Quantum Algorithms
Lecture 12
Bases for quantum circuits II

Zhejiang University

Norm notation

Finite basis – challenges

We now pass to finite bases. In this case it is only possible to obtain an approximate representation of operators as products of basis elements. In order to define the approximate realization, we need a norm on the operator space.

Euclidean norm

On the space of vectors there is the Euclidean norm $\||\xi\rangle\| = \sqrt{\langle\xi|\xi\rangle}$. By the definition of a norm, it satisfies the following conditions:

$$\begin{aligned}
\||\xi\rangle\| &\begin{cases} = 0 & \text{if } |\xi\rangle = 0, \\ > 0 & \text{if } |\xi\rangle \neq 0, \end{cases} \\
\||\xi\rangle + |\eta\rangle\| &\leq \||\xi\rangle\| + \||\eta\rangle\|, \\
\|c|\xi\rangle\| &= |c|\,\||\xi\rangle\|.
\end{aligned}$$

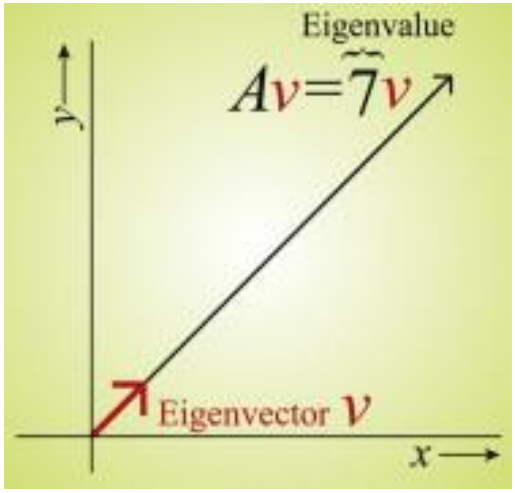
If $|\xi\rangle = a|0\rangle + b|1\rangle$, then $\langle\xi|\xi\rangle = a^*a + b^*b$

Eigenvector and eigenvalue

In linear algebra, an eigenvector or characteristic vector of a linear transformation is a nonzero vector that changes at most by a scalar factor when that linear transformation is applied to it. The corresponding eigenvalue, often denoted by λ, is the factor by which the eigenvector is scaled.

Eigenvector and eigenvalue

Geometrically, an eigenvector, corresponding to a real nonzero eigenvalue, points in a direction in which it is stretched by the transformation and the eigenvalue is the factor by which it is stretched. If the eigenvalue is negative, the direction is reversed. Loosely speaking, in a multidimensional vector space, the eigenvector is not rotated.



Eigenvector and eigenvalue

Letting $\mathbf{A}$ be a $k \times k$ square matrix

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix}$$

with eigenvalue λ, then the corresponding eigenvectors satisfy

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix},$$

Supremum

Formally, the supremum $sup_{x \in S} x$ for S a (nonempty) subset of the affinely extended real numbers $\bar{R} = R \cup \{\pm\infty\}$ is the smallest value y in $\bar{R}$ such that for all x in S we have $x \leq y$. Using this definition, $sup_{x \in S} x$ always exists and, in particular, $supR = \infty$.

Supremum

Can be also considered as unique smallest upper bound.

Example:

$$A = \left\{\frac{1}{2}, \frac{3}{4}, \frac{7}{8}, \dots\right\} = \{1 - 2^{-n} \mid n \in \mathbb{N}, n > 0\}$$

Here $\sup(a) = 1$.

Norm of an operator

The norm of an operator X (the so-called operator norm; in general, there are others) is

$$\|X\| = sup_{|\xi\rangle \neq 0} \frac{\|X|\xi\rangle\|}{\||\xi\rangle\|}$$

We note that $\|X\|^2$ is the largest eigenvalue of the operator $X^\dagger X$.

Norm properties

$$\|X\| = sup_{|\xi\rangle \neq 0} \frac{\|X|\xi\rangle\|}{\||\xi\rangle\|}$$

$$\|XY\| \leq \|X\| \, \|Y\|,$$

$$\|X^\dagger\| = \|X\|,$$

$$\|X \otimes Y\| = \|X\| \, \|Y\|,$$

$$\|U\| = 1 \qquad \text{if } U \text{ is unitary.}$$

$$\||\xi\rangle\| \begin{cases} = 0 & \text{if } |\xi\rangle = 0, \\ > 0 & \text{if } |\xi\rangle \neq 0, \end{cases}$$

$$\||\xi\rangle + |\eta\rangle\| \leq \||\xi\rangle\| + \||\eta\rangle\|,$$

$$\|c|\xi\rangle\| = |c| \, \||\xi\rangle\|.$$

Operator approximation

Approximate realization

If the operator in question is U, then its approximate realization will be denoted by $\widetilde{U}$.

The operator $\widetilde{U}$ approximates the operator U with precision δ if

$$\left\| \widetilde{U} - U \right\| \leq \delta$$

Corollaries

$$\left\|\widetilde{U} - U\right\| \leq \delta$$

This definition has two noteworthy corollaries. First, if $\widetilde{U}$ approximates U with precision δ, then $\widetilde{U}^{-1}$ approximates U^{-1} with the same precision δ. Indeed, if we multiply the expression $\widetilde{U} - U$ by $\widetilde{U}^{-1}$ on the left and by U^{-1} on the right, the norm does not increase. Thus we obtain a corollary of the inequality:

$$\left\|\widetilde{U}^{-1} - U^{-1}\right\| \leq \delta$$

Corollaries

$$\left\|\widetilde{U} - U\right\| \leq \delta$$

The second property is as follows. Consider the product of several operators, $U = U_L \cdots U_2 U_1$. If each U_k has an approximation $\widetilde{U_k}$ with precision δ_k, then the product of these approximations, $\widetilde{U} = \widetilde{U_L} \cdots \widetilde{U_2}\widetilde{U_1}$, approximates U with precision $\sum \delta_k$ (i.e., errors accumulate linearly):

$$\left\|\widetilde{U_L} \cdots \widetilde{U_2}\widetilde{U_1} - U_L \cdots U_2 U_1\right\| \leq \sum_j \delta_j$$

Remark

$$\left\|\widetilde{U} - U\right\| \leq \delta$$

$$\left\|\widetilde{U_L} \cdots \widetilde{U_2}\widetilde{U_1} - U_L \cdots U_2 U_1\right\| \leq \sum_j \delta_j$$

We have used the fact that the norm of a unitary operator is 1. (With nonunitary operators, the approximation errors could accumulate much faster, e.g., exponentially.)

Example with 2 operators

$$
\begin{aligned}
\left\|\tilde{U}_2\tilde{U}_1 - U_2U_1\right\| &= \left\|\tilde{U}_2(\tilde{U}_1 - U_1) + (\tilde{U}_2 - U_2)U_1\right\| \\
&\leq \left\|\tilde{U}_2(\tilde{U}_1 - U_1)\right\| + \left\|(\tilde{U}_2 - U_2)U_1\right\| \\
&\leq \left\|\tilde{U}_2\right\| \left\|\tilde{U}_1 - U_1\right\| + \left\|\tilde{U}_2 - U_2\right\| \left\|U_1\right\| \\
&= \left\|\tilde{U}_1 - U_1\right\| + \left\|\tilde{U}_2 - U_2\right\|.
\end{aligned}
$$

Error accumulation

Every model that aims at solving computational problems by real physical processes, has to be scrutinized for stability to approximation errors. (In real life the parameters of any physical process can be given only with certain precision.) In particular, computation with exponential error accumulation is almost definitely useless from the practical point of view.

Approximation with ancillas

The operator U: $B^{\otimes n} \to B^{\otimes n}$ is approximated by the operator $\widetilde{U}$: $B^{\otimes N} \to B^{\otimes N}$ with precision δ using ancillas if, for arbitrary $|\xi\rangle$ in $B^{\otimes n}$, the following inequality is satisfied:

$$\left\| \widetilde{U}(|\xi\rangle \otimes |0^{N-n}\rangle - U|\xi\rangle \otimes |0^{N-n}\rangle) \right\| \leq \delta \||\xi\rangle\|$$

Approximation with ancillas

We can formulate this definition in another way. Let us introduce a linear map V: $B^{\otimes n} \to B^{\otimes N}$ which acts by the rule V: $|\xi\rangle \to |\xi\rangle \otimes |0^{N-n}\rangle$. The map V is not unitary, but isometric, i.e., $V^{\dagger}V = I_{B\otimes n}$. The condition from the last definition may be written as

$$\left\| \widetilde{U}V - VU \right\| \leq \delta$$

The basic properties of approximation remain true for approximation using ancillas.

Basis of operators

Choosing basis

What bases allow the realization of an arbitrary unitary operator with arbitrary precision? What is the size of the circuit that is needed to achieve a given precision δ? How to construct this circuit efficiently? Unfortunately, we cannot give a universal answer to these questions.

Standard basis

In constructing quantum algorithms, we will use the following (widely adopted) standard basis.

The standard basis:

$Q = \{H, K, K^{-1}, \Lambda(\sigma^x), \Lambda^2(\sigma^x)\}$, where

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad K = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

K-gate currently is widely called as the S-gate.

Realization over standard basis

Any unitary operator U on a fixed number of qubits can be realized with precision δ by a $poly(\log(1/\delta))$ -size, $poly(\log\log(1/\delta))$ -depth circuit over the standard basis, using ancillas. There is a polynomial algorithm that constructs this circuit on the description of U.

$$Q = \{H, K, K^{-1}, \Lambda(\sigma^x), \Lambda^2(\sigma^x)\}$$

Complete basis

Let A be a gate set that is closed under inversion. We call A a complete basis (or a universal gate set) if the applications of its elements generate a dense subgroup in the group $U(B^{\otimes k})/U(1)$ for some $k \geq 2$. (Here $U(1)$ corresponds to multiplication by phase factors.)

Remark

The phase factors are unimportant from the physical point of view, as well as for the definition of quantum computation that will be given after two lectures.

Suppose that a basis A is closed under inversion and allows the realization of any one-qubit operator up to a phase factor (e.g., $A = SU(2)$). The multiplication by a phase factor can be realized over A using one ancilla.

Remark

Why don’t we accept ancillas in the definition of a complete basis? Indeed, it seems more natural to call a basis A complete if any unitary operator U can be realized with an arbitrary precision δ by a quantum circuit over this basis, using ancillas. Unfortunately, with this definition it is not clear how to estimate the size of the circuit in question. On the contrary, mentioned definition provides a rather general way of obtaining such an estimate. It is not known whether the two definitions of a complete basis are equivalent.

Remark – encoded qubits

There is yet another definition of a complete basis, which is based on an even more general notion of realization of a unitary operator than the realization using ancillas. A basis is called complete if it can effect an arbitrary unitary operator on "encoded qubits" with any given precision (we will discuss quantum codes (Chapter 15) in June 2022).

Remark – encoded qubits

The idea is that the quantum state of each qubit is represented by a state of several qubits; it is even permitted to have multiple representations of the same state. This situation is characterized by an isometric map $V: B^{\otimes F} \to B^{\otimes k}$, in which case we say that a single logical qubit is represented by k physical qubits (the space F corresponds to the nonuniqueness of the representation). The gates of the basis act on physical qubits, whereas the operator we want to realize acts on logical qubits.

General model

In such a general model, it is again possible to estimate the size of the circuit that is needed to achieve the given precision δ. Moreover, the gates of the basis can be specified with a constant precision δ_0, yet arbitrarily accurate realization is possible. This fundamental result is called the threshold theorem for fault-tolerant quantum computation.

Quantum threshold theorem

In quantum computing, the quantum threshold theorem (or quantum fault-tolerance theorem) states that a quantum computer with a physical error rate below a certain threshold can, through application of quantum error correction schemes, suppress the logical error rate to arbitrarily low levels.

Quantum threshold theorem

Threshold theorem for quantum computation: A quantum circuit on n qubits and containing $p(n)$ gates may be simulated with probability of error at most ε using

$$O(\log^{c}(p(n)/\varepsilon)p(n))$$

gates (for some constant c) on hardware whose components fail with probability at most p, provided p is below some constant threshold, $p < p_{th}$, and given reasonable assumptions about the noise in the underlying hardware.

Quantum threshold theorem

Current estimates put the threshold for the surface code on the order of 1%, though estimates range widely and are difficult to calculate due to the exponential difficulty of simulating large quantum systems. At a 0.1% probability of a depolarizing error, the surface code would require approximately 1,000-10,000 physical qubits per logical data qubit, though more pathological error types could change this figure drastically.

Theorem

The standard basis:

$$Q = \{H, K, K^{-1}, \Lambda(\sigma^x), \Lambda^2(\sigma^x)\}$$

is complete.

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad K = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

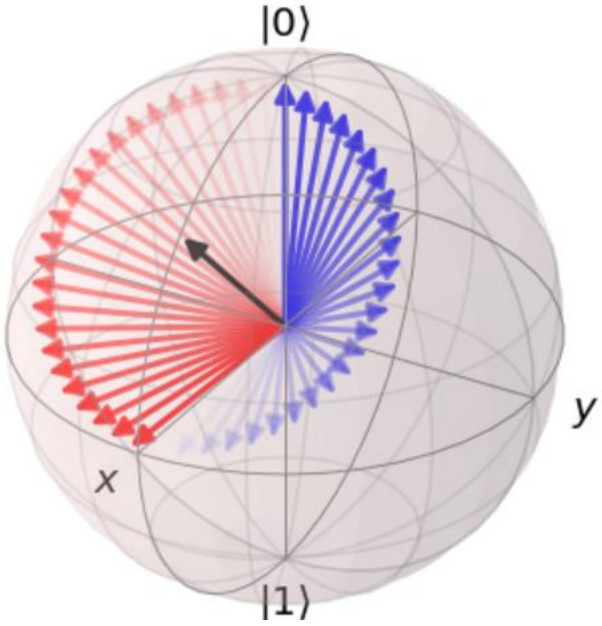
Basis Q without Toffoli

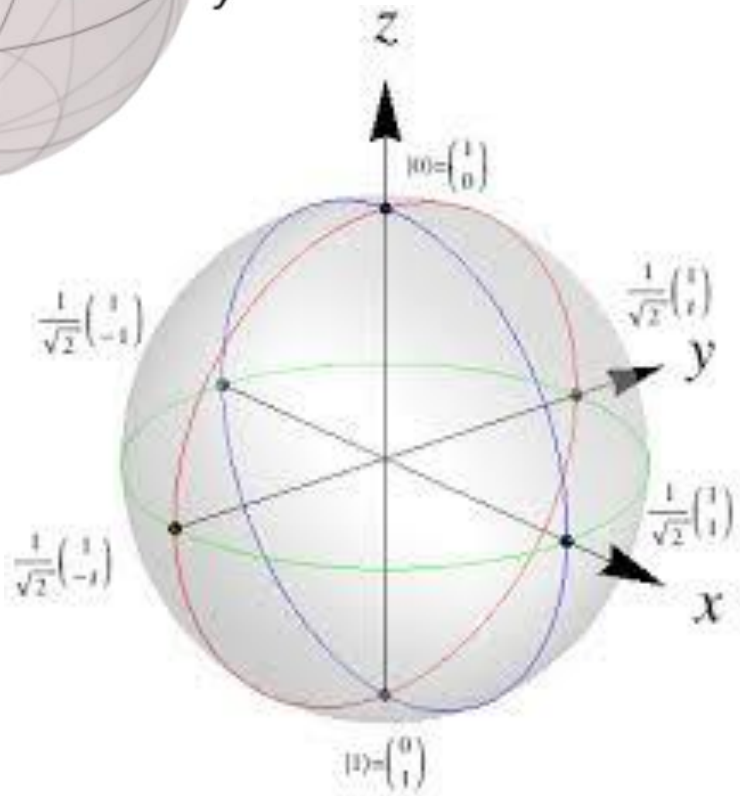
If we remove the Toffoli gate from the basis Q, it ceases to be complete. However, many important computations can be done even with such a reduced basis. In particular, as will be evident later, error-correcting circuits for quantum codes can be realized without the Toffoli gate.

One-qubit gates for basis

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$





Another complete basis

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Solovay–Kitaev theorem

Introduction

An arbitrary unitary operation U may be implemented on a quantum computer using a circuit consisting of single qubit and controlled-NOT gates. Such universality results are important because they ensure the equivalence of apparently different models of quantum computation. For example, the universality results ensure that a quantum computer programmer may design quantum circuits containing gates which have four input and output qubits, confident that such gates can be simulated by a constant number of controlled-NOT and single qubit unitary gates.

Introduction

An unsatisfactory aspect of the universality of controlled-NOT and single qubit unitary gates is that the single qubit gates form a continuum, while the methods for fault-tolerant quantum computation work only for a discrete set of gates. Fortunately, any single qubit gate may be approximated to arbitrary accuracy using a finite set of gates, such as the controlled-NOT gate, Hadamard gate H, phase gate S, and $\pi/8$ gate. The heuristic argument states that approximating the chosen single qubit gate to an accuracy ϵ required only $\Theta(1/\epsilon)$ gates chosen from the finite set. Furthermore, controlled-NOT, Hadamard, phase and $\pi/8$ gates may be implemented in a fault-tolerant manner.

Theorem

In quantum information and computation, the Solovay–Kitaev theorem says, roughly, that if a set of single-qubit quantum gates generates a dense subset of $SU(2)$ then that set is guaranteed to fill $SU(2)$ quickly, which means any desired gate can be approximated by a fairly short sequence of gates from the generating set.

Theorem

The Solovay–Kitaev theorem shows that for any gate U on a single qubit, and given any $\epsilon > 0$, it is possible to approximate U to a precision ϵ using $\Theta(log^c(1/(\epsilon))$ gates from a fixed finite set, where c is a small constant approximately equal to 2.

Efficient approximation over a complete basis

Introduction

How can one estimate the complexity of realizing a unitary operator U over a complete basis A with a given precision δ? How to construct the corresponding circuit efficiently? This questions arise if we want to simulate circuits over another basis C by circuits over A. We would like to prove that such simulation does not increase the size of the circuit too much. In this regard, we may assume that $U \in C$ is fixed, while δ tends to zero.

Operator with some precision

Let $U: B^{\otimes n} \to B^{\otimes n}$ be an arbitrary unitary operator. It can be represented by a matrix with complex entries, where each entry is a pair of real numbers, and each number is an infinite sequence of binary digits. We set the question of computing these digits aside.

Operator with some precision

We assume that each matrix entry is specified with a suitable precision, namely, $\delta/2^{n+1}$. In this case the overall error in U, measured by the operator norm, does not exceed $\delta/2$. (Taking this input error into account, the algorithm itself should work with precision $\delta/2$, but we will rather ignore such details.)

Constructing unitary

The problem can be divided into two parts. First, we realize U over the infinite basis A_0 that consists of all one-qubit and two-qubit gates. Second, we approximate each gate V of the resulting circuit C_0 by a circuit C over the basis A.

Constructing unitary

The first part is mostly done in the proof of Theorem 8.1; we just need to add some details. By examining the proof, we find that the circuit C_0 has size $L_0 = \exp(O(n))$.

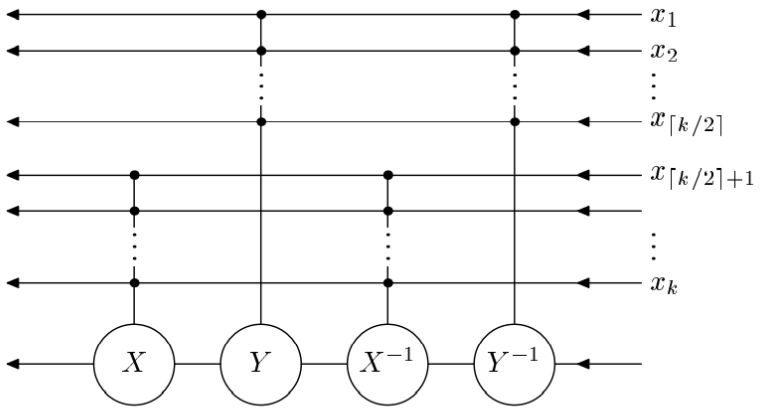


Fig. 8.4. Implementation of the operator $\Lambda^k(i\sigma^x)$ without ancillas.

$$\begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots \\ \vdots & \ddots & 0 & \cdots & \cdots & \cdots \\ 0 & \cdots & 1 & 0 & \cdots & \cdots \\ 0 & \cdots & \cdots & \begin{pmatrix} a & b \\ c & d \end{pmatrix} & 0 & \cdots \\ 0 & \cdots & \cdots & \cdots & 1 & 0 \\ \cdots & \cdots & \cdots & \cdots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & 1 \end{pmatrix}, \quad where \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathbf{U}(2)$$

Constructing unitary

If we want to represent U with precision δ, we need to compute all gates of the circuit with precision $\delta' = \delta/L = \exp(-O(n))\delta$, which amounts to computing the entries of the corresponding matrices with precision $\delta'' = \delta'/2^n = \exp(-O(n))\delta$.

Constructing unitary

The realization of U over the infinite basis A_0 that consists of all one-qubit and two-qubit gates, can be done in time $T = \exp(O(n)) \cdot poly(\log(1/\delta))$. The presence of the exponential factor should not bother us, since in the practical application U is fixed, and so is n. Thus the first part is finished and we proceed to the second part.

Thank you for your attention!