QUANTUM ALGORITHMS

Equation solving by simulation

Quantum computers can outperform their classical counterparts at some tasks, but the full scope of their power is unclear. A new quantum algorithm hints at the possibility of far-reaching applications.

Andrew M. Childs

Quantum mechanical computers have the potential to quickly perform calculations that are infeasible with present technology. There are quantum algorithms to simulate efficiently the dynamics of quantum systems[1] and to decompose integers into their prime factors[2], problems thought to be intractable for classical computers. But quantum computation is not a magic bullet — some problems cannot be solved dramatically faster by quantum computers than by classical ones. Is a quantum computer necessarily a special-purpose device then, suitable only for number-theoretic calculations or problems that are inherently quantum? There might be hope for more generality. Writing in *Physical Review Letters*[3], Aram Harrow, Avinatan Hassidim and Seth Lloyd describe how quantum computers can extract information about the solutions to linear equations, a fundamental task with broad applications.

The basic problem of finding a vector $\mathbf{x}$ satisfying $A\mathbf{x} = \mathbf{b}$ for some given matrix A and vector $\mathbf{b}$ arises throughout science and engineering. For example, signal processing, convex optimization and finite-element analysis all rely on solving linear equations. Owing to the importance of linear systems, considerable effort has been spent on finding fast algorithms for solving them. One simple approach is known as Gaussian elimination, a method that was already used in ancient China, and that today is standard fare in courses on linear algebra. Alternative methods offer improved speed and better numerical stability, and some can take advantage of special properties such as sparsity of A. However, if A is an $N \times N$ matrix (or a rectangular matrix for which the larger dimension is N), all of these methods use a number of operations at least proportional to N.

The authors[3] suggest approaching this problem using quantum simulation. Given an $N \times N$ sparse Hermitian matrix A, a quantum state $|b\rangle$ and a time t, quantum simulation provides a method for preparing the quantum state $e^{-iAt}|b\rangle$ using a number of operations that is only polynomial in $\log(N)$ (ref. 4). In the approach of Harrow *et al.*, the vector $\mathbf{b}$ is first encoded into a quantum state $|b\rangle$. Then, by a well-known technique called phase estimation[5], the ability to produce $e^{-iAt}|b\rangle$ is leveraged to create a quantum state $|x\rangle$ proportional to $A^{-1}|b\rangle$. (A similar approach can be applied when the matrix A is non-Hermitian, or even when A is non-square.) The result is a solution to the system of linear equations encoded as the quantum state $|x\rangle$.

Producing a quantum state proportional to $A^{-1}|b\rangle$ does not, by itself, solve the task at hand. To extract information from a quantum state, we must perform a measurement. Learning all N amplitudes of an N-dimensional quantum state requires a number of measurements at least proportional to N. Thus, if our goal is to completely reconstruct a solution $\mathbf{x}$, there is no hope for a quantum algorithm to offer a significant advantage over classical methods. However, for some problems we might not be interested in the entire vector $\mathbf{x}$, but rather in some special feature of it, such as an expectation value. Then a quantum computer may be able to solve the problem rapidly.

In addition to readout, the approach suffers from other significant limitations. The vector $\mathbf{b}$ must be given in a way that allows quick preparation of the quantum state $|b\rangle$. Methods are available for doing this if $\mathbf{b}$ has a suitable implicit description, but the approach is useless if $\mathbf{b}$ is given by explicit classical data. Furthermore, as in classical methods for solving linear equations, the performance depends crucially on the condition number κ, a measure of how close A is to being singular. The running time of the quantum algorithm is polynomial in $\log(N)$ and in κ, so the quantum advantage is only significant when κ is not too large.

Having lowered the bar for the sense in which we hope to solve a system of linear equations, one might wonder whether this quantum algorithm[3] offers a real advantage over classical computing at all. To address this concern, in perhaps the most interesting aspect of their work, Harrow, Hassidim and Lloyd prove that the problem they solve is as hard as anything a quantum computer can do. In particular, they show that any quantum computation can be encoded into an instance of solving linear equations, even with the restrictions required for their quantum solver to be efficient. Therefore, either ordinary classical computers can efficiently simulate quantum ones — a highly unlikely proposition — or the quantum algorithm for solving linear equations performs a task that is beyond the reach of classical computation.

Proving 'hardness' results of this kind is a widely used strategy for establishing the non-triviality of quantum algorithms. Similar constructions are known for problems such as finding ground states by adiabatic evolution[6], computing invariants of knots[7], estimating entries of powers of matrices[8] and contracting tensor networks[9], among other tasks. But what sets linear equations apart is their ubiquity in scientific computing and engineering applications.

Will the quantum solution of linear equations[3] turn out to be a widely used tool, or are its limitations too great for the technique to be of practical significance? Unfortunately, no concrete task has yet been proposed for which the quantum algorithm provides a clear advantage. But it will be exciting to explore the impact of this and related applications of quantum computing, in particular as the methods can be put into practice. ❐

Andrew M. Childs is in the Department of Combinatorics and Optimization and at the Institute for Quantum Computing, University of Waterloo, 200 University Avenue West, Waterloo, Ontario N2L 3G1, Canada.
e-mail: amchilds@uwaterloo.ca

References

1. Feynman, R. P. *Int. J. Theor. Phys.* **21,** 467–488 (1982).
2. Shor, P. W. *SIAM J. Comput.* **26,** 1484–1509 (1997).
3. Harrow, A. W., Hassidim, A. & Lloyd, S. *Phys. Rev. Lett.* **103,** 150502 (2009).
4. Aharonov, D. & Ta-Shma, A. in *Proc. 35th ACM Symp. Theor. Comput.* 20–29 (ACM, 2003).
5. Cleve, R., Ekert, A., Macchiavello, C. & Mosca, M. *Proc. R. Soc. Lond. A* **454,** 339–354 (1998).
6. Aharonov, D. *et al.* in *Proc. 45th IEEE Found. Comput. Sci.* 42–51 (IEEE, 2004).
7. Bordewich, M., Freedman, M., Lovász, L. & Welsh, D. *Comb. Probab. Comput.* **14,** 737–754 (2005).
8. Janzing, D. & Wocjan, P. *Theor. Comput.* **3,** 61–79 (2007).
9. Arad, I. & Landau, Z. Preprint at <http://www.arxiv.org/abs/0805.0040> (2008).