave function in a 2^n grid, where n is the number of qubits. The position grid is $x_k = -d + \left(k + \frac{1}{2}\right)\Delta x$ where $\Delta x = \frac{2d}{2^n}$ and $0 \leq k \leq 2^n - 1$. Thus $\psi(x_k, 0)$ is

calculated for every x_k and the obtained amplitudes are multiplied for the correspondent basis state of the initial state vector and normalized, i.e.

$$\sum_{k=0}^{2^n-1} c_k(t)|\, k\,\rangle = \frac{1}{N}\sum_{k=0}^{2^n-1} \psi(x_k,t)|\, k\,\rangle \tag{2}$$

where $|\, k\,\rangle = |\, k_{n-1}\,\rangle|\, k_{n-2}\,\rangle \ldots |\, k_0\,\rangle$ is a state of the computational basis.

To make the particle evolves for an ϵ time step from a certain time t, we write

$$\psi(x, t+\epsilon) = e^{-i[H_0+V(x)]\epsilon/\hbar}\psi(x,t), \tag{3}$$

where if ϵ is small enough, we can use the Trotter decomposition and write the time evolution operator as

$$e^{-i[H_0+V(x)]\epsilon/\hbar} \approx e^{-iH_0\epsilon/\hbar}e^{-iV(x)\epsilon/\hbar}. \tag{4}$$

Then, in order to have the free energy time evolution operator diagonalized, we need to write in the p representation by means of the Fourier transform. This means that the time evolution operator is given by

$$F^{-1}e^{-i\left(p^2/2m\right)\epsilon/\hbar}Fe^{-iV(x)\epsilon/\hbar}, \tag{5}$$

where F is the Quantum Fourier Transform. When such an operator is applied l times to the initial state, this evolves for a time $t = l\epsilon$. Let us now see the implementation of the operators.

4.2 Method

The approach is similar for the different operators, except for the coefficients and the grid used.

Quantum Fourier Trasform

We used the already implemented QFT and TFQ $(= QFT^{-1})$ of Quantum++. However, in order to transform the operator properly we need to apply the CQFT (centered QFT), as in [2], i.e. we need to apply a Pauli-X gate on the most significant qubit before and after the QFT.

It is important to notice that the convention about most significant qubit is inverted between [1] and Quantum++. Indeed the paper consider as most significant qubit k_{n-1}, instead in the code it is necessary to consider as most significant qubit k_0.

Free energy operator

For the free energy time evolution operator we are in the momentum representation and so the p grid, similar to how is reported in [2], is

$$p_k = \left(k - \frac{2^n}{2}\right)\Delta p, \qquad 0 \le k \le 2^n - 1 \tag{6}$$

where $\Delta p = \frac{\pi}{d}$. Similarly to the calculation for the x grid in [1], we can write the p grid as

$$p_k = \alpha\sum_{j=0}^{n-1}(k_j 2^j + \beta) \tag{7}$$

with $\alpha = \Delta p$ and $\beta = -\frac{2^n}{2n}$. Thus we can write the free energy time evolution operator as

$$e^{-i\left(p^2/2m\right)\epsilon/\hbar} = \prod_{j,l=0}^{n-1} e^{-i\gamma(k_j 2^j+\beta)(k_l 2^l+\beta)} \tag{8}$$

with the coefficient $\gamma = \frac{\epsilon\alpha^2}{2m\hbar} = \frac{\epsilon\alpha^2}{2}$, where for sake of simplicity we consider $m = \hbar = 1$.

Expanding the product, we obtain three different phase operators:

- a two-qubits controlled-phase gate (j control qubit, l target qubit)

$$\exp(-i\gamma 2^{j+l} k_j k_l), \tag{9}$$

which target-qubit matrix is

$$R_{j,l} = \begin{pmatrix} 1 & 0 \\ 0 & \exp\left\{-i\gamma 2^{j+l}\right\}; \end{pmatrix} \tag{10}$$

- a single-qubit phase gate

$$\exp(-i\gamma\beta n 2^{j+1}\ k_j) \tag{11}$$

which single-qubit matrix is

$$R_j = \begin{pmatrix} 1 & 0 \\ 0 & \exp\left\{-i\gamma\beta n\, 2^{j+1}\right\} \end{pmatrix} \tag{12}$$

- a global phase gate

$$\exp(-i\gamma n^2\beta^2) \tag{13}$$

which single-qubit matrix is

$$R_{glob} = \begin{pmatrix} \exp\left\{-i\gamma n\beta^2\right\} & 0 \\ 0 & \exp\left\{-i\gamma n\beta^2\right\} \end{pmatrix} \tag{14}$$

As it is reported in subsection 4.2 the significancy of the qubits is inverted between the paper and Quantum++, thus we need to apply the j-gate to the $(n-1-j)$-qubit (obviously for the global phase doesn't change anything).
As it is shown also in the following circuit implementation, the number of gates scales as $O(n^2)$, with n number of qubits.

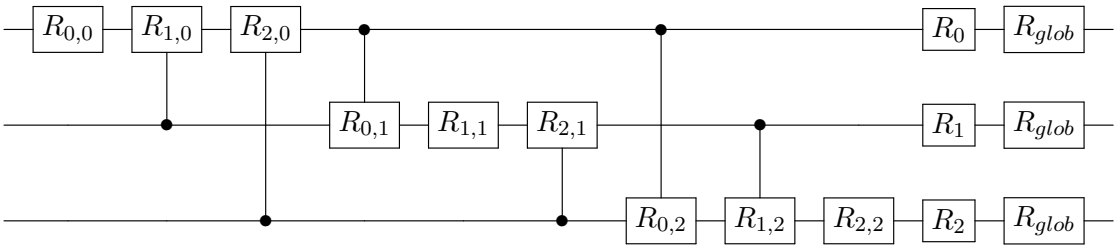


Figure 1: free energy operator for $n = 3$ qubits

Linear potential

The linear potential $V(x) = \omega x$ is implemented in the position representation then the grid is

$$x_k = -d + \left(k + \frac{1}{2}\right)\Delta x \tag{15}$$

where $\Delta x = \frac{2d}{2^n}$. As it was done in subsection 4.2, it can be written as

$$x_k = \alpha \sum_{j=0}^{n-1} (k_j 2^j + \beta) \tag{16}$$

with $\alpha = \Delta x$ and $\beta = \frac{-d+\Delta/2}{\alpha n}$. Then the product has just one index

$$e^{-iV(x)\epsilon/\hbar} = \prod_{j=0}^{n-1} e^{-i\gamma(k_j 2^j+\beta)}, \tag{17}$$

where $\gamma = \alpha\omega\epsilon/\hbar$. The two operators are:

- a single-qubit phase gate

$$\exp(-i\gamma 2^j\ k_j) \tag{18}$$

which single-qubit matrix is

$$R_j = \begin{pmatrix} 1 & 0 \\ 0 & \exp\{-i\gamma\, 2^j\} \end{pmatrix} \tag{19}$$

- a global phase gate

$$\exp(-i\gamma\beta n) \tag{20}$$

which single-qubit matrix is

$$R_{glob} = \begin{pmatrix} \exp\{-i\gamma\beta\} & 0 \\ 0 & \exp\{-i\gamma\beta\} \end{pmatrix} \tag{21}$$

As it is explained in subsection 4.2, the j-gate needs to be applied to the $(n-1-j)$-qubit. In this case, the number of gates scales linearly as $O(n)$ with the number of qubits.

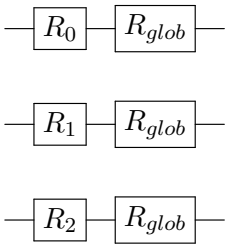


Figure 2: linear potential operator for $n = 3$ qubits

Harmonic potential

The grid used is the same as in the linear potential 4.2. The potential is $V(x) = \frac{x^2}{2}$ and the coefficient $\gamma = \frac{m\omega^2\alpha^2\epsilon}{2\hbar}$. The implementation of the three phase operator is the same as 4.2.

Laser field

The Hamiltonian of a particle in a laser field can be written as

$$H = \frac{p^2}{2m} + \frac{f(t)}{2m}p \qquad f(t) = \left[\cos^2\left(\frac{\omega t}{2N}\right)\right]\sin(\omega t). \tag{22}$$

Then the implementation of the p^2 part is as in 4.2, meanwhile the implementation of the p-linear part is as in 4.2, using the p-grid and setting $\gamma = \frac{\epsilon\alpha f(t)}{2m\hbar}$.

Superposition of two counterpropagating wave packets in harmonic potential

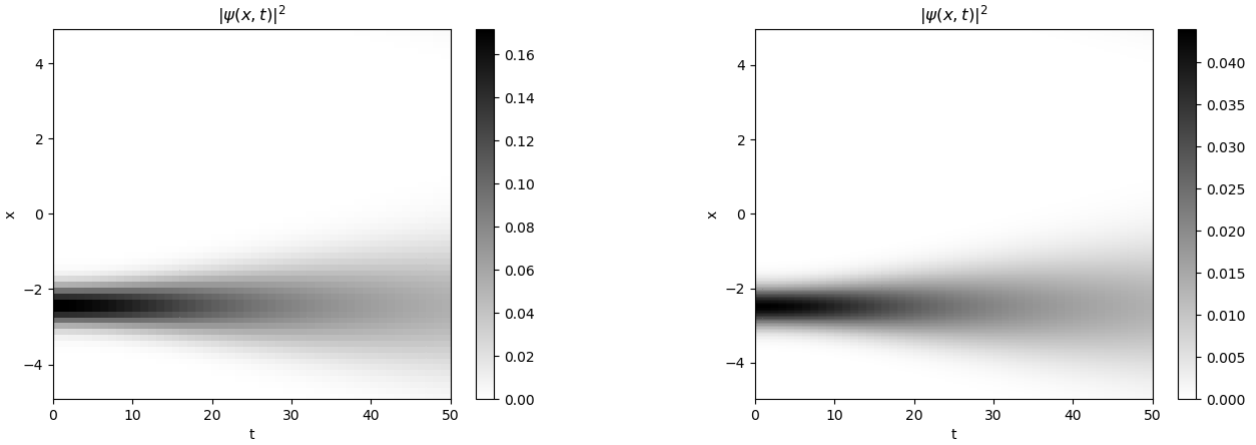
The implementation is the same as the harmonic potential for one Gaussian wave packet. The initial state is the normalized sum of two symmetric Gaussian wave packets, i.e. $\psi(x) = \psi_{x_0} + \psi_{-x_0}$.

5 Results

As it is shown below, plots are compared for $n = 6$ and $n = 8$ qubits for different potentials and parameters.

Free particle

$$d = 5 \qquad \epsilon = \frac{\pi}{200} \qquad x_0 = -2.5 \qquad p_0 = 0 \qquad \sigma = 0.5$$

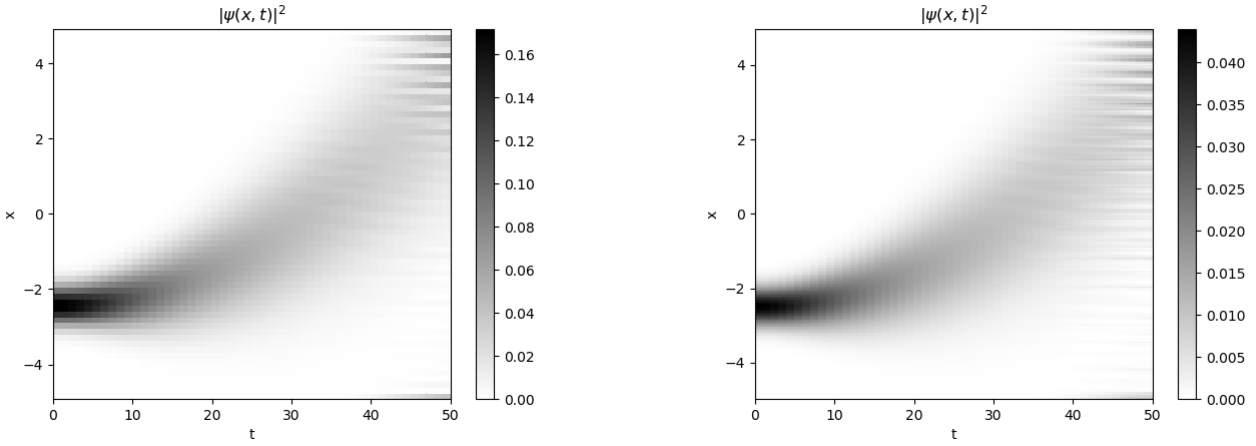


(a) $n = 6$, running time $\sim 1.21\ s$ (b) $n = 8$, running time $\sim 6.98\ s$

Figure 3: Free particle

Linear potential

$$d = 5 \qquad \epsilon = \frac{\pi}{100} \qquad x_0 = -2.5 \qquad p_0 = 0 \qquad \sigma = 0.5 \qquad V(x) = -4.8x$$

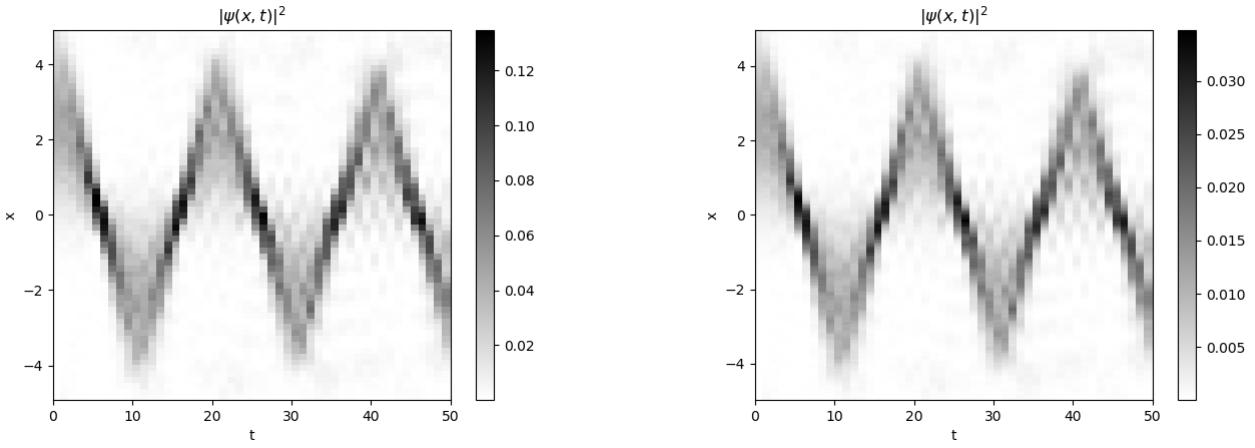


(a) $n = 6$, running time $\sim 1.40\ s$ (b) $n = 8$, running time $\sim 7.35\ s$

Figure 4: Linear potential

Harmonic potential

$$d = 5 \qquad \epsilon = \frac{\pi}{10} \qquad x_0 = 2.5 \qquad p_0 = 0 \qquad \sigma = 2 \qquad V(x) = \frac{x^2}{2}$$

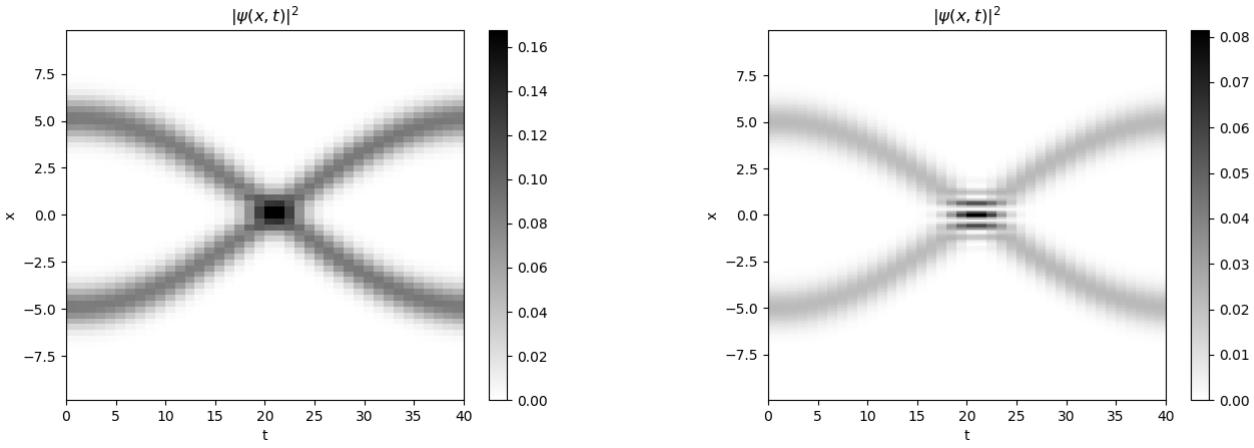


(a) $n = 6$, running time $\sim 1.74\ s$ (b) $n = 8$, running time $\sim 10.32\ s$

Figure 5: Harmonic potential

Superposition of two counterpropagating wave packets in harmonic potential

$$d = 10 \qquad \epsilon = \frac{\pi}{40} \qquad x_0 = \pm 5 \qquad p_0 = 0 \qquad \sigma = 1 \qquad V(x) = \frac{x^2}{2}$$



(a) $n = 6$, running time $\sim 1.80\ s$ (b) $n = 8$, running time $\sim 8.78\ s$

Figure 6: Superposition of two counterpropagating wave packets in harmonic potential

6 Conclusion

In this report, it has been presented a new implementation of a quantum simulator in different potentials using the C++ library Quantum++. The running times are exposed and, as it is easy to notice, they increase exponentially with the number of qubits.
For future works, it can be possible to implement other types of potential, i.e. the anharmonic potential. In such a case the difficulties consist of developing a circuit that, given a power of three for the potential, includes controlled-phase operators $R_{j,k,l}$ with two target qubits added to a similar implementation as in the harmonic potential, having then a number of gates that scales as $O(n^3)$.
However, for a generic potential there is currently no known method for achieving an exact implementation without experiencing an exponential growth in gate count in terms of qubits involved. The only way out here would be to use gate sequences that are approximate, meaning the discretized potential they resemble is not exactly the mathematical one, even in the limit of zero lattice spacing. In fact, this seems to be the biggest formal challenge when doing simulating quantum dynamics on a quantum computer.

Bibliography

[1] Giuliano Benenti and Giuliano Strini. Quantum simulation of the single-particle schrödinger equation. *American Journal of Physics*, 76(7):657–662, 2008.

[2] Taichi Kosugi, Yusuke Nishiya, Hirofumi Nishi, and Yu-ichiro Matsushita. Imaginary-time evolution using forward and backward real-time evolution with a single ancilla: First-quantized eigensolver algorithm for quantum chemistry. *Physical Review Research*, 4(3):033121, 2022.