Preconditioned Quantum Linear System Algorithm

B. D. Clader,[*] B. C. Jacobs, and C. R. Sprouse

The Johns Hopkins University Applied Physics Laboratory, Laurel, Maryland 20723, USA

(Received 9 January 2013; revised manuscript received 30 May 2013; published 18 June 2013; corrected 2 July 2013)

We describe a quantum algorithm that generalizes the quantum linear system algorithm [Harrow *et al.*, Phys. Rev. Lett. **103**, 150502 (2009)] to arbitrary problem specifications. We develop a state preparation routine that can initialize generic states, show how simple ancilla measurements can be used to calculate many quantities of interest, and integrate a quantum-compatible preconditioner that greatly expands the number of problems that can achieve exponential speedup over classical linear systems solvers. To demonstrate the algorithm's applicability, we show how it can be used to compute the electromagnetic scattering cross section of an arbitrary target exponentially faster than the best classical algorithm.

DOI: 10.1103/PhysRevLett.110.250504 PACS numbers: 03.67.Ac, 41.20.Jb

The potential power of quantum computing was first described by Feynman, who showed that the exponential growth of the Hilbert space of a quantum computer allows efficient simulations of quantum systems, whereas a classical computer would be overwhelmed [1]. Shor extended the applicability of quantum computing when he developed a quantum factorization algorithm that also provides exponential speedup over the best classical algorithm [2].

More recently, Harrow *et al.* [3] demonstrated a quantum algorithm for solving a linear system of equations. In that Letter, the authors demonstrated how to invert a sparse matrix to solve the quantum linear system $A|x\rangle = |b\rangle$, with a stated run time of roughly $O(d^4\kappa^2 \log N/\epsilon)$ [4], where N is the size of the $N \times N$ matrix A, d is the number of nonzero entries per row, κ is the condition number of the matrix, and ϵ is the desired precision of the calculation. Meanwhile, the best classical sparse-matrix solving algorithm, conjugate gradient, has a run time of $O[Nd\kappa \log(1/\epsilon)]$ [6]. The requirements for achieving exponential speedup were (1) the elements of A be efficiently computable via an oracle (i.e., a black-box input or output function), (2) the matrix A must to be sparse, or efficiently decomposable into sparse form, and (3) the condition number of A must scale as polylog N, where N is the size of the linear system, since both quantum and classical solvers scale linearly with κ.

As presented, the algorithm had three features that made it difficult to apply to generic problem specifications and achieve the promised exponential speedup. These included the following. *State preparation*—preparing the generic state $|b\rangle$ is an unsolved problem [7–11], and no mention on how one might do this was provided. *Solution readout*—since the solution is stored in a quantum state $|x\rangle$, measurement of it is impractical. The authors suggested that it could be used to calculate some expectation values of an arbitrary operator $\langle x|\hat{R}|x\rangle$. However, no measurement procedure was specified, and estimating $\langle x|\hat{R}|x\rangle$ is not trivial in general. *Condition number*—in order for the quantum algorithm to achieve exponential speedup, the condition number can scale at most polylogarithmically with the size of the matrix A. This is a very strict condition that greatly limits the class of problems that can achieve exponential speedup.

In this Letter, we provide solutions to these three problems, greatly expanding the applicability of the quantum linear systems algorithm (QLSA). In addition, we show how our new techniques enable the first start-to-finish application of the QLSA to a problem of broad interest and importance. Namely, we show how to solve for the scattering cross section of an arbitrary target exponentially faster than the best classical algorithm.

Before we begin, we first review the original scheme of Harrow *et al.* [3]. One begins by preparing a quantum state $|\Psi\rangle = \sum_{\tau=0}^{N-1}|\tau\rangle|b\rangle$ (state normalization is ignored here and in subsequent steps for clarity). Next, perform a phase-estimation routine by simulating the matrix A as a Hamiltonian for time $|\tau\rangle$ giving

$$|\Psi\rangle \rightarrow \sum_{j=0}^{N-1}\sum_{\tau=0}^{T-1}|\tau\rangle e^{i\lambda_j\tau t_0/T}\beta_j|u_j\rangle, \tag{1}$$

with $t_0 = O(\kappa/\epsilon)$. This value is determined by error requirements and implies that the total quantum algorithm complexity scales linearly with κ. To obtain Eq. (1) we have expanded the state vector $|b\rangle$ in the eigenbasis of A with eigenvalues λ_j, eigenvalues $|u_j\rangle$, and probability amplitudes β_j. Apply a quantum Fourier transform to the first register yielding

$$|\Psi\rangle \rightarrow \sum_{j=0}^{N-1}|\tilde{\lambda}_j\rangle\beta_j|u_j\rangle, \tag{2}$$

where $\tilde{\lambda}_j$ is related to the eigenvalues of A through a constant scaling. Apply a rotation to an adjoined ancilla qubit, controlled off the value of the first register giving

$$|\Psi\rangle \rightarrow \sum_{j=0}^{N-1}|\tilde{\lambda}_j\rangle\beta_j|u_j\rangle\left(\sqrt{1-\frac{C}{\lambda_j}}|0\rangle_a + \frac{C}{\lambda_j}|1\rangle_a\right), \tag{3}$$

where C is a normalization constant chosen to ensure rotations are less than 2π, and the ancilla qubit is denoted by the subscript a. Uncompute the first register by reversing the previous steps and measure the ancilla qubit. If the measurement result is $|1\rangle$, we obtain

$$|\Psi\rangle \to C' \sum_{j=0}^{N-1} \frac{\beta_j}{\lambda_j} |u_j\rangle \equiv |x\rangle, \tag{4}$$

the solution to $A|x\rangle = |b\rangle$, with normalization factor C'.

With this starting point, we present robust approaches to issues highlighted regarding state preparation, solution readout, and condition number that are not addressed in the scheme outlined in Eqs. (1)–(4). Direct preparation of the state $|b\rangle$, required by Eq. (1), is not possible in general. Consider instead the state

$$|b_T\rangle = \cos\phi_b|\tilde{b}\rangle|0\rangle_a + \sin\phi_b|b\rangle|1\rangle_a \tag{5}$$

that contains our desired arbitrary state, $|b\rangle$, entangled with an ancilla qubit in state $|1\rangle_a$. This can be prepared efficiently in the following manner: initialize three quantum registers and an ancilla qubit as

$$|\Psi\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle|0\rangle|0\rangle|0\rangle_a. \tag{6}$$

Query an oracle that calculates the amplitude and phase components, denoted as b_j and ϕ_j, respectively, of the vector $|b\rangle = \sum_{j=0}^{N-1} b_j e^{i\phi_j}|j\rangle$, controlled off the value in the first register. Apply a controlled phase gate to the ancilla qubit, controlled by the calculated phase, and finally rotate the fourth ancilla controlled by the calculated amplitude. Uncompute registers 2 and 3 by recomputing b_j and ϕ_j in the same register leaving

$$|\Psi\rangle \to \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{i\phi_j}|j\rangle\left(\sqrt{1 - C_b^2 b_j^2}|0\rangle_a + C_b b_j|1\rangle_a\right), \tag{7}$$

where $C_b \leq 1/\max(b_j)$ to ensure that all rotations are less than 2π. State (7) is exactly the state (5) with $\sin^2\phi_b = (C_b^2/N)\sum_{j=0}^{N-1} b_j^2$, $\cos^2\phi_b = (1/N)\sum_{j=0}^{N-1}(1 - C_b^2 b_j^2)$, $|\tilde{b}\rangle = (1/\sqrt{N}\cos\phi_b)\sum_j \sqrt{1 - C_b^2 b_j^2}\, e^{i\phi_j}|j\rangle$, and $|b\rangle = (C_b/\sqrt{N}\sin\phi_b)\sum_j b_j e^{i\phi_j}|j\rangle$. The query complexity to prepare this state is $O(1)$.

Next, we apply Eqs. (1)–(3) of the original QLSA to the state $|b_T\rangle$. We modify the original algorithm by removing the last postselection step in Eq. (4), such that the our implementation is unitary. This yields

$$|\Psi\rangle = (1 - \sin^2\phi_b \sin^2\phi_x)^{1/2}|\Phi_0\rangle + \sin\phi_b \sin\phi_x |x\rangle|1\rangle_a|1\rangle_a, \tag{8}$$

where $\sin\phi_x$ is a normalization term resulting from the QLSA, $|\Phi_0\rangle$ is a garbage state in an expanded Hilbert space spanned by the solution vector and two ancilla qubits which are not in the state $|1\rangle$ simultaneously, and $|x\rangle$ is the normalized solution to the linear systems problem for an arbitrary input state $|b\rangle$ entangled with two ancilla qubits in the state $|1\rangle_a$.

We now provide a partial resolution to the readout problem and show how to unentangle the solution $|x\rangle$ from the rest of state (8). While access to the entire solution is impossible since it lies in an exponentially large space, we provide examples of calculable quantities from Eq. (8). These include the overlap of the solution with an arbitrary vector $|R\rangle$ and individual values of the solution vector denoted $x_j = \langle j|x\rangle$. We also note that the solution can be used in conjunction with the quantum data-fitting algorithm [12] to perform approximate quantum state tomography to access information from the entire Hilbert space.

To estimate the overlap, we prepare the state $|R_T\rangle = \cos\phi_r|\tilde{R}\rangle|0\rangle_a + \sin\phi_r|R\rangle|1\rangle_a$ using the same method we used to prepare the state $|b_T\rangle$. Adjoin this state to Eq. (8) along with a fourth ancilla qubit initialized to state $|0\rangle_a$. Apply a Hadamard gate to the fourth ancilla qubit, and use it to perform a controlled swap operation between the registers containing the solution vector $|x\rangle$ and the vector $|R\rangle$, followed by a second Hadamard operation on the ancilla. In doing so, we compute the overlap between $|x\rangle$ and $|R\rangle$,

$$|\langle R|x\rangle|^2 = \frac{P_{1110} - P_{1111}}{\sin^2\phi_b \sin^2\phi_x \sin^2\phi_r}, \tag{9}$$

where P_{1110} and P_{1111} refer to the probability of measuring a 1 in the first three ancilla qubits and a 0 or 1 in the last adjoined ancilla, respectively. More details on this process can be found in the Supplemental Material [13].

To calculate a particular solution value, one simply sets $|R\rangle = |j\rangle$ and uses Eq. (9). Finally, one could use the quantum data-fitting algorithm [12] to access many other features of the solution state vector $|x\rangle$. To ensure that the data-fitting subroutine only operates on the solution vector and not the associated garbage states, one would apply the subroutine as a unitary operator controlled by the two ancilla qubits in Eq. (8) being both in state $|1\rangle$ simultaneously.

The last and most critical issue in the original specification in Ref. [3] relates to the spectral condition number κ. The Hamiltonian simulation step in Eq. (1) causes the quantum algorithm query complexity to scale linearly with κ. Thus, in order for the quantum algorithm to scale as $O(\log N)$ and achieve exponential speedup, κ must scale in the worst case polylogarithmically with the size of the $N \times N$ matrix A. However, for most matrices one typically has linear or even exponential scaling with N [14,15], greatly limiting the class of problems that can achieve exponential speedup.

We provide a solution to the condition number scaling problem through a technique known as preconditioning [16]. When preconditioning, rather than solving the system

$Ax = b$, one instead solves the modified linear system $MAx = Mb$. Convergence is improved if one can find a matrix M such that the condition number of MA is much lower than the original matrix A. The best preconditioner is obviously $M = A^{-1}$. However, finding A^{-1} is equivalent to solving the linear system, so using this as a preconditioner provides no speedup. One solution is to find an efficiently computable approximate inverse $M \approx A^{-1}$. Unfortunately, two constraints make many classical preconditioners unusable. These are (1) only local knowledge of A can be obtained, and (2) the preconditioned matrix MA must itself be sparse for Hamiltonian simulation.

A class of preconditioners that satisfy both these constraints are sparse approximate inverse (SPAI) preconditioners [17,18]. We integrate this method with the quantum algorithm as follows. One attempts to find the matrix M by minimizing

$$\|MA - I\|_F^2 = \sum_{k=0}^{N-1} \|(MA - I)e_k\|_2^2, \qquad (10)$$

where the subscript F refers to the Frobenius norm and e_k is the kth column of the identity matrix. Equation (10) separates into N independent least squares problems

$$\min_{\hat{m}_k} \|\hat{A}\hat{m}_k - \hat{e}_k\|_2 \qquad (11)$$

for $k = 0, \ldots, N-1$, where the circumflex denotes that we have removed rows and columns containing all zeros. One imposes sparsity constraints on matrix M. As an example, if one chooses the sparsity pattern of A as the sparsity pattern of M (a common choice), then the least squares problem in Eq. (11) is very small, of order $n \times d$, where n is the number of nonzero rows in column k and d is the number of nonzero elements per row of the least squares problem in Eq. (11). Thus, we now have N independent $n \times d$ sized least squares problems to compute the SPAI preconditioner. We have an oracle that can compute the elements and locations of nonzero terms in the matrix A for a given row. One queries this oracle to setup Eq. (11) controlled by a supplied row index. Since the matrix A is highly sparse, both n and d are small.

Within the quantum algorithm, to simulate the matrix A, one requires a unitary $U^{(c)}$ that calculates the elements of MA, denoted as a_k, and its column index y_k for a specific graph edge color c (see Supplemental Material [13] and Refs. [8,19,20] for more information), conditioned on a row index k. This operates as $U^{(c)}|k, 0\rangle = |k, a_k, y_k\rangle$. The matrix preconditioner step can fit neatly within this unitary operator. The techniques used to calculate the SPAI require only local accesses of A, which we have access to via its oracle, and the matrix M can be calculated for each row independently. The sparsity structure of M is either calculated efficiently or set *a priori*, and thus we can calculate y_k. Therefore, the oracle for the matrix MA can be created by combining Eq. (11) together with the original oracle for A with only modest overhead of $O(d^3)$ in run time and $O(d^2)$ in query complexity.

To prepare the state $M|b_T\rangle$ as opposed to $|b_T\rangle$ alone requires a slight modification to the oracles presented in Eqs. (6) and (7). For each row index j, the preconditioner M must be computed using Eq. (11). This adds the same constant overhead to the query and computational complexity as stated in the previous paragraph.

The condition number of the preconditioned matrix can be shown to be constrained to lie in a circle of radius $\sqrt{d}\epsilon_{\rm pre}$, where $\epsilon_{\rm pre} > \|Am_k - e_k\|$ is the largest residual of any preconditioned matrix row from the identity [17]. If $\sqrt{d}\epsilon_{\rm pre} < 1$, then the spectral condition number satisfies the inequality

$$\kappa \equiv \left|\frac{\lambda_{\rm max}}{\lambda_{\rm min}}\right| \leq \frac{1 + \sqrt{d}\epsilon_{\rm pre}}{1 - \sqrt{d}\epsilon_{\rm pre}}. \qquad (12)$$

We now show how our algorithm can achieve exponential speedup over the best classical algorithm. On a classical computer the run time is dominated by the linear systems solving operation that requires many matrix vector products. As noted in the introduction, the best sparse-matrix solving algorithm, conjugate gradient, is $O[Nd\kappa\log(1/\epsilon)]$.

In the quantum algorithm, estimation of $\sin\phi_b$ and $\sin\phi_r$ requires $O(1/\epsilon)$ iterations to estimate to accuracy ϵ with amplitude estimation (AE). The QLSA uses Hamiltonian simulation. Berry *et al.* [5] show that when using the Suzuki higher-order integrator method [21], this step requires $N_{\rm exp} \leq 2m^2\tau\exp[2\sqrt{\ln 5\ln(m\tau/\epsilon)}]$ exponential operator applications, where m is the number of submatrices needed to decompose the sparse matrix A into 1-sparse form ($m = 6d^2$ using the decomposition technique in Ref. [5], d is the sparsity of A, where sparsity is defined as the maximum number of nonzero elements per row), and $\tau = \kappa\|A\|/\epsilon$. For the algorithm to be accurate to within ϵ, $\tau = O(\kappa/\epsilon)$ [3]. Since we estimate $\sin\phi_x$ using AE, multiple applications of Hamiltonian simulation with different times are required. Thus, the query complexity required to estimate $\sin\phi_x$ as well as P_{1110} and P_{1111} to accuracy ϵ is $\tilde{O}(d^4\kappa\log N/\epsilon^2)$, where the tilde indicates that we are neglecting more slowly growing terms in the exponent of $N_{\rm exp}$. Our implementation is quadratically better in κ than in the original QLSA due to our removal of the postselection step.

The overhead to estimating the preconditioner varies depending on which technique one uses for estimating the sparsity pattern. As an example, if one uses an *a priori* sparsity pattern [18] then one must simply solve a small $O(n \times d)$ linear system, which takes $O(d^3)$ operations and $O(d^2)$ A matrix oracle queries. Therefore, creating $M|b\rangle$ has a run time overhead of $O(d^3)$ and requires $O(d^2)$ oracle queries, compared with $O(1)$ queries for nonpreconditioned state preparation.

Combining all steps, including the preconditioner, the overall quantum algorithm run time complexity is $\tilde{O}(d^7\kappa \log N/\epsilon^2)$. The additional factor of d^3 over the query complexity quoted above is due to the overhead of using the preconditioner. When the SPAI is used, this algorithm provides exponential speedup over the best classical algorithm, since the condition number is bounded by Eq. (12). The SPAI preconditioner is known to be applicable to a wide class of problems [16–18,22,23], providing a large number of applications that can achieve exponential speedup.

To demonstrate the algorithm's usefulness, we now show how it can be used to calculate the electromagnetic scattering cross section of an arbitrary target using the finite element method (FEM) [24]. Calculation of the scattering cross section is routinely used in the electromagnetics modeling community to characterize detectability by radar. In particular, the calculations are used to drive design considerations of low-observable (stealth) objects. The FEM approach to solving an electromagnetic scattering problem is to break up the computational domain into small volume elements and apply boundary conditions at neighboring elements. This allows one to cast the solution of Maxwell's equations into a linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$.

The matrix $\mathbf{A}$ is constructed from a discretization of Maxwell's equation together with appropriate boundary conditions due to the scattering object under consideration. The vector $\mathbf{b}$ consists of the known electric field components on the scattering boundary. The matrix $\mathbf{A}$ and vector $\mathbf{b}$, which contain information about the scattering object, can be efficiently derived from the components of a matrix $\mathbf{F}$ that is dependent only upon the form of the discretization chosen to break up the computational domain (see Ref. [24] and the Supplemental Material [13]) together with boundary conditions that include the scattering geometry. Edge basis vectors [25], denoted as $\mathbf{N}_i$, are highly popular for electromagnetic scattering applications. They give a form of $\mathbf{F}$ as

$$F_{lj} = \int_V [(\nabla \times N_l) \cdot (\nabla \times N_j) - k^2 \mathbf{N}_l \cdot \mathbf{N}_j] dV + ik \int_S (N_l)_t \cdot (N_j)_t dS, \tag{13}$$

where V is the volume of the computational region, S is the outer surface of the computational region, k is the electric field wave number, the subscript t denotes the tangential component, and the indices l and j denote the numbering of all the edges contained in the volume V. The surface integral is an absorbing term used to prevent reflections off the artificial computational boundary. On the inner scattering surface the correct boundary condition for the scattered field on metallic scatterers is $\hat{\mathbf{n}} \times \mathbf{E} = -\hat{\mathbf{n}} \times \mathbf{E}^{(i)}$, where $\mathbf{E}^{(i)}$ is the incident field, $\mathbf{E}$ is the scattered field, and $\hat{\mathbf{n}}$ is the unit vector normal to the surface is applied.

Using the edge basis expansion, the far-field radiation in direction $\mathbf{s}$ is

$$\mathbf{E}(\mathbf{s}) \cdot \hat{\mathbf{p}} = \frac{e^{-iks}}{4\pi s} \sum_k R_k(\hat{\mathbf{s}}) x_k, \tag{14}$$

where $\hat{\mathbf{p}}$ is the radar polarization (with $\hat{\mathbf{p}} \cdot \hat{\mathbf{s}} = 0$) and

$$R_k(\hat{\mathbf{s}}, \hat{\mathbf{p}}) = \hat{\mathbf{p}} \cdot \int_S \hat{\mathbf{s}} \times \{\hat{\mathbf{s}} \times [(\nabla \times \mathbf{N}_k) \times \hat{\mathbf{n}}] + ik\mathbf{N}_k \times \hat{\mathbf{n}}\} e^{ik\hat{\mathbf{s}}\cdot\mathbf{r}} dS, \tag{15}$$

where the index k here is the global edge index. The radar scattering cross section (RCS) in the direction $\hat{\mathbf{s}}$ is given by

$$\text{RCS} = \lim_{s\to\infty} 4\pi s^2 |\mathbf{E}(\mathbf{s}) \cdot \hat{\mathbf{p}}|^2 = \frac{1}{4\pi} |\mathbf{R} \cdot \mathbf{x}|^2, \tag{16}$$

or simply the dot product of $\mathbf{R}$ with the solution $\mathbf{x}$, where we have assumed an incident plane wave with unit electric field amplitude without loss of generality.

The edge basis elements can take a simple functional form, which allows one to analytically evaluate the integrals in Eqs. (13) and (15). This allows for fast computation of the matrix and vector elements, allowing for efficient oracles. Because of the local nature of the finite element expansion, the volume and surface integrals extend only over the region encompassed by the finite element. As a result $\mathbf{A}$ is highly sparse, also a necessary condition for the quantum algorithm.

To obtain the cross section using the quantum algorithm, one uses the oracles just defined, together with the SPAI, to create the A matrix and $|b\rangle$ and $|R\rangle$ state vectors. Then one must restore units to the normalized output received from the quantum algorithm. Doing so yields the following equation for the cross section in terms of outputs from the quantum computation:

$$\text{RCS} = \frac{1}{4\pi} \frac{N^2 \sin^2\phi_b \sin^2\phi_r}{C_b^2 C_r^2 \sin^2\phi_x} (P_{1110} - P_{1111}), \tag{17}$$

where $C_b = 1/\max(\mathbf{b})$ and $C_r = 1/\max(\mathbf{R})$ are known parameters. Thus, to compute the cross section, we estimate each $\sin^2\phi_{(b,x,r)}$ term as well as the P_{1110} and P_{1111} terms independently using AE.

Finally, we comment on the efficiency of the scattering cross section calculation. With no preconditioning, finite element condition numbers scale as $N^{2/n}$ [14,15], where n is the number of dimensions of the problem, implying that even in the most general case our algorithm scales better than its classical counterpart for a three-dimensional finite element problems. However, by applying the quantum preconditioner, the eigenvalues of the finite element matrix can be bounded achieving exponential speedup, since the FEM admits an efficient SPAI [22,23].

We have demonstrated a quantum algorithm that generalizes the QLSA to solve arbitrary linear systems. We show how simple ancilla measurements can efficiently calculate many useful quantities of interest from the exponentially large solution space. Additionally, we have greatly expanded the class of problems that can be solved with exponential speedup, by incorporating matrix preconditioning into the quantum algorithm. To demonstrate its functionality we showed how one could use it to solve an electromagnetic scattering problem using the finite element method and estimate the scattering cross section. We show that this can be done in a time exponentially faster than the best classical algorithm. This opens up the potential for quantum computing to be applied to a broad class of problems of practical interest to the computational physics community.

This project was supported by the Intelligence Advanced Research Projects Activity via Department of Interior National Business Center Contracts No. N00024-03-D-6606 and No. 2012-12050800010, with additional support provided by a Stuart S. Janney Fellowship from the Applied Physics Laboratory. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government. Many thanks to Joan Hoffmann and Nathan Wiebe for helpful comments and discussions.

*dave.clader@jhuapl.edu

[1] R. Feynman, Int. J. Theor. Phys. **21**, 467 (1982).

[2] P. Shor, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (IEEE, New York, 1994), pp. 124–134.

[3] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).

[4] Reference [3] incorrectly cites the results of Theorem 2 of [5]. Therefore they quote the cost of linear inversion scales as $O(d^2)$ rather than the correct scaling of $O(d^4)$.

[5] D. Berry, G. Ahokas, R. Cleve, and B. Sanders, Commun. Math. Phys. **270**, 359 (2007).

[6] Y. Saad, *Iterative Methods for Sparse Linear Systems* (Society for Industrial and Applied Mathematics, Philadelphia, 2003).

[7] L. Grover and T. Rudolph, arXiv:quant-ph/0208112v1.

[8] D. Aharonov and A. Ta-Shma, in *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC '03)* (ACM, New York, 2003), pp. 20–29.

[9] N. J. Ward, I. Kassal, and A. Aspuru-Guzik, J. Chem. Phys. **130**, 194105 (2009).

[10] A. N. Soklakov and R. Schack, Phys. Rev. A **73**, 012307 (2006).

[11] P. Kaye and M. Mosca, arXiv:quant-ph/0407102v1.

[12] N. Wiebe, D. Braun, and S. Lloyd, Phys. Rev. Lett. **109**, 050505 (2012).

[13] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.110.250504 for detailed descriptions of the quantum subroutines and electromagnetic scattering oracles.

[14] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods* (Springer-Verlag, New York, 2008), 3rd ed., Chap. 9.6.

[15] R. E. Bank and L. R. Scott, SIAM J. Numer. Anal. **26**, 1383 (1989).

[16] M. Benzi, J. Comput. Phys. **182**, 418 (2002).

[17] M. Grote and T. Huckle, SIAM J. Sci. Comput. **18**, 838 (1997).

[18] E. Chow, SIAM J. Sci. Comput. **21**, 1804 (2000).

[19] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, in *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC '03)* (ACM, New York, 2003), pp. 59–68.

[20] N. Wiebe, D. W. Berry, P. Hoyer, and B. C. Sanders, J. Phys. A: Math. Theor. **44**, 445308 (2011).

[21] M. Suzuki, Phys. Lett. A **146**, 319 (1990).

[22] S. Li, P. Rui, and R. Chen, in *Proceedings of the IEEE Antennas and Propagation Society International Symposium, 2006* (IEEE, New York, 2006), pp. 1765–1768.

[23] X. W. Ping and T.-J. Cui, Prog. Electromagn. Res. **98**, 15 (2009).

[24] J. Jin, *The Finite Element Method in Electromagnetics* (John Wiley and Sons, Inc., New York, 2002), 2nd ed.

[25] A. Chatterjee, J. Jin, and J. Volakis, IEEE Trans. Antennas Propag. **41**, 221 (1993).