J. FAC. SCI., SHINSHU UNIVERSITY, Vol. 40, pp.41-48 2005

Precise probability that Grover's quantum search finds a solution

By Yoko IKUMI* and Yoshiki OTOBE**

**System Development Division, Hitachi Information Systems, Ltd.*
4-9-1 Futago, Takatsu-ku, Kawasaki 213-8521, Japan
***Department of Mathematical Science, Faculty of Science, Shinshu University.*
3-1-1 Asahi, Matsumoto 390-8621, Japan

(Received November 24, 2005)

Abstract

We will give a precise estimate for Grover's extended quantum search algorithm. It is shown by Grover himself that his search mechanism can find a solution with $O(\sqrt{N})$ steps under a hypothesis that the coefficients of a unitary matrix is sufficiently small. We, however, give a precise expression of the probability that the algorithm reaches a solution for any unitary matrix. Finally, we will show the behavior of the probability by providing some graphs.

1 Introduction

The theory of quantum computation has been well developed in the last decade. It is now well understood that there are at least two effective algorithms that quantum computers can solve. One is Shor's prime factorization[9], which can break the RSA public-key cryptosystem since it requires only $O(n^3)$ operations (exponentially faster than classical computers can do) to factorize n-bit integers. The other is Grover's search algorithm[3, 4, 5], which can find a solution from N elements by $O(\sqrt{N})$ operations, while classical computers need $O(N)$ operations.

Here, the terminology "classical computers" denotes the computer system widely used at the present time in the world. They have n-bit registers which are denoted by B^n, where $B=\{0, 1\}$. An element of B^n is sometimes interpreted as an integer by the following manner that $(j_1, j_2, \ldots, j_n)\in B^n$ is displayed on the computer screen as $2^{n-1}j_1 + 2^{n-2}j_2 + \cdots + 2^0 j_n$. We may sometimes interpret the element of B^n as a "signed" integer or a "floating point" real number. We, however, emphasize that these all are

2000 *Mathematics Subject Classification.* Primary 81P68; Secondary 68W40.
Key words and phrases. Grover's algorithm, quantum search.

just a problem of interpretation, and classical computers are generally defined as maps $B^n \to B^n$.

Quantum computers, defined in the next section, are a theoretical model which quantizes B to $\mathbf{C}^2$, of which idea was originally introduced by Feynman[2] and Deutsch [1]. As mentioned above, there are some algorithms that quantum computers can solve essentially faster than classical ones. One reason, from a theoretical point of view, why quantum computers are much faster than classical ones is that they can compute quantum Fourier transforms very quickly. However, as far as our knowledge, there cannot be found any literature that explains it as mathematicians easily read. Hence we will give a brief sketch of the quantum Fourier transform in section 3. Finally we shall analyze Grover's quantum search algorithm in section 4. The present paper is a part of a master's thesis of the first author[7].

2 Quantum Computers

Quantum computers are the systems that have n-quantum bit (qubit) registers, which are quantum mechanical analogue of B^n. First, we define a qubit. However, the n-qubit register is not just an n-direct product of copies of a qubit.

Let us prepare a two-dimensional vector space $\mathbf{Q}$ over $\mathbf{C}$. Then an element $q \in \mathbf{Q}$ have a form that $q = q_0 e_0 + q_1 e_1$, where $\{e_1, e_2\}$ is a fixed orthonormal base of $\mathbf{Q}$ and $q_i \in \mathbf{C}$ $(i=0, 1)$. The basis e_0 and e_1 are sometimes denoted by Dirac notations $|0\rangle$and $|1\rangle$, respectively, and called computational basis states. Since we fixed the base, we can identify $q \in \mathbf{Q}$ by $(q_0, q_1) \in \mathbf{C}^2$, and $\mathbf{Q}$ by $\mathbf{C}^2$. In this manner, we may, as usual, take $e_0 = (1, 0)^t$ and $e_1 = (0, 1)^t$, and q is called a superposition (linear combination) of computational basis states. A state of a qubit is defined as an element of $\mathbf{Q}$.

We assume that, if $q = e_i$, $(i=0, 1)$, we obtain the value $i \in B$ after an observation of the qubit, respectively. Moreover, if the state of a qubit is $q = (q_0, q_1)^t \in \mathbf{C}^2$, we obtain a result 0 (resp. 1) with probability $\frac{|q_0|^2}{|q_0|^2 + |q_1|^2}$ (resp. $\frac{|q_1|^2}{|q_0|^2 + |q_1|^2}$). That is to say, the observation is a random variable $\mathbf{Q} \to B$. Operations allowed on a qubit need to be unitary from the hypothesis of quantum mechanics.

It is now natural that an n-qubit is expressed by an element of $\mathbf{Q}^n$. However, the operations $\tilde{U} : \mathbf{Q}^n \to \mathbf{Q}^n$ must be unitary on each component of $\mathbf{Q}^n$, that is, $\tilde{U}$ is an n-linear map. Hence we are lead to a definition that n-qubit state is an element of $\mathbf{Q}^{\otimes n} = \overbrace{\mathbf{Q} \otimes \mathbf{Q} \otimes \cdots \otimes \mathbf{Q}}^{n}$ rather than $\mathbf{Q}^n$, of which base forms $\{e_{j_1, j_2, \ldots, j_n} = e_{j_1} \otimes e_{j_2} \otimes \cdots \otimes e_{jn},\ j_i \in B,\ i = 1, 2, \ldots, n\}$. The base $e_{j_1, j_2, \ldots, j_n}$ is sometimes denoted by $|2^{n-1} j_1 + 2^{n-2} j_2 + \cdots + 2^0 j_n\rangle$ or $|j_1 j_2 \cdots j_n\rangle$ (binary notation) in the Dirac notation. Therefore we can say that a quantum computer is a unitary operator on 2^n-dimensional complex linear space.

3 Quantum Fourier Transform

A key property to explain why quantum computers can solve problems much faster than classical ones is based on the quantum Fourier transform, which is essentially same with the usual discrete Fourier transform. Although the quantum Fourier transform itself does not explicitly appear in the present paper, we believe it worthy to give a brief review here as was mentioned in section 1. A systematic treatment of this topic can be found in [8, Chapter 5].

Let us consider a complex vector space $\mathbf{C}^N$ with the usual complex inner product $\langle\cdot,\cdot\rangle$. We fix two orthonormal basis $\{e^k\,;\ k=0, 1, \ldots, N-1\}$ and $\{f^k\,;\ k=0, 1, \ldots, N-1\}$ defined by $e^k_j=\delta_{k,j}$ and $f^k_j=\frac{1}{\sqrt{N}}\exp\left\{-\frac{2\pi\sqrt{-1}kj}{N}\right\}$, namely,

$$e^k=\begin{bmatrix}0\\ \vdots\\ 0\\ 1\\ 0\\ \vdots\\ 0\end{bmatrix},\ \text{and}\ f^k=\frac{1}{\sqrt{N}}\begin{bmatrix}\exp\left\{-\frac{2\pi\sqrt{-1}k\cdot 0}{N}\right\}\\ \exp\left\{-\frac{2\pi\sqrt{-1}k\cdot 1}{N}\right\}\\ \exp\left\{-\frac{2\pi\sqrt{-1}k\cdot 2}{N}\right\}\\ \vdots\\ \exp\left\{-\frac{2\pi\sqrt{-1}k\cdot(N-1)}{N}\right\}\end{bmatrix}. \tag{1}$$

Definition 3.1. For an element $x=\sum_{k=0}^{N-1}x_ke^k$ of $\mathbf{C}^N$, let $c\equiv(c_0, c_1, \ldots, c_{N-1})^t\in\mathbf{C}^N$ be a coordinate of x with respect to $\{f^k\}$, that is,

$$c_k:=\langle x, f^k\rangle=\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}x_je^{\frac{2\pi\sqrt{-1}\,kj}{N}}.$$

We call a map $\mathbf{C}^N\to\mathbf{C}^N$ defined by $x\mapsto c$ the discrete Fourier transform denoted by $c=\mathfrak{F}x$.

Remark 3.1. The discrete Fourier transform is a linear map $\mathbf{C}^N\to\mathbf{C}^N$ and its matrix representation F is given by

$$F=\frac{1}{\sqrt{N}}\begin{bmatrix} e^{\frac{2\pi\sqrt{-1}\,0\cdot 0}{N}} & e^{\frac{2\pi\sqrt{-1}\,0\cdot 1}{N}} & \cdots & e^{\frac{2\pi\sqrt{-1}\,0\cdot(N-1)}{N}}\\ e^{\frac{2\pi\sqrt{-1}\,1\cdot 1}{N}} & e^{\frac{2\pi\sqrt{-1}\,1\cdot 1}{N}} & \cdots & e^{\frac{2\pi\sqrt{-1}\,1\cdot(N-1)}{N}}\\ e^{\frac{2\pi\sqrt{-1}\,2\cdot 1}{N}} & e^{\frac{2\pi\sqrt{-1}\,2\cdot 1}{N}} & \cdots & e^{\frac{2\pi\sqrt{-1}\,2\cdot(N-1)}{N}}\\ \vdots & \vdots & \ddots & \vdots\\ e^{\frac{2\pi\sqrt{-1}\,(N-1)\cdot 0}{N}} & e^{\frac{2\pi\sqrt{-1}\,(N-1)\cdot 1}{N}} & \cdots & e^{\frac{2\pi\sqrt{-1}\,(N-1)(N-1)}{N}}\end{bmatrix}.$$

Now let us regard the $\mathbf{C}^N$, $N=2^n$, as a tensor product $(\mathbf{C}^2)^{\otimes n}$. Then, using a base $\{e_0\equiv(1, 0)^t, e_1\equiv(0, 1)^t\}$ of $\mathbf{C}^2$, the base $\{e^j, j=0, 1, \ldots, N-1\}$ of $(\mathbf{C}^2)^{\otimes n}$ has the following representation: If $j=\sum_{l=1}^{n}2^{n-j}j_l$, $j_l\in B$, $e^j=e_{j_1}\otimes e_{j_2}\otimes\cdots\otimes e_{j_n}$.

Definition 3.2. The discrete Fourier transform on $\mathbf{C}^{2^n}$ is called the quantum Fourier transform when $\mathbf{C}^{2^n}$ is identified with $\mathbf{C}^{\otimes n}$.

In this setting, we have the following proposition.

Proposition 3.1 ([8]). *For the base* $e^j = e_{j_1} \otimes e_{j_2} \otimes \cdots \otimes e_{j_n}$ of $\mathbf{C}^{\otimes n}$, *we have*

$$\mathfrak{F}e^j = \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^{n} \begin{pmatrix} 1 \\ e^{\frac{2\pi\sqrt{-1}\,j}{2^l}} \end{pmatrix}$$

Proof. It is clear by definition that, if $k = \sum_{l=1}^{n} 2^{n-l} k_l$ and $N = 2^n$,

$$\mathfrak{F}e^j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp\left\{\frac{2\pi\sqrt{-1}\,j(2^{n-1}k_1 + 2^{n-2}k_2 + \cdots + 2^0 k_n)}{N}\right\}$$

$$(2) \qquad = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi\sqrt{-1}\,j2^{n-1}k_1}{N}} e^{\frac{2\pi\sqrt{-1}\,j2^{n-2}k_2}{N}} \cdots e^{\frac{2\pi\sqrt{-1}\,j2^0 k_n}{N}} (e_{k_1} \otimes e_{k_2} \otimes \cdots \otimes e_{k_n})$$

$$= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \bigotimes_{l=1}^{n} \left(e^{\frac{2\pi\sqrt{-1}\,jk_l 2^{n-1}}{2^n}} e_{k_l} \right).$$

However, since $\sum_{k=1}^{2^n-1} = \sum_{k_n=0}^{1} \sum_{k_{n-1}=0}^{1} \cdots \sum_{k_1=0}^{1}$, we can rewrite (2) as

$$\mathfrak{F}e^j = \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^{n} \sum_{k_l=0}^{1} e^{\frac{2\pi\sqrt{-1}\,jk_l}{2^l}} e_{k_l}. \qquad \square$$

We will introduce the following conventions to see the above theorem can be rewritten in an easier form : $(j_1 j_2 \ldots j_n)_2 := \sum_{l=1}^{n} 2^{n-l} j_l$, and $(0.\, j_1 j_2 \ldots j_n)_2 := \sum_{l=1}^{n} 2^{n-l} j_l$, which are just binary notations of j. We sometimes omit the subscript 2 if there is no confusion.

Corollary 3.2. *We have the following formula :*

$$\mathfrak{F}e^{j_1 j_2 \ldots j_n} = \frac{1}{\sqrt{2^n}} \begin{pmatrix} 1 \\ e^{2\pi\sqrt{-1}\,0.j_n} \end{pmatrix} \otimes \begin{pmatrix} 1 \\ e^{2\pi\sqrt{-1}\,0.j_{n-1}j_n} \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 \\ e^{2\pi\sqrt{-1}\,0.j_1 j_2 \ldots j_n} \end{pmatrix}.$$

4 Grover's algorithm

Grover's quantum search algorithm is to find a solution from an indexed database. The elements of the database is expressed as a tensor product of two state vectors (consisting of index and datum), and the goal is to find an index with a high probability. In other words, the algorithm distinguish knowing the solution to a search problem from being able to recognize the solution. Generally speaking, to recognize the solution is much easier than knowing it. For example, Eratosthenes' sieve is a simple search-based algorithm for prime factorization. It is very easy to check a prime number in a database is a divisor for the specified number. The detailed discussion of this topic can be found in [8].

The first algorithm for quantum search is discovered by Grover[3] in 1996, and extended later by himself[6] to be able to apply it to a multi-solution case, which is also applicable to rapid sampling obeying arbitrary probability distribution.

Let $s \in \{0, 1, \ldots, 2^n-1\}$ be a source and define e^s in a same manner with (1). We assume that we have m targets and they are given by $t_i \in \{0, 1, \ldots, 2^n-1\}$, $i=1, 2, \ldots, m$. For given e^s, we define a matrix I_s by $I-2e^s \otimes e^s$, where, to avoid the complication of the notations, $e^s \otimes e^s$ denotes a matrix defined by $e^s \cdot (e^s)^*$. Note that, in the Dirac notation, $e^s \otimes e^s = |s\rangle\langle s|$. Similarly we define I_t by $I - \sum_{i=1}^{m} e^{t_i} \otimes e^{t_i} \equiv I - \sum_{i=1}^{m} |t_i\rangle\langle t_i|$. For a fixed unitary matrix $U=(U_{ij})_{i,j=0,1,\ldots,2^n-1}$, we moreover define a matrix $Q := -I_s U^{-1} I_t U$, which we call Grover's operator.

Then, simple computations lead us to

$$\begin{cases} Qe^s = (1-4\sum_{i=1}^{m}|U_{t_i s}|^2)e^s + 2\sum_{i=1}^{m} U_{t_i s} U^{-1} e^{t_i}, \\ Q\sum_{i=1}^{m} U_{t_i s} U^{-1} e^{t_i} = -2\sum_{i=1}^{m}|U_{t_i s}|^2 e^s + \sum_{i=1}^{m} U_{t_i s} U^{-1} e^{t_i}. \end{cases}$$

We can summarize it as

$$Q\left(e^s \ \frac{1}{u}\sum_{i=1}^{m} U_{t_i s} U^{-1} e^{t_i}\right) = \left(e^s \ \frac{1}{u}\sum_{i=1}^{m} U_{t_i s} U^{-1} e^{t_i}\right)\begin{pmatrix} 1-4u^2 & -2u \\ 2u & 1 \end{pmatrix},$$

where $u := \sqrt{\sum_{i=1}^{m}|U_{t_i s}|^2}$, $(0 \leq u \leq 1)$.

Lemma 4.1. *For $\eta \in \mathbf{N}$, we have*

$$Q^\eta\left(e^s \ \frac{1}{u}\sum_{i=1}^{m} U_{t_i s} U^{-1} e^{t_i}\right) = \left(e^s \ \frac{1}{u}\sum_{i=1}^{m} U_{t_i s} U^{-1} e^{t_i}\right)A^\eta,$$

where $A \equiv A(u) := \begin{pmatrix} 1-4u^2 & -2u \\ 2u & 1 \end{pmatrix}$.

Proof. By induction. □

The goal of the searching problem is to find t_i's. That is to say, it would be ideal if we got η satisfying $Q^\eta e^s = \frac{1}{u}\sum_{i=1}^{m} U_{t_i s} U^{-1} e^{t_i}$. In general, it is impossible so that we will try to maximize the probability with which we can observe the target states (t_i's). Grover[6] showed $\eta = \pi/(4u)$ is optimal under a condition that u is sufficiently small, which is achieved if the number N of items in the database is very large, and this time $u \sim \sqrt{M/N}$. We shall, in the present paper, establish the probability to be able to observe target states without any restriction.

Lemma 4.2. *We can observe the target states with probability*

$$P(u,\eta) := \frac{|\lambda_1^\eta + \lambda_2^\eta|^2}{|(\lambda_1^\eta(u\sqrt{-1}+\sqrt{1-u^2}) + \lambda_2^\eta(-u\sqrt{-1}+\sqrt{1-u^2}))^2| + |\lambda_1^\eta + \lambda_2^\eta|^2},$$

where $\lambda_1 = 1-2u^2+2u\sqrt{u^2-1}$, *and* $\lambda_2 = 1-2u^2-2u\sqrt{u^2-1}$.

Proof. It is clear that λ_1 and λ_2 are eigenvalues of A and their corresponding eigenvectors are given by $\begin{pmatrix} 1 \\ -u-\sqrt{u^2-1} \end{pmatrix}$ and $\begin{pmatrix} 1 \\ -u+\sqrt{u^2-1} \end{pmatrix}$, respectively. Hence the lemma comes immediately. □

Theorem 4.3. *Set* $X \equiv X(u, \eta) := (\eta-1)\arccos(1-2u^2)$. *Then we have*

$$(3)\quad P(u,\eta)=\Big[(1-2u^2)\sin X+2u\sqrt{1-u^2}\,\cos X\Big]\Big/ \\ \Big[\Big(\sqrt{1-u^2}(1-4u^2)\cos X+u(4u^2-3)\sin X\Big)^2 \\ +\Big((1-2u^2)\sin X+2u\sqrt{1-u^2}\,\cos X\Big)^2\Big].$$

Proof. Put $\lambda_1^\eta=:\ a_{1,\eta}+\sqrt{-1}b_{1,\eta},\ (a_{1,\eta},\ b_{1,\eta}\in\mathbf{R})$. Then we get the following recurrence formula:

$$(4)\qquad \begin{cases} a_{1,\eta+1}=(1-2u^2)a_{1,\eta}-2u\sqrt{1-u^2}\,b_{1,\eta}, \\ b_{1,\eta+1}=2u\sqrt{1-u^2}\,a_{1,\eta}+(1-2u^2)b_{1,\eta}. \end{cases}$$

Since $\begin{pmatrix} 1-2u^2 & -2u\sqrt{1-u^2} \\ 2u\sqrt{1-u^2} & 1-2u^2 \end{pmatrix}$ is a rotation matrix, we can easily solve (4). Similar computations can be done on λ_2, and combining them ensures the conclusion. □

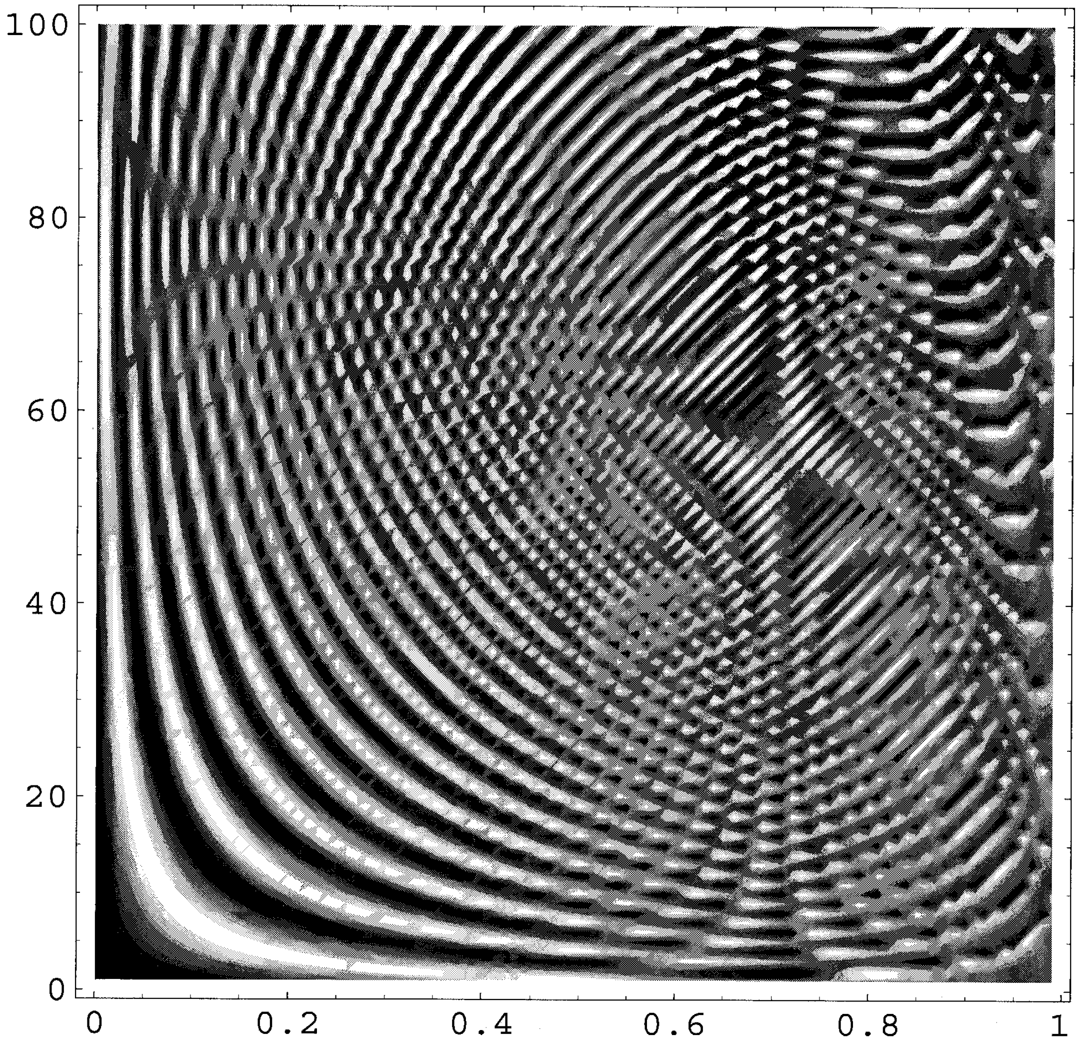
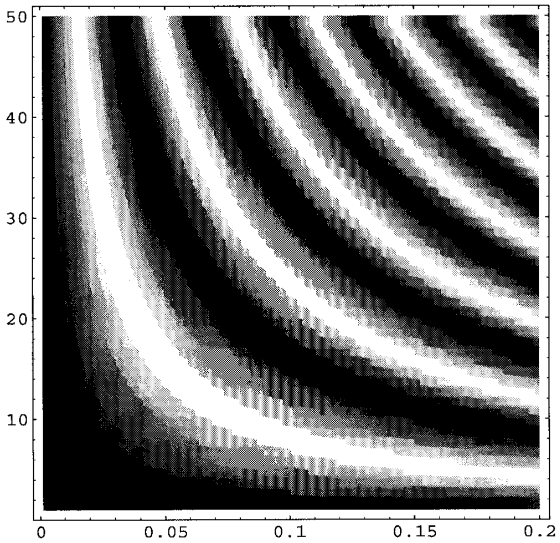


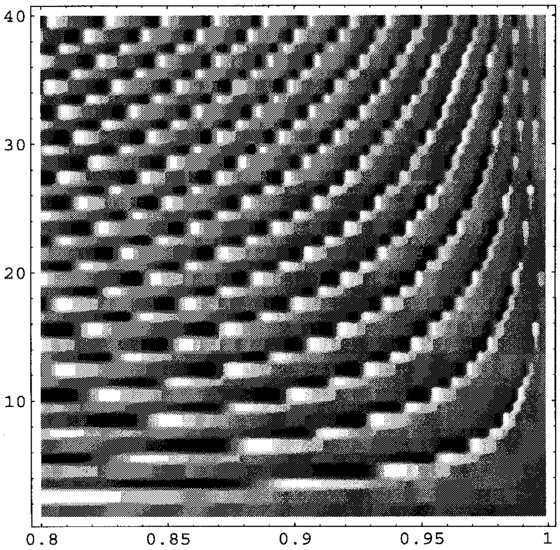
Fig. 1

Although $P(u, \eta)$ is defined on $[0, 1] \times \mathbf{N}$, it is clear that we can extend it on $[0, 1] \times \mathbf{R}_+$. We omit to show $\partial P / \partial \eta (u, \eta)$ since it is rather complicated, but it is possible to show $P(u, \eta)$ is smooth with respect to η. It seems hard to write down an explicit formula of η to maximize $P(u, \eta)$ for each fixed u. For our purpose, however, it is enough to draw reliable pictures. Finally, we conclude the paper with showing some of them.

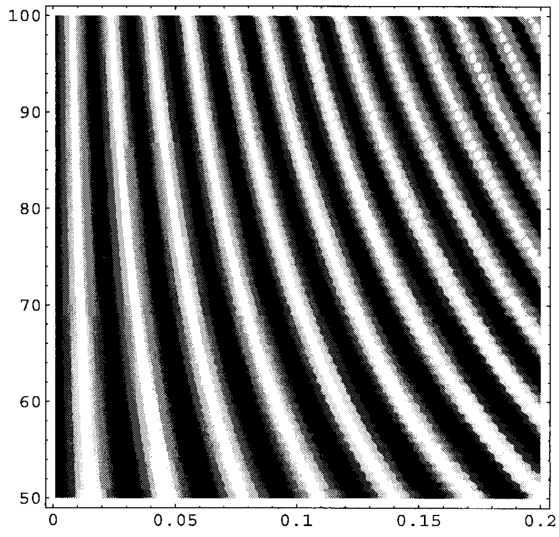
The Fig. 1 on the former page is the contour of (3) with $\eta \in \mathbf{N}$ (left axis) and $u \in (0, 1)$ (bottom axis), and the dark area means $P(u, \eta) \sim 0$ and the bright (white) area means $P(u, \eta) \sim 1$.

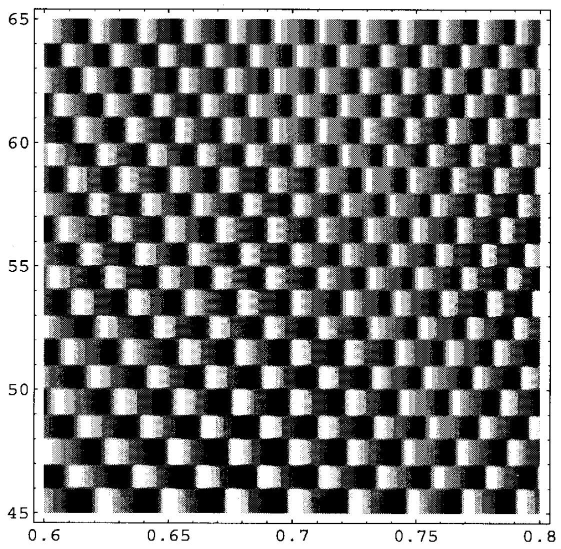
We may observe that, the bright area behaves as Grover already showed in [6] in the left-bottom corner (left), but it does not in the right-bottom corner (right).





Finally we will end the paper with showing two pictures of other regions. We may see that the Grover's algorithm is very sensitive in the sense that the bright and dark are neighbors in many places.





References

1 . D. Deutsch, *Quantum theory, the Church-Turing principle and the universal quantum computer,* Proc. R. Soc. Lond. A (1985), 400 : 97.

2 . R. P. Feynman, *Feynman lectures on computation,* Addison-Wesley, 1996.

3 . L. K. Grover, *A fast quantum mechanical algorithm for database search,* Proceedings of 28th Annual ACM Symposium on Theory of Computing (STOC), 1996, quant-ph/9605043, pp. 212-219.

4 . ____, *Quantum mechanics helps in searching for a needle in a haystack,* Phys. Rev. Letters **79** (1997), no. 2, 325-328, quant-ph/9706033.

5 . ____, *A framework for fast quantum mechanical algorithms,* Proceedings of 30th Annual ACM Symposium on Theory of Computing (STOC), 1998, quant-ph/9711043, pp. 53-62.

6 . ____, *Rapid sampling through quantum computing,* Proceedings of 32th Annual ACM Symposium on Theory of Computing (STOC), 2000, quant-ph/9912001, pp. 618-626.

7 . Yoko Ikumi, *Sampling by quantum search algorithm,* Master's thesis, Shinshu University, 2005, in Japanese.

8 . Michael A. Nielsen and Issac L. Chung, *Quantum computation and quantum information,* Cambridge, 2000.

9 . P. W. Shor, *Algorithms for quantum computation : discrete logarithms and factoring,* Proceedings of 35th Annual Symposium on Foundations of Computer Science, 1994, p. 116.