JOURNAL OF APPLIED
COMPUTER SCIENCE
Vol. 20 No. 2 (2012), pp. 95-106

Quantum Simulations of Particle Scattered by a Rectangular Potential

Marcin Ostrowski[1]

[1]*Lodz University of Technology*
Institute of Information Technology
Wolczanska 215, 90-924 Lodz, Poland
marcin.ostrowski@p.lodz.pl

Abstract. *In this paper we examine whether a quantum computer can efficiently simulate time evolution of a one dimensional Schrödinger particle. Two cases are considered: free particle and particle dispersed on a rectangular potential. In order to simulate the Schrödinger particle, we use a quantum algorithm based on the procedure of diagonalisation of time evolution operator. The procedure of diagonalisation is based on the Quantum Fourier Transform (QFT) algorithm. The effects of simulation are presented in the form of figures.*

We also compare the results obtained from quantum algorithm with the results of classical simulations (Cayley's method).
Keywords: *quantum computations, quantum simulations, Schrödinger particle.*

1. Introduction

There is a growing need for computing power. Currently existing technologies, based on miniaturisation, are about to reach the end of their possibilities. Therefore, we are searching for new solutions. One alternative is a quantum computer [1]. Although its practical implementations have not been built yet, its existence

seems to be possible. Therefore, it is worth examining the properties of such machines.

Today we know some quantum algorithms that are faster than their best classical counterparts. They include Shor [2] and Grower [3] algorithms. Another promising application of quantum computer are quantum simulations, i.e. the computer modelling of behaviour of physical quantum systems.

As is well known, simulations of quantum systems performed using conventional computers are not very effective. This follows from the fact that the number of states that a quantum system can take grows exponentially with its size. For example, a system composed of n-spins (with each individual having two base states) may receive a total of 2^n base states. This means that for classical computer the memory resources and time required to simulate grow exponentially with n. In the case of a quantum computer, the situation is different. The heart of such computer is n-qubit register, which in itself is a quantum system that could take the 2^n base states. This means that the relationship between the size of quantum computer (register) and the size of the simulated quantum system is linear. Also the number of elementary operations (quantum gates) to be performed on the register as a rule does not depend exponentially on n. So we have a potentially large profit, namely the ability to efficiently simulate quantum systems with a relatively small quantum register.

Why are simulations of quantum systems interesting for us? Microscope systems such as elementary particles or chemical molecules are quantum systems. This means that accurate simulation of their behaviour may be conducted only on the basis of quantum mechanics. Among the particles that the researchers are particularly interested in are, for example, complex polymers used in medicine (e.g. drugs). However, for the time being, such complex cases are not analysed in this work. The present discussion has been limited to the analysis of the free Schrödinger particle in one dimension or particle dispersed on a rectangular potential (Fig. 1). Our aim is to examine the effectiveness of such an approach.

In order to simulate quantum register we wrote a program (in C++ language). It allows us to test (of course, for the register of small size) the software of the future quantum computer. Applications of classical computers in simulations of quantum computing is widely presented in the literature. In particular, it is worth mentioning here of many core computer architectures (many core CPU [4, 5, 6], many cores GPU [7], clusters etc.). There are also special packages for commercial software (e.g. for Mathematica [8]). An introduction to the basic computational models used in quantum information theory can be found in [9].

2. Simulations of the Schrödinger particle

The simulation algorithm used by us (described in the literature [10, 11, 12, 13] is based on the procedure of diagonalisation of time evolution operator. The procedure of diagonalisation is based on the the Quantum Fourier Transform (QFT).

Before we present the algorithm, let us briefly recall the basic properties of the Schrödinger particle and outline the basic ideas on which this algorithm is based.

Time evolution of the Schrödinger particle is described by the Schrödinger equation. In one-dimensional case, it takes the form:

$$i\hbar\frac{d}{dt}\Psi(x,t) = H\Psi(x,t) \tag{1}$$

where

$$H = H_0 + V(x) \tag{2}$$

is the Hamiltonian consisting of a free part $H_0 = p^2/(2m)$ and potential $V(x)$ describing interactions with external force. In our considerations confine ourselves to the (stationary) rectangular potential (Fig. 1).

The formal solution of the Eq. (1) is a expression in following form:

$$\Psi(x,t_1) = \exp(-iH\,\Delta t/\hbar)\,\Psi(x,t_0) = \exp(-i(H_0 + V(x))\Delta t)\Psi(x,t_0); \tag{3}$$

which describes the evolution of the system from the initial state at the time t_0 to state at time $t_1 = t_0 + \Delta t$.

In the case $\Delta t \to 0$ (which corresponds appropriately short time-step of the simulation) we can use the approximation:

$$\exp(-i(H_0 + V(x))\Delta t/\hbar) = \exp(-i\,H_0\Delta t/\hbar)\exp(-i\,V(x)\Delta t/\hbar) \tag{4}$$

which allows a separate simulation of the free evolution and the contribution from the potential.

Operator $\exp(-i\,V(x)\Delta t/\hbar)$ acting on the wave function $\Psi(x,t)$ only multiplies it by the phase factor. The situation is different for the free evolution operator $\exp(-i\,H_0\Delta t/\hbar)$. However, if the wave function $\Psi(x,t)$ is presented in the momentum representation $\tilde{\Psi}(p,t)$ then the action of free evolution operator also comes down to multiplying the wave function by the phase factor:

$$\exp(-i\,H_0\Delta t/\hbar)\tilde{\Psi}(p,t) = \exp(-i\,p^2/(2m\hbar))\tilde{\Psi}(p,t). \tag{5}$$

The transition between the two representations is described by Fourier transform:

$$\tilde{\Psi}(p,t) = \mathcal{F}\{\Psi(x,t)\}. \tag{6}$$

This means that the action of the time evolution operator (4) on the state of the system can be written as follows:

$$\exp(-i(H_0+V(x))\Delta t/\hbar)\Psi(x,t) = \exp(-i\,V(x)\Delta t/\hbar)\mathcal{F}^{-1}\{\exp(-i\,p^2/(2m\hbar))\mathcal{F}\{\Psi(x,t)\}\} \tag{7}$$

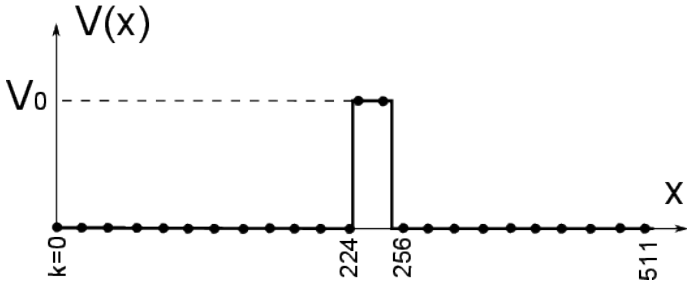

Figure 1. Rectangular potential. The numerical values of the x-axis denote the number of samples

3. Realisation of the quantum algorithm

An arbitrary state of the n-qubit register can be written as:

$$|\Psi\rangle = \sum_{k=0}^{2^n-1} c_k\,|k\rangle, \tag{8}$$

where c_k are complex coefficients, and $|k\rangle$ are base states of register which are tensor products of base states of individual qubits:

$$|k\rangle = |q_{n-1}\rangle \ldots |q_1\rangle|q_0\rangle, \tag{9}$$

where q_i take the values 0 or 1.

In order to encode the wave function of the Schrödinger particle $\Psi(x)$ in the quantum register we have to perform the discretisation. Then, we make the following assignment:

$$c_k = \Psi(x_k) = \Psi(\Delta x\, k) \tag{10}$$

where Δx is a parameter determining the spatial distance between adjacent samples of the wave function $\Psi(x)$. Parameter k=0, 1, ... $2^n - 1$ numbers the successive samples.

Simulation of time evolution of the particle is based on a rule set by Eq. (7) wherein the continuous Fourier transform $\mathcal{F}$ is replaced by a discrete quantum Fourier transform (QFT).

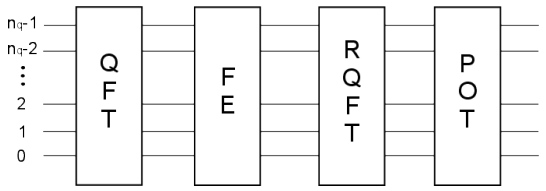

Figure 2. The scheme of a quantum algorithm simulating time evolution of the Schrödinger particle

Block diagram of the algorithm is shown in Fig. 2. Blocks QFT and RQFT implement respectively: Quantum Fourier Transform and Inverse Quantum Fourier Transform. Blocks FE and POT implement phase shift operations $\exp(-ip^2/(2m\hbar))$ and $\exp(-iV_0\Delta t/\hbar)$, respectively.

In our simulation we use the standard implementation of QFT and RQFT algorithms. Their brief description is given in Appendix A. Implementation of the free evolution block FE and potential block POT are shown sequentially in Figs. 3-6. In the case of FE block we use controlled phase shift gates R_f:

$$\begin{aligned}
|0\rangle|0\rangle &\rightarrow |0\rangle|0\rangle \\
|0\rangle|1\rangle &\rightarrow |0\rangle|1\rangle \\
|1\rangle|0\rangle &\rightarrow |1\rangle|0\rangle \\
|1\rangle|1\rangle &\rightarrow \exp(i\phi)\,|1\rangle|1\rangle.
\end{aligned} \tag{11}$$

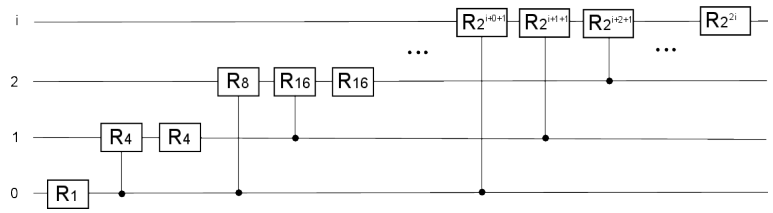

Figure 3. Implementation of FE block - free evolution for $p > 0$. Gates labeled by R_f are controlled phase shift gates with angle $\phi = -2\pi^2\hbar\Delta t/(mx_{max}^2)f$

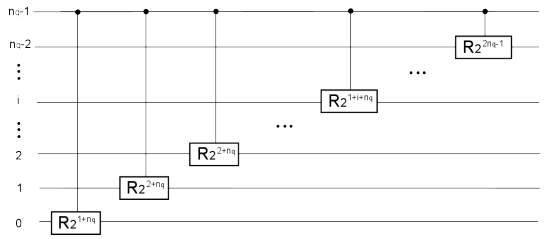

Figure 4. Implementation of the second part of FE block - free evolution for $p < 0$. Gates labeled by R_f are controlled phase shift gates with angle $\phi = 2\pi^2\hbar\Delta t/(mx_{max}^2)f$

In the case of the POT block, we additionally used NOT gates:

$$\begin{aligned} |0\rangle &\rightarrow |1\rangle \\ |1\rangle &\rightarrow |0\rangle, \end{aligned}$$

and controlled negation (CNOT) gates:

$$\begin{aligned} |0\rangle|0\rangle &\rightarrow |0\rangle|0\rangle \\ |0\rangle|1\rangle &\rightarrow |0\rangle|1\rangle \\ |1\rangle|0\rangle &\rightarrow |1\rangle|1\rangle \\ |1\rangle|1\rangle &\rightarrow |1\rangle|0\rangle. \end{aligned}$$

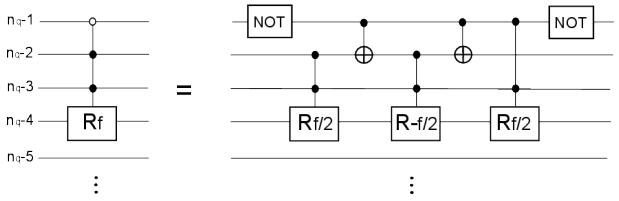

Figure 5. Implementation of POT block - implementation of rectangular potential shown in Fig. 1. The value of f parameter is given by $f = -\Delta t\, V_0/\hbar$

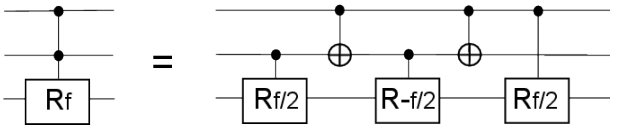

Figure 6. Implementation of the three-input controlled phase shift gate on two-input gates (from [14])

4. Results of the simulations. Comparison with classical algorithm

The algorithm described in previous section was used to simulate Schrödinger particle in the register of $n = 9$ qubits length (512 sample points). The initial state of the particle corresponds to a Gaussian wave packet in the form:

$$c_k = \Psi(x_k) = C_n \exp\Big(-\frac{(x_k - \langle x\rangle)^2}{4\,dx^2} + \frac{i\langle p\rangle x_k}{\hbar}\Big), \tag{12}$$

where C_n is a normalisation constant, and $x_k = k\,\Delta x$.

Position of the particle ranged from $x_{min} = 0$ to $x_{max} = 50$nm which gives $\Delta x = x_{max}/2^n = 98$pm. Initial parameters of the Gaussian packet (12) were as follows: $\langle x\rangle = 5$nm, $dx = 1.0$nm, $E_s = \langle p\rangle^2/(2m) = 1.0$eV where $m = 9.1 * 10^{-31}$kg is mass of electron. In all simulations time step value was equal to $\Delta t = 5 * 10^{-17}$s.

For comparison, classical simulation for exactly the same parameters has also been done. We used Cayley's method (described in Appendix B) for $\Delta t = 5{*}10^{-17}$s and for 512 sample points.

4.1. Simulation of the free particle

The results of the simulations for $V_0 = 0$ are shown in Fig. 7. Initial state corresponds to black curve. Results of quantum simulations (for times: $t_1 = 360\Delta t$, $t_2 = 720\Delta t$ and $t_3 = 1080\Delta t$) are given by blue curves. Results of Cayley's method (for the same values of t) are given by red curves.

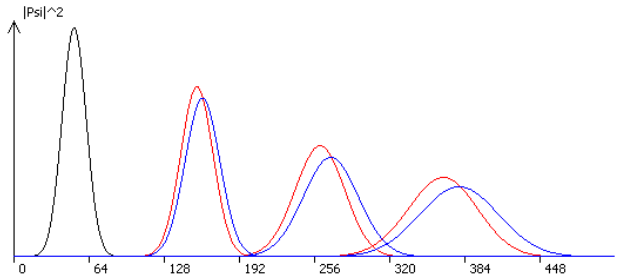

Figure 7. Free evolution of Gaussian wave packet in position representation

4.2. Simulation in the presence of a rectangular potential

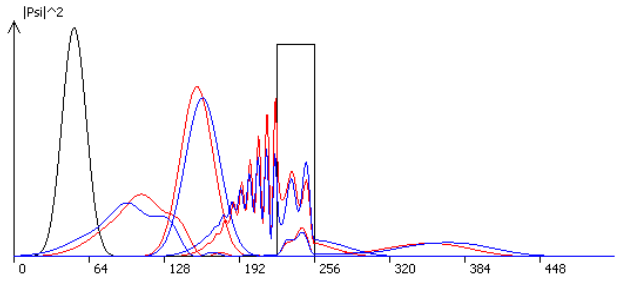

Figure 8. Evolution of Gaussian wave packet in position representation for rectangular potential

The results of simulations for $V_0 = 1.0$eV are shown in Fig. 8. Initial state corresponds to black curve. Results of quantum simulation (for times: $t_1 = 360\Delta t$,

$t_2 = 720\Delta t$ and $t_3 = 1080\Delta t$) are given by blue curves. Results of Cayley's simulation (for the same values of t) are given by red curves.

5. Conclusions and comments for the future

- We see the potential usefulness of the presented algorithm to simulate quantum systems. For one time step in the register of n qubits length ($N = 2^n$ spatial samples) we need only $3/2n^2 + 5/2n + 19$ quantum logic gates.[1] It gives time complexity equal to $O(\log^2 N)$. In comparison, the classical algorithms provide time complexity of a polynomial character.

- Both methods of simulation (quantum and Cayley's) give similar results. Some discrepancies are due to numerical errors of both methods. Examination the relationship between value of the time step Δt (or the density of spatial sampling Δx) and the accuracy of the results will be made in subsequent work.

- In our work we have conducted no analysis of the process of initial data entry in the register and outputting the final results from the register. It is well known that from a quantum register storing the 2^n states (data), only n states (data) can be read as a result of quantum measurement (no-cloning of the quantum state). Ability to read useful information from the final state of the register (as, for example, transition and reflection coefficients of the packet from threshold of the potential) will be examined in the next work.

- We do not review here factors affecting the accuracy of the results such as accuracy of quantum gates realization or noise in the register (decoherence). The analysis of these problems we will take in future research. In particular, it seems interesting to test the usefulness of Quantum Error Correcting Codes in quantum simulations.

Appendix A. Quantum Fourier Transform

In our simulation the standard implementation of QFT and RQFT algorithms has been used. Both of them are well known and widely presented in the literature

[1] We need $1/2(n^2 + n)$ gates for each of QFT, RQFT and FE (for $p > 0$). Additionally we need n gates for implementation of second part of FE block, and 19 gates for implementation of POT block.

(including textbooks such as [14, 15]). We present them merely as a reminder.

Implementation of the QFT and RQFT are shown in Fig. 9 and Fig. 10 respectively. Gates denoted by H are the Hadamard gates:

$$|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$|1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

and gates denoted by R_f are controlled phase shift gates (Eq. (11)) where $\phi = \pi/f$.

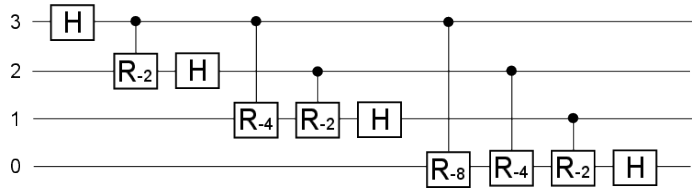

Figure 9. Block diagram of QFT

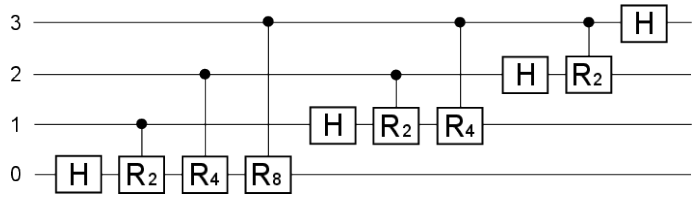

Figure 10. Block diagram of RQFT

Appendix B. Brief description of the classical algorithm used for comparison

As a comparative method, we used the classical Cayley's method [16]. It is based on Schrödinger equation of motion written in the following form:

$$(1 + 1/2i\,H\,dt/\hbar)\Psi(t + dt) = (1 - 1/2i\,H\,dt/\hbar)\Psi(t) \tag{13}$$

where H is the hamiltonian (2) with $p = -i\hbar\partial_x$ and second derivative is carried out by three-point approximation. In such a situation equation for one time step of simulation takes the form:

$$-iA\Psi_{n+1}(t_2) + C_n\Psi_n(t_2) - iA\Psi_{n-1}(t_2) = iA\Psi_{n+1}(t_1) + C_n^*\Psi_n(t_1) + iA\Psi_{n-1}(t_1) \tag{14}$$

where $t_2 = t_1 + dt$, $A = \hbar dt/(4m\Delta x^2)$, $C_n = 1 + i(2A + V_n dt/(2\hbar))$ while V_n are sampled values of the potential.

References

[1] Feynman, R., Internat. J. Theor. Phys., Vol. 21, 1982, pp. 467–488.

[2] Shor, P. W., Proc 35th Ann. Symp. Found. Comp. Sci., IEEE Comp.Soc. Pr., Vol. 124, 1994.

[3] Grover, L. K., *From Schrodinger equation to the quantum search algorithm*, Am. J. Phys., Vol. 69, 2001, pp. 769–777.

[4] De Raedt, K., Michielsen, K., De Raedt, H., Trieu, B., Arnold, G., Richter, M., Lippert, T., Watanabe, H., and Ito, N., *Massively Parallel Quantum Computer Simulator*, Computer Physics Communications, Vol. 176, 2007, pp. 121–136.

[5] Niwa, J., Matsumoto, K., and Imai, H., *General-Purpose Paralel Simulator for Quantum Computing*, Phys. Rev. A, Vol. 66(6), 2002, pp. 623171–62317111.

[6] Sawerwain, M. and Pilecki, J., *Parallel implementation of a quantum computing simulator*, Journal of Applied Computer Science, Vol. 14, No. 2, 2006.

[7] *NVIDIA CUDA Homepage*, http://developer.nvidia.com/object/cuda.html.

[8] Touchette, H. and Dumai, P., *QuCalc - The quantum computation package for Mathematica*, http://crypto.cs.mcgill.ca/QuCalc/, 2000.

[9] Miszczak, J. A., *Models of quantum computation and quantum programming languages*, Bulletin of The Polish Academy of Sciences, Technical Sciences, Vol. 59, No. 3, 2011.

[10] Wiesner, S., *Simulation of many-body quantum systems by a quantum computer*, http://xxx.lanl.gov/quant-ph/9603028.

[11] Zalka, C., *Efficient simulation of quantum system by quantum computers*, Fortschr. Phys., Vol. 46, 1998, pp. 877–879.

[12] Strini, G., *Error sensitivity of a quantum simulator I: a first example*, Fortschr. Phys., Vol. 50, 2002, pp. 171–183.

[13] Benenti, G. and Strini, G., *Quantum simulation of the single-particle Schrodinger equation*, http://xxx.lanl.gov/arXiv:0709.1704v2.

[14] Preskill, J., http://www.theory.caltech.edu/ preskill/ph229.

[15] Hirvensalo, M., *Quantum Computing*, Springer-Verlag Berlin Heidelberg, 2001 and 2003.

[16] Saleida, W., Tyc, M. H., and Just, M., *Algebraiczne metody rozwiazywania rownania Schrodingera*, PWN Warszawa, 2002.