Ostrowski *EPJ Quantum Technology* (2024) 11:12
https://doi.org/10.1140/epjqt/s40507-024-00223-3

EPJ Quantum Technology
a SpringerOpen Journal

RESEARCH Open Access

Application of the QDST algorithm for the Schrödinger particle simulation in the infinite potential well

Marcin Ostrowski[1*]

*Correspondence:
marcin.ostrowski@p.lodz.pl
[1]Institute of Information Technology FTIMS, Technical University of Łódź, Wólczańska 215, Łódź, 90-924, Poland

Abstract

This paper examines whether a quantum computer can efficiently simulate the time evolution of the Schrödinger particle in a one-dimensional infinite potential well. In order to solve the Schrödinger equation in the quantum register, an algorithm based on the Quantum Discrete Sine Transform (QDST) is applied. The paper compares the results obtained in this way with the results given by the previous method (based on the QFT algorithm).

Keywords: Quantum Discrete Sine Transform; Quantum simulations; Schrödinger particle

1 Introduction

In the near future, quantum calculations can make a major contribution to the development of informatics [1]. Nowadays some institutions claim to have a quantum computer and offer its computing power. Therefore, it is worth examining the properties of such machines.

For many years we have known Shor [2] and Grover [3, 4] algorithms which are of lower computational complexity than their best classical counterparts. Another promising application of quantum computer are quantum simulations [5–8], 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 [10–12], Heisenberg chains [13], Lattice Boltzmann methods [14] quantum chemistry [15, 16], field theories [17, 18], gravitational waves [19], and paradoxes in quantum mechanics [20].

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 sampling the wave function and build an algorithm based on the Quantum Fourier Transform (QFT). This case was tested in [21–24], which examined the free particle and the harmonic oscillator.

In our previous works algorithms based on the QFT have been used. We have successfully simulated both the Schrödinger particle with a continuous spectrum [25], and a particle trapped in a potential well with finite length [26]. In this paper we examine a modification of that approach. We show that algorithm based on the Quantum Discrete Sine Transform (QDST) can be successfully used for simulation of the Schrödinger particle inside infinite potential well. We also demonstrate that this approach has significant advantages.

The physical system we simulate here is simple and is successfully simulated on classical computers. However, the presented algorithm can be easily generalized to the three-dimensional case. Moreover, it may be part of a larger simulation of multi-molecular processes. In such situations, the use of quantum computer offers significant advantages.

The possibility of using a quantum register to simulate the Schrödinger particle has already been discussed in the literature. For example, in [27] simulation of noisy Schrödniger equation is examined. It is also worth mentioning about [28, 29], where technique based on the quantum lattice-gas model was examined. Our approach is completely different. We examine the problem on purely algorithmic grounds, using abstract model of quantum system. We abstract completely from specified physical implementation of the quantum register.

As is well known, quantum transforms are widely used in quantum processing. We can mention here the Quantum Fourier Transform [30, 31] and the cosine transform [32], which is used in image compression issues [33, 34]. The wavelet [35, 36] (e.g. in image processing [37, 38]) and the Walsh-Hadamard [39] transforms are also considered.

In order to simulate a quantum register, we used an environment written in C++ language. We also performed several tests of the algorithm with the use of IBM hardware [40]. In both cases we obtained the same results. Python (Qiskit) code of the algorithm main blocks can be found in Appendix E.

2 Description of the simulated system

In this work we consider the one-dimensional Schrödinger particle in a infinite potential well, as shown in Fig. 1. Our main purpose is the simulation of time evolution of such system in the quantum register.

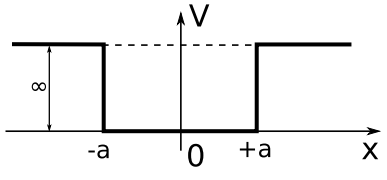

Figure 1 The infinite one-dimensional potential well used in the simulation

Time evolution of the Schrödinger particle in a one-dimensional case takes the form:

$$i\hbar\frac{d}{dt}\psi_S(x,t) = \big(H_0 + V(x)\big)\psi_S(x,t), \tag{1}$$

where $H_0 = p^2/(2m)$ is free Hamiltonian and $V(x)$ is the potential of the well from Fig. 1. The time evolution operator corresponding to Eq. (1) can be approximated as follows:

$$\begin{aligned} U(\Delta t)\psi_S(t) &= \exp\big(-i\big(H_0 + V(x)\big)\Delta t/\hbar\big)\psi_S(t) \\ &= \exp\big(-iV(x)\Delta t/\hbar\big)\mathcal{F}^{-1}\big\{\exp\big(-ip^2/(2m\hbar)\big)\mathcal{F}\big\{\psi_S(t)\big\}\big\}, \end{aligned} \tag{2}$$

where $\mathcal{F}$ is the Quantum Fourier Transform. This method has been tested in paper [26]. In this work we show that algorithm based on the Quantum Discrete Sine Transform (QDST) can be used as well. Moreover, there are advantages to this approach. Both the QFT and the QDST algorithms applied to the quantum particle state allow the transition from the position to the momentum representation (stationary states). However, the QDST transformation is dedicated to states that meet the Dirichlet boundary conditions. Thus, it maps much more accurately the state of the particle to stationary states in the well (Eq. (C.4)). The accuracy of the mapping is not the only advantage. The fulfillment of the condition $\psi_S(-a) = \psi_S(a) = 0$ means that the potential $V(x)$ from Fig. 1 is implemented automatically. The block implementing $\exp(-iV(x)\Delta t/\hbar)$ operator is not required. For this reason the algorithm can be run only in the energy domain and for time step of any value (i.e. the Trotter time step dt with error $O(\mathrm{d}t^2)$ [41] is not required). The algorithm based on the QFT (examined in an earlier work [26]) does not show the advantages described above.

3 Description of the algorithm

In the simulation a n_q-qubit register has been used. The state of the Schrödinger particle is encoded in all qubits exept the last one. Initial state of the last qubit is $|1\rangle$, as shown in Fig. 2. It is an auxiliary qubit for the QDST algorithm (see Appendix B). The state of the Schrödinger particle is sampled and encoded in the subregister state in the following way:

$$\psi_i = \psi_S(\Delta x \cdot i - a) \quad \text{for } i = 0, 1 \ldots 2^{n_x} - 1, \tag{3}$$

where $n_x = n_q - 1$, $\psi_S(x)$ is the Schrödinger particle wave function, $\Delta x = 2a/2^{n_x}$ is the distance between adjacent spatial samples. The ψ_i numbers are probability amplitudes of i-th subregister base state (i.e. $\psi_0 = {}_{n_x-1}\langle 0|\ldots{}_1\langle 0|_0\langle 0|\Psi\rangle$, $\psi_1 = {}_{n_x-1}\langle 0|\ldots{}_1\langle 0|_0\langle 1|\Psi\rangle$, $\psi_2 = {}_{n_x-1}\langle 0|\ldots{}_1\langle 1|_0\langle 0|\Psi\rangle$, etc., where $|\Psi\rangle$ is the subregister state). Thus, amplitude ψ_0 corresponds to $\psi_S(-a)$, while amplitude $\psi_{2^{n_x}-1}$ corresponds to $\psi_S(+a - \Delta x)$.

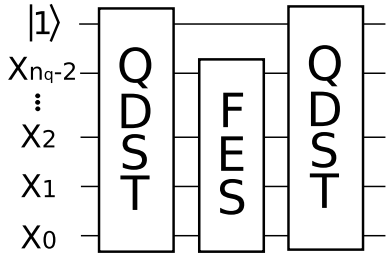

Figure 2 The scheme of the algorithm

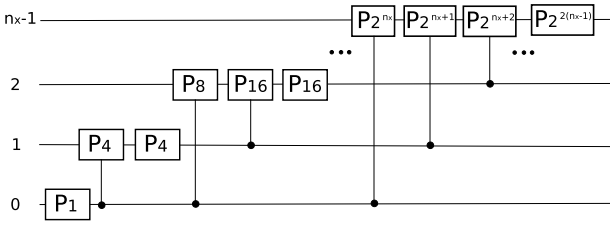

Figure 3 The scheme of the FES block implementing free evolution of the particle. Gates labeled by P_k are controlled phase shift gates with angle $\phi_k = -\pi^2 k\hbar t/(8ma^2)$

The scheme of the algorithm is shown in Fig. 2. The first QDST block implements the QDST algorithm which changes the state of the particle from the position to the energy representation (the stationary states given by Eq. (C.4)). A detailed description of the QDST algorithm implementation can be found in [32] (see also Appendix B). The FES block implements free evolution of the particle ($\hat{U}(t) = \exp(-it\hat{p}^2)/(2\hbar m)$ operator). Its implementation is shown in Fig. 3. Gates P_ϕ are phase shift gates ($|1\rangle \rightarrow e^{i\phi}|1\rangle$), and controlled phase shift gates ($|1,1\rangle \rightarrow e^{i\phi}|1,1\rangle$). Implementation of the FES block is based on decomposition of the $\hat{U}(t)$ operator in the following way:

$$\begin{aligned}\exp\left(-\frac{itp^2}{2\hbar m}\right) &= \exp\left(-i\alpha n^2\right)\\ &= \exp\left(-i\alpha \sum_{k_1=0}^{n_x-1}\sum_{k_2=0}^{n_x-1} 2^{k_1+k_2} i_{k_1} i_{k_2}\right)\\ &= \prod_{k=0}^{n_x-1} \exp\left(-i\alpha 2^{2k} i_k\right) \prod_{k_1=1}^{n_x-1}\prod_{k_2=0}^{k_1-1} \exp\left(-i\alpha 2^{k_1+k_2+1} i_{k_1} i_{k2}\right),\end{aligned} \tag{4}$$

where $\alpha = \pi^2\hbar t/(8ma^2)$, n is level number in the well (see Eq. (C.3)), and i_k are binary digits of n ($n = \sum_k 2^k i_k$). The first product from Eq. (4) corresponds to one-qubit gates from Fig. 3, while the second one (over k_1 and k_2) corresponds to two-qubits gates (k_1 is position of the gate, while k_2 is its control qubit number). The second QDST block implements the QDST^{-1} transformation which is identical with the QDST transformation.

4 The simulation results

The algorithm has been tested in the $n_q = 9$ qubit quantum register. We examined both the QDST algorithm from Fig. 2 and the algorithm based on the QFT transformation presented in [26]. In both cases we simulated a particle with the mass of electron in a well with length $2a = 4$ nm. In the case of the QFT algorithm time step is $dt = 10^{-18}s$. The results of simulation are shown in Figs. 4–7. In all plots only single curves are visible because the results of both algorithms coincide with each other.

Plots from Fig. 4 show example of the stationary state given by Eq. (C.4) for $n = 3$. In the next figures propagation of the localized Gaussian state (given by Eq. (D.1)) is tested. In the case from Fig. 5 initial packet parameters take the following values: $dx = 0.04a$, $\langle x\rangle = -0.6a$ and $T = 25$ eV. In the case from Fig. 6 packet with parameters: $dx = 0.04a$, $\langle x\rangle = +0.5a$ and $T = 150$ eV has been used. In all figures the left plots show the particle state in the position representation. Numbers on the horizontal axes correspond to

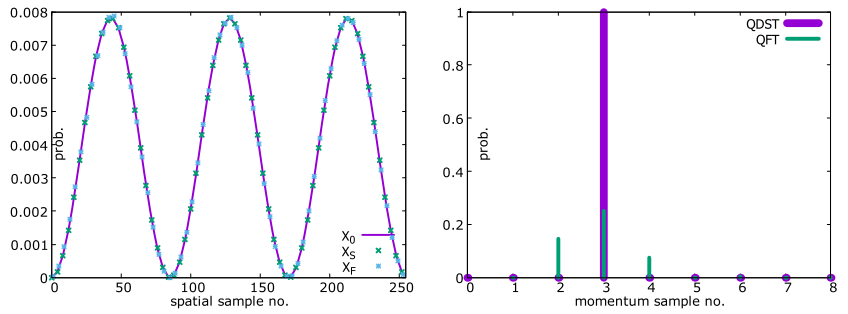

Figure 4 Simulation of the particle stationary state ($n = 3$) in the well. The X_0 curve is the initial state of the particle (for t = 0 s). Other curves are result of simulation for time $t = 4.5 \cdot 10^{-16}$ s. The X_S curve is result of the QDST simulation while the X_F curve is result of simulation based on the QFT algorithm. The right plot shows the momentum representation of the particle state (for the QFT method only positive values of momentum are shown)

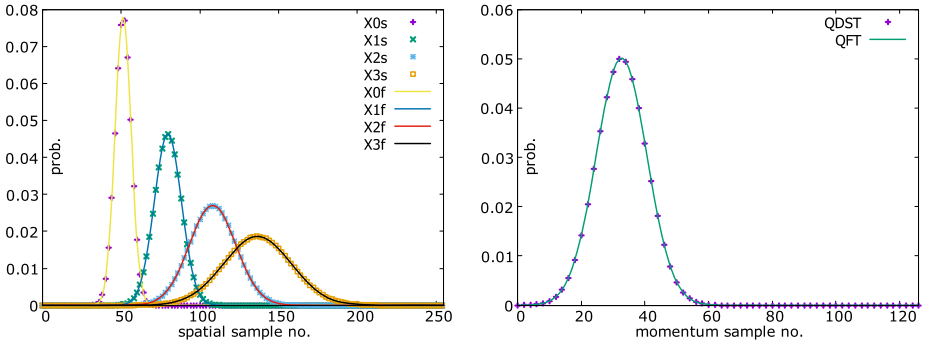

Figure 5 Simulation of the localized Gaussian state inside the well. On the left plot the initial state and three phases of motion are shown (with time step $1.5 \cdot 10^{-16}$ s). Dotted curves (denoted by "s") are results of the QDST simulation. Solid curves (denoted by "f") are results of simulation based on the QFT algorithm. The right plot shows the momentum representation of the particle state

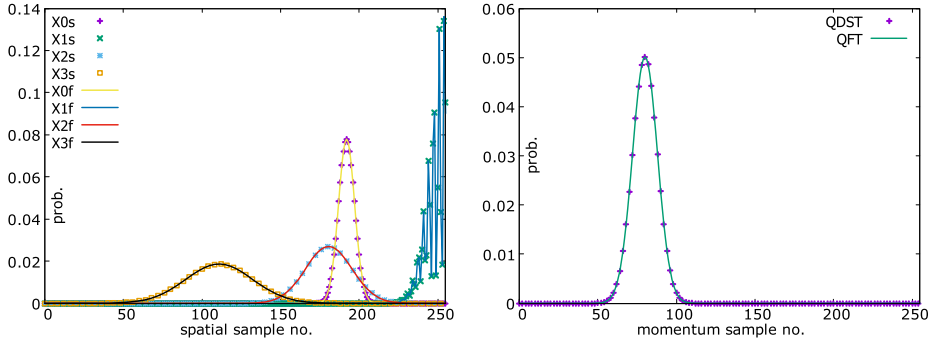

Figure 6 Simulation of the Gaussian state collision with edge of the well. On the left plot the initial state and three phases of motion are shown (with time step $1.5 \cdot 10^{-16}$ s). Dotted curves (denoted by "s") are results of the QDST simulation. Solid curves (denoted by "f") are results of simulation based on the QFT algorithm. The right plot shows the momentum representation of the particle state

spatial sample values (sample no. 0 corresponds to $x = -a$ while sample no. 256 corresponds to $x = +a$). The right plots show the particle state in the momentum representa-

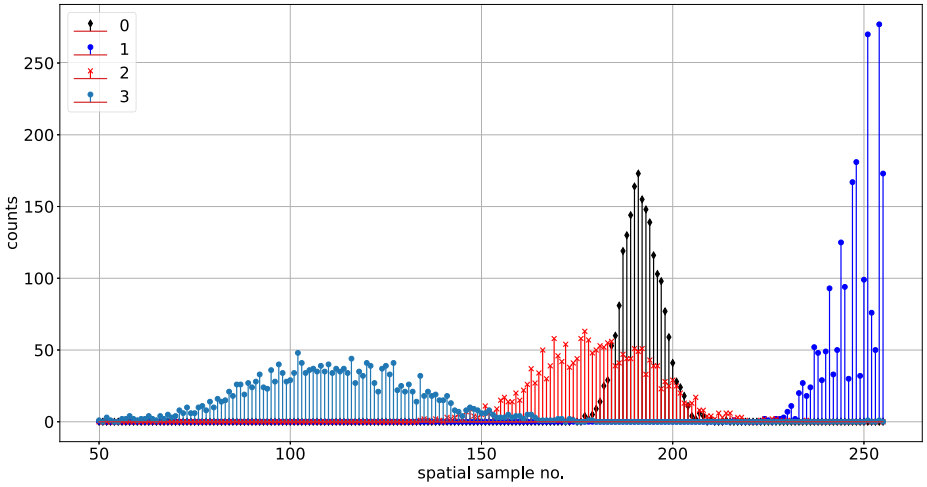

Figure 7 Simulation of the Gaussian state collision with edge of the well. The IBM hardware ('ibmq_qasm_simulator') has been used (for shots = 2048). The initial state and three phases of motion are shown (with time step $1.5 \cdot 10^{-16}$ s). Exactly the same case as in Fig. 6 is examined. Numbers on the horizontal axis correspond to the numbers of the spatial samples

tion. Numbers on the horizontal axes correspond to n (number of the stationary state in the well).

Plot from Fig. 7 shows results from the IBM simulator (ibmq_qasm_simulator) [40]. We examine exactly the same case as in Fig. 6. As we can see, the results from both simulations are similar.

5 Conclusions

- Both methods (the QDST and the QFT) give convergent results. The states in the position representation coincide with each other for all three cases from Figs. 4-6.
- The QFT method does not calculate exactly the number of the stationary level inside the well (the right plot from Fig. 4). This affects the accuracy of the simulation as shown in the left plot from Fig. 4.
- Preparation of the initial state was excluded from the discussion. A simple algorithm for inputting these types of states into the quantum register has been proposed in [42].
- Implementation of the QDST transform requires more gates than implementation of the QFT transform (according to [32]). Therefore, in the case of a free Schrödinger particle, a QFT-based simulation is faster. In the case of a particle trapped inside a well (regardless of its initial state), a simulation based on QDST is definitely better because it does not require the Trotter time step.
- In the QDST method the well potential $V(x)$ is implemented automatically. For this reason the algorithm can be run only in the momentum representation and the simulation can be performed in one time step. This is a particularly important advantage for simulations on currently available quantum computers, which are characterized by high error rates.

Appendix A: The Discrete Sine Transform (DST-1)

- formal definition of the DST-1 transform:

$$X_k = \sqrt{\frac{2}{N+1}} \sum_{n=0}^{N-1} x_n \sin\left(\frac{\pi}{N+1}(n+1)(k+1)\right) \quad \text{for } k = 0, \ldots, N-1 \tag{A.1}$$

- inverse transformation is given by the same formula:

$$x_n = \sqrt{\frac{2}{N+1}} \sum_{k=0}^{N-1} X_k \sin\left(\frac{\pi}{N+1}(n+1)(k+1)\right) \quad \text{for } n = 0, \ldots, N-1 \tag{A.2}$$

- inserting $X_k = (1, 0 \ldots 0)$ to Eq. (A.2) we obtain:

$$x_n = \sqrt{\frac{2}{N+1}} \sin\left(\frac{\pi}{N+1}(n+1)\right). \tag{A.3}$$

  Samples $x_{-1} = x_N = 0$. Therefore N is number of non-zero samples. Function period is equal to $N+1$.
- k-th sample X_k from Eq. (A.1) corresponds to the stationary state $\psi_n(x)$ from Eq. (C.4) for $k = n-1$.

Appendix B: The QDCT and the QDST algorithms

In our simulation the QDST1 algorithm has been used. The algorithm (shown in Fig. 2 from [32]) implements both the DCT-1 and the DST-1 transforms. In the first 2^{n_q-1} samples the DCT-1 transform of the input state is computed, while in the last 2^{n_q-1} samples the DST-1 transform is obtained (with multiplication by i factor). Hence, in our simulation the state of the last (auxiliary) qubit is set to $|1\rangle$.

Important note: in the DST-1 transform from Appendix A n-th sample codes ψ_{n+1} state. For the QDST algorithm n-th sample stores ψ_n state.

Appendix C: The Schrödinger particle in the infinite potential well

Let us consider the Schrödinger particle with the mass m inside the infinite potential well with a length $2a$ (as showni in Fig. 1). The stationary states of the particle can be predicted in the following form:

$$\psi(x) = A\exp(ikx) + B\exp(-ikx), \tag{C.1}$$

where $k = \sqrt{2Em}/\hbar$.

By imposing boundary conditions in the form $\psi(-a) = \psi(a) = 0$ we get:

$$\psi_n(x) = A\big(\exp\big(ik_n(x-a)\big) - \exp\big(-ik_n(x-a)\big)\big), \tag{C.2}$$

where

$$k_n = \frac{\pi n}{2a} \quad \text{for } n = 1, 2 \ldots. \tag{C.3}$$

The normalization condition allows to calculate $A = \frac{1}{2\sqrt{a}}$.

Finally, we obtain the stationary states in the well in the following form:

$$\psi_n(x) = \frac{1}{2\sqrt{a}}\left(e^{ik_n(x-a)} - e^{-ik_n(x-a)}\right) = \frac{i}{\sqrt{a}}\sin\left(k_n(x-a)\right). \tag{C.4}$$

Appendix D: The Gaussian state

The localized Gaussian state of the Schrödinger particle takes the form:

$$\psi_S(x) = C\exp\left(-\frac{(x-\langle x\rangle)^2}{4dx^2} + \frac{i\langle p\rangle x}{\hbar}\right), \tag{D.1}$$

where $\langle x\rangle$ is the expected value of the position, $\langle p\rangle$ is the expected value of the momentum, dx is standard deviation of the position while C is a normalization constant. In our work, instead of explicit value of $\langle p\rangle$, kinetic energy of the particle $T = \langle p\rangle^2/(2m)$ is used.

Appendix E: The simulation code in Qiskit

Listing 1 The FES block from Fig. 3

```
alfa = (np.pi**2)*hbar*dt/(8*m*a*a)
regX = QuantumRegister(size=nx, name='regX')
regP = QuantumRegister(size=1, name='regP')
circ = QuantumCircuit(regX, regP, name='Step')

# FES block:
for i in range(nx):
    for j in range(0,i):
        circ.cp(2**(i+j+1)*alfa, i, j)
    circ.p(2**(2*i)*alfa, i)
```

Listing 2 The QDST-1 block

```
listaC=[i for i in range(nq)]# control-qubits list
circ.u(np.pi/2, 0, -np.pi/2, regP) # B gate
circ.x(regX)
 # controlled B^+ gate:
circ.append(UGate(np.pi/2, -np.pi/2, np.pi).control(nx), listaC)
circ.x(regX)
circ.cx(regP,regX) # CNOT block
listaCC=[nx] # permutation block
for i in range(nx):
    listaCC.append(i)
for i in range(nx,0,-1):
    circ.append(XGate().control(i), listaCC)
    listaCC.pop(i)
circ.barrier() #QFT algorithm:
for i in range(nq-1,-1,-1):
```

```
    for j in range(nq-1,i,-1):
        circ.cp(np.pi/2**(j-i),i,j)
    circ.h(i)
    circ.barrier()
for i in range(nq//2):
    circ.swap(i, nx-i)
for i in range(nx): # permutation block^{-1}
    listaCC.append(i)
    circ.append(XGate().control(i+1), listaCC)
circ.cx(regP,regX) # CNOT block
circ.x(regX)
# controlled B gate:
circ.append(UGate(np.pi/2, 0, -np.pi/2).control(nx), listaC)
circ.x(regX)
circ.u(np.pi/2, -np.pi/2, np.pi, regP) # B gate
```

Funding
The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Abbreviations
DCT, Discrete Cosine Transform; DST, Discrete Sine Transform; FES, free evolution block; QDCT, Quantum Discrete Cosine Transform; QDST, Quantum Discrete Sine Transform; QFT, Quantum Fourier Transform.

Data availability
No datasets were generated or analysed during the current study.

Code availability
Not applicable.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Author contributions
This paper is written entirely by Marcin Ostrowski

Received: 30 January 2024 Accepted: 14 February 2024 Published online: 22 February 2024

References

1. Feynman R. Int J Theor Phys. 1982;21:467–88.
2. Shor PW. Algorithms for quantum computation: discrete logarithms and factoring. In: Proc 35th Ann. Symp. Found. Comp. Sci. Los Alamitos: IEEE Comput. Soc.; 1994. p. 124. https://doi.org/10.1109/SFCS.1994.365700.
3. Grover LK. From Schrödinger equation to the quantum search algorithm. Am J Phys. 2001;69:769–77. https://doi.org/10.1119/1.1359518.
4. Sakhouf H, Daoud M, Laamara RA. Implementation of Grover's search algorithm in the QED circuit for two superconducting qubits. Int J Theor Phys. 2020;59:3436–48. https://doi.org/10.1007/s10773-020-04602-1.
5. Lloyd S. Universal quantum simulators. Science. 1996;273(5278):1073–8. https://doi.org/10.1126/science.273.5278.1073.
6. Johnson TH, Clark SR, Jaksch D. What is a quantum simulator? EPJ Quantum Technol. 2014;1:10. https://doi.org/10.1140/epjqt10.
7. Schaetz T, Monroe CR, Esslinger T. Focus on quantum simulation. New J Phys. 2013;15:085009. https://doi.org/10.1088/1367-2630/15/8/085009.
8. Lanyon BP, Hempel C, Nigg D, Müller M, Gerritsma R, Zähringer F, Schindler P, Barreiro JT, Rambach M, Kirchmair G, Hennrich M, Zoller P, Blatt R, Roos CF. Universal digital quantum simulation with tapped ions. 2011. https://doi.org/10.1126/science.1208001. http://xxx.lanl.gov/arXiv:1109.1512v2.

9. Childs AM, Maslov D, Nam Y, Ross NJ, Su Y. Toward the first quantum simulation with quantum speedup. Proc Natl Acad Sci. 2018;115(38):9456–61. https://doi.org/10.1073/pnas.1801723115.
10. Wecker D, Hastings MB, Wiebe N, Clark BK, Nayak C, Troyer M. Solving strongly correlated electron models on a quantum computer. Phys Rev A. 2015;92:062318. https://doi.org/10.1103/PhysRevA.92.062318.
11. Kokail C, Maier C, van Bijnen R, Brydges T, Joshi MK, Jurcevic P, Muschik CA, Silvi P, Blatt R, Roos CF, Zoller P. Self-verifying variational quantum simulation of lattice models. Nature. 2019;569:16. https://doi.org/10.1038/s41586-019-1177-4.
12. Lamata L, Mezzacapo A, Casanova J et al. Efficient quantum simulation of fermionic and bosonic models in trapped ions. EPJ Quantum Technol. 2014;1:9. https://doi.org/10.1140/epjqt9.
13. Grass T, Lewenstein M. Trapped-ion quantum simulation of tunable-range Heisenberg chains. EPJ Quantum Technol. 2014;1:8. https://doi.org/10.1140/epjqt8.
14. Succi S, Fillion-Gourdeau F, Palpacelli S. Quantum lattice Boltzmann is a quantum walk. EPJ Quantum Technol. 2015;2:12. https://doi.org/10.1140/epjqt/s40507-015-0025-1.
15. Wecker D, Bauer B, Clark BK, Hastings MB, Troyer M. Gate count estimates for performing quantum chemistry on small quantum computers. Phys Rev A. 2014;90:022305. https://doi.org/10.1103/PhysRevA.90.022305.
16. Hempel C, Maier C, Romero J, McClean J, Monz T, Shen H, Jurcevic P, Lanyon BP, Love P, Babbush R, Guzik AA, Blatt R, Roos CF. Quantum chemistry calculations on a trapped-ion quantum simulator. Phys Rev X. 2018;8:031022. https://doi.org/10.1103/PhysRevX.8.031022.
17. Jordan SP, Lee KSM, Preskill J. Quantum algorithms for quantum field theories. Science. 2012;336:1130–3. https://doi.org/10.1126/science.1217069.
18. Sinha S, Russer P. Quantum computing algorithm for electromagnetic field simulation. Quantum Inf Process. 2010;9:385–404.
19. Bravo T, Sabín C, Fuentes I. Analog quantum simulation of gravitational waves in a Bose-Einstein condensate. EPJ Quantum Technol. 2015;2:3. https://doi.org/10.1140/epjqt16.
20. Hou T, Ding D, Wang C et al. Quantum simulation of generalized Hardy's paradox and corresponding Hardy's inequality via quantum programming. Int J Theor Phys. 2021;60:1972–9. https://doi.org/10.1007/s10773-021-04815-y.
21. Wiesner S. Simulation of many-body quantum systems by a quantum computer. http://xxx.lanl.gov/quant-ph/9603028.
22. Zalka C. Efficient simulation of quantum system by quantum computers. Fortschr Phys. 1998;46:877–9. https://doi.org/10.1002/(SICI)1521-3978(199811)46:6/8<877::AID-PROP877>3.0.CO;2-A.
23. Strini G. Error sensitivity of a quantum simulator I: a first example. Fortschr Phys. 2002;50:171–83. https://doi.org/10.1002/1521-3978(200203)50:2<171::AID-PROP171>3.0.CO;2-B.
24. Benenti G, Strini G. Quantum simulation of the single-particle Schrödinger equation. http://xxx.lanl.gov/arXiv:0709.1704v2.
25. Ostrowski M. Quantum simulation of two interacting Schrödinger particles. Open Syst Inf Dyn. 2016;23(4):1650020. https://doi.org/10.1142/S1230161216500207.
26. Ostrowski M. Simulation of diffusion of a single Schrödinger particle in the quantum register. Acta Phys Pol A. 2020;137(6):1182–6. https://doi.org/10.12693/APhysPolA.137.1182.
27. Garg N, Parthasarathy H, Upadhyay DK. Real-time simulation of H-P noisy Schrödinger equation and Belavkin filter. Quantum Inf Process. 2017;16:121.
28. Yepez J, Boghosian B. An efficient and accurate quantum lattice-gas model for the many-body Schrödinger wave equation. Comput Phys Commun. 2002;146:280–94. https://doi.org/10.1016/S0010-4655(02)00419-8.
29. Yepez J, Vahala G, Vahala L. Quantum Inf Process. 2006;4(6):457–69. https://doi.org/10.1007/s11128-005-0008-8.
30. Yin H, Lu D, Zhang R. Quantum windowed Fourier transform and its application to quantum signal processing. Int J Theor Phys. 2021;60:3896–918. https://doi.org/10.1007/s10773-021-04933-7.
31. Zhu H, Zhang Y, Li Z. Efficient quantum blind signature scheme based on quantum Fourier transform. Int J Theor Phys. 2021;60:2311–21. https://doi.org/10.1007/s10773-021-04854-5.
32. Klappenecker A, Rötteler M. Discrete cosine transforms on quantum computers. 2001. arXiv:quant-ph/0111038v1.
33. Pang CY, Zhou RG, Hu BQ, Hu WW, El-Rafei A. Signal and image compression using quantum dicrete cosine transform. Inf Sci. 2019;473:121–41.
34. Jiang N, Lu X, Hu H, Dang Y, Cai Y. A novel quantum image compression method based on JPEG. Int J Theor Phys. 2018;57:611–36.
35. Fijany A, Williams CP. Quantum wavelet transform: fast algorithms and complete circuits. 1998. arXiv:quant-ph/9809004.
36. Li HS, Fan P, Xia Hy, Song S. Quantum multi-level wavelet transforms. Inf Sci. 2019;504:113–35.
37. Wang J, Geng YC, Liu JQ. Adaptive quantum image encryption method based on wavelet transform. 2019. arXiv:1901.07762.
38. Chakraborty S, Shaikh SH, Chakrabarti A et al. An image denoising technique using quantum wavelet transform. Int J Theor Phys. 2020;59:3348–71. https://doi.org/10.1007/s10773-020-04590-2.
39. Hoyer P. Efficient quantum transforms. 1997. arXiv:quant-ph/9702028.
40. https://quantum-computing.ibm.com/.
41. Childs AM, Su Y, Tran MC, Wiebe N, Zhu S. Theory of Trotter error with commutator scaling. Phys Rev X. 2021;11:011020. https://doi.org/10.1103/PhysRevX.11.011020.
42. Ostrowski M. Loading initial data into the quantum register. J Appl Comput Sci. 2020;28(1):39–49.

Publisher's Note