REDUCING QUANTUM COMPUTATIONS TO ELEMENTARY UNITARY OPERATIONS

Quantum computations are intimately connected with unitary operators. This article shows that standard techniques from numerical linear algebra can be used to represent quantum computations as sequences of simple quantum operations, called quantum Givens operators, on single quantum bits.

Quantum computing has enormous potential for introducing fundamentally new capabilities to computational science and engineering, primarily through exponential parallelism.[1,2] One of the many challenges in building practical quantum computers is to reduce a general quantum computation to some set of elementary operations that simple quantum devices can implement. By analogy, in the case of classical computing, it is well known that the elementary operations AND, OR, and NOT are sufficient to implement any finite classical computation. In fact, the single NOR operation by itself is sufficient.[3]

Over the past 10 years, several researchers have addressed the question of reducing quantum computations to elementary operations.[4–6] My goal here is to collect those results, simplify them, and present the basic ideas in terms of traditional linear algebraic operations.[7]

1521-9615/01/$10.00 © 2001 IEEE

GEORGE CYBENKO
Dartmouth College

Some background

Our starting point is the observation that a quantum computer implements a unitary matrix operation on the quantum "state" of the quantum computer's register. In classical linear algebra, any unitary matrix can be expressed as a product of unitary operations acting in 2D planes. The quantum computing analog of this result is similar in spirit but different in the details, specifically about what constitutes a primitive operation and how permutations are implemented using bit exchanges.

To begin, consider the Schrödinger equation with a time-invariant Hamiltonian that governs the evolution of a quantum computer:

$$i\hbar \frac{\partial x(t)}{\partial t} = Hx(t)$$

where x is the quantum state of the computer's registers, H is the total system energy Hamiltonian (a Hermitian operator), and $\hbar$ is Planck's constant. It is well known that the solution to this equation is

$$x(t) = e^{-iHt/\hbar} x(0).$$

Because H is Hermitian, $U = e^{-iH/\hbar}$ is a unitary

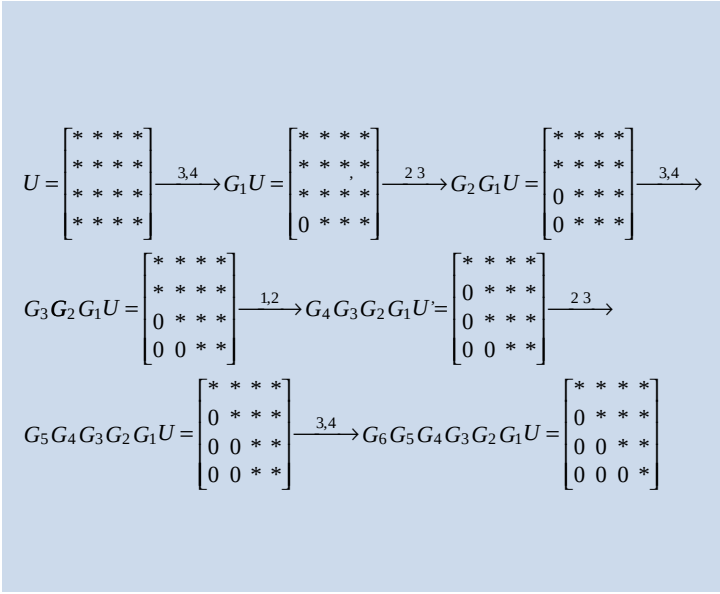

Figure 1. A 4 × 4 example of a classical triangularization or QR factorization of U using complex 2D unitary operations.

operator, as we can easily verify:

$$U^* = e^{iH^*/\hbar} = U^{-1} \text{ and so } U^*U = UU^* = I.$$

A quantum computer has a register consisting of n quantum bits or qubits. Each qubit has classical states 0 and 1 so that the n qubit register has 2^n classical states. The state of the quantum computer is a 2^n dimensional vector, x, as above, indexed by the classical state values, i = 000...00, 000...01, 000...10, ..., 111...11 in binary notation. Moreover,

$$\|x\|^2 = \sqrt{\sum_j |x_j|^2} = 1$$

and the squared norm of each component of x, namely $|x_j|^2$, can be interpreted as the probability that the register is in state j. Clearly, $\|Ux\| = \|x\| = 1$, because U is unitary. We call x the wavefunction of the register.

In linear algebra terms, U is merely a $2^n \times 2^n$ unitary matrix that can be represented by $2^{n-1}(2^n - 1)$ unitary matrices operating on 2D planes. We can see this by performing a classical triangularization or QR-factorization of U using complex 2D unitary operations.[7] Figure 1 shows a 4×4 example, where * denotes a possibly nonzero element. The notation $\xrightarrow{3,4}$ denotes the application of a unitary operation in dimensions 3 and 4 from the right, namely GU.

For example,

$$G_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & \delta & \gamma \end{bmatrix}.$$

G_1 is unitary, requiring that

$$\begin{bmatrix} \alpha & \beta \\ \delta & \gamma \end{bmatrix} \begin{bmatrix} \alpha & \beta \\ \delta & \gamma \end{bmatrix}^* = I$$

and we choose it to transform

$$\begin{bmatrix} u_{31} \\ u_{41} \end{bmatrix} \text{ to } \begin{bmatrix} \sqrt{|u_{31}|^2 + |u_{41}|^2} \\ 0 \end{bmatrix}.$$

For instance,

$$\begin{bmatrix} \alpha & \beta \\ \delta & \gamma \end{bmatrix} = \frac{1}{\sqrt{|u_{31}|^2 + |u_{41}|^2}} \begin{bmatrix} \overline{u_{31}} & \overline{u_{41}} \\ u_{41} & -u_{31} \end{bmatrix}$$

works just fine. The order of the sequence of steps is very important to maintain 0s where they have already been introduced.

This is the standard QR-factorization algorithm, typically based on so-called Givens rotations in the real case (see other texts for information on numerical linear algebra[6]). Calling this a quantum Givens operation, we see that in this example, we require 4*3/2 = 6 quantum Givens operations. After applying these six operations, the final matrix,

$$\prod_{i,j} G_{i,j} U = D$$

is unitary and upper triangular, and we can readily see it must therefore be diagonal. (Just consider the consequences of $UU^* = I$ on the off-diagonal entries of a triangular unitary matrix U.) This example shows the general approach to representing a unitary matrix as a product of elementary 2D quantum Givens operations together with a diagonal scaling matrix, namely

$$U = \left(\prod_{i,j} G_{i,j} \right)^{-1} D.$$

However, we can't implement quantum Givens operations on two arbitrary coordinate planes of the quantum register state vector, as shown below, so some modifications of the classical algorithm are required.

Elementary quantum computer operations

In keeping with standard notation used to describe quantum computer registers and operations, let the register consist of n input qubits, as depicted in Figure 2. Think of control as progressing from left to right, and the devices in the path affect the state of the quantum register.

This register has $2^3 = 8$ classical states, whereas the quantum state, the wavefunction, is an 8D complex vector. In the following sections, I present some elementary quantum gates and how they operate on the qubits and wavefunction. Elementary quantum gates operate on a single qubit at a time, with control optionally provided by another qubit in some cases.

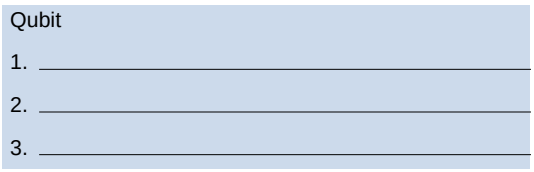

Figure 2. Graphical depiction of a three-qubit quantum register. Control and time flow from left to right.

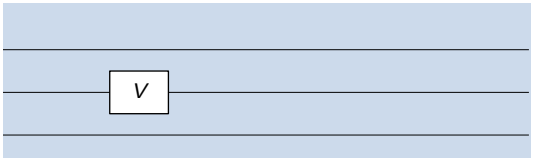

Figure 3. A qubit operation with no control bits.

Qubit operation with no control bits

Figure 3 represents this operation. In the figure,

$$V = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$$

is a single qubit quantum operation applied to the second qubit. In the 8D wavefunction representation with the standard ordering, this operation is simultaneously performed on all pairs of planes defined by states that differ only in their second bits. So, the 8×8 matrix representation of this operation is given by

$$\begin{bmatrix} \alpha & 0 & \beta & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha & 0 & \beta & 0 & 0 & 0 & 0 \\ \gamma & 0 & \delta & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma & 0 & \delta & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha & 0 & \beta & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha & 0 & \beta \\ 0 & 0 & 0 & 0 & \gamma & 0 & \delta & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma & 0 & \delta \end{bmatrix}$$

This operates simultaneously and identically on the (0, 2), (1, 3), (4, 6) and (5, 7) planes. Note, I am numbering rows and columns starting with 0, not 1, to make some later derivations easier to express. This is an elementary quantum gate with no control bits. You can think of the register representation and quantum gate operation as a shorthand for the unitary operation presented above—I use this notation throughout the article, and it might require some getting used to. When in doubt, write out a simple example (3×3 is typically enough) to get some intuition.

Qubit operation with control bits

Adding a control bit effectively means applying the operation only to states in which the corresponding control bit is set (that is, equal to 1 or "True"). You can think of this operation as follows. All eight classical states are present in the quantum computer, and the controlled quantum operation only acts on the wavefunction coordinates in which the classical states have the requisite classical bit set to 1. Figure 4a, for example, means apply V only to the states in which the first qubit is set, so the operation in matrix terms is effectively

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha & 0 & \beta & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha & 0 & \beta \\ 0 & 0 & 0 & 0 & \gamma & 0 & \delta & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma & 0 & \delta \end{bmatrix}.$$

Similarly,

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha & 0 & \beta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma & 0 & \delta & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha & 0 & \beta \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma & 0 & \delta \end{bmatrix}$$

represents Figure 4b.

Finally, the elementary quantum operation on

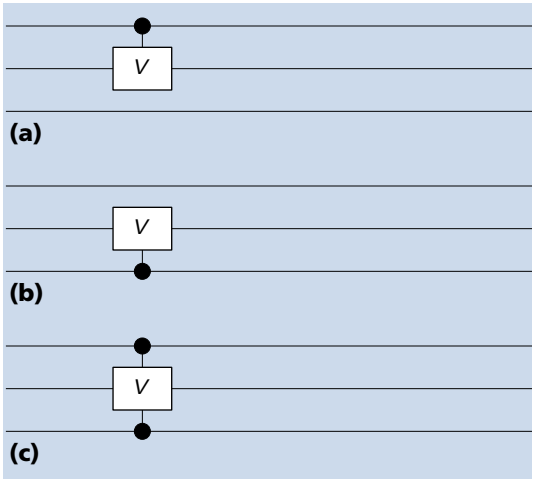

Figure 4. Elementary quantum operations ((a) and (b)) with two different control bits. (c) An elementary quantum operation on the second qubit with both the first and third qubits as controls.

the second qubit with both the first and third qubits as controls is depicted as shown in Figure 4c, which applies V only to the dimensions in which both the first and third qubits are set. In matrix terms, this is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha & 0 & \beta \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma & 0 & \delta \end{bmatrix}$$

The Toffoli gate

A special quantum gate is the so-called Toffoli gate,

$$T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

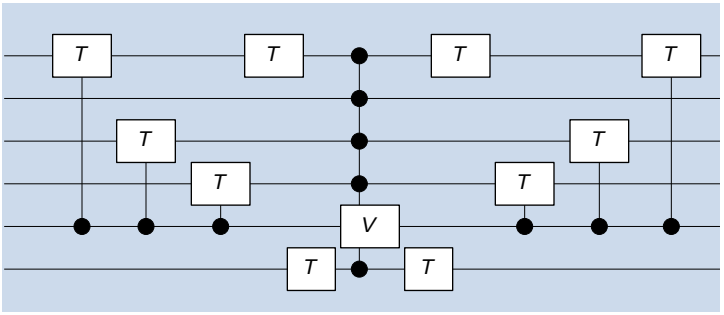

Figure 5. Graphical representation of elementary quantum operations required to implement a 2 × 2 quantum Givens operation in the $i = 011100$, $j = 110010$ plane.

In classical terms, this is a qubit negation, because it maps classical states to their negations. The Toffoli gate has many interesting properties in the quantum computing context.[1]

Reducing a general quantum computation to elementary quantum operations

Recall that we are ordering coordinates of the wavefunction vector from 0 to $2^n - 1$. This makes it easier to demonstrate how to implement a 2D unitary operator acting on planes i and j with $0 \le i < j \le 2^n - 1$ using only the above repertoire of elementary quantum operations. Because i and j are different, they have at least one bit different in their binary representations, say, the kth bit. Using the kth bit as a control bit, apply the Toffoli gate to every qubit for which the binary representations of i and j differ, except the kth bit, of course. Then, apply the Toffoli operator, uncontrolled, to all bits of the thus permuted i and j binary representations that are 0. This permutation of the register state space effectively makes all the bits of the permuted representations of i and j equal to 1 except for the kth bit. Now apply a quantum operator on the kth qubit using all other qubits as controls. Finally, apply all the Toffoli gates that were originally applied in the reverse order to effectively undo all the permutations.

For example, suppose we want to apply the 2×2 unitary,

$$V = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix},$$

in the i, j plane. Let $n = 6$, let $i = 011100$ and $j = 110010$ be indices in binary numerical notation, and let $k = 5$. Then we claim that the sequence of elementary operations in Figure 5 implements precisely the unitary operation given by V in the i,j planes. Table 1 shows the sequence of transformations that i and j undergo as a result of these operations.

The transformations involving T are merely permutations that reorder the coordinates of the wavefunction, while the operation of V uses all other qubits as control. That is, the operation of V only affects two coordinates of the wavefunction. The sequence of permutations after applying V undoes the permutation of coordinates. This construction effectively shows that we can reduce any 2D unitary operation

to a sequence of Toffoli gates with a single control bit—or no control bits at all—and 2D unitary operators acting on a single qubit with multiple control bits. The final step in the reduction is to show that we can reduce all elementary qubit quantum operations to Toffoli gates with a single control qubit and uncontrolled quantum gates. This step is recursive and goes as follows.

Without losing generality, consider the elementary qubit operation depicted in Figure 6a by V. Because V is unitary, it has a square root, namely W, so that $WW = V$. Then, the circuit in Figure 6a is identical to the circuit in Figure 6b.

If the first $n - 1$ qubits are set, then we apply $WW = V$ as required. If any of the first $n - 2$ qubits are not set, but the $n - 1$ qubit is, then we apply $WW^* = I$. If any of the first $n - 2$ qubits are not set and neither is the $n - 1$ qubit, we apply none of the transformations, so that the transformation on those coordinates is effectively the identity. Finally, if the first $n - 2$ qubits are set but the $n - 1$ qubit is not, we apply $W^*W = I$.

Applying this construction recursively, we see that we can reduce the operation of a general V with any number of control bits to quantum gate operations acting on one qubit with only a single control qubit. The final reduction shows that we can reduce a general V with a single control bit to Toffoli gates (with single control bits) and elementary unitary operations (with no control bits).

To see this, consider the operation depicted in Figure 7a, where V is a general single qubit operation. Because V is a 2×2 unitary, we can write it as

$$V = \begin{bmatrix} e^{i\delta} & 0 \\ 0 & e^{i\delta} \end{bmatrix} \begin{bmatrix} e^{-i\alpha/2} & 0 \\ 0 & e^{i\alpha/2} \end{bmatrix} \times \begin{bmatrix} \cos\theta/2 & \sin\theta/2 \\ -\sin\theta/2 & \cos\theta/2 \end{bmatrix} \begin{bmatrix} e^{-i\beta/2} & 0 \\ 0 & e^{i\beta/2} \end{bmatrix}$$

Introduce matrices A, B, and C, defined as

$$A = \begin{bmatrix} e^{-i\alpha/2} & 0 \\ 0 & e^{i\alpha/2} \end{bmatrix} \begin{bmatrix} \cos\theta/4 & \sin\theta/4 \\ -\sin\theta/4 & \cos\theta/4 \end{bmatrix}$$

$$B = \begin{bmatrix} \cos-\theta/4 & \sin-\theta/4 \\ -\sin-\theta/4 & \cos-\theta/4 \end{bmatrix} \begin{bmatrix} e^{i(\alpha+\beta)/4} & 0 \\ 0 & e^{-i(\alpha+\beta)/4} \end{bmatrix}$$

$$C = \begin{bmatrix} e^{i(\alpha-\beta)/4} & 0 \\ 0 & e^{-i(\alpha-\beta)/4} \end{bmatrix}$$

Then $ABC = I$, while at the same time

$$ATBTC = \begin{bmatrix} e^{-i\alpha/2} & 0 \\ 0 & e^{i\alpha/2} \end{bmatrix} \times \begin{bmatrix} \cos\theta/2 & \sin\theta/2 \\ -\sin\theta/2 & \cos\theta/2 \end{bmatrix} \begin{bmatrix} e^{-i\beta/2} & 0 \\ 0 & e^{i\beta/2} \end{bmatrix}$$

where T is the Toffoli gate. Finally, letting

Table 1. The sequence of transformations that *i* and *j* undergo as a result of elementary operations.

i	*j*
011100	110010
011100	010010
011100	011010
011100	011110
111100	111110
111101	111111
Transformation on the *k*th qubit by *V* using all other qubits as control	
111100	111110
011100	011110
011100	011010
011100	010010
011100	110010

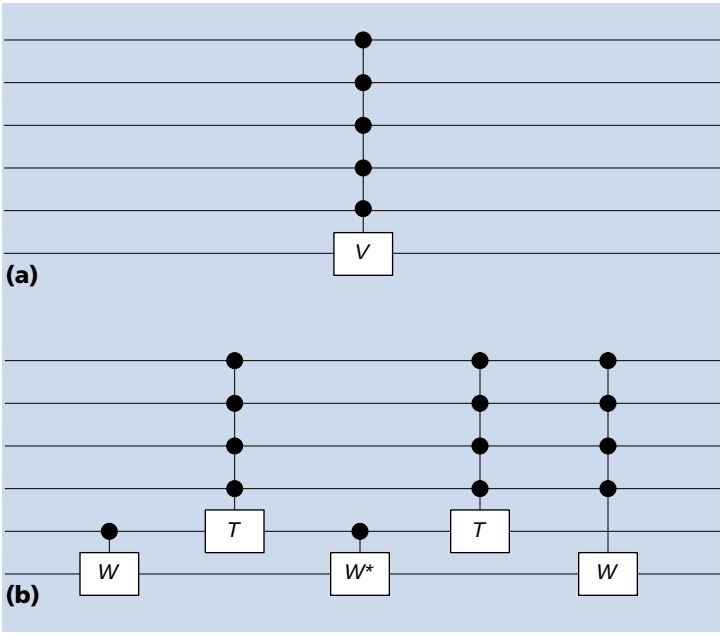

Figure 6. Elementary qubit operations with a multiple control bit can be reduced to Toffoli gates with multiple control bits and elementary gates with single control bits.

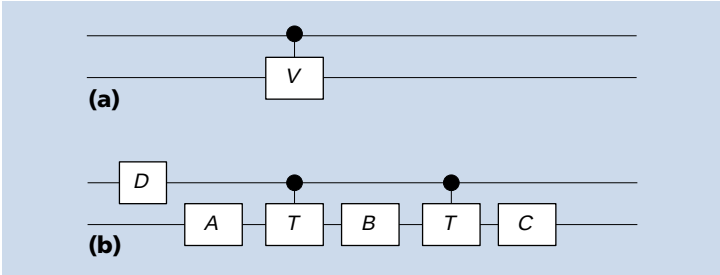

Figure 7. An elementary quantum gate with one control qubit, as depicted in Figure 7a, can be implemented as a sequence of elementary quantum gates with no control bits and Toffoli gates with only one control qubit, as depicted in Figure 7b.

$$D = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\delta} \end{bmatrix},$$

you can see by direct multiplication that Figure 7a has the same operation as Figure 7b.

This series of reductions shows that a general quantum computation can be implemented using two extremely simple quantum operations: Toffoli gates with single control qubits and elementary quantum gates operating on single qubits, using no control qubits.

The functional representation of a quantum computation is intimately related to unitary matrices. It has previously been shown that general quantum computations can be reduced to simple, elementary quantum gates acting on single qubits with additional control provided by other qubits.

In this article, I used classical ideas from numerical linear algebra to simplify this reduction and make the presentation self-contained. Several tantalizing questions remain to be explored. To implement a general quantum computation on a register with k qubits, the known reductions including the one presented in this article, use an exponential (in k) number of elementary quantum devices. Can this exponential number be reduced? Can it be reduced for a special but large class of interesting quantum operations? With respect to the physics and engineering of quantum computers, how can these elementary quantum gates be built reliably and in large quantities? Mastering quantum computation requires an intellectual investment, but the paradigm opens up exciting new avenues to basic research and engineering design, making that investment worthwhile.

Acknowledgments

I would like to thank Daniel Bilar and the anonymous reviewers for suggesting significant improvements.

References

1. C.P. Williams and S.H. Clearwater, *Explorations in Quantum Computing*, Springer-Verlag, New York, 1998.
2. P.M.B. Vitanyi, "Physics and the New Computation," *Proc. 20th Int'l Symp./Mathematical Foundations of Computer Science, MFCS '95, Lecture Notes in Computer Science*, vol. 969, Springer-Verlag, Heidelberg, Germany, 1995, pp. 106–128.
3. R.H. Katz, *Contemporary Logic Design*, Addison-Wesley, Reading, Mass., 1993.
4. A. Barenco et al., "Elementary Gates for Quantum Computation," *Physical Rev. A*, vol. 52, 1995, pp. 3457–3467.
5. D. Deutsch, A. Barenco, and A. Ekert, "Universality in Quantum Computation," *Proc. Royal Soc. of London A*, vol. 449, 1995, pp. 669–677.
6. A. Peres, "Reversible Logic and Quantum Computers," *Physical Rev. A*, vol. 32, no. 6, 1985, pp. 3266–3276.
7. G.H. Golub and C.F. Van Loan, *Matrix Computations*, Johns Hopkins Press, Baltimore, 1996.

George Cybenko is the Dorothy and Walter Gramm Professor of Engineering at Dartmouth College and the former editor in chief of *Computing in Science & Engineering*. His current research interests include quantum computing and computer security. He received a BSc from the University of Toronto and a PhD from Princeton University, both in mathematics. He is a fellow of the IEEE. Contact him at Dartmouth College, Hanover, NH, 03755; gvc@dartmouth.edu.