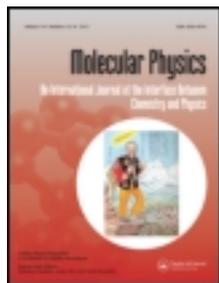


This article was downloaded by: [Purdue University]

On: 11 December 2013, At: 11:43

Publisher: Taylor & Francis

Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



Molecular Physics: An International Journal at the Interface Between Chemistry and Physics

Publication details, including instructions for authors and subscription information:

<http://www.tandfonline.com/loi/tmph20>

Quantum circuit design for solving linear systems of equations

Yudong Cao ^a, Anmer Daskin ^b, Steven Frankel ^a & Sabre Kais ^c

^a Department of Mechanical Engineering, Purdue University

^b Department of Computer Science, Purdue University

^c Department of Chemistry, Physics and Birck Nanotechnology Center, Purdue University, West Lafayette, IN 47907, USA

Accepted author version posted online: 21 Feb 2012. Published online: 15 May 2012.

To cite this article: Yudong Cao, Anmer Daskin, Steven Frankel & Sabre Kais (2012) Quantum circuit design for solving linear systems of equations, *Molecular Physics: An International Journal at the Interface Between Chemistry and Physics*, 110:15-16, 1675-1680, DOI: [10.1080/00268976.2012.668289](http://dx.doi.org/10.1080/00268976.2012.668289)

To link to this article: <http://dx.doi.org/10.1080/00268976.2012.668289>

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the "Content") contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms & Conditions of access and use can be found at <http://www.tandfonline.com/page/terms-and-conditions>

INVITED ARTICLE

Quantum circuit design for solving linear systems of equations

Yudong Cao^a, Anmer Daskin^b, Steven Frankel^a and Sabre Kais^{c*}

^aDepartment of Mechanical Engineering, Purdue University; ^bDepartment of Computer Science, Purdue University;
^cDepartment of Chemistry, Physics and Birck Nanotechnology Center, Purdue University, West Lafayette, IN 47907, USA

(Received 9 January 2012; final version received 14 February 2012)

Recently, it has been demonstrated that quantum computers can be used for solving linear systems of algebraic equations with exponential speedup compared with classical computers. Here, we present an efficient and generic quantum circuit design for implementing the algorithm for solving linear systems. In particular, we show the detailed construction of a quantum circuit which solves a 4×4 linear system with seven qubits. It consists of only the basic quantum gates that can be realized with present physical devices, implying great possibility for experimental implementation. Furthermore, the performance of the circuit is numerically simulated and its ability to solve the intended linear system is verified.

Keywords: quantum algorithm; quantum computing; linear systems

Quantum computers are devices that take direct advantage of quantum mechanical phenomena such as superposition and entanglement to perform computations [1]. Because they compute in ways that classical computers cannot, for certain problems quantum algorithms provide exponential speedups over their classical counterparts. For example, in solving problems related to factoring large numbers [2] and simulation of quantum systems [3–17], quantum algorithms are able to find the answer exponentially faster than classical algorithms. Recently, Harrow *et al.* [18] proposed a quantum algorithm for solving linear systems of equations with exponential speedup over the best known classical algorithms.

With the theoretical potential of the algorithm, an equally important issue is to render the algorithm viable for experimental implementation. Therefore, in this work we present a generic quantum circuit design based on the theoretical framework outlined by the algorithm. Particularly, the original work [18] did not elaborate on the circuit design for implementing λ_j^{-1} -controlled rotation when only λ_j is available. Hence in this work we present a scalable design of the quantum circuit for finding the reciprocals of the eigenvalues λ_j and stored them in a quantum register with a superposition of states that encode the values of λ_j^{-1} in binary format. The runtime scaling of this inverting circuit is $O(\log N)$, which retains the advantageous $O(\log N)$ runtime for the overall algorithm. For the purpose of showing a concrete numerical

example of how this scheme works, we choose a specific linear system (described by a matrix A of dimension 4×4) as the numerical example of the algorithm. Since our circuit involves only seven qubits and is composed of only basic quantum gates, our work bridges the theoretical development of the algorithm with the possibility of physical implementation by experimentalists.

The algorithm [18] solves the problem $Ax = b$ where A , a Hermitian d -sparse $N \times N$ matrix, and b , a unit vector, are given. The major steps of the algorithm can be summarized as the following: (1) Represent the vector b as a quantum state $|b\rangle = \sum_{i=1}^N b_i |i\rangle$ stored in a quantum register (termed B), where $|i\rangle$ represents a basis state of the register B where the states of the qubits represent the binary coding of i . In a separate quantum register (termed C) of t qubits, initialize the qubits by transforming the register to state $\sum_{\tau} |\tau\rangle$ from $|0\rangle$. Here $|\tau\rangle$ represents a basis state of the register C where the states of the qubits represent the binary coding of τ . (2) Apply the conditional Hamiltonian evolution $\sum_{\tau=0}^{T-1} |\tau\rangle\langle\tau|^C \otimes \exp[iA\tau t_0/T]$. (3) Apply the quantum inverse Fourier transform to the register C . Denote the basis states after quantum Fourier transform as $|k\rangle$. At this stage, the amplitudes of the basis states are concentrated on k values that satisfy $\lambda_k \approx 2\pi k/t_0$, where λ_k is the k th eigenvalue of the matrix A . (4) Add an ancilla qubit and apply conditional rotation on it, controlled by the register C with $|k\rangle \approx |\lambda_k\rangle$. The rotation transforms

*Corresponding author. Email: kais@purdue.edu

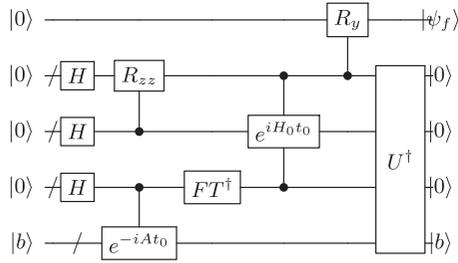


Figure 1. Overall circuit design. The registers from top to bottom are respectively the ancillary qubit, register L , M , C and B . The initial state is denoted as $|0000b\rangle$. $|\psi_f\rangle = [1 - (C^2/\lambda_j^2)]^{1/2}|0\rangle + (C/\lambda_j)|1\rangle$. U^\dagger represents uncomputation.

the qubit to $[1 - (C^2/\lambda_j^2)]^{1/2}|0\rangle + (C/\lambda_j)|1\rangle$. This is a key step of the algorithm and it involves finding the reciprocal of the eigenvalue λ_j quantum mechanically, which is not a trivial task on its own. We will elaborate on how to achieve this later in this work. (5) Uncompute the registers b and C . (6) Measure the ancilla bit. If it returns 1, the register b of the system is in the state $\sum_{j=1}^n \beta_j \lambda_j^{-1} |u_j\rangle$ up to a normalization factor, which is equal to the solution $|x\rangle$ of the linear system $Ax = b$. Here $|u_j\rangle$ represents the j th eigenvector of the matrix A and let $|b\rangle = \sum_{i=1}^n \beta_i |u_i\rangle$.

Based on the theoretical framework of the algorithm outlined above, here we present a generic quantum circuit design that implements the algorithm to find the solution $|x\rangle$ to the linear equation $A|x\rangle = |b\rangle$. The circuit contains registers B , C , M , L and an ancillary qubit (Figure 1). The detailed matrix forms of the quantum gates are presented in Appendix 1. Here the time parameter t_0 for all Hamiltonian simulations is defined to be 2π . Register B is used to store the value of $|b\rangle$. Register C , M and L contain t , m and l qubits respectively. the Walsh–Hadamard transform on Register C , the controlled Hamiltonian simulation and the inverse Fourier transform ideally gives a state $\sum_j \beta_j |\lambda_j\rangle |u_j\rangle$ [18] in Register B and C . At the same time, Walsh–Hadamard transform and controlled global phase shift R_{zz} is applied to register L and M (Figure 1) and the state of the two registers becomes $\sum_p \sum_s \exp[i(p/2^m)t_0] |s\rangle |p\rangle$, where $|p\rangle$ and $|s\rangle$ represent the basis states of the register M and L respectively. Details of this part of the circuit is shown in Figure 2. After the inverse Fourier transform is executed on register C , we use its $|\lambda_j\rangle$ states stored in register C as a control for a Hamiltonian simulation $\exp(iH_0 t_0)$ that is applied on register M (details in Figure 3). Here H_0 is defined as a diagonal matrix whose diagonal elements are $(1, 2, \dots, 2^{m-1})$. Due to the unitary nature of $\exp(iH_0 t_0)$, the operation $\exp(iH_0 t_0)$ can be readily decomposed into a quantum

circuit [19,20] that consists of only basic quantum gates.

We further establish the control relationship between Register L and the $\exp(iH_0 t_0)$ by using the the k th qubit (Figure 3) of the register L to control a $\exp[-ip((\lambda_j/2^m)(1/2^{l-k})t_0)]$ simulation that acts on register M . The values of binary numbers stored in register L are then able to determine the time parameter t in the overall Hamiltonian simulation $\exp(-iH_0 t)$. Following such construction, after the Hamiltonian simulation $\exp(-iH_0 t_0)$ the state of the system is

$$|0\rangle \otimes \sum_{j=1}^n \sum_{p=0}^{2^m-1} \sum_{s=0}^{2^l-1} \beta_j \exp\left[i \frac{p}{2^{m+l}} t_0 (2^l - \lambda_j s)\right] |s\rangle |p\rangle |\lambda_j\rangle |u_j\rangle.$$

With $t_0 = 2\pi$, the states $|s\rangle$ is concentrated on values of $s = 2^l/\lambda_j$ in the above expression (denote such $|s\rangle$ states as $|2^l/\lambda_j\rangle$). This is because of the fact that sums of the form $\sum_{k=0}^{N-1} \exp[2\pi i k(r/N)]$ vanish unless $r = 0 \pmod N$. Denote the components of the superposition states in register L as $|2^l/\lambda_j\rangle$. Rotate the ancilla bit with the angle shift controlled by the $|2^l/\lambda_j\rangle$ states stored in register L :

$$R_y(2^l/\lambda_j)|0\rangle \approx \left(1 - \frac{C^2}{\lambda_j^2}\right)^{1/2} |0\rangle + \frac{C}{\lambda_j} |1\rangle$$

with C being a constant. Uncomputing the three registers (Figure 1), we are left with a state in register B with the ancilla qubit proportional to

$$\sum_{j=1}^n \left[(1 - C^2/\lambda_j^2)^{1/2} |0\rangle + C/\lambda_j |1\rangle \right] \otimes |000\rangle \otimes \beta_j |u_j\rangle.$$

Measuring the ancilla qubit and if we obtain $|1\rangle$, the state of the system will collapse to $|1000\rangle \otimes \sum_{j=1}^n \beta_j \lambda_j^{-1} |u_j\rangle$, where $|u_j\rangle$ the first register represents the solution. If the ancilla bit measures $|0\rangle$, the system is back to its original state $|0000b\rangle$. If the ancilla bit measures $|1\rangle$, the state stored in the register B is proportional to the solution of the linear system of equations $A|x\rangle = |b\rangle$ and the register B is no longer entangled with any other registers in the system. Therefore we have obtained the solution of the linear system of equations in the form of a quantum superposition state stored in the register B .

There are mainly two factors that influence the number of qubits needed in register L and M . The first one is the desired bit precision for encoding both λ_j and $1/\lambda_j$. Clearly if one desires to encode a number with precision $\epsilon = O(2^q)$, at least q qubits are needed for the register, which is true for any generic binary registers. In addition to this factor, for the particular roles that the register L and M play in the circuit, the condition

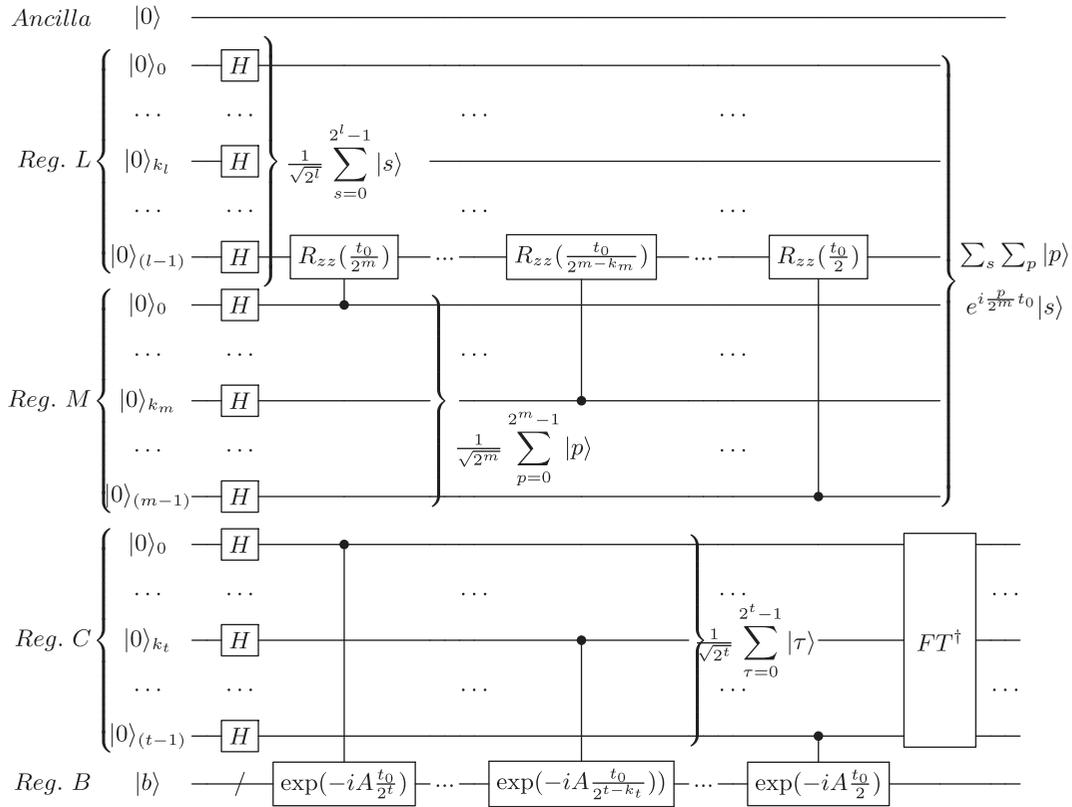


Figure 2. Detailed quantum circuit of the generic circuit implementation. This figure shows the Walsh–Hadamard transform, $|\tau\rangle$ -controlled Hamiltonian simulation $\exp(iAt)$ and Fourier transform on the register C .

number κ of A and the condition number of the matrix A is also important. In order for the inversion circuit to work, 2^m must be larger than any eigenvalue of A , i.e. $m \geq \lceil \log \lambda_{\max} \rceil$. On the other hand in order to encode all values of $1/\lambda_j$ register L must have enough qubits such that even the largest one $1/\lambda_{\min}$ can be encoded, i.e. $l \geq \lceil \log \lambda_{\min} \rceil$. Therefore $m - l \approx \log \kappa$ is a general constraint that m and l must comply with.

A detailed runtime analysis in [18] shows that the runtime scales as $O(\log(N)d^2)$ where d is the sparsity of A . The circuit that we present here adds to the algorithm the additional complexity of controlled Hamiltonian simulation $\exp(iH_0 t_0)$. However, it is shown in [21] that $\exp(iAt)$ can be simulated in time $O(\log(N)d^2 t)$. Therefore the Hamiltonian simulations in this circuit design only add at most a constant overhead to the cost scaling of the algorithm.

With the generic circuit model outlined above, now we present a seven-qubit circuit for solving a system with A of dimension 4×4 as a numerical example. For the numerical example we specifically choose the form of matrix A such that it allows for a very simple *ad hoc* design of the circuit for finding the reciprocal modulo

2^l (in the seven-qubit case $2^l = 16$). For this example we choose

$$A = \frac{1}{4} \begin{pmatrix} 15 & 9 & 5 & -3 \\ 9 & 15 & 3 & -5 \\ 5 & 3 & 15 & -9 \\ -3 & -5 & -9 & 15 \end{pmatrix}. \quad (1)$$

Hence A is a Hermitian matrix with the eigenvalues $\lambda_j = 2^{i-1}$ and corresponding eigenvector $|u_i\rangle = \frac{1}{2} \sum_{j=1}^4 (-1)^{\delta_{ij}} |j\rangle^C$, where $|j\rangle^C$ represents the state of register C which encodes the number j in binary form, δ_{ij} is the Kronecker delta, and the index i runs from 1 to 4. Furthermore, we let $\mathbf{b} = \frac{1}{2}(1, 1, 1, 1)^T$. Therefore, $|b\rangle = \sum_{j=1}^4 \beta_j |u_j\rangle$ and each $\beta_j = \frac{1}{2}$. To compute the reciprocals of the eigenvalues, a quantum *swap* gate is used (Figure 4) to exchange the values of the first and third qubit. Therefore there is no need for specific auxiliary registers M and L (Figure 1) in this example. The eigenvalues are inverted to their reciprocals with the *swap* gate in the following fashion. For example, the eigenvalue $\lambda_4 = 8$ is encoded as $|1000\rangle$ in register C , after applying the *swap* gates it is transformed to $|0010\rangle$, which is equal to $\frac{1}{8} \times 2^4 = 2$.

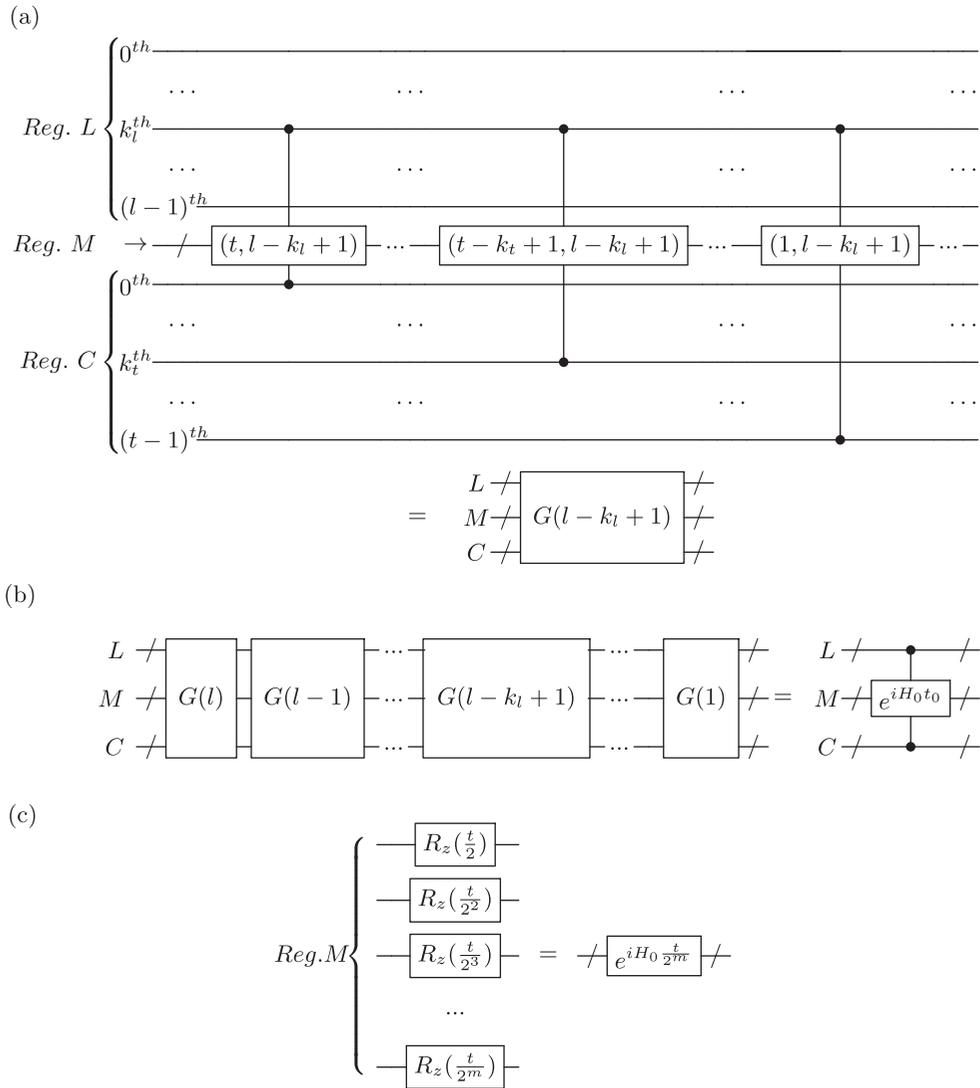


Figure 3. Hamiltonian Simulation $\exp(iH_0 t)$ which is controlled by register L and C . (a) A subroutine of the Hamiltonian simulation of the matrix H_0 . Every gate labelled (u, v) in this diagram represents a quantum gate $\exp(-iH_0 t_0 / 2^{u+v})$. We denote this subroutine as $G(l - k_l + 1)$. (b) The overall circuit as a cascade of the subroutines $G(l - k_l + 1)$. (c) Implementation of $\exp[iH_0(t/2^m)]$ in register M . The (u, v) gate can be realized by letting $t = t_0 / 2^{u+v-m}$.

Figure 4 shows the circuit. We use the Group Leader Optimization Algorithm to find the circuit decomposition of the Hamiltonian simulation operator $\exp[iA(2\pi/16)]$ (see [19,20] for details, also see [1,22] for general methods). Here we let $t_0 = 2\pi$. The resulting quantum circuit for $\exp[iA(2\pi/16)]$ is shown in Figure 4. With the decomposition of $\exp[iA(2\pi/16)]$ readily available, the operators $\exp[iA(2\pi/8)]$, $\exp[iA(2\pi/4)]$ and $\exp[iA(2\pi/2)]$ can be implemented by simply multiplying the angle shifts in all the rotation gates in the circuit for $\exp[iA(2\pi/16)]$ by a factor of 2, 4, and 8 respectively [19].

The final state of system, conditioned on obtaining $|1\rangle$ in the ancilla bit, is $(1/85^{1/2})(8|u_1\rangle + 4|u_2\rangle +$

$2|u_3\rangle + |u_4\rangle) \otimes |0000\rangle \otimes |1\rangle$. Written in the standard bases, it becomes $(1/340^{1/2})(-|00\rangle + 7|01\rangle + 11|10\rangle + 13|11\rangle)$, which is proportional to the exact solution of the system $\mathbf{x} = (1/32)(-1, 7, 11, 13)^T$ (Figure 5). The quantum circuits (Figure 4) outlined above are then simulated and the statistics of final states are computed from the simulations (Figure 6).

The exponent r in the rotation gates $R(n\pi/2^{r-1})$ (Figure 4) is an important parameter that determines the accuracy of the final state of the system, which contains the solution to the linear system of equations. As can be seen in Figure 6, when the value of r is sufficiently large, say larger than 6, the probabilities of measuring each basis state $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$

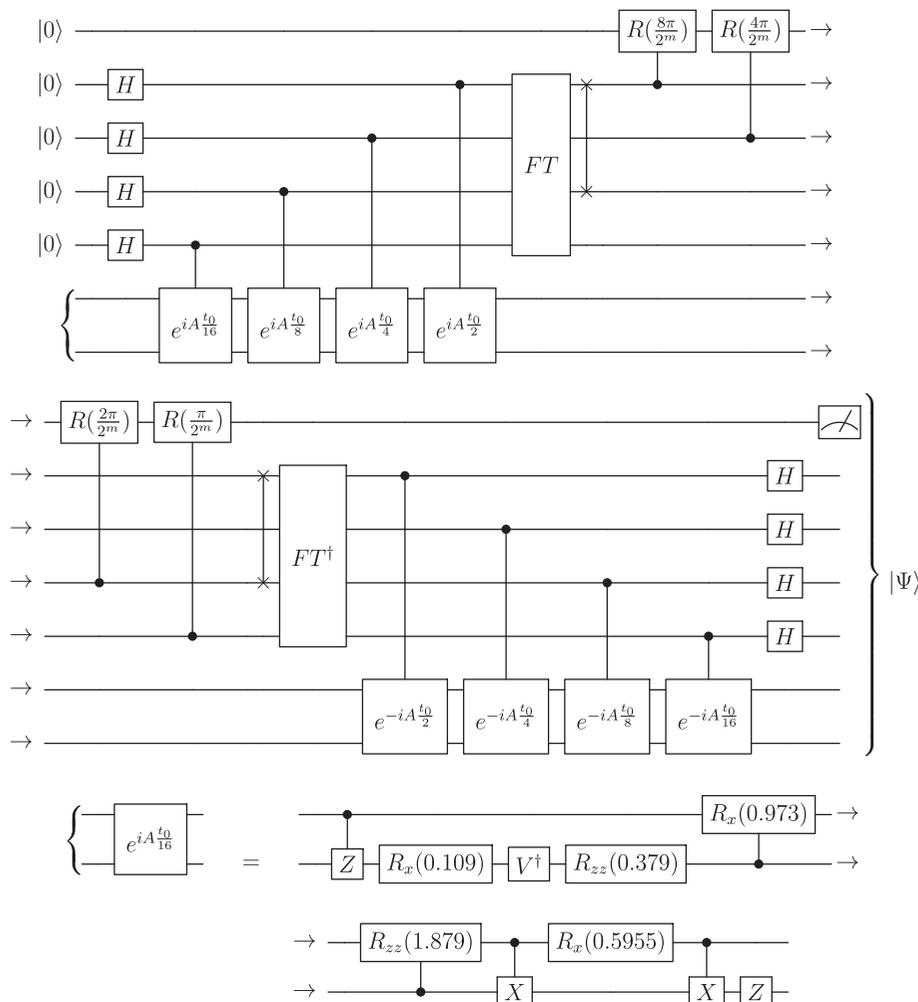


Figure 4. Quantum circuit for solving $Ax = b$ with $A_{4 \times 4}$ being the matrix shown in Equation (1). The first qubit at the top of the figure is the ancilla bit. The four qubits in the middle stand for the register C . The two qubits at the bottom is the register that stores the vector b . The detailed matrix forms of the quantum gates are presented in Appendix 1.

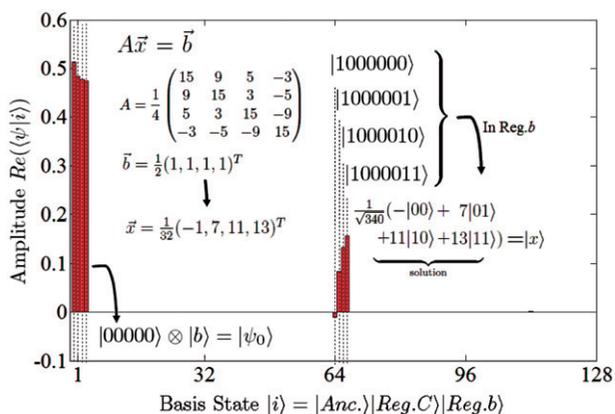


Figure 5. The final state of the algorithm for solving the 4×4 system. The horizontal axis represents the decimal value that corresponds to a basis state of the seven-qubit system. The vertical axis represents the real parts of the probability amplitudes that corresponds to a certain basis.

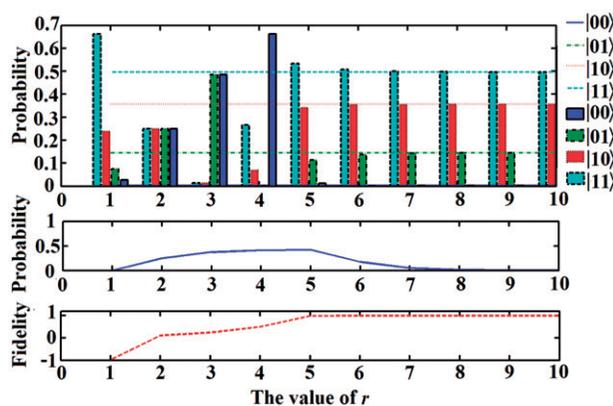


Figure 6. Simulation results on the dependence of the final state on the value of the parameter r . The first bar plot shows the probability distributions over the four basis states of Register M with different values of r in the controlled rotation gates acted upon the ancilla bit. The horizontal dashed lines show the analytical values of the probabilities. The two plots following show the probability of getting the solution as well as the fidelity $\langle x'|x \rangle$ as functions of r .

converge to the analytical values. Those analytical values correspond to the analytical solution of the 4×4 system $\mathbf{x} = \frac{1}{64}(-1, 7, 11, 13)^T$. Furthermore, the fidelity of the results $\langle x|x \rangle$ increases as r is increased. However, in Figure 6, as r grows beyond around 5, the probability of measuring the ancilla bit as $|1\rangle$ decays, which indicates that as r is increased, the solutions obtained in the final state in register b becomes more accurate yet less probable to attain.

In conclusion, in this work we present constructions of quantum circuits, both general and particular, for implementing the algorithm for solving linear systems of equations. Our results may motivate experimentalists with the capability of addressing seven or more qubits and execute basic quantum gates on their setups to implement the algorithm and verify its results. Future work regarding the improvement upon the algorithm needs to be focused on topics such as extending the algorithm to cases where the matrix A has a high condition number [23].

Acknowledgement

This work is dedicated to professor Dudley Herschbach in appreciation of his contribution to science and celebration of his 80th birthday. Also we thank the NSF Center for Quantum Information and Computation for Chemistry, Award number CHE-1037992.

References

- [1] M.A. Nielsen and I.L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).
- [2] P.W. Shor, in *Proceedings of the 35th Annual Symposium Foundations of Computer Science*, edited by S. Goldwasser (IEEE Computer Society Press, New York, 1994), pp. 124–134.
- [3] S. Lloyd, *Science* **273** (5278), 1073 (1996).
- [4] D.S. Abrams and S. Lloyd, *Phys. Rev. Lett.* **83** (24), 5162 (1999).
- [5] D.S. Abrams and S. Lloyd, *Phys. Rev. Lett.* **79** (13), 2586 (1997).
- [6] A. Papageorgiou, I. Petras, J.F. Traub, and C. Zhang, preprint, arXiv:1008.4294v2 (2010). <<http://arxiv.org/abs/1008.4294>>
- [7] A. Papageorgiou and C. Zhang, preprint, arXiv:1005.1318v3 (2010). <<http://arxiv.org/abs/1005.1318>>
- [8] H. Wang, S. Kais, A. Aspuru-Guzik and M.R. Hoffmann, *Phys. Chem. Chem. Phys.* **10**, 5388 (2008).
- [9] A. Aspuru-Guzik, A.D. Dutoi, P.J. Love and M. Head-Gordon, *Science* **379** (5741), 1704 (2005).
- [10] D. Lidar and H. Wang, *Phys. Rev. E* **59**, 2429 (1999).

- [11] H. Wang, S. Ashhab, and F. Nori, *Phys. Rev. E*, preprint, arXiv:1108.5902 (2011). <<http://arxiv.org/abs/1108.5902>>
- [12] J. You and F. Nori, *Nature* **474**, 589 (2011).
- [13] J. You and F. Nori, *Phys. Today* **58** (11), 42 (2005).
- [14] I. Buluta and F. Nori, *Science* **326**, 108 (2009).
- [15] J. Dowling, *Nature* **439**, 919 (2006).
- [16] L. Veis and J. Pittner, *J. Chem. Phys.* **133**, 194106 (2010).
- [17] L. Veis, J. Visnak, T. Fleig, S. Knecht, T. Saue, L. Visscher, and J. Pittner, preprint, arXiv:1111.3490v1 (2011). <<http://128.84.158.119/abs/1111.3490v2>>
- [18] A.W. Harrow, A. Hassidim and S. Lloyd, *Phys. Rev. Lett.* **15** (103), 150502 (2009).
- [19] A. Daskin and S. Kais, *J. Chem. Phys.* **134**, 144112 (2011).
- [20] A. Daskin and S. Kais, *Mol. Phys.* **109** (5), 761 (2011).
- [21] D.W. Berry, G. Ahokas, R. Cleve and B.C. Sanders, *Commun. Math. Phys.* **270**, 359 (2007).
- [22] V.V. Shende, S.S. Bullock and I.L. Markov, *IEEE Trans. Comput.-Aided Des.* **25**, 1000 (2006).
- [23] A.M. Childs, *Nat. Phys.* **5**, 861 (2009).

Appendix 1

The matrix representations of the quantum gates and algorithms used in this work are the following: X , Y and Z gates which are the Pauli operators σ_x , σ_y and σ_z and Hadamard gate:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad (2)$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad H = \frac{1}{2^{1/2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \quad (3)$$

The S gate and T gate (or $8/\pi$ gate) are

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 0 \\ 0 & \exp(i\frac{\pi}{4}) \end{pmatrix}. \quad (4)$$

Square root of X gate (or NOT gate) and its conjugate are

$$V = \frac{1}{2} \begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix}, \quad V^\dagger = \frac{1}{2} \begin{pmatrix} 1-i & 1+i \\ 1+i & 1-i \end{pmatrix}. \quad (5)$$

Rotation gates are

$$R_x(\theta) = \begin{pmatrix} \cos(\theta/2) & i \sin(\theta/2) \\ i \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}, \quad (6)$$

$$R_y(\theta) = \begin{pmatrix} \cos(\theta/2) & \sin(\theta/2) \\ -\sin(\theta/2) & \cos(\theta/2) \end{pmatrix}, \quad (7)$$

$$R_z(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & \exp(i\theta) \end{pmatrix}, \quad (8)$$

$$R_{zz}(\theta) = \begin{pmatrix} \exp(i\theta) & 0 \\ 0 & \exp(i\theta) \end{pmatrix}. \quad (9)$$