

Simulation of the Schrödinger particle non-elastic scattering with emission of photon in the quantum register

M. OSTROWSKI*

Institute of Information Technology FTIMS, Technical University of Łódź, ul. Wólczańska 215, 90-924 Łódź, Poland

Abstract. This paper investigates whether a quantum computer can efficiently simulate the non-elastic scattering of the Schrödinger particle on a stationary excitable shield. The return of the shield to the ground state is caused by photon emission. An algorithm is presented for simulating the time evolution of such a process, implemented on standard two-input gates. The algorithm is used for the computation of elastic and non-elastic scattering probabilities. The results obtained by our algorithm are compared with those obtained using the standard Cayley's method.

Key words: quantum simulations, nonelastic scattering, Schrödinger particle.

1. Introduction

In the near future, quantum calculations are likely to make a major contribution to the development of informatics [1]. Although practical implementations of quantum computer have not been built yet, the existence of such seems to be possible. Therefore, it is worth examining the properties of such machines [2, 3].

Today we know Shor [4] and Grover [5] algorithms which are faster than their best classical counterparts. Another promising application of quantum computer are quantum simulations [6–8], i.e. the computer modeling of behavior of physical quantum systems. It gives the possibility of effective modeling quantum processes, which is not possible using classical computers [9]. Quantum computers can simulate a wide variety of quantum systems, including fermionic lattice models [10, 11], quantum chemistry [12, 13], and quantum field theories [14].

As is well known, simulations of quantum systems performed using conventional computers are not effective. This means that for classical computer the memory resources and time required to simulate grow exponentially with the size of quantum system. In the case of a quantum computer, the situation is different. The relationship between the size of quantum computer (register) and the size of the simulated quantum system is linear. Therefore, a very important task is to find the appropriate algorithms that can properly simulate complex quantum systems and non-trivial interactions between them. This is a difficult issue, because most of the interesting quantum systems is feasible in infinitely-dimensional Hilbert spaces. In such situations, we can use the technique of the wave function sampling and build an algorithm based on Quantum Fourier Transform. This case was tested in [15–18], which examined the free particle and the harmonic oscillator. The main limita-

tion of this coding method of the particle state is that it does not enable implementing an arbitrary potential $V(x)$. It allows only a few special cases such as the $V(x) \sim x^2$ potential.

In our earlier works we have shown that also rectangular potentials (like thresholds and wells) can be simulated using this method. This provides the ability to examine other interesting processes, such as the tunnel effect [19] and scattering of two Schrödinger particles to each other [20]. The present study also makes use of this kind of potential to simulate interaction between Schrödinger particle and shield.

In order to simulate the Schrödinger particle in the quantum register a different technique can also be used. For example, in [21, 22] method based on the quantum lattice-gas model was examined. It is worth mentioning about [23], where simulation of noisy Schrödinger equation is examined.

Another important issue is the simulation of quantum fields, which are the systems with an infinite number of degrees of freedom. In this case we can replace a continuous band of energy levels with its discrete counterpart. We use this method in the current publication and in [24] where we investigated the possibility of quantum simulation of the excited state decay with photon emission. Similar problems are also examined in the works of other authors. For example, processes such as: beta decay of helium atom [25] and decay of two-level atom in crystal [26] are tested.

In the present study we investigate whether it is possible to simulate in the quantum register a more complex process, which is a combination of processes presented in our previous works. It focuses on the simulation of non-elastic scattering of the Schrödinger particle on stationary, massive shield with photon emission. The process is shown in Fig. 1. The Schrödinger particle (subsystem A) in the form of the Gaussian packet collides with the massive and motionless shield (subsystem B), which goes to the excited state $(|1\rangle_B)$. Next, the excited system B^* returns to the ground state $(|0\rangle_B)$ with emission of photon (subsystem C).

The simulation conducted in this study aims at imitating common natural phenomena, e.g. excitation of gas atoms due

*e-mail: marcin.ostrowski@p.lodz.pl

Manuscript submitted 2020-03-06, revised 2020-06-02, initially accepted for publication 2020-06-30, published in October 2020

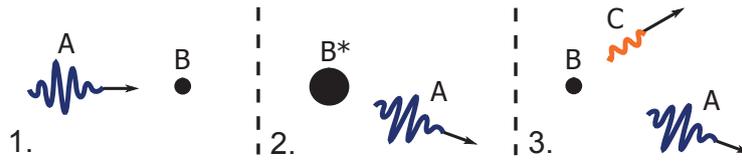


Fig. 1. Three stages of the nonelastic scattering process simulated in the quantum register

to nonelastic scattering of free electrons. Such processes take place, for example, in gas-discharge lamps. The toy model presented in this work was developed by the author. But the phenomenon simulated here has been known for a long time (e.g. Franck–Hertz experiment). Therefore, in literature it is described using many other approaches (e.g. S-matrix formalism).

Due to the specificity of quantum algorithms, a considerable simplification of the problem was required. However, an analysis of such a simplified model is still valuable, and allows us to draw conclusions regarding the possible application of quantum simulations to more real and complex processes.

In our previous work [20] we examined only the elastic collision of the Schrödinger particles. Here, for the first time, we simulate an inelastic scattering and return of the shield to the ground state by the photon emission. The algorithm simulating the deexcitation process has been developed by the author and was pre-tested in [24].

In order to simulate a quantum register, we used a simple environment written in C language for a single processor. However, there is possibility of using parallel computation methods for the simulation of a quantum computer [27, 28]. Moreover, some quantum algorithms can also be studied using neural networks and machine learning models [29].

2. Description of the simulated system

The simulation presented here required numerous simplifications:

- the motion of the Schrödinger particle (subsystem A) and photon propagation (subsystem C) are simulated only in one spatial dimension,
- we assume that the interaction between subsystems A and B is expressed as a rectangular potential (Fig. 2),
- we assume that $m_B \gg m_A$ and that position in the space of system B is well-defined. For this reason, system B is simu-

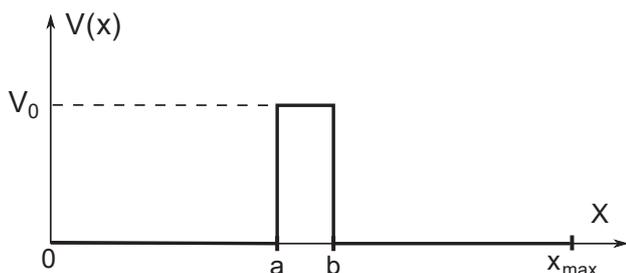


Fig. 2. Potential $V(x)$. In our numerical simulation we took: $a = 7/16 x_{max}$ and $b = 8/16 x_{max}$, where x_{max} is the length of the simulated area

lated only as a two base state system. State $|0\rangle_B$ is a ground state with energy $E_{B0} = 0$. State $|1\rangle_B$ is an excited state with energy $E_{B1} = \Delta E$.

- the process of returning system B to a ground state with photon emission is also simulated with simplifications. It has been examined and presented in [24]. We assume that system C has ground state $|0\rangle_C$ and n_c excited levels $|i\rangle_C$ which form a band. The energies of the levels are given by the formula:

$$E_{C_i} = \Delta E + \Delta \left(i - \frac{n_c + 1}{2} \right) \quad \text{for } i = 1, \dots, n_c, \quad (1)$$

where Δ is the gap between adjacent levels, and $\Delta E = E_{B1} - E_{B0}$.

The Hamiltonian of subsystems A and B are chosen in the following way:

$$\hat{H}_{AB} = \frac{\hat{p}_A^2}{2m_A} + \Delta E \hat{b}^\dagger \hat{b} + V(x) \otimes (\hat{I} + g_b^* \hat{b}^\dagger + g_b \hat{b}), \quad (2)$$

where $\hat{p}_A = -\frac{1}{2\pi} i \hbar \partial_x$, m_A is the mass of particle A, \hat{b} is a decreasing energy operator of subsystem B ($\hat{b}|1\rangle_B = |0\rangle_B$ and $\hat{b}^\dagger|0\rangle_B = |1\rangle_B$), g_b is a complex coupling constant between subsystems A and B.

The Hamiltonian of subsystem C and its interaction with subsystem B is chosen as follows:

$$\hat{H}_{BC} = \sum_{i=1}^{n_c} E_{C_i} \hat{c}_i^\dagger \hat{c}_i + \Delta^{\frac{1}{2}} \sum_{i=1}^{n_c} (\tilde{g} \hat{b}^\dagger \hat{c}_i + \tilde{g}^* \hat{b} \hat{c}_i^\dagger), \quad (3)$$

where \hat{c}_i is a decreasing energy operator of subsystem C ($\hat{c}_i|i\rangle_C = |0\rangle_C$ and $\hat{c}_i^\dagger|i\rangle_C = |i\rangle_C$), \tilde{g} is a coupling constant. The energies E_{C_i} are given by (Eq. 1).

The total Hamiltonian of the system is given by:

$$\hat{H}_{tot} = \hat{H}_{AB} + \hat{H}_{BC}. \quad (4)$$

3. The algorithm simulating time evolution of the system

The algorithm is implemented in a n_q -qubit register. In the first n_a qubits, the state of subsystem A is stored. In the middle qubit (denoted as B), the state of subsystem B is stored. The state of subsystem C is stored in the last n_c qubits. The above numbers fulfill the following relation: $n_a + n_c + 1 = n_q$.

In the present method, the wave function of particle A ($\psi_A(x)$) is sampled and encoded in the state of the n_a -qubit subregister in the following way:

$$\psi_k = \psi_A(k\Delta x), \quad (5)$$

where ψ_k is probability amplitude of k -th base state of the n_a -qubit subregister ($k = 0 \dots, 2^{n_a} - 1$), $\Delta x = x_{max}/2^{n_a}$ is sampling interval. Thus, sample ψ_0 corresponds to $\psi_A(0)$, while sample $\psi_{2^{n_a}-1}$ corresponds to $\psi_A(x_{max} - \Delta x)$.

The algorithm implementing the Hamiltonian given by Eq. (2) is shown in Fig. 3. The first three blocks (QFT, FE, QFT †) implement free evolution of particle A. We use a method based on the time evolution operator $\hat{U}(dt)$ approximation:

$$\begin{aligned} \hat{U}(dt)\psi_A(x, t) &= \exp(-i\hat{p}_A^2/(2m_A)2\pi dt/h)\psi_A(x, t) = \\ &= \mathcal{F}^{-1}\{\exp(-ip_A^2 dt/(m_A h/\pi))\mathcal{F}\{\psi_k(t)\}\}, \end{aligned} \quad (6)$$

where \mathcal{F} is the Quantum Fourier Transform and $\psi_k(t)$ is spatially sampled wave function of particle A. Blocks QFT and QFT † implement the Quantum Fourier Transform and its reverse. The FE block implements operator $\exp(-ip_A^2 dt/(m_A h/\pi))$. The above-mentioned blocks have been examined and described in several previous works, e.g. [20].

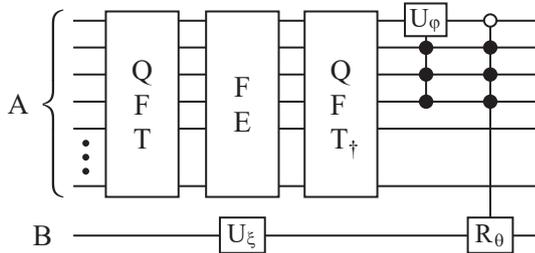


Fig. 3. The algorithm simulating subsystems A and B and their interaction

Gate U_ϕ is a controlled phase shift gate which operates according to the scheme: $|0, 1, 1, 1\rangle \rightarrow e^{i\phi}|0, 1, 1, 1\rangle$ where $\phi = 2\pi dt V_0/h$. It implements the potential $V(x)$ from Fig. 2. Gate U_ξ is a standard phase shift gate ($|1\rangle \rightarrow e^{-i\xi}|1\rangle$) where $\xi = 2\pi dt \Delta E/h$. It implements free evolution of subsystem B. Gate R_θ (controlled by four control qubits) implements the following operation:

$$|01110\rangle \rightarrow \cos\theta|01110\rangle + \sin\theta|01111\rangle, \quad (7)$$

$$|01111\rangle \rightarrow \cos\theta|01111\rangle + \sin\theta|01110\rangle, \quad (8)$$

where $\theta = 2\pi g_b V_0 dt/h$ and V_0 is the maximum value of the potential $V(x)$ (see Fig. 2). The number of control qubits in U_ϕ and R_θ gates is independent of n_a . The control qubits values

(0111 $_2 = 7$) correspond to the selection of the potential $V(x)$ area in $x \in (7/16 x_{max}, 8/16 x_{max})$ as shown in Fig. 2.

In the n_c -qubit subregister the state of photon (subsystem C) is stored. Each qubit stores information only about single energy level ($|0\rangle$ – free level, $|1\rangle$ – occupied level) from Eq. (1). This coding method increases the number of qubits necessary for the simulation, but it simplifies the implementation of the algorithm on the quantum gates.

The algorithm implementing the Hamiltonian given by Eq. (3) is shown in Fig. 4. Gates $U_{\phi C_i}$ are phase shift gates. Blocks R_η (shown in Fig. 5) implement the interaction between subsystem B and i -th level of subsystem C (components $\tilde{g}\hat{b}^\dagger\hat{c}_i + \tilde{g}^*\hat{b}\hat{c}_i^\dagger$ from Eq. (3)) for parameter $\eta = -2\pi\tilde{g}\Delta^{1/2}dt/h$.

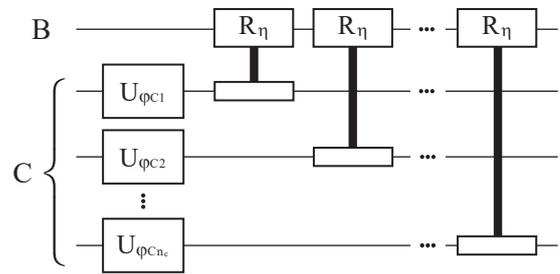


Fig. 4. The algorithm simulating free evolution of subsystem B and the interaction between subsystems B and C

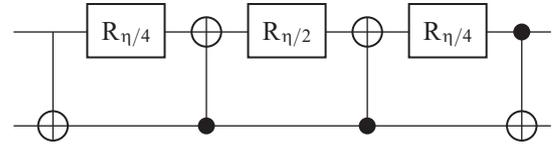


Fig. 5. The implementation of the interacting R_η module

4. The simulation results

This part of the paper examines the implementation of the algorithm using ideal quantum gates. As the initial state of system A we choose the Gaussian state with the following parameters: $m_A = 9.11 \cdot 10^{-31}$ kg (mass of particle A), $dX = 0.02 \cdot x_{max}$ (initial uncertainty of the position of the packet), $x_s = 0.15 \cdot x_{max}$ (center of the packet), $T = 8.0$ eV (kinetic energy of the packet center), where $x_{max} = 20$ nm is the length of the simulated area. As the initial states of subsystems B and C we take their ground states (i.e. $|0\rangle_B$ and $|0\rangle_C$). The time step for every simulation is equal to $dt = 2 \cdot 10^{-18}$ s.

Initially, we examine only the evolution of subsystem AB. We simulate a 10-qubit register (9 qubits for system A and 1 qubit for subsystem B). Examples of results are presented in Figs. 6–8 and the upper part of Fig. 10.

The probabilities of elastic/non-elastic reflection and transmission of particle A (shown in Fig. 7) are obtained by simultaneous measurement of the states of two qubits: the oldest qubit coding state of system A (which determines whether the

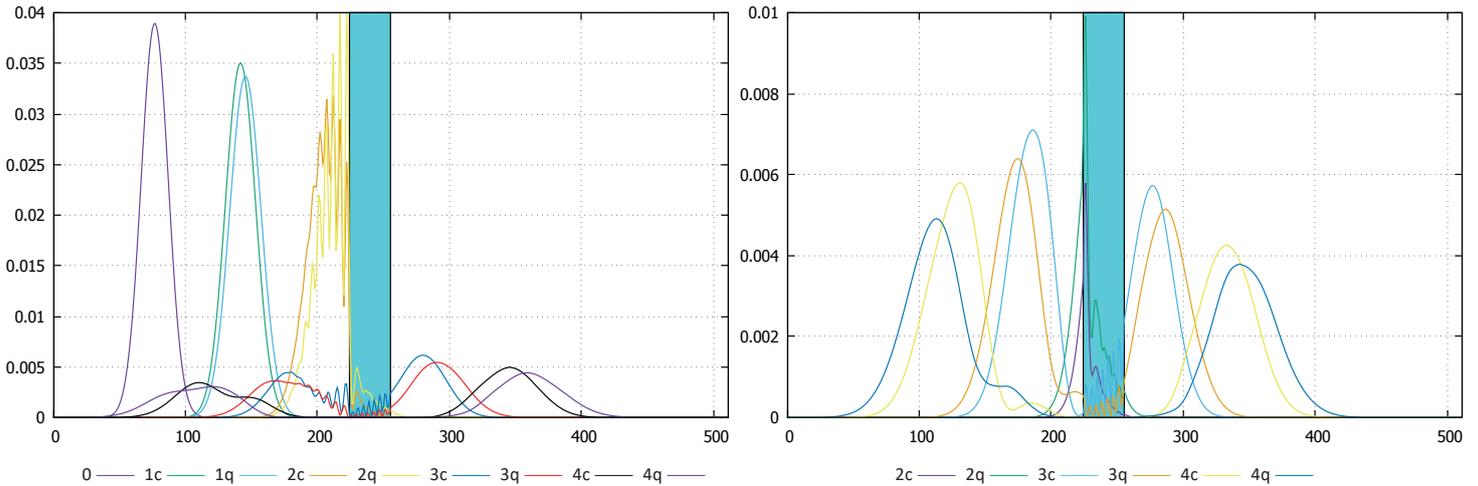


Fig. 6. Spatial probability distribution for particle A. The left plot shows part of the state which is entangled with state $|0\rangle_B$, whereas the right plot shows part of the state entangled with state $|1\rangle_B$. Four successive phases of motion (for $t = 1.6 \cdot 10^{-15}$ s (denoted by “1”), $3.2 \cdot 10^{-15}$ s (denoted by “2”), $4.8 \cdot 10^{-15}$ s (denoted by “3”) and $6.4 \cdot 10^{-15}$ s (denoted by “4”)) are shown. The initial state is denoted by “0”. Curves denoted by “q” are the results of quantum simulation, curves denoted by “c” are the results of a comparative Cayley’s simulation. The numbers on the horizontal axis correspond to the numbers of the spatial samples of the wave function $\psi_A(x)$ (sample 0 corresponds to $x = 0$, while sample 511 corresponds to $x = x_{max} - \Delta x$). The other parameters of the system are: $\Delta E = 2$ eV, $V_0 = 2$ eV and $g_b = -5$

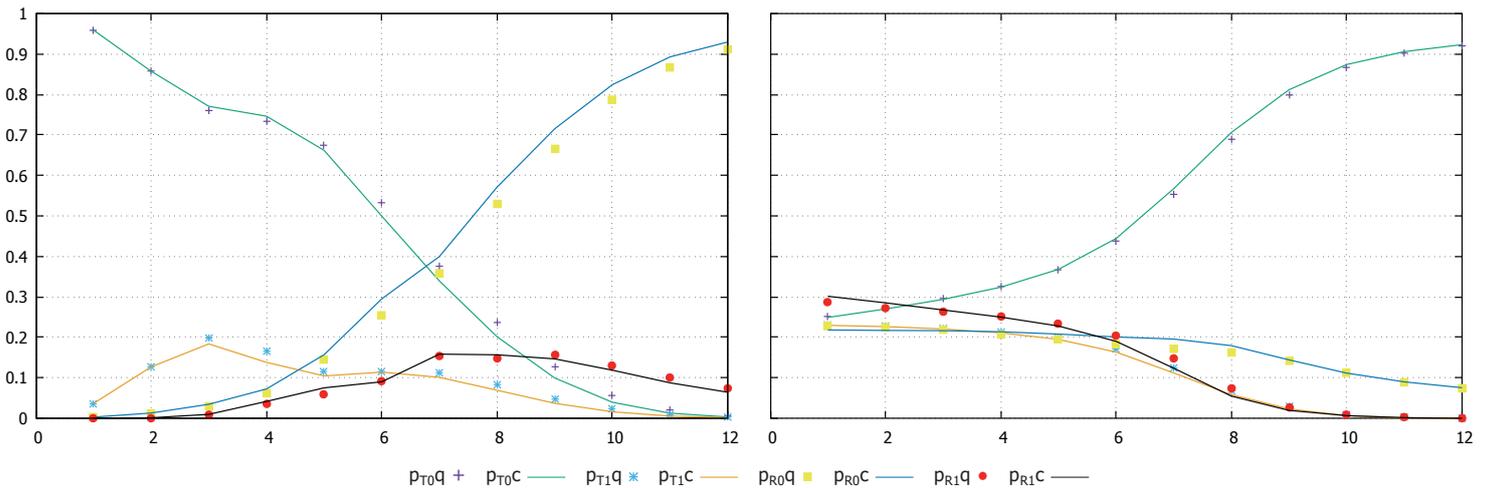


Fig. 7. Probabilities: p_{T0} (elastic transmission of particle A), p_{T1} (nonelastic transmission of particle A), p_{R0} (elastic reflection of particle A) and p_{R1} (nonelastic reflection of particle A). The left plot is made for $\Delta E = 2$ eV, $g_b = -0.2$ and $\Delta = 4 \cdot 10^{-20}$ J and the horizontal axis represents V_0 parameter (measured in eV). The right plot is made for $V_0 = 2$ eV, $g_b = -5$ and $\Delta = 4 \cdot 10^{-20}$ J and the horizontal axis represents excitation energy ΔE (measured in eV). Broken lines (denoted additionally by “c”) are the results of Cayley’s simulation, whereas dotted lines (denoted additionally by “q”) represent the results of quantum simulation

particle is on the left or on the right side of the shield) and the qubit coding the state of shield B.

In order to perform quantitative analysis of the results we have introduced ε parameter, defined as follows:

$$\varepsilon = \arccos\left(\left|\langle \psi_c | \psi_q \rangle\right|\right), \quad (9)$$

where $|\psi_q\rangle$ is the state of the system obtained by the quantum algorithm, while $|\psi_c\rangle$ has been obtained by the Cayley’s method. The values of ε parameter for the results from Fig. 8 are shown in Table 1.

Table 1

The values of the ε parameter for the cases from Fig. 8. The Roman numerals correspond to successive phases of motion shown in Fig. 8. The last row shows the case of $T = 8.0$ eV again for $n_q = 11$

| n_q | $T(\text{eV})$ | I | II | III | IV |
|-------|----------------|-------|-------|-------|-------|
| 10 | 2.0 | 0.038 | 0.136 | 0.594 | 0.671 |
| 10 | 4.0 | 0.074 | 0.142 | 0.554 | 0.802 |
| 10 | 8.0 | 0.185 | 0.469 | 0.974 | 1.094 |
| 11 | 8.0 | 0.056 | 0.465 | 0.812 | 0.844 |

Simulation of the Schrödinger particle non-elastic scattering with emission of photon in the quantum register

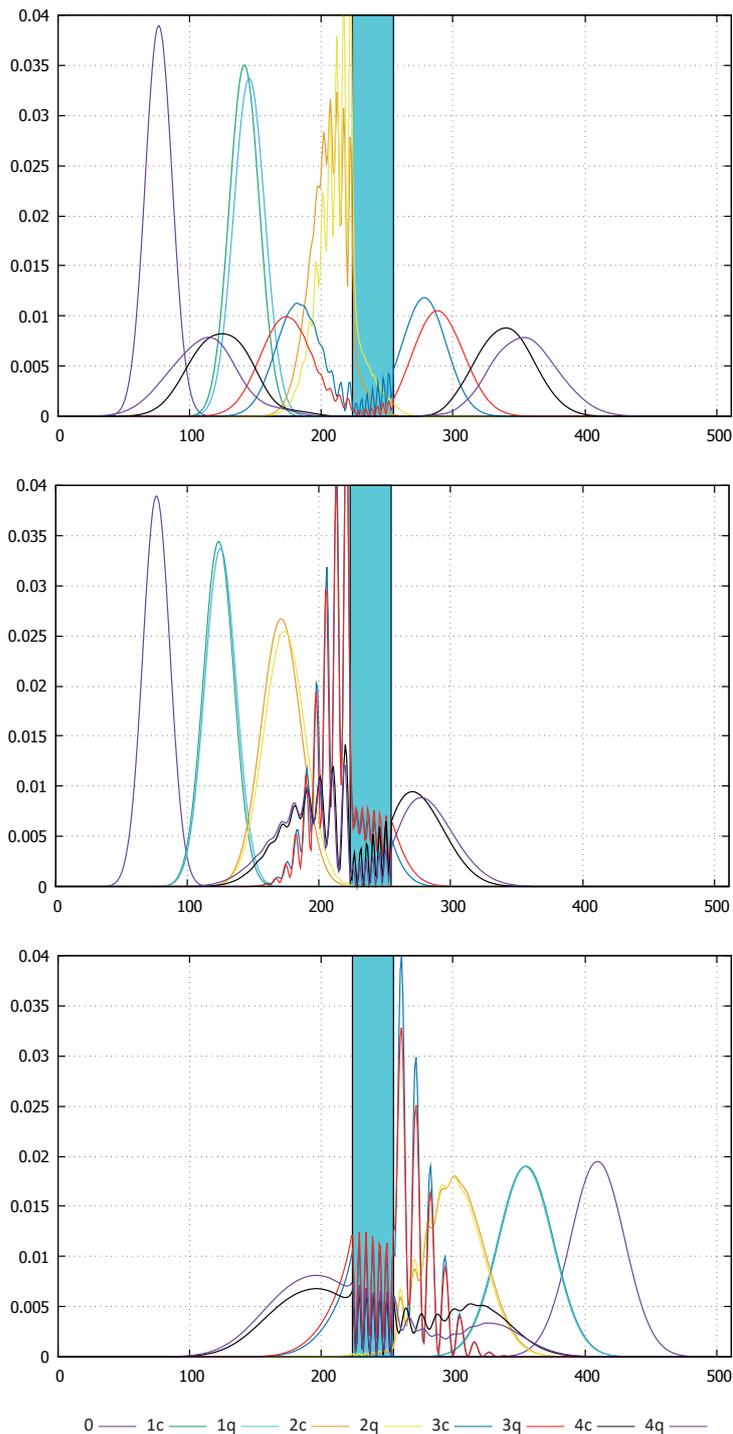


Fig. 8. Spatial probability distribution for particle A. The top plot is made for $T = 8$ eV, the middle for $T = 4$ eV and the bottom for $T = 2$ eV. In the bottom plot particle speed is negative, $dX = 0.04 \cdot x_{max}$ and $x_s = 0.8 \cdot x_{max}$. The other parameters are the same as in Fig. 6. Four successive phases of motion are shown (with time interval equal to $t = 1.6 \cdot 10^{-15}$ s (for the top and the middle plot) and $t = 2.56 \cdot 10^{-15}$ s (for the bottom plot)). The initial state is denoted by “0”. Curves denoted by “q” are the results of quantum simulation, curves denoted by “c” are the results of a comparative Cayley’s simulation. The numbers on the horizontal axis correspond to the numbers of the spatial samples of the wave function $\psi_A(x)$ (sample 0 corresponds to $x = 0$, while sample 511 corresponds to $x = x_{max} - \Delta x$

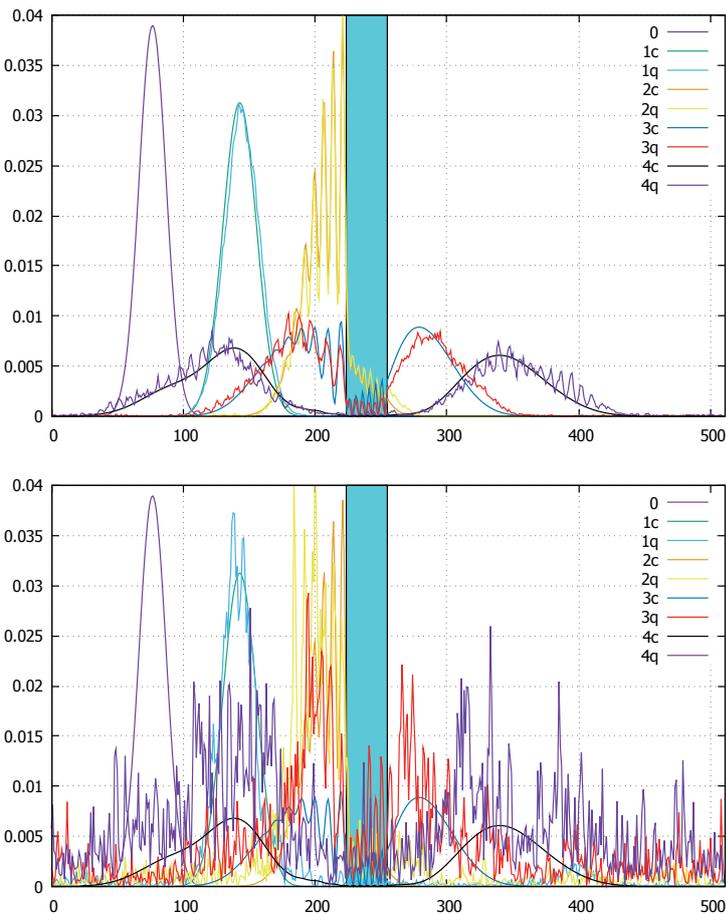


Fig. 9. The influence of quantum gates accuracy on the quality of the obtained results. Markings on both drawings and the values of all parameters are the same as in the middle plot from Fig. 8. The top plot is made for $\varepsilon_R = 10^{-3}$ and the bottom for $\varepsilon_R = 5 \cdot 10^{-3}$

Figure. 9 shows the influence of quantum gates accuracy on the quality of the obtained results. The continuous parameters of U_ϕ and R_θ gates are burdened with errors and drawn from $(\phi_{opt} - \varepsilon_R, \phi_{opt} + \varepsilon_R)$ and $(\theta_{opt} - \varepsilon_R, \theta_{opt} + \varepsilon_R)$ intervals, respectively, where ϕ_{opt} and θ_{opt} are optimal values and ε_R is an error level.

In the next part of the work, the full simulation of system ABC is performed on a 16-qubit register (9 qubits for system A, 1 for system B and 6 for system C). The results are presented in Fig. 10 (the bottom plot) and Fig. 11.

Energy values (T and ΔE) chosen for the simulation correspond to typical processes with visible light photon emission. The shield size $x_{max}/16 = 1.25$ nm is similar to the typical chemical molecule diameter. The total simulation time ($dt \cdot 3200$ (number of time steps)) allows the particle A move a distance approximately equal to x_{max} . Unfortunately, the value of the \tilde{g} parameter does not correspond to the values typical for atom deexcitation process. As is well known, the lifetime of such an excited state is in the order of 10^{-8} s. This is far too much in comparison to time of particle A simulation (the order of 10^{-14} s). Therefore, the time scale of deexcitation process has been reduced by several orders of magnitude.

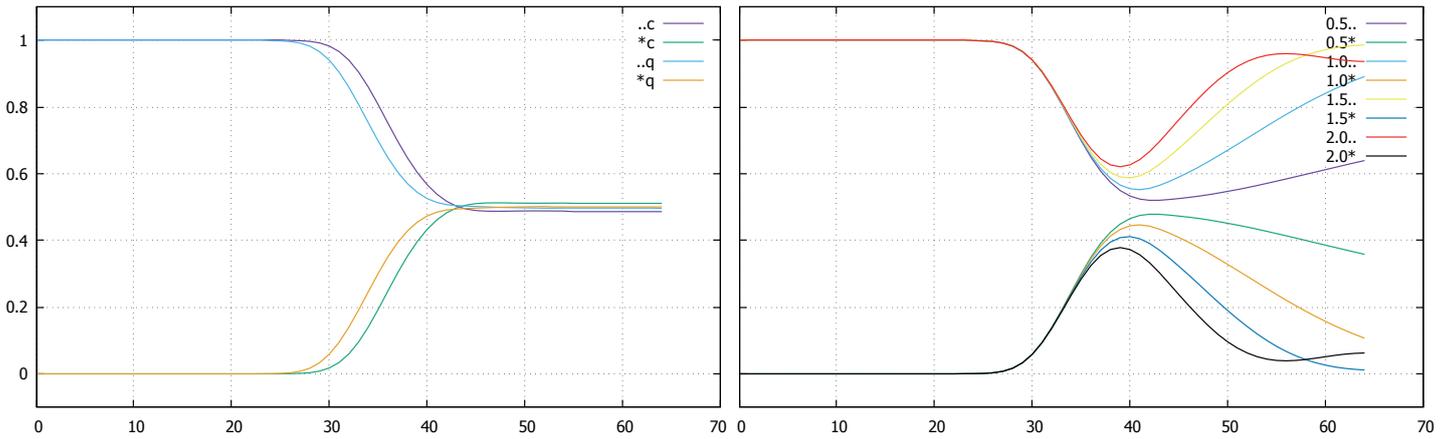


Fig. 10. The probability of occupying by shield B the ground state (denoted by “..”) and the excited state (denoted by “**”) as a function of time. The horizontal axis represents time in units equal to 10^{-16} s. The left plot is made for the simulation of subsystem AB (a 10-qubit register), the right plot is made for the full simulation of system ABC (a 16-qubit register). In the left plot curves denoted by “q” represent the results of the quantum simulation, whereas those denoted by “c” are the results obtained using Cayley’s simulation. In the right plot successive curves for $\tilde{g} = 0.5 \cdot 10^{-10} \text{ J}^{1/2}$ (denoted by “0.5”), $1.0 \cdot 10^{-10} \text{ J}^{1/2}$ (denoted by “1.0”), $1.5 \cdot 10^{-10} \text{ J}^{1/2}$ (denoted by “1.5”) and $2.0 \cdot 10^{-10} \text{ J}^{1/2}$ (denoted by “2.0”) are presented. The other parameters of the system: $\Delta E = 2 \text{ eV}$, $V_0 = 2 \text{ eV}$, $g_b = -5$ and $\Delta = 4 \cdot 10^{-20} \text{ J}$

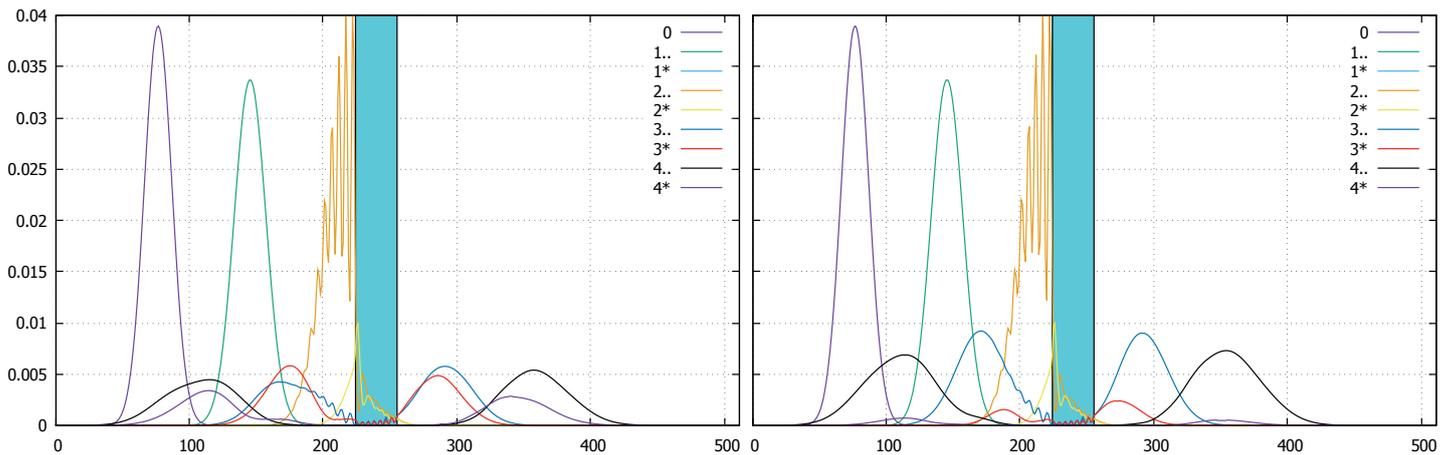


Fig. 11. Spatial probability distribution for particle A in four successive phases of motion (for $t = 1.6 \cdot 10^{-15} \text{ s}$ (denoted by “1”), $3.2 \cdot 10^{-15} \text{ s}$ (denoted by “2”), $4.8 \cdot 10^{-15} \text{ s}$ (denoted by “3”) and $6.4 \cdot 10^{-15} \text{ s}$ (denoted by “4”). Index 0 denotes an initial state of the particle. The curves denoted by “..” are the parts of states entangled with state $|0\rangle_B$, whereas those denoted by “*” are the parts of states entangled with state $|1\rangle_B$. The left plot is made for $\tilde{g} = 0.5 \cdot 10^{-10} \text{ J}^{1/2}$, whereas the right one is made for $\tilde{g} = 2.0 \cdot 10^{-10} \text{ J}^{1/2}$. The curve denoted by “1*” is not visible because it is equal to zero. The numbers on the horizontal axis correspond to the numbers of the spatial samples of the wave function $\psi_A(x)$ (sample 0 corresponds to $x = 0$, while sample 511 corresponds to $x = x_{max} - \Delta x$). The other parameters of the system: $\Delta E = 2 \text{ eV}$, $V_0 = 2 \text{ eV}$, $g_b = -5$ and $\Delta = 4 \cdot 10^{-20} \text{ J}$

5. Conclusions

- The evidence from this study suggests that even for $n_q = 16$ qubits it is possible to obtain satisfying results. However, the algorithm is scalable – an increase in the number of qubits means a higher sampling density of the wave function and, consequently, more accurate results. Moreover, standard quantum gates, like NOT-gate and controlled phase-shift gates with precision close to float number (mantissa 2^{-23} , which gives the accuracy of phase argument $\Delta\phi$ at the level of the order of 10^{-7}) are needed. We expect that such quantum computers will be built in the near future, perhaps with the help of semiconductor [30], quantum dots [31, 32] or spin systems [33] technology.
- Analyzing the values in Table 1, we can conclude that the results are worse for higher energies of particle A. In this case, higher values of the particle momentum must be sampled. Therefore, by increasing the number of qubits (the last row from Table 1), we obtain improvement of the accuracy.
- We should remember that the error value (Eq. 9) is also influenced by the state of the shield.
- On the bottom plot in Fig. 10 (for $\tilde{g} = 2.0 \cdot 10^{-10}$) we can see an unnatural rise of the curve. It is caused by an insufficient number of qubits used for the simulation of subsystem C.

- As can be observed from the comparison of plots in Fig. 11, the decay rate of state entangled with $|1\rangle_B$ increases as the value of parameter \tilde{g} (connected with the lifetime of the excited state of B) rises.
- As shown in Fig. 7, the divergence of results obtained by the quantum and Cayley's algorithms amounts to several percent.
- Figure 7b shows an obvious decay of p_{T1} and p_{R1} probabilities for $\Delta E > T$.
- In order to perform the algorithm on a quantum computer, initial Gaussian state of the particle A must be entered to the quantum register. In this work we do not present the appropriate algorithm. (In Listing 2 from Appendix B we present only a script preparing an initial state for the needs of a classic simulator). A simple algorithm for inputting these types of states into the quantum register has been proposed in [34]. Such problems have also been considered in other works, including both standard approaches [35] and those based on genetic algorithms [36].
- Analyzing the results from Fig. 9 we can conclude that gates accuracy should be better than $\varepsilon_R = 10^{-3}$.
- In this work we study only gates unitary errors (coherent errors). Errors caused by the decoherence are more important, but the density matrix formalism must be applied in this case. It causes significant increase in computational effort.
- The quantum algorithm presented here is a little faster than Cayley's method even when it is simulated using a classical computer.

6. Appendix A. Brief description of the classical algorithm used for comparison

As a comparative method (only for systems A and B), we used the classical Cayley's algorithm [37, 38]. It is based on the Schrödinger equation of motion written in the following form:

$$\left(1 + \frac{1}{2}iHdt/\hbar\right)\Psi(t + dt) = \left(1 - \frac{1}{2}iHdt/\hbar\right)\Psi(t) \quad (10)$$

where H is the Hamiltonian of the system given by Eq. 2). After sampling of the wave function of subsystem A and performing the second derivative by a three-point approximation we obtain:

$$\begin{aligned} \frac{1}{2}iHdt/\hbar \begin{pmatrix} \Psi_0^n \\ \Psi_1^n \end{pmatrix} &= -iA \begin{pmatrix} \Psi_0^{n+1} \\ \Psi_1^{n+1} \end{pmatrix} + 2iA \begin{pmatrix} \Psi_0^n \\ \Psi_1^n \end{pmatrix} - \\ &+ iA \begin{pmatrix} \Psi_0^{n-1} \\ \Psi_1^{n-1} \end{pmatrix} + iB \begin{pmatrix} 0 \\ \Psi_1^n \end{pmatrix} + iC(x) \begin{pmatrix} \Psi_0^n \\ \Psi_1^n \end{pmatrix} + \\ &+ iD^*(x) \begin{pmatrix} 0 \\ \Psi_0^n \end{pmatrix} + iD(x) \begin{pmatrix} \Psi_1^n \\ 0 \end{pmatrix}, \end{aligned} \quad (11)$$

where $A = \hbar dt / (8\pi m_A \Delta x^2)$, $B = \pi \Delta E dt / \hbar$, $C(x) = \pi V(x) dt / \hbar$ and $D(x) = \pi V(x) g_b dt / \hbar$. Ψ_0^n and Ψ_1^n are n -th space samples of the subsystem A wave functions entangled with $|0\rangle_B$ and $|1\rangle_B$ states, respectively.

7. Appendix B. The simulation code in pyQuil

All results presented in this work were obtained using the algorithm described in Section 3. Its pyQuil version is shown in the Listing 1. It was performed on a classic computer using the autor's simulator. Listing 2 presents Python script which calculates the initial state of the quantum register in the form of complex list ("Psi" variable).

An electronic version of this code can be obtained from: <https://zenodo.org/record/3872769#.XtYHpcDgqUk>

Listing 1
One step of the simulation

```
#phys const:
hbar = 1.0522e-34
me = 9.11e-31
eV = 1.6e-19

#simulation control:
nq = 14
nb = 2**nq
nc = 4
na = nq-nc-1
dt = 2e-18
xmax = 2.0e-8#20nm

#system parameters:
alfa = -2.0*math.pi*math.pi*dt*hbar/(xmax*xmax*me)
V0 = 2.0*eV
beta = -V0*dt/hbar
dE = 2.0*eV#energy of B excited state
xi = -dE*dt/hbar#free evol of B
gb = -5.0
theta = -gb*V0*dt/hbar
gtilda = 1e-10#decay const
delta = 4e-20
gg = math.sqrt(delta)*gtilda

#tables:
TabPhi=list()
for i in range(na):
    TabPhi.append(math.pi/(2**i))
TabPhi2=list()
for i in range(2*na):
    TabPhi2.append(alfa*(2**(i+1)))
TabPhi3=list()
for i in range(na):
    TabPhi3.append(alfa*(2**(2*i)))
EC1=list()
for i in range(nc):
    energia=dE+delta*(i+1-0.5*(nc+1))
    EC1.append(energia)
phiC1=list()
for i in range(nc):
    phi=-EC1[i]*dt/hbar
    phiC1.append(phi)
phiT=-gg*dt/hbar

p = Program()
#QFT algorithm:
ki=0
for i in range(nq-1,nq-na-1,-1):
    kj=ki
    for j in range(nq-1,i,-1):
        p.inst(CPHASE(-TabPhi[kj], i, j))
        kj-=1
    p.inst(H(i))
    ki+=1
```

```

#FE algorithm:
ki=0
for i in range(nq-1, nq-na-1, -1):
    kj=0
    for j in range(nq-1, i, -1):
        p.inst(CPHASE(TabPhi2[kj+ki], i, j))
        kj+=1
    p.inst(PHASE(TabPhi3[ki], i))
    ki+=1
ki=0
for i in range(nq-1, nq-na, -1):
    p.inst(CPHASE(-TabPhi2[ki+na], i, nq-na))
    ki+=1
#ROFT algorithm:
for i in range(nq-na, nq, 1):
    p.inst(H(i))
    kj=1
    for j in range(i+1, nq, 1):
        p.inst(CPHASE(TabPhi[kj], i, j))
        kj+=1
#POT algorithm:
p.inst(X(nq-1))
p.inst(PHASE(beta, nq-1).controlled([nq-2, nq-3, nq-4])
p.inst(RY(2*theta, nc).controlled([nq-1, nq-2, nq-3, nq-4])
p.inst(X(nq-1))
#subsystem B evol:
p.inst(PHASE(xi, nc))
#subsystem C evol:
for i in range(nc):
    p.inst(PHASE(phiC1[i], i))
    p.inst(CNOT(i, nc))
    p.inst(RY(phiT/2, nc))
    p.inst(CNOT(nc, i))
    p.inst(RY(phiT/1, nc))
    p.inst(CNOT(nc, i))
    p.inst(RY(phiT/2, nc))
    p.inst(CNOT(i, nc))
print(p)

```

Listing 2
The initial state preparation

```

import numpy as np
#phys const:
hbar = 1.0522e-34
me = 9.11e-31
eV = 1.6e-19

#symulation control:
nq=8#number of qubits
nc=0#system C qubits
na=nq-nc-1#system A qubits
nb=2**nq#base states
npp=2**na#number of spatial samples

#parameters of the initial Gaussian state:
xmax = 2.0e-8;#20nm
dX = 0.02*xmax#packet position uncertainty
Xs = 0.15*xmax;#center of the packet
T = 8.0*eV#kinetic energy of the packet
dE = 2.0*eV#excitation energy of B

PsiA=np.zeros(npp, dtype=complex)
delta = xmax/npp
Psh = np.sqrt(2.0*me*T)/hbar

def norma(reg):
    wynik=0
    for i in range(npp):
        wynik+=(reg[i].real)**2+(reg[i].imag)**2
    return np.sqrt(wynik)

```

```

for i in range(npp):
    xi=delta*i
    probka=-(xi-Xs)*(xi-Xs)/(4*dX*dX)+1*j*Psh*xi
    PsiA[i]=np.exp(probka)
PsiA/=norma(PsiA)

Psi=np.zeros(nb, dtype=complex)
for i in range(npp):
    Psi[i*(2**(nc+1))]=PsiA[i]

```

REFERENCES

- [1] R. Feynman, "Simulating physics with computers", *Int. J. Theor. Phys.* 21, 467–488 (1982).
- [2] S. Wegrzyn, J. Klamka, L. Znamirovski, R. Winiarczyk, and S. Nowak, "Nano and quantum systems of informatics", *Bull. Pol. Ac.: Tech.* 52(1), 1–10 (2004).
- [3] J.A. Miszczak, "Models of quantum computation and quantum programming languages", *Bull. Pol. Ac.: Tech.* 59(3), 305–324 (2011).
- [4] P.W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring", *Proceedings 35th Annual Symposium on Foundations of Computer Science*, Santa Fe, NM, USA, 1994, pp. 124–134.
- [5] L.K. Grover, "From Schrödinger equation to the quantum search algorithm", *Am. J. Phys.* 69, 769–777 (2001).
- [6] S. Lloyd "Universal Quantum Simulators", *Science* 273(5278), 1073–1078 (1996), doi:10.1126/science.273.5278.1073.
- [7] T. Schaeetz, C.R. Monroe, and T. Esslinger, "Focus on quantum simulation", *New J. Phys.* 15, 085009, (2013).
- [8] B.P. Lanyon, *et al.*, "Universal digital quantum simulation with trapped ions", *Science* 334(6052), 57–61 (2011). doi: 10.1126/science.1208001.
- [9] A.M. Childs, D. Maslov, Y. Nam, N.J. Ross, and Y. Su, "Toward the first quantum simulation with quantum speedup", *PNAS* 115(38), 9456–9461 (2018), doi:10.1073/pnas.1801723115.
- [10] D. Wecker, M.B. Hastings, N. Wiebe, B.K. Clark, C. Nayak, and M. Troyer, "Solving strongly correlated electron models on a quantum computer", *Phys Rev A* 92, 062318 (2015), doi:10.1103/PhysRevA.92.062318.
- [11] C. Kokail *et al.*, "Self-verifying variational quantum simulation of lattice models", *Nature* 569, 355–360 (2019), doi:10.1038/s41586-019-1177-4.
- [12] D. Wecker, B. Bauer, B.K. Clark, M.B. Hastings, and M. Troyer, "Gate count estimates for performing quantum chemistry on small quantum computers", *Phys Rev A* 90, 022305 (2014), doi:10.1103/PhysRevA.90.022305.
- [13] C. Hempel *et al.*, "Quantum Chemistry Calculations on a Trapped-Ion Quantum Simulator", *Phys. Rev. X* 8, 031022 (2018), doi:10.1103/PhysRevX.8.031022.
- [14] S.P. Jordan, K.S.M. Lee, and J. Preskill, "Quantum algorithms for quantum field theories", *Science* 336, 1130–1133 (2012), doi:10.1126/science.1217069.
- [15] S. Wiesner, "Simulation of many-body quantum systems by a quantum computer", arXiv:quant-ph/9603028.
- [16] C. Zalka, "Efficient simulation of quantum system by quantum computers", *Fortschr. Phys.* 46, 877–879 (1998).
- [17] G. Strini, "Error sensitivity of a quantum simulator I: a first example", *Fortschr. Phys.* 50, 171–183 (2002).
- [18] G. Benenti and G. Strini, "Quantum simulation of the singleparticle Schrödinger equation", *Am. J. Phys.* 76, 657–663 (2008), doi:10.1119/1.2894532.

- [19] M. Ostrowski, “Quantum simulation of the tunnel effect”, *Bull. Pol. Ac.: Tech.* 63(2), 379–383, (2015).
- [20] M. Ostrowski, “Quantum Simulation of Two Interacting Schrödinger Particles”, *Open Syst. Inf. Dyn.* 23(4), 1650020 (2016).
- [21] J. Yepez and B. Boghosian, “An efficient and accurate quantum lattice-gas model for the many-body Schrödinger wave equation”, *Comp. Phys. Commun.* 146, 280–294, (2002).
- [22] J. Yepez, G. Vahala, and L. Vahala, “Lattice quantum algorithm for the Schrödinger wave equation in 2+1 dimensions with a demonstration by modeling soliton instabilities”, *Quantum Inf. Process.* 4, 457–469 (2005), doi: 10.1007/s11128-005-0008-8.
- [23] N. Garg, H. Parthasarathy, and D.K. Upadhyay, “Real-time simulation of H-P noisy Schrödinger equation and Belavkin filter”, *Quantum Inf. Process.* 16, 121, (2017).
- [24] M. Ostrowski, “Simulation of the excited state decay in the quantum register”, *Prz. Elektrotechniczny* 10, 167–169, (2020).
- [25] S. Vaintraub *et al.*, “Simulations of β - decay of ${}^6\text{He}$ in an Electrostatic Ion Trap”, <https://arxiv.org/ftp/arxiv/papers/1402/1402.3978.pdf>
- [26] V. Buzek, G. Drobny, M.G. Kim, M. Havukainen, and P.L. Knight, “Numerical simulations of atomic decay in cavities and material media”, *Phys. Rev. A* 60(1), 582–592 (1999).
- [27] M. Sawerwain and J. Pilecki, “Parallel implementation of a quantum computing simulator”, *J. Appl. Comp. Sci.* 14(2), 79-90 (2006).
- [28] R. Nowotniak and J. Kucharski, “GPU-based tuning of quantum-inspired genetic algorithm for a combinatorial optimization problem”, *Bull. Pol. Ac.: Tech.* 60(2), 323–330 (2012).
- [29] W. Citko and W. Sieńko, “The Feasibility of Quantum Algorithms Using Machine Learning Models”, *Prz. Elektrotechniczny* 9/2019, 146(2019) [in Polish].
- [30] J.R. Tucker and T.C. Shen, “The road to a Silicon Quantum Computer”, *Quantum Inf. Process.* 3, 105–113 (2005), doi:10.1007/s11128-004-0417-0.
- [31] D. Loss and D.P. DiVincenzo, “Quantum computation with quantum dots”, *Phys. Rev. A.* 57(1), 120–126 (1998), doi:10.1103/PhysRevA.57.120.
- [32] J. Łuczak and B.R. Bułka, “Dynamics of a Charge Qubit Encoded in a Double Quantum Dot”, *Acta Phys. Polon. A* 133(3), 748–750 (2018), doi:10.12693/APhysPolA.133.748.
- [33] M. Twardy and D. Olszewski, “Realization of controlled NOT quantum gate via control of a two spin system”, *Bull. Pol. Ac.: Tech.* 61(2), 379–390 (2013).
- [34] M. Ostrowski, “Loading initial data into the quantum register”, *XV International Conference “System Modelling and Control”*, Łódź, Poland, 2013.
- [35] L. Grover and T. Rudolph, “Creating superpositions that correspond to the efficiently integrable probability distributions”, arXiv:quant-ph/0208112 (2002).
- [36] D. Ventura, “Learning quantum operators”, in *Proceedings of the International Conference on Computational Intelligence and Neuroscience*, 2000, pp. 750–752.
- [37] W. Salejda, M.H. Tyc, and M. Just, *Algebraiczne metody rozwiązywania równania Schrödingera*, PWN, Warszawa (2002) [in Polish].
- [38] F.L. Dubeube, “Solving the time-dependent Schrödinger equation with absorbing boundary conditions and sourceterms in Mathematica 6.0”, arXiv:1005.0044v3, (2010).