

## チュートリアル

## 量子力学と情報処理(Ⅲ)

丸山 耕司

## 1 計 算 量

量子コンピュータは、現在我々が日々使っている古典コンピュータよりも計算能力が高い、と言われる。これは同じ計算をさせたときに、量子コンピュータの方が処理速度が速いということだ。しかし、ここで言う処理速度とは、今のパソコンは10年前のものより速い、などという場合の速さのことではもちろんない。量子コンピュータの方が速いというのは、同じ計算の実行を、古典コンピュータで行うよりも短い時間で完了できる量子アルゴリズムが存在する、という意味である。そこで、アルゴリズム実行に必要な全計算時間(計算量)を評価する方法が必要になる。深入りはしないが、一般に計算量とは基本計算操作の数のことである。

古典的情報処理の場合には、簡単に言うと数えるべき基本操作とは2つのビットを足し合わせることだ(繰り上がり分も含む)。他にも定義の仕方はあるのだが、本質的には同じことであるし、またより複雑な処理もこの単純なビット演算に分解できることが知られている。2つの  $m$  ビットの整数(2進数)を足し合わせる演算を考えると<sup>†1</sup>、各桁ごとに基本ビット演算を行うのでこの計算に必要な計算量は  $m$  である。では、 $m$  ビットの整数  $k$  と  $n$  ビットの整数  $l$  をかけあわせる演算  $k \cdot l$

はどうだろうか。図1のように  $k, l$  の順に縦に並べる筆算にしてみると分かるが、この乗算処理に必要な計算量は高々  $mn$  である。

もちろん、個々のビット演算にかかる実時間は個々の計算機システムに依存するが、違いはせいぜい比例定数であり、計算量の議論では気にしない。多くの場合、与えられた問題を計算するのに必要な計算量が、入力ビット数(桁数)  $n$  のべき乗( $n^3, n^{10}$  など)なのか指数関数的( $2^2, 2^{n/2}$  など)なのかが重要なのであって、何時間かかるかなどはどうでもいいことなのである。

ここで、計算量評価を簡潔に記述するのに便利な方法を導入しておこう。 $f(n)$  および  $g(n)$  は任意の正整数  $n$  に対して非負の値をとる関数とする。ある定数  $c, n_0$  が存在して、 $n > n_0$  なるすべての  $n$  に対して  $f(n) \leq cg(n)$  が成り立つとき、 $f(n) = O(g(n))$ 、あるいは  $f = O(g)$  と表す。たとえば、 $3n^2 + 2n - 15 = O(n^2)$  である。平たく言えば、大きな  $n$  に対して、 $cg(n)$  が  $f(n)$  の上界を与える。上の  $k$  と  $l$  の乗算の例では、計算量は  $O(\log k \cdot \log l) = O(mn)$  と表されることになる。他に、不等号の向きなどによって  $\Omega$  や  $\Theta$  などの記号も定義されるが、通常の議論では上の  $O$  記号だけで十分である。

「効率のよい(efficient)」アルゴリズムとは、 $n$  ビットの入力に対して計算量が  $O(n^k)$  ( $k$  は任意の整数)として表わされるものを言い、「多項式時間(polynomial time)で実行可能であるという。逆に、「効率の悪い(inefficient)」アルゴリズムと

†1 片方の数が  $(n < m)$  ビットだったら  $m$  ビットになるように上位のビットに0を書き加えればよい。

$$\begin{array}{r}
 1011 \\
 \times 110 \\
 \hline
 10110 \\
 1011 \\
 \hline
 100010
 \end{array}$$

図1 4ビットの数  $k=1011$  と3ビットの数  $l=110$  の乗算

$l$  の2進表記中の '1' の位置に応じて,  $k$  を左側へシフトさせたものを足し合わせる。 $k$  をシフトさせたり, それを必要数コピーして記憶しておいたりする作業に要する時間は, ビット数が大きくなったときには実質的に無視できる。

は,  $O(k^n)$  のように「指数的時間(exponential time)」が必要な場合を指す。ここで,  $k$  の大小は気にしない。あくまで  $n$  が非常に大きくなった場合の振る舞いの違いを問題にするからである。

効率の悪いアルゴリズムしか知られていない「難しい」問題はゴマンとあるのだが, 量子コンピュータの威力の説明によく引き合いに出される(素)因数分解を例にとろう。もっとも単純で確実なのは, 入力  $N$  ( $\log N$  ビット) を2および3から  $\sqrt{N}$  までの奇数で順に  $N$  を割って, 割り切れるかどうかを調べていく方法だろう。この計算には  $O(\sqrt{N}) = O(2^{(\log N)/2})$  回の割り算が必要であり, 指数的時間かかることが明らかである。

こうした場合には作戦を変えて, 確率的に計算に成功するアルゴリズムを探す方が現実的だ。一度のアルゴリズム実行での計算成功確率が  $1-\epsilon$  ( $\epsilon \in (0, 1)$  は任意の実数) だとして。このアルゴリズムを  $k$  回実行して, 1回以上成功する確率は  $1-\epsilon^k$  である。因数分解のような問題の場合は, 得られた答えが正解かどうか(因数を掛け合わせて  $N$  になるかどうか)を調べるのは簡単であり(図1),  $k$  回実行するうち1度でも成功すればそこで計算を終了すればよい。さらに, 1回の計算量が多項式時間であれば,  $k$  回繰り返しても多項式時間であるから, そうしたアルゴリズムが見つければ多項式時間で実行しつつ, 成功確率を任意の精度で1に近づけることができる。しらみつぶしに計算するよりはるかに効率よく答えにたどり

つけるわけだ。

しかし, このアプローチをもってしても因数分解は手ごわく, 古典計算機の場合, 現在知られている最も効率のよいアルゴリズムでも  $O(\exp[(\log N)^{1/3}(\log \log N)^{2/3}])$  の時間を要し, とても現実的とは言えない<sup>12</sup>。冒頭で述べた短い時間で計算を完了する量子アルゴリズムとは, ほとんどすべての場合, このように成功確率  $1-\epsilon$  で正しい答えを出力する「確率的」アルゴリズムである。量子力学が本質的に確率事象を扱うことを考えるとこれはごく自然なことである。

量子情報処理では計算量をどう見積もるのだろうか。これにもいくつかの定義があるが, 上述の古典情報処理の場合との対比で考えてもっとも自然なのは, 基本操作として2つの qubit に対して行う制御を採用する流儀だろう。実際これが重要なのだが, 古典的ビットと違い量子の場合は単一の量子ビットに対する操作を含めることも多い。古典の場合は自明でない単一ビット操作にはビット反転しかないが, 量子の場合は任意の軸の周りの任意の角度分の回転操作(SU(2)の自由度に対応)が可能だからだ。このあたりの基本操作のことを「量子論理ゲート」とよぶ。したがって,  $n$  qubit に対するある量子アルゴリズムが多項式時間で実行可能であるということは, そのアルゴリズム実行には量子論理ゲートが高々  $O(n^k)$  個必要であるということである。

## 2 量子計算の基本要素

さて, これまで量子コンピュータが具体的に何をするのかを述べずに「確率  $p$  で正しい答え  $a$  が得られる」というような言い方をしてしまったが, 量子系のどこにどう答えが出てくるのか疑問に思われた方も多いと思う。実装の語題に進む前に, 以下のようなざっくりとした直感的イメージを持っておくのは悪くないだろう。既知の初期状態, たとえば  $|0\dots 0\rangle$ , からスタートした  $n$  qubit

<sup>12</sup> この極端な難しさがRSA暗号の安全性を支えている。

状態が様々な外的制御と時間発展によって $|0\dots 0\rangle$ から $|1\dots 1\rangle$ までの最大 $2^n$ 個の状態の重ね合わせとなる。量子アルゴリズムは、この膨大な数の状態間の干渉を(問題のパラメータに応じて)制御して、答えに相当する状態の確率振幅だけを大きくする量子操作を行う<sup>13</sup>。さらに簡潔に言い換えれば次のようになる。入力 $x$ に対する出力 $f(x)$ を求めたいとする( $x$ も $f(x)$ も $n$ ビット整数とする)。量子コンピュータは初期状態 $|0\dots 0\rangle$ にユニタリー演算子 $U_x$ を作用させ、 $n$  qubit 全てを基底 $\{|0\rangle, |1\rangle\}$ で観測する。正しい答え $f(x)$ が得られる確率 $|\langle f(x)|U_x|0\dots 0\rangle|^2$ が、 $x$ の値によらず高くなるように $U_x$ を設計するのがキモである。

量子アルゴリズムを見つけるという作業は、ここで言う大きなユニタリー演算子 $U_x$ を見つけることであり、高度に非自明な仕事である。しかし、一度アルゴリズムが構築されれば、あとはそれをいかに物理系で実行するかだけを問題にすればよい。つまり、アルゴリズムとしてどんなユニタリー演算子が与えられても、それを正確に実行できる系を作ることができれば、それが量子コンピュータとなる。ところが、 $n$  qubit の系は $2^n$ 次元の Hilbert 空間に相当し、そこに作用するユニタリー演算子も $2^n \times 2^n$ の行列で表わされる大きなものである。そのままでは物理的には非常に見通しが悪く、どの制御パラメータをどう変化させればよいかなど見当もつかないのが普通だ。そこで、これを物理的に実行可能な基本要素に分解する必要がある。この分解が常に可能であることを述べる重要な定理がある：

$U$ を $n$  qubit に作用する( $2^n$ 次元の)任意のユニタリー変換とする。 $U$ は、単一 qubit に対する操作と 2 qubit に対する制御 NOT (controlled-NOT)ゲートを適宜組み合わせることで、任意の

<sup>13</sup> 実際には、答えの正当性のチェックが効率よく行える限り成功確率が $1-\epsilon$ であればいいので、正解状態の確率振幅だけを大きくする必要はない。ただし、アルゴリズム実行の必要回数が指数関数的に大きくなってしまいう事態を避けるために、 $1-\epsilon$ は $(\log N(\text{入力の桁数の多項式})^{-1})$ 以上でないといけない。

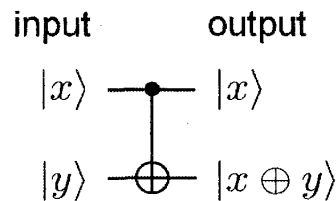


図2 制御 NOT (CNOT)ゲートのダイアグラム表示

制御ビット $|x\rangle$ とターゲットビット $|y\rangle$ が入力されると、 $|x\rangle \otimes |x \oplus y\rangle$ を出力する。ここで、 $\oplus$ 記号は modulo 2 の和を表す。

精度で近似できる。

Qubit とは複素数体 $\mathbb{C}$ 上の2次元ベクトルであるから、単一 qubit に対する操作とは2次元ユニタリー群( $U(2)$ )の要素に相当する回転操作に他ならない。制御 NOT ゲート(普通 CNOT と略される)は、制御ビットとターゲットビットの2つに作用し、制御ビットの状態が $|0\rangle$ にあるときはターゲットビットには何の操作もしないが、制御ビットが $|1\rangle$ にあるときはターゲットビットを反転させる。言い換えれば、2 qubit 状態を基底 $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ で表わしたとき、CNOT とはその行列表現が

$$U_{\text{CNOT}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (1)$$

となるものである。しばしば図2のように図示する。

さらに、任意の1 qubit 操作は少数の特定の操作を組み合わせ、任意の精度で近似できる。たとえば、Hadamard ゲート

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

と $\pi/8$ ゲート

$$S = \begin{pmatrix} 1 & 0 \\ 0 & \exp(i\pi/4) \end{pmatrix}$$

の組などである。Hadamard と $\pi/8$ ゲートでなくとも、少なくとも(Bloch 球の) $y, z$ 軸の周りの任意の回転が実行できれば、任意の回転 $R$ のオイラー分解によって $R = R_z(\alpha)R_y(\beta)R_z(\gamma)$ によ

って容易に実現できる<sup>14</sup>。ここで、 $R_i(\theta)$ は*i*-軸のまわりの角  $\theta$  回転を表す。

ここで、2 qubit 操作としてなぜ CNOT という特定の操作が取り上げられるのか、釈然としない方も多いかもしれない。実は上の定理は CNOT の代わりに「(2 qubit に作用する)任意のエンタングル操作」と言い換えても成立する。これは、2つの独立な qubit をエンタングルさせる能力をもつ操作のことであり、無限に多く存在する。それでも CNOT がしばしば語られる理由は、 $\{|0\rangle, |1\rangle\}$  基底での見通しのよさにつきる。式(1)の形式的単純さがなければ、目標となる  $U \in U(2^n)$  を分解する作業が煩雑を極めるだろう。また、実験で目指すべきものとしても明快だし、到達度評価もしやすい。そこで以下でも、2 qubit 操作の象徴的存在として CNOT を考える。

CNOT と等価な操作の例として、制御ビット、ターゲットビットが共に  $|1\rangle$  にあるときにだけ位相を  $\pi$  だけずらす CZ 操作 ( $|11\rangle \rightarrow -|11\rangle$ ) もよく用いられる (CPHASE とよばれる)。これは、ターゲットビット側で CZ を Hadamard ゲート  $H$  ではさむことで、CNOT と等しくなるからである。つまり、 $\text{CNOT} = (I \otimes H) \text{CZ} (I \otimes H)$  と表わせる ( $I$  は恒等変換)。

以上の 1 qubit 回転操作と CNOT が任意の qubit (対) に対して実行できれば、どんなユニタリー変換  $U_x$  でも実行できる。が、上述したように問題は分解によって生じるゲートの総数である。これが常に  $O(n^k)$  ならどんな量子計算も効率がよいことになってすばらしいのだが、そこまで都合よくできてはいない。任意の  $n$  qubit ゲート (上の  $U_x$ ) を実行するのに必要なゲートの総数は、一般に  $n$  に対して指数関数的であることが知られている [1]。

ゲート数が  $n$  の多項式で表わされるような便利なアルゴリズム (ユニタリー変換) が見つければ、「古典計算機より速い量子アルゴリズムを見つけ

た」と胸を張れるわけだ。そういう都合のいいユニタリー変換の代表格が量子フーリエ変換 (QFT) :

$$\text{QFT}: |j\rangle \mapsto \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi i j k / 2^n} |k\rangle \quad (2)$$

であり、 $O(n^2)$  個のゲートで実行することができる ([1] や [2] を参照)。この変換が、Shor の因数分解アルゴリズムをはじめとする多くの量子アルゴリズムの「メインモジュール」をなす重要な要素となっている。Shor のアルゴリズムの詳細の理解には、整数論の基本的知識などもいくらか必要であるので詳細は上にも挙げた [1] や [2] などを参照していただきたい。

### 3 量子計算の物理的実現と課題

必要な基本要素の組が分かったので、あとはそれを現実の物理系で実行すれば量子計算機の完成だ。まずは、モノとして何を選ぶかから始めなければならない。どんな物理系のどの量子力学的自由度に qubit の  $\{|0\rangle, |1\rangle\}$  を割り当てるか、CNOT をどう実現するか、である。代表的なものを以下に挙げる<sup>15</sup>。

- 光子 (非線形光学) : 1 ビットにつき 2 つの光路を設け、単一光子がどちらの光路を通るかで  $|0\rangle, |1\rangle$  を表す。2 つの qubit の  $|1\rangle$  光路を非線形光学素子に通し、CNOT を実現する。非線形光学効果は生起確率が非常に低いのが問題。偏光状態を qubit とすることもできる。
- 光子 (線形光学) : 線形光学素子のみで CNOT を確率的に実現することも可能。あらかじめ別の qubit に対して成功した CNOT を「テレポート」して任意の qubit 対に適用する KLM scheme が基本 [3]。CNOT のテレポーテーションの成功確率向上が難しい。
- NMR (核磁気共鳴) : 分子を構成する原子の核スピン。隣接原子間のスピン相互作用によ

<sup>14</sup>  $R$  に位相因子  $e^{i\theta}$  をかければ任意のユニタリー変換 ( $\in U(2)$ ) を構成できる。

<sup>15</sup> 特に参考文献を示していないものについては、[2] などの標準的教科書に詳細な記述がある。

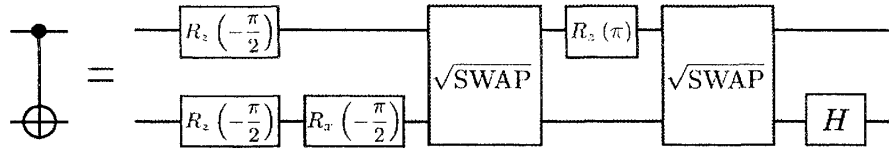


図3 スピン間の交換作用による CNOT 構成法

$H$  は Hadamard ゲートを表す. 各  $x, y, z$ -軸は Pauli 行列  $\sigma_x, \sigma_y, \sigma_z$  の固有状態に対応する軸として定まる.

り CNOT を実行する. 室温で機能し, CNOT も高精度で実証済みであることが強みだが, 個々の qubit の読み出し, 集積化に難.

- イオントラップ: 1次元内に束縛したイオン電子状態の超微細構造で qubit を表現. イオン列の格子振動(フォノン)を介して離れた qubit 間で CNOT を実現, 集積化が課題.
- 光学格子中の中性原子: 複数のレーザーを重ねて卵パック型のポテンシャル(光学格子)を作り, 1つの谷に1つの原子を束縛. 電子状態の超微細構造準位で qubit を表現し, 光学格子の制御により隣接原子同士を相互作用させる. 現状の構成では, 特定の qubit 同士だけを選択的に相互作用させることができない [4].
- 量子ドット: 基盤上の微小電極によるポテンシャルに単一電子を束縛. 電子スピンを qubit として用い, 隣接する束縛ポテンシャルとの障壁高さを制御して 2 qubit 操作. CNOT はスピン間交換相互作用による [5].
- ダイヤモンド中の N-V 中心: ダイヤモンド結晶格子中で隣接する炭素原子が窒素原子と空孔で置き換えられたものを N-V 中心とよぶ. N-V 中心に束縛される電子のスピンを qubit として用い, N-V 中心のネットワークを組むことで量子コンピュータを構成する [6]. 常温でも動作可能だが, N-V 中心の配置は現在のところ制御できない.
- 超伝導 qubit: ジョセフソン接合により隔離された超伝導体の「島」中のクーパー対の数で qubit を表現, 島と島をコンデンサーやジョセフソン接合を介して相互作用させると実

質的にスピン間相互作用と等価となる. ジョセフソン接合を含む超伝導体のリング中に流れる永久電流の向きを qubit とするなど, いくつか方式がある [7].

直接 qubit 同士を相互作用させる代わりに, qubit とは別に量子情報を一時的にやり取りするための「バス」を設ける方法もある. バスと qubit の間の相互作用を制御して, 実質的に 2 qubit 間で CNOT を実現するわけだ. イオントラップや超伝導 qubit の一部がこれに相当する. 空間的に離れた qubit 同士を選択的に相互作用させる場合などに有効な方法である.

上のリストではさらりと流してしまっただが, 粒子間相互作用を CNOT 実現にいかにも利用するかが, どの方式においても知恵の出どころとなる. 個々の詳細は参考文献に譲るが, スピン 1/2 による qubit を用いた例を紹介しておこう. 2つの近接したスピン 1/2 粒子の間には交換相互作用

$$H = -J\vec{S}_1 \cdot \vec{S}_2 \quad (3)$$

がはたらく.  $\vec{S}=1/2\vec{\sigma}=1/2(\sigma_x, \sigma_y, \sigma_z)$  はスピン演算子,  $J>0$  は結合定数である(簡単のため強磁性を仮定,  $\hbar=1$  とする). この  $H$  を生成子とする時間発展演算子  $U(t)=\exp[-iHt]$  は,  $t=\pi/J$  で 2つの qubit 状態をそっくり交換する SWAP 演算子:

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4)$$

となる. この半分の時間だけの時間発展で得られる演算子  $U(\pi/(2J))$  を  $\sqrt{\text{SWAP}}$  とよぶ. 図3のように,  $\sqrt{\text{SWAP}}$  と単一 qubit 操作を組み合わせると CNOT が実現できる.

さて、上のどの実現方式にも多くの改良型、応用型があり、これですべてというわけでは決してない。どれも一長一短(多短?)があり、どれが「本命」なのかははっきりしないのが現状だ。光子を使うものは外乱に対する強さという大きな魅力があるものの(とは言っても外乱と無縁ではない)、肝心のプロセスが確率的で、実際の光学系が非常に複雑になって集積化しにくい面がある<sup>16</sup>。誤り訂正も含めて意味のある量子計算を行うには $10^3 \sim 10^4$  qubit 必要と言われており、集積化という点からは固体ベースの系(上のリスト最後の3つ)が有利といえる。しかし、現実には数( $\sim 4$ ) qubit がせいぜいで、量子計算の強みをフルに生かすには前途は厳しい。どの候補技術においても、課題を克服する大胆な方法が見出されれば、一気に実現へ向けたレースの先頭に躍り出る可能性がある。

また、すでにチラホラと触れたが、物理系として何を選ぼうが、常についてまわる難問がデコヒーレンスである。「量子性」を十分に活用して情報処理を実行するには、全系が純粋状態になければならないが、環境との(予測・追跡不能な)相互作用によって純粋さ(コヒーレンス)が時々刻々失われていってしまう。情報を担う光子や電子そのものが環境中へ逃げていってしまう可能性すらある。環境から完全に系を隔離すればコヒーレンスは保たれるだろうが、それでは我々の制御も系に届かないことになってしまう。光子のようにデコヒーレンスが起こりにくい場合に、qubit 同士の相互作用もさせにくいのも同じ理屈である。量子コンピュータに我々が求めているのは、環境とは相互作用しないが制御系との相互作用はきれいに受け入れて意のままに機能してほしいという、実にワガママな要求なのだ。そう簡単に自然がそんな虫のいい要求を受け入れてくれるわけはなく、そこに知恵と努力が必要とされているのである。

集積化も大きな課題である。NMR のように本

質的に qubit の数に限界がある方式だと、量子コンピュータの「うまみ」を最大限引き出すことができない。かと言って、固体ベースの系で、遠く離れた2つの qubit(たとえば、1番めと1000番め)をうまく(他の qubit には影響を与えずに)カップリングさせられるだろうか。

数 qubit からなる系しか実現できていない現状からすれば、 $10^3$  とか  $10^4$  qubit などというのは夢物語に聞こえるかもしれない。課題を知れば知るほど実現までの道のりは遠いという印象が強くなる。しかし、現状での様々な困難を知るからこそ、「あっ」と驚くゲリラ的新アイデアによって、一気に実用化に近づく可能性があるとは言えないだろうか。逆に言えば、地道な改良だけではおそらく相当な時間をかけても(汎用量子コンピュータの)実現は難しいだろう。

ソフトの面からは、有用な量子アルゴリズムとして、実質的に上で触れた Shor の因数分解と、もう1つ Grover のサーチアルゴリズムしか見つかっていないのが問題だ。実のところ、なぜ量子コンピュータが速いのかとか、量子コンピュータにしか効率的に解けない問題が本当にあるのか、などといった根源的な問題は未解決なのだ。苦労して量子コンピュータを作っても、古典コンピュータをしのぐ性能を発揮する他の用途がなければ魅力薄になってしまう。ただ、たとえ汎用量子コンピュータでなくても、大規模量子系を効率的にシミュレートできる「量子シミュレーター」[8]としての高度な有用性は常に残る。

以上、駆け足で量子計算の概要を説明したが、技術的な詳細は別にしても、量子系に情報処理をさせるということのイメージをつかんでいただければ幸いである。繰り返しになるが、物理的には、初期状態の準備、ユニタリー変換(制御含む)、観測であり、量子計算アルゴリズムは、求めるべき計算結果が最後の観測で確率  $1-\epsilon$  で得られるようなユニタリー変換として実行される。次回は、「(多体)量子系の制御」のより原理的側面に注目して、制御可能性についての数学的条件や、関連す

<sup>16</sup> 原理的には集積化できるが、量子情報を光子以外の物理系に移した上での長時間保存・蓄積など別の技術が必要となる。

る最近の結果を紹介する予定である。

### 参考文献

- [ 1 ] Ekert, A. and Jozsa, R., *Rev. Mod. Phys.*, 68 (1996), 733.
- [ 2 ] Nielsen, M. A. and Chuang, I. L., *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.
- [ 3 ] Knill, E., Laflamme, R. and Milburn, G. J., *Nature*, 409(2001), 49.
- [ 4 ] Mandel, O., et al., *Nature*, 425(2003), 937.
- [ 5 ] Loss, D. and DiVincenzo, D. P., *Phys. Rev. A*, 57 (1998), 120. Petta, J. R., et al., *Science*, 309(2005), 2180.
- [ 6 ] Jelezko, F., et al., *Phys. Rev. Lett.*, 93(2004), 130501. Wrachtrup, J. and Jelezko, F., *J. Phys. Condens. Matter*, 18(2006), S807. Gaebel, T. et al., *Nature Phys.*, 2(2006), 408.
- [ 7 ] Makhlin, Y., Schön, G. and Schnirman, A., *Rev. Mod. Phys.*, 73(2001), 357. You, J. Q. and Nori, F., *Phys. Today*, 58(2005), 42.
- [ 8 ] Feynman, R., *Int. J. Theor. Phys.*, 21(1982), 467.