Eur. Phys. J. B **27**, 15–28 (2002)
DOI: 10.1140/epjb/e20020126

THE EUROPEAN PHYSICAL JOURNAL B
EDP Sciences

Quantum spin dynamics as a model for quantum computer operation

H. De Raedt[1,a], K. Michielsen[1], A. Hams[1], S. Miyashita[2], and K. Saito[2]

[1] Institute for Theoretical Physics and Materials Science Centre, University of Groningen, Nijenborgh 4, 9747 AG Groningen, The Netherlands
[2] Department of Applied Physics, School of Engineering, University of Tokyo, Bunkyo-ku, Tokyo 113, Japan

Received 5 November 2001 and Received in final form 8 February 2002

Abstract. We study effects of the physical realization of quantum computers on their logical operation. Through simulation of physical models of quantum computer hardware, we analyze the difficulties that are encountered in programming physical realizations of quantum computers. Examples of logically identical implementations of the controlled-NOT operation and Grover's database search algorithm are used to demonstrate that the results of a quantum computation are unstable with respect to the physical realization of the quantum computer. We discuss the origin of these instabilities and discuss possibilities to overcome this, for practical purposes, fundamental limitation of quantum computers.

PACS. 75.10.Jm Quantized spin models – 76.20.+q General theory of resonances and relaxations – 03.65.-w Quantum mechanics – 03.67.Lx Quantum computation

1 Introduction

Recent theoretical work has shown that a quantum computer has the potential of solving certain computationally hard problems such as factoring integers [1] and searching databases much faster than a conventional computer [2]. In most theortical work the operation of a quantum computer is described in terms of highly idealized transformations on the qubits [3–6]. The impact of the physical implementation of a quantum computer on its computational efficiency is largely unexplored.

The logical operation of conventional digital circuits does not depend on their hardware implementation (*e.g.* semiconductors, relays, vacuum tubes, etc.). Dissipative processes suppress the effects of the internal, non-ideal (chaotic) dynamics and drive the circuits into regions of stable operation. Conventional digital computers, built from these digital circuits, are in one particular state at a time and are able to perform logical operations that do not depend on their hardware implementation. From the point of view of programming the computer this is very important. Implementations of algorithms designed to run on a conventional computer will give results that do not depend on the hardware used to build the computer.

A quantum computer differs from a conventional digital computer in many respects. A quantum computer exploits the fact that a quantum system can be in a superposition of states. Interference of these states allows exponentially many computations to be done in parallel [3–7]. The presence of the superposition of states is a direct manifestation of the internal quantum dynamics of the elementary units of the quantum computer, the qubits. In an ideal quantum computer the qubits are assumed to be ideal two-state quantum systems. Therefore, the operation of an ideal quantum computer does not depend on the intrinsic dynamics of its qubits.

A physically realizable quantum computer is a many-body system in which the quantum dynamics of the qubits is essential to its operation. Manipulation of one qubit may cause unwanted motion of other qubits. It is difficult to suppress these effects by dissipation because in contrast to the case of conventional digital circuits, dissipation processes have a devastating effect on the coherent quantum dynamical motion of the qubits. Therefore a quantum algorithm may yield quantum computation results that depend on the specific physical realization of the quantum computer. Although quantum algorithms can be designed independent of the quantum computer hardware, the implementation of a quantum algorithm on a physical realization of a quantum computer (*i.e.* the programming of the quantum computer) very much depends on the hardware of which the quantum computer is built from. We refer to the problem of programming quantum computers as the quantum programming problem.

Due to the quantum programming problem it may be very difficult to develop a non-trivial quantum program for a physical realization of a quantum computer. Moreover, there is no guarantee that an implementation of a

[a] e-mail: deraedt@phys.rug.nl

quantum algorithm, that works well on one quantum computer will also perform well on other physical realizations of quantum computers. As mentioned above, there are several factors that contribute to the quantum programming problem:

1) Differences between the theoretically perfect and physically realizable one- and two-qubit operations; qubits cannot be kept still during the time that other qubits are being addressed; precision needed to implement operations on the qubits.
2) The effect of coupling of the qubits to other degrees of freedom (dissipation, decoherence).

How does a quantum programming problem reveal itself? Consider two logically independent operations (O_1 and O_2) of the machine. On a conventional computer or ideal quantum computer, the order in which we execute these two mutually independent instructions does not matter: $O_1O_2 = O_2O_1$. However, on a physically realizable quantum computer *sometimes* the order does matter, even if there are no logical dependencies in these two program steps. In some cases, due to practical problems in manipulating individual qubits $O_1O_2 \neq O_2O_1$ and the quantum computer may give wrong answers. Note the qualifier *sometimes*. There seems to be no general rule to decide beforehand which operation and at what stage of the quantum algorithm the quantum programming problem leads to incorrect results. At present the only way to find out seems to be to actually carry out the calculations and check the results.

In most theoretical work on quantum computers and quantum algorithms [1–6,8] one considers theoretically ideal (but physically unrealizable) quantum computers. Then the quantum programming problem is not an issue. The quantum programming problem is also fundamentally different from the error propagation previously studied in quantum algorithms implemented on ideal quantum computers [9–12] since the quantum programming problem is due to the specific realization of the quantum computer and leads to systematic instead of random errors. In principle the theory of fault-tolerant quantum computation can be used to deal with random errors, this to the extent that the noise satisfies certain criteria [8]. However, it is good to realize that it is *not* the case that it is possible to protect a quantum computation against the effects of completely random noise [8]. According to optimistic estimates the probability of error per quantum gate should be less than 10^{-4} for fault-tolerant quantum computation [8]. Disregarding the fact that the systematic errors considered in this paper are far from being random, the typical size of these errors is much larger than 10^{-4} and therefore there is no indication that these errors can be compensated for by current fault-tolerant quantum computation techniques. Furthermore one also has to take into account that quantum error correction requires the manipulation of additional qubits. On a physical quantum computer, these extra qubits suffer from exactly the same quantum programming problem as the other qubit(s). In principle the quantum error correction part should be treated on the same physical footing as the other qubits. Actually this is a highly non-trivial simulation because a physical model of 2 qubits with minimal quantum error correction contains at least 10 qubits. In view of the difficulties encountered with only 2 qubits, as reported in this paper, we relegate the simulation of a physical model with noise and quantum error correction to a future project.

In this paper we study the relation between the physical realization of quantum computers and their logical operation. We investigate various aspects of the quantum programming problem by simulating quantum computer hardware. In this work we only consider effects of the intrinsic quantum dynamics of the quantum computer (item 1, see above). The study of the effect of the coupling of the qubits to other degrees of freedom (item 2, see above) is left for future research. We demonstrate that the programming of a physical, non-ideal implementation of a quantum computer is difficult, even if the quantum computer consists of only two qubits. Berman *et al.* [13] investigated the influence of the Ising spin interaction on the quantum dynamics of NMR systems. Although they did not address the quantum programming problem, the work is similar in spirit to [14] and the present paper as it explores the consequences of the difference between the ideal quantum computers and physical realizations of quantum computers. As far as we know no experimental data has been published that specifically addresses this, for potential applications, very important and intrinsic problem of programming quantum computers. However, with the quantum computer hardware currently available a test of correct quantum computation on a physical realization of a quantum computer is definitely within reach. In this paper we propose two simple quantum algorithms that may be used to study the quantum programming problem in physical realizations of quantum computers. We also discuss methods to enlarge the region(s) of reliable operation.

The paper is organized as follows: In Section 2 we describe a physical model of a quantum computer. Our choice is largely inspired by NMR quantum computer experiments [15–22], mainly because other candidate technologies for building quantum computers [23–37] are not yet developed to the point that they can execute computationally non-trivial quantum algorithms. As the basic example of a quantum algorithm we take the Controlled-NOT (CNOT) gate [38]. In Section 3 we discuss the implementation of the $CNOT$ gate on an ideal two-qubit quantum computer and describe simple, non-trivial quantum algorithms that consist of repetitions of $CNOT$ operations. As an illustration of the general nature of the quantum programming problem, we also consider a more complicated example, namely Grover's quantum algorithm to search for an item in a database [2]. In Section 4 we derive the conditions for which the physical two-qubit quantum computer will exhibit ideal quantum computer behavior and discuss the generalization of these ideas to n-qubit quantum computers. Also in Section 4 we describe the implementation of the quantum algorithms discussed in Section 3 on a physical realization of a quantum computer. In Section 5 we demonstrate and analyze the quantum programming problem by simulating the time-evolution

of (= execute quantum algorithms on) the physical model of the quantum computer by solving the time-dependent Schrödinger equation. In Section 6 we summarize our findings.

2 Physical model of a quantum computer

Generic quantum computer hardware can be modeled in terms of quantum spins (qubits) that evolve in time according to the time-dependent Schrödinger equation

$$\mathrm{i}\frac{\partial}{\partial t}|\Psi(t)\rangle = H(t)|\Psi(t)\rangle, \tag{1}$$

in units such that $\hbar = 1$. For present purposes it is sufficient to consider two-qubit quantum computers only. The state

$$\begin{aligned}|\Phi(t)\rangle = a(\downarrow,\downarrow;t)|\downarrow,\downarrow\rangle + a(\uparrow,\downarrow;t)|\uparrow,\downarrow\rangle \\ + a(\downarrow,\uparrow;t)|\downarrow,\uparrow\rangle + a(\uparrow,\uparrow;t)|\uparrow,\uparrow\rangle,\end{aligned} \tag{2}$$

describes the state of the quantum computer at time t. The complex coefficients $a(\downarrow,\downarrow;t),\ldots,a(\uparrow,\uparrow;t)$ completely specify the state of the quantum system. In the absence of interactions with other degrees of freedom this spin-1/2 system can be modeled by the time-dependent Hamiltonian

$$\begin{aligned}H(t) = &-JS_1^zS_2^z - h_1^zS_1^z - h_2^zS_2^z - h_1^xS_1^x - h_2^xS_2^x \\ &-h_1^yS_1^y - h_2^yS_2^y \\ &-(\tilde{h}_1^xS_1^x + \tilde{h}_2^xS_2^x)\sin(\omega t + \phi_x) \\ &-(\tilde{h}_1^yS_1^y + \tilde{h}_2^yS_2^y)\sin(\omega t + \phi_y),\end{aligned} \tag{3}$$

where S_j^α, $\alpha = x, y, z$ denotes the αth component of the spin-1/2 operator representing the jth qubit, J determines the strength of the interaction between the two qubits, h_j^α and $\tilde{h}_j^\alpha$ represent the strength of the applied static (magnetic) and applied sinusoidal field (SF) acting on the jth spin respectively. For a physical system, $h_2^\alpha = \gamma h_1^\alpha$ and $\tilde{h}_2^\alpha = \gamma\tilde{h}_1^\alpha$, for $\alpha = x, y, z$ where γ is a constant. The frequency and the phase of the sinusoidal field are denoted by ω and ϕ_α. As the Ising model, *i.e.* the first term of (3), is known to be a universal quantum computer [39,40], model (3) is sufficiently general to serve as a physical model for a generic quantum computer at zero temperature. In terms of spin matrices, the operator Q_j measuring the state of qubit j is given by

$$Q_j = \frac{1}{2} - S_j^z. \tag{4}$$

For numerical purposes it is necessary to fix as many model parameters as possible. We have chosen to simulate the two nuclear spins of the ^{1}H and ^{13}C atoms in a carbon-13 labeled chloroform, a molecule that has been used in NMR quantum computer experiments [17,18]. In these experiments $h_1^z/2\pi \approx 500$ MHz, $h_2^z/2\pi \approx 125$ MHz, and $J/2\pi \approx -215$ Hz [17]. In the following we will use model parameters rescaled with respect to $h_1^z/2\pi$, *i.e* we put

$$J = -0.43\times 10^{-6}, \quad h_1^z = 1, \quad h_2^z = 0.25. \tag{5}$$

With this choice of units, time divided by 2π is measured in units of 2 ns. Note that there is a difference of many orders of magnitude between the interaction J and the fields h_j^z. If the duration of the sinusoidal field pulses is much shorter than $2\pi/|J|$, the effects of J on the time evolution during these pulses are very small. Our numerical experiments (see below) are all performed under this condition. We will only consider quantum computers at zero temperature without coupling to the environment. In this sense we simulate highly idealized NMR experiments on a closed quantum system at zero temperature. This allows us to study a concrete physical realization of a quantum computer and at the same time focus on the intrinsic quantum dynamics of the quantum computer.

A quantum algorithm for quantum computer model (3) consists of a sequence of elementary operations that change the state $|\Psi\rangle$ of the quantum processor according to the time-dependent Schrödinger equation, *i.e.* by (a product of) unitary tranformations. Each elementary operation transforms the input state $|\Psi(t)\rangle$ into the output state $|\Psi(t+\tau)\rangle$ where τ denotes the execution time of the elementary operation. The action of an elementary operation on the state $|\Psi\rangle$ of the quantum processor is defined by specifying how long it acts (*i.e.* the time interval τ during which it is active), and the values of J and all h's. During the execution of an elementary operation the values of J and all h's are kept fixed.

The time evolution of quantum model (3) is obtained by solving time-dependent Schrödinger equation (1) for model (3). The simulations have been carried out with a software tool called Quantum Computer Emulator [41]. The quantum computer software simulates physical models of quantum computer hardware by a Suzuki product-formula [42,43], *i.e.* in terms of elementary unitary operations [44–46]. For all practical purposes, the numerical results obtained by this technique are exact. A detailed description of the quantum computer software tool can be found elsewhere [47].

3 Ideal quantum computer

3.1 Single-qubit operations

One qubit or one spin-1/2 system is a two-state quantum system. The two basis states spanning the Hilbert space are denoted by $|\uparrow\rangle \equiv |0\rangle$ and $|\downarrow\rangle \equiv |1\rangle$. Rotations of spin j about $\pi/2$ around the x and y-axis are basic quantum computer operations. We will denote them by X_j and Y_j respectively. In matrix notation, they are given by

$$X_j \equiv \mathrm{e}^{\mathrm{i}\pi S_j^x/2} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & \mathrm{i}\\ \mathrm{i} & 1\end{pmatrix}, \tag{6}$$

Table 1. Input and output states and the corresponding expectation values (a, b) of the qubits for the $CNOT$ operation.

Input state	a	b	Output state	a	b
$\|00\rangle$	0	0	$\|00\rangle$	0	0
$\|10\rangle$	1	0	$\|11\rangle$	1	1
$\|01\rangle$	0	1	$\|01\rangle$	0	1
$\|11\rangle$	1	1	$\|10\rangle$	1	0

and

$$Y_j \equiv \mathrm{e}^{\mathrm{i}\pi S_j^y/2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}. \tag{7}$$

Clearly operations such as (6) and (7) can be implemented in terms of the time evolution of model (3) by a proper choice of the model parameters. Writing $|a\rangle = a_0|00\rangle + a_1|10\rangle + a_2|01\rangle + a_3|11\rangle$ with $|b_1 b_2\rangle \equiv |b_1\rangle|b_2\rangle$ and $b_i = 0, 1$ we have

$$X_1|a\rangle = X_1 \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & \mathrm{i} & 0 & 0 \\ \mathrm{i} & 1 & 0 & 0 \\ 0 & 0 & 1 & \mathrm{i} \\ 0 & 0 & \mathrm{i} & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}. \tag{8}$$

For example, $X_1|11\rangle = (|11\rangle + i|01\rangle)/\sqrt{2}$. Using the same labeling of the basis states as in (8) we have

$$Y_2 \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}, \tag{9}$$

e.g. $Y_2|11\rangle = (|10\rangle + |11\rangle)/\sqrt{2}$. The matrix expressions for the inverse of the rotations X_1 and Y_2, denoted by $\overline{X}_1$ and $\overline{Y}_2$ respectively, are obtained by taking the hermitian conjugates of the matrices in (8) and (9). For example, $\overline{Y}_2|11\rangle = (|11\rangle - |10\rangle)/\sqrt{2}$.

3.2 Two-qubit operations: CNOT gate

Computation requires some form of communication between the qubits. A basic two-qubit operation is provided by the $CNOT$ gate. The $CNOT$ gate flips the second spin if the first spin is in the down state, *i.e.* the first qubit acts as a control qubit for the second one, see Table 1. The procedure that we use to construct the $CNOT$ gate may seem a little ad hoc and indeed to considerable extent it is. There is no unique method to construct quantum computer gates.

On an ideal quantum computer the $CNOT$ gate can be implemented by a combination of single-qubit operations and a two-qubit phase shift operation P defined by the matrix

$$P \equiv \begin{pmatrix} \mathrm{e}^{\mathrm{i}\phi_0} & 0 & 0 & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\phi_1} & 0 & 0 \\ 0 & 0 & \mathrm{e}^{\mathrm{i}\phi_2} & 0 \\ 0 & 0 & 0 & \mathrm{e}^{\mathrm{i}\phi_3} \end{pmatrix}. \tag{10}$$

Assume that the quantum computer is in a state

$$|\Psi\rangle = a_0|00\rangle + a_1|10\rangle + a_2|01\rangle + a_3|11\rangle. \tag{11}$$

First we apply to $|\Psi\rangle$ the rotation Y_2, as defined in (9). This gives

$$Y_2|\Psi\rangle = \frac{1}{\sqrt{2}}\left[(a_0 + a_2)|00\rangle + (a_1 + a_3)|10\rangle + (a_2 - a_0)|01\rangle + (a_3 - a_1)|11\rangle\right]. \tag{12}$$

Next we apply to $Y_2|\Psi\rangle$ the phase shift P

$$PY_2|\Psi\rangle = \frac{1}{\sqrt{2}}\left[\mathrm{e}^{\mathrm{i}\phi_0}c_0|00\rangle + \mathrm{e}^{\mathrm{i}\phi_1}c_1|10\rangle + \mathrm{e}^{\mathrm{i}\phi_2}c_2|01\rangle + \mathrm{e}^{\mathrm{i}\phi_3}c_3|11\rangle\right], \tag{13}$$

where $c_0 = a_0 + a_2$, $c_1 = a_1 + a_3$, $c_2 = a_2 - a_0$ and $c_3 = a_3 - a_1$. Finally we apply the inverse of the rotation Y_2

$$\overline{Y}_2 PY_2|\Psi\rangle = \frac{1}{2}\left[(\mathrm{e}^{\mathrm{i}\phi_0}c_0 - \mathrm{e}^{\mathrm{i}\phi_2}c_2)|00\rangle + (\mathrm{e}^{\mathrm{i}\phi_1}c_1 - \mathrm{e}^{\mathrm{i}\phi_3}c_3)|10\rangle + (\mathrm{e}^{\mathrm{i}\phi_0}c_0 + \mathrm{e}^{\mathrm{i}\phi_2}c_2)|01\rangle + (\mathrm{e}^{\mathrm{i}\phi_1}c_1 + \mathrm{e}^{\mathrm{i}\phi_3}c_3)|11\rangle\right]. \tag{14}$$

We now determine the angles ϕ_i such that the sequence (14) performs the $CNOT$ operation. Since the $CNOT$ gate will not change a_0 and a_2 (see Tab. 1) we can choose $\phi_0 = \phi_2$. This gives

$$\overline{Y}_2 PY_2|\Psi\rangle = \mathrm{e}^{\mathrm{i}\phi_0}\left[a_0|00\rangle + a_2|01\rangle + \mathrm{e}^{\mathrm{i}\beta}(a_1\cos\alpha + \mathrm{i}a_3\sin\alpha)|10\rangle + \mathrm{e}^{\mathrm{i}\beta}(a_3\cos\alpha + \mathrm{i}a_1\sin\alpha)|11\rangle\right], \tag{15}$$

where $\beta = \alpha + \phi_3 - \phi_0$ and $\alpha = (\phi_2 - \phi_3)/2$. The global phase factor $e^{i\phi_0}$ is physically irrelevant.

The simplest way to implement the phase shift P is to use the time evolution, *i.e.* $P = \mathrm{e}^{-\mathrm{i}\tau H_I}$, of the Ising model

$$H_I = -JS_1^z S_2^z - hS_1^z - hS_2^z, \tag{16}$$

where the external fields acting on both spins are the same. From (16) it follows immediately that $\phi_0 = \tau(J/4 + h)$, $\phi_1 = \phi_2 = -\tau J/4$ and $\phi_3 = \tau(J/4 - h)$. Taking into account our choice $\phi_0 = \phi_2$, (15) becomes

$$\overline{Y}_2 PY_2|\Psi\rangle = \mathrm{e}^{\mathrm{i}\alpha/2}\left[a_0|00\rangle + a_2|01\rangle + \mathrm{e}^{-\mathrm{i}\alpha}(a_1\cos\alpha + \mathrm{i}a_3\sin\alpha)|10\rangle + \mathrm{e}^{-\mathrm{i}\alpha}(a_3\cos\alpha + \mathrm{i}a_1\sin\alpha)|11\rangle\right]. \tag{17}$$

Using the same labeling of states as in (8) we have

$$\overline{Y}_2 P Y_2 = \mathrm{e}^{\mathrm{i}\alpha/2} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{e}^{-\mathrm{i}\alpha}\cos\alpha & 0 & \mathrm{i}\mathrm{e}^{-\mathrm{i}\alpha}\sin\alpha \\ 0 & 0 & 1 & 0 \\ 0 & \mathrm{i}\mathrm{e}^{-\mathrm{i}\alpha}\sin\alpha & 0 & \mathrm{e}^{-\mathrm{i}\alpha}\cos\alpha \end{pmatrix}. \quad (18)$$

Comparing the truth table of the $CNOT$ gate (see Tab. 1) with the matrix in (18), it is clear that putting $\alpha = \pi/2$ will do the job (upto an irrelevant global phase factor). In terms of Hamiltonian (16), $-\tau J = \pi$ and $h = -J/2$. The sequence

$$CNOT = \overline{Y}_2 I Y_2 = \mathrm{e}^{\mathrm{i}\pi/4} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad (19)$$

performs the $CNOT$ operation on qubit 2 with qubit 1 acting as the control variable. Here we introduced the symbol I to represent the time evolution $\mathrm{e}^{-\mathrm{i}\tau H_I}$ with $\tau = -\pi/J$.

3.3 Quantum algorithms

Any quantum algorithm can be written as a sequence of the one- and two-qubit operations discussed above. As a simple example of a quantum algorithm we take $(CNOT)^5$. On an ideal quantum computer, $CNOT^2$ is the identity operation and hence $(CNOT)^5 = CNOT$ but on a physical quantum computer this is not always the case, see below. To illustrate the dependence of the quantum computation on the physical implementation and on the choice of the input state we consider two quantum algorithms, QA_1 and QA_2, defined by

$$QA_1|b_1b_2\rangle \equiv (CNOT)^5|b_1b_2\rangle, \quad (20)$$
$$QA_2|singlet\rangle \equiv Y_1(CNOT)^5|singlet\rangle, \quad (21)$$

where $|singlet\rangle = (|01\rangle - |10\rangle)/\sqrt{2}$. We have

$$(|01\rangle - |11\rangle)/\sqrt{2} = (CNOT)^5|singlet\rangle, \quad (22)$$

and hence $\langle singlet|(CNOT)^5 QA_1 (CNOT)^5|singlet\rangle = 1/2$. We can obtain a clear-cut answer in terms of expectation values of the qubits by applying a $\pi/2$ rotation to spin 1

$$|11\rangle = Y_1(CNOT)^5|singlet\rangle. \quad (23)$$

Therefore in (21), the $CNOT$ operations are followed by a $\pi/2$ rotation of spin 1.

As a more complicated example of a quantum algorithm, we consider Grover's database search algorithm to find the needle in a haystack. On a conventional computer, finding an item out of N elements requires $\mathcal{O}(N)$ queries [48]. Grover has shown that a quantum computer can find the item using only $\mathcal{O}(\sqrt{N})$ attempts [2]. Assuming a uniform probability distribution for the needle, for $N = 4$ the average number of queries required by a conventional algorithm is 9/4 [17,48]. With Grover's quantum algorithm the correct answer can be found in a single query [15,17].

Experimentally Grover's quantum algorithm has been implemented on a 2-qubit NMR quantum computer for the case of a database containing four items [15,17]. In experiments [15,17] the sequences

$$U_0 = X_1\overline{Y}_1X_2\overline{Y}_2 G X_1\overline{Y}_1X_2\overline{Y}_2 G\overline{X}_1\overline{X}_1\overline{Y}_1\overline{X}_2\overline{X}_2\overline{Y}_2, \quad (24)$$
$$U_1 = X_1\overline{Y}_1X_2\overline{Y}_2 G X_1\overline{Y}_1\overline{X}_2\overline{Y}_2 G\overline{X}_1\overline{X}_1\overline{Y}_1\overline{X}_2\overline{X}_2\overline{Y}_2, \quad (25)$$
$$U_2 = X_1\overline{Y}_1X_2\overline{Y}_2 G\overline{X}_1\overline{Y}_1X_2\overline{Y}_2 G\overline{X}_1\overline{X}_1\overline{Y}_1\overline{X}_2\overline{X}_2\overline{Y}_2, \quad (26)$$
$$U_3 = X_1\overline{Y}_1X_2\overline{Y}_2 G\overline{X}_1\overline{Y}_1\overline{X}_2\overline{Y}_2 G\overline{X}_1\overline{X}_1\overline{Y}_1\overline{X}_2\overline{X}_2\overline{Y}_2, \quad (27)$$

have been chosen to implement Grover's search algorithm. The subscript j of U_j corresponds to the position of the searched-for item in the database. In all four cases the input state is $|00\rangle$. The two-qubit operation G is defined by

$$G = \begin{pmatrix} \mathrm{e}^{-\mathrm{i}\pi/4} & 0 & 0 & 0 \\ 0 & \mathrm{e}^{+\mathrm{i}\pi/4} & 0 & 0 \\ 0 & 0 & \mathrm{e}^{+\mathrm{i}\pi/4} & 0 \\ 0 & 0 & 0 & \mathrm{e}^{-\mathrm{i}\pi/4} \end{pmatrix}, \quad (28)$$

and performs a conditional phase shift.

On an ideal quantum computer the quantum algorithms (24–27) are by no means unique: Various alternative expressions can be written down by using the algebraic properties of the X's and Y's. This feature has been exploited to eliminate redundant elementary operations [17]. On an ideal quantum computer sequences (24–27) return the correct answer, *i.e.* the position of the searched-for item. This is easily verified on the quantum computer by selecting the elementary operations that implement an ideal quantum computer.

4 Physical quantum computer

In this section we recapitulate some elementary quantum mechanics that is useful to understand how the ideal one- and two-qubit operations can be implemented by controlling the time evolution of the quantum spin system. We use the NMR system described above as a concrete example and use analytical instead of numerical techniques to discuss the salient features. However at this point we want to stress that in contrast to the analytical treatment presented below, our numerical simulations do not in any way rely on one or more approximations that are necessary to make the problem analytically tractable. Thus, the analytical results presented below should be considered as useful, qualitative information that is helpful to understand the behavior of the quantum spin system, as obtained from the simulation.

4.1 Single-qubit operations

NMR uses sinusoidal field pulses to rotate the spins. By tuning the frequency of the sinusoidal field to the precession frequency of a particular spin (h_j^z in our case), the power of the applied pulse (= intensity times duration) controls how much the spin will rotate. The axis of the rotation is determined by the direction of the applied sinusoidal field. The elementary model of an NMR experiment on a single spin (qubit 1 for example) subject to a constant magnetic field along the z-axis and a sinusoidal field along the x-axis reads [49]

$$\mathrm{i}\frac{\partial}{\partial t}|\Phi(t)\rangle = -\left(h_1^z S_1^z + \tilde{h}_1^x S_1^x \sin\omega t\right)|\Phi(t)\rangle, \tag{29}$$

where $|\Phi\rangle = |\uparrow\rangle\langle\Phi|\uparrow\rangle + |\downarrow\rangle\langle\Phi|\downarrow\rangle$, $|\Phi(t=0)\rangle$ is the initial state of the two-state system and we have set the phase $\phi_x = 0$. Substituting $|\Phi(t)\rangle = \mathrm{e}^{\mathrm{i}th_1^z S_1^z}|\Psi(t)\rangle$ yields

$$\mathrm{i}\frac{\partial}{\partial t}|\Psi(t)\rangle = -\tilde{h}_1^x\,(S_1^x \sin\omega t\cos h_1^z t + S_1^y \sin\omega t \sin h_1^z t)\,|\Psi(t)\rangle. \tag{30}$$

At resonance, *i.e.* $\omega = h_1^z$, we find

$$\mathrm{i}\frac{\partial}{\partial t}|\Psi(t)\rangle = -\frac{\tilde{h}_1^x}{2}\;\;(S_1^y + S_1^x \sin 2\omega t - S_1^y \cos 2\omega t)\,|\Psi(t)\rangle. \tag{31}$$

Assuming that the effects of the higher harmonic terms (*i.e.* the terms in $\sin 2\omega t$ and $\cos 2\omega t$) are small [49], (31) is easily solved to give

$$|\Psi(t)\rangle \approx \mathrm{e}^{\mathrm{i}t\tilde{h}_1^x S_1^y/2}|\Psi(t=0)\rangle, \tag{32}$$

so that the overall action of a sinusoidal field pulse of duration τ can be written as

$$|\Phi(t+\tau)\rangle \approx \mathrm{e}^{\mathrm{i}\tau h_1^z S_1^z}\mathrm{e}^{\mathrm{i}\tau\tilde{h}_1^x S_1^y/2}|\Phi(t)\rangle. \tag{33}$$

Hence it follows that application of a sinusoidal field pulse of power $\tau\tilde{h}_1^x = \pi$ will have the effect of rotating spin 1 by an angle of $\pi/2$ about the y-axis, as is clear by comparing (7) with (33).

In deriving (33), higher harmonics have been neglected, as indicated by the "$\approx$" sign. Note that as already mentioned above, in our simulations we solve for the time evolution of the quantum system *exactly*, *i.e.* without making any approximation. Instead of applying sinusoidal fields along the x or y direction, one may also consider using sinusoidal fields that rotate in the x-y plane. This leads to the time-dependent Schrödinger equation [49]

$$\mathrm{i}\frac{\partial}{\partial t}|\Phi(t)\rangle = -\left[h_1^z S_1^z + \tilde{h}_1^x(S_1^x \sin\omega t + S_1^y \cos\omega t)\right]|\Phi(t)\rangle, \tag{34}$$

and instead of (33) we obtain

$$|\Phi(t+\tau)\rangle = \mathrm{e}^{\mathrm{i}\tau h_1^z S_1^z}\mathrm{e}^{\mathrm{i}\tau\tilde{h}_1^x S_1^y}|\Phi(t)\rangle. \tag{35}$$

A quantum computer contains at least two spins. If in experiments it is difficult to shield a particular spin from the sinusoidal field, an application of a sinusoidal field pulse affects not only the state of the resonant spin but changes the state of the other spins too (unless they are perfectly aligned along the z-axis). A general analytical, quantitative analysis of this many-body problem is rather difficult. We will study the limiting case in which the interaction between the spins has neglegible impact on the time evolution of the spins during application of the sinusoidal field pulse. As our numerical results (see below) demonstrate, this is the case that is relevant to the model system considered in the present paper and also to experiments [15–18].

We consider the two-spin system described by the time-dependent Schrödinger equation

$$\mathrm{i}\frac{\partial}{\partial t}|\Phi(t)\rangle = -\Big[h_1^z S_1^z + h_2^z S_2^z + \tilde{h}_1^x(S_1^x \sin\omega t + S_1^y \cos\omega t) + \tilde{h}_2^x(S_2^x \sin\omega t + S_2^y \cos\omega t)\Big]|\Phi(t)\rangle. \tag{36}$$

Substituting $|\Phi(t)\rangle = \mathrm{e}^{\mathrm{i}t\omega(S_1^z+S_2^z)}|\Psi(t)\rangle$ we obtain

$$\mathrm{i}\frac{\partial}{\partial t}|\Psi(t)\rangle = -\Big[(h_1^z-\omega)S_1^z + (h_2^z-\omega)S_2^z + \tilde{h}_1^x S_1^y + \tilde{h}_2^x S_2^y\Big]|\Psi(t)\rangle. \tag{37}$$

Our aim is to rotate spin 1 about an angle φ_1 without affecting the state of spin 2. This can be accomplished as follows. First we choose

$$\omega = h_1^z, \tag{38}$$

i.e. the frequency of the sinusoidal field pulse is tuned to the resonance frequency of spin 1. Then (37) can easily be integrated. The result is

$$|\Phi(t)\rangle = \mathrm{e}^{\mathrm{i}th_1^z(S_1^z+S_2^z)}\mathrm{e}^{\mathrm{i}t\tilde{h}_1^x S_1^y}\mathrm{e}^{\mathrm{i}t\mathbf{S}_2\cdot\mathbf{v}_{1,2}}|\Phi(0)\rangle, \tag{39}$$

where $\mathbf{v}_{n,m} \equiv (0, \tilde{h}_m^x, h_m^z - h_n^z)$.

The third factor in (39) rotates spin 2 around the vector $\mathbf{v}_{1,2}$. This factor can be expressed as

$$\mathrm{e}^{\mathrm{i}t\mathbf{S}_m\cdot\mathbf{v}_{n,m}} = \begin{pmatrix}1&0\\0&1\end{pmatrix}\cos\frac{t|\mathbf{v}_{n,m}|}{2} + \mathrm{i}|\mathbf{v}_{n,m}|^{-1}\begin{pmatrix}h_m^z-h_n^z & -\mathrm{i}h_m^x\\ \mathrm{i}h_m^x & h_n^z-h_m^z\end{pmatrix}\sin\frac{t|\mathbf{v}_{n,m}|}{2}, \tag{40}$$

and we see that the sinusoidal field pulse will not change the state of spin 2 if and only if the duration t_1 of the pulse satisfies

$$t_1|\mathbf{v}_{1,2}| = t_1\sqrt{(h_1^z-h_2^z)^2 + (\tilde{h}_2^x)^2} = 4\pi n_1, \tag{41}$$

where n_1 is a positive integer.

The second factor in (39) is a special case of (40). It is easy to see that if

$$t_1\tilde{h}_1^x = \varphi_1, \tag{42}$$

the second factor in (39) will rotate spin 1 about φ_1 around the y-axis. Therefore, if conditions (38), (41), and (42) are satisfied we can rotate spin 1 about φ_1 without affecting the state of spin 2, independent of the physical realization of the quantum computer. However, the first factor in (39) can still generate a phase shift. Although it drops out of the expression of the expectation value of the qubits, in general it has to be taken into account in a quantum computer calculation because this phase shift depends on the state of the spins. Adding the condition

$$t_1 h_1^z = 4\pi k_1, \tag{43}$$

where k_1 is a positive integer ($h_i^z > 0$ by definition), the first factor in (39) is always equal to one. Summarizing: If conditions (38, 41, 42), and (43) are satisfied we can rotate spin 1 about φ_1 without affecting the state of spin 2 and without introducing unwanted phase shifts.

A last constraint on the choice of the pulse parameters comes from the fact that

$$h_2^\alpha = \gamma h_1^\alpha \quad , \quad \tilde{h}_2^\alpha = \gamma \tilde{h}_1^\alpha \quad ; \quad \alpha = x, y, z. \tag{44}$$

Without loss of generality we will assume that $0 < \gamma < 1$, in concert with the choice of parameters (5).

Using constraint (44) and conditions (38, 41, 42), and (43) we have

$$(1-\gamma)^2 k_1^2 + \frac{\gamma^2}{4}\left(\frac{\varphi_1}{2\pi}\right)^2 = n_1^2, \tag{45}$$

and reversing the role of spin 1 and spin 2 we obtain

$$(1-\frac{1}{\gamma})^2 k_2^2 + \frac{1}{4\gamma^2}\left(\frac{\varphi_2}{2\pi}\right)^2 = n_2^2, \tag{46}$$

where k_1, k_2, n_1, and n_2 are positive integers. The angles of rotation about the y-axis can be chosen such that $0 \leq \varphi_1 \leq 2\pi$ and $0 \leq \varphi_2 \leq 2\pi$.

In general (45) or (46) have no solution but a good approximate solution may be obtained if γ is a rational number and k_1 and k_2 are large. Let $\gamma = N/M$ where N and M are integers satisfying $0 < N < M$. It follows that the representation $k_1 = kMN^2$ and $k_2 = kNM^2$ will generate sufficiently accurate solutions of (45) and (46) if the integer k is chosen such that

$$2kNM(M-N) \gg 1. \tag{47}$$

In terms of k, N, and M, the relevant physical quantities are then given by

$$\frac{t_1 h_1^z}{2\pi} = 2kMN^2 \quad , \quad \frac{\tilde{h}_1^x}{h_1^z} = \frac{1}{2kMN^2}\frac{\varphi_1}{2\pi}, \tag{48}$$

and

$$\frac{t_2 h_1^z}{2\pi} = 2kM^3 \quad , \quad \frac{\tilde{h}_2^x}{h_1^z} = \frac{1}{2kM^3}\frac{\varphi_2}{2\pi}\cdot \tag{49}$$

In our numerical experiments we use (48) and (49) to determine the duration of the sinusoidal field pulses for both the static and rotating sinusoidal fields. In the latter case the sinusoidal field pulses will be optimized in the sense that a pulse that rotates spin 1 (2) will hardly affect spin 2 (1) if k satisfies condition (47).

The assumption of a pure sinusoidal time dependence of the applied fields serves to simplify the analytical analysis given above. In experiment there is no good reason to stick to a simple time dependence of the pulses [50,51]. In general

$$\begin{aligned}\mathrm{i}\frac{\partial}{\partial t}|\Phi(t)\rangle = -\Big[&h_1^z S_1^z + h_2^z S_2^z + w(t)\tilde{h}_1^x \\ &\times(S_1^x \sin\omega t + S_1^y \cos\omega t) + w(t)\tilde{h}_2^x \\ &\times(S_2^x \sin\omega t + S_2^y \cos\omega t)\Big]\,|\Phi(t)\rangle,\end{aligned} \tag{50}$$

where $w(t)$ can be almost any waveform. For $\omega = h_1^z$, the formal solution of (50) reads

$$\begin{aligned}|\Phi(t)\rangle = \mathrm{e}^{\mathrm{i}t h_1^z (S_1^z + S_2^z)} &\exp\left(\mathrm{i}\int_0^t \mathrm{d}u\, w(u)\tilde{h}_1^x S_1^y\right) \\ &\times \exp_+\left\{\mathrm{i}\int_0^t \mathrm{d}u\,\left[(h_2^z - h_1^z)S_2^z + w(u)\tilde{h}_2^x S_2^y\right]\right\}|\Phi(0)\rangle,\end{aligned} \tag{51}$$

where $\exp_+\{\ldots\}$ denotes the time-ordered exponential. Expression (51) is an explicit Floquet-operator representation of the time-evolution operator [50]. The introduction of a general form of $w(t)$ replaces condition (41) by

$$\exp_+\left\{\mathrm{i}\int_0^{t_1} \mathrm{d}u\,\left[(h_2^z - h_1^z)S_2^z + w(u)\tilde{h}_2^x S_2^y\right]\right\} = 1, \tag{52}$$

and condition (42) becomes

$$\tilde{h}_1^x \int_0^{t_1} \mathrm{d}u\, w(u) = \varphi_1, \tag{53}$$

expressing the fact that the rotation angle φ_1 is determined by the power of the pulse only. Conditions (38) and (43) remain the same. There are many forms of $w(u)$ that will satisfy (53), so in this respect there is a lot of freedom in the choice of $w(u)$. The average-Hamiltonian theory can be used to find approximations to $w(u)$ but this approach is not exact (an average Hamiltonian only has a finite number of transitions) and has limited applicability [50]. Finding the form of $w(u)$ such that also (52) holds *exactly* is a complicated non-linear optimization problem, in particular when the quantum computer contains several qubits.

To summarize: If conditions (38, 41, 42), and (43) are satisfied we can rotate spin 1 about φ_1 without affecting the state of spin 2 and without introducing unwanted phase shifts. In practice we may replace (41) and (42), by (52) and (53) respectively.

4.2 Two-qubit operations: CNOT gate

As the $CNOT$ sequence (19) has been constructed on the basis of model (16), some modification is necessary to account for the fact that the two nuclear spins feel different static fields (see (3)). In general the Hamiltonian reads

$$H_{\rm NMR} = -JS_1^zS_2^z - h_1^zS_1^z - h_2^zS_2^z. \tag{54}$$

Comparison of (16) with (54) shows that the implemention of the $CNOT$ operation requires additional rotations:

$$\begin{aligned} CNOT &= \overline{Y}_2 \mathrm{e}^{-\mathrm{i}\tau(h_1^z-h)S_1^z} \mathrm{e}^{-\mathrm{i}\tau(h_2^z-h)S_2^z} \mathrm{e}^{-\mathrm{i}\tau H_{\rm NMR}} Y_2, \\ &= \overline{Y}_2 \mathrm{e}^{-\mathrm{i}\tau(h_1^z-h)S_1^z} \mathrm{e}^{-\mathrm{i}\tau(h_2^z-h)S_2^z} Y_2\overline{Y}_2 \mathrm{e}^{-\mathrm{i}\tau H_{\rm NMR}} Y_2, \end{aligned} \tag{55}$$

where we used the fact that $Y_2\overline{Y}_2 = 1$. The extra phase shifts in (55) can be expressed in terms of single-qubit operations. The identities

$$\mathrm{e}^{-\mathrm{i}\tau(h_1^z-h)S_1^z} = Y_1X_1'\overline{Y}_1 = \overline{X}_1Y_1'X_1, \tag{56}$$

$$\mathrm{e}^{-\mathrm{i}\tau(h_2^z-h)S_2^z} = Y_2X_2'\overline{Y}_2, \tag{57}$$

define the single-spin rotations X_1', Y_1', and X_2'.

As (56) and (57) suggest, there are many different, logically equivalent sequences that implement the $CNOT$ gate on an NMR quantum computer. We have chosen to limit ourselves to the respresentations

$$CNOT_1 = Y_1X_1'\overline{Y}_1X_2'\overline{Y}_2I'Y_2, \tag{58}$$

$$CNOT_2 = Y_1X_1'X_2'\overline{Y}_1\overline{Y}_2I'Y_2, \tag{59}$$

$$CNOT_3 = \overline{X}_1Y_1'X_2'\overline{Y}_2X_1I'Y_2, \tag{60}$$

where we introduced the symbol I' to represent the time evolution $\mathrm{e}^{-\mathrm{i}\tau H_{NMR}}$ with $\tau = -\pi/J$. On an ideal quantum computer the sequences (58–60) give identical results. On an NMR-like quantum computer they do not because on the physically realizable quantum computer $X_1X_2 \neq X_2X_1$ unless (47, 48) and (49) are satisfied. We will use sequences (58–60) to demonstrate the quantum programming problem.

4.3 Quantum algorithms

On a conventional computer an algorithm is a sequence of logical operations that defines a one-to-one relation between the input and output data. We expect that a conventional computer always returns the correct result, irrespective of the input. Also a quantum computer should have correct (input, output) relationships. In contrast to a conventional computer, a quantum computer accepts as input linear superpositions of basis states and can return superpositions as well. If a quantum gate correctly operates on each of the basis states, it will also do so on any general linear superposition unless the operation generates additional phase factors that depend on the input state. Of course this does not happen on an ideal quantum computer but on a realistic one it may. Above we have shown how to reduce unwanted phase errors that result from imperfections of the one- and two-qubit operations.

For each realization of quantum computer hardware, there is a one-to-one correspondence between the quantum algorithm and the unitary matrix that transforms the superposition accordingly. A quantum algorithm will operate correctly under *all* circumstances if the *whole* unitary matrix representing the quantum algorithm is a good approximation to the ideal one. In other words, the magnitude and the phase of *all* matrix elements should be close to their ideal values. It is not sufficient to have for example two different $CNOT$ gates that operate correctly by themselves: Also the relative phases that they produce should match. For n qubits there are $2^n(2^n-1)$ complex numbers that specify the unitary matrix corresponding to a quantum algorithm. All these numbers should be close to their ideal values, otherwise the quantum algorithm is bound to produce wrong answers.

In general on a physical quantum computer, $CNOT^2 \neq 1$ and hence $(CNOT)^5$ in (20) and (21) is not *exactly* equal to the $CNOT$ operation. The effect of the physical implemention of a quantum computer on the logical operation of a quantum algorithm will be most clear if we can distinguish errors due to faulty input data from those that are intrinsic to the physics of the qubits. Therefore we will provide the exact input state to the quantum algorithm and compare the result returned by the quantum algorithm with the exact answer. This procedure simplifies the analysis but does not touch the essence of the matter. We prepare $|b_1b_2\rangle$ and $|singlet\rangle = (|01\rangle - |10\rangle)/\sqrt{2}$ by starting from the state $|00\rangle$ and by performing exact rotations of the spins.

In the case of Grover's database search algorithm, the representation of G in terms of the time evolution of (54) reads

$$\begin{aligned} G &= \mathrm{e}^{-\mathrm{i}\pi S_1S_2} = \mathrm{e}^{-\mathrm{i}\tau h_1^zS_1^z} \mathrm{e}^{-\mathrm{i}\tau h_2^zS_2^z} \mathrm{e}^{-\mathrm{i}\tau H_{\rm NMR}} \\ &= Y_2X_2''\overline{Y}_2Y_1X_1''\overline{Y}_1 \mathrm{e}^{-\mathrm{i}\tau H_{\rm NMR}}, \end{aligned} \tag{61}$$

where $\tau = -\pi/J$. This choice of τ also fixes the angles of the rotations, and through relations (48) and (49) also all parameters of the operations X_1'' and X_2''.

4.4 Generic features

For our choice (5) of the model parameters, $\gamma = 1/4$ such that $N = 1$ and $M = 4$. In general there is no good reason to expect that γ will be a ratio of two small integers but of course it may be approximated to any desired precision by a rational number. Let us consider the hypothetical case ($N = 11$, $M = 40$) such that $\gamma = 11/40 = 0.275$. Then (47) reads $25520k \gg 1$ so that the choice $k = 1$ already yields an accurate solution to (45) and (46). However as $t_1h_1^z/2\pi = 9680$ and $t_2h_1^z/2\pi = 128\,000$, rather long (in units of 2 ns) sinusoidal field pulses are required to perform these nearly ideal, single-qubit operations. As this example shows, the duration of the pulses that implement accurate single-qubit operations will be determined

Table 2. Model parameters for the elementary operations on the ideal quantum computer. Parameters of model (3) that do not appear in this table are zero, except for the interaction $J = -0.43 \times 10^{-6}$. The time-dependent Schrödinger equation is solved using a time step $\delta/2\pi = 1$.

	$\tau/2\pi$	h_1^x	h_2^x	h_1^y	h_2^y	h_1^z	h_2^z
X_1	0.25	1	0	0	0	0	0
X_2	0.25	0	1	0	0	0	0
Y_1	0.25	0	0	1	0	0	0
Y_2	0.25	0	0	0	1	0	0
X_1'	1	−0.4477	0	0	0	0	0
X_2'	1	0	−1.4244	0	0	0	0
Y_1'	1	0	0	0.4477	0	0	0
X_1''	1	−0.6977	0	0	0	0	0
X_2''	1	0	−1.6744	0	0	0	0
I	$-1/2J$	0	0	0	0	$-J/2$	$-J/2$
I'	$-1/2J$	0	0	0	0	1	0.25
G	$-1/2J$	0	0	0	0	1	0.25

by the representation of γ as a ratio of two (small) integers.

Another complication of using these rather long pulses is that non-resonant effects due to the presence of the (small) spin-spin interaction J will also affect the accuracy of the single-qubit operations. For instance if $t_2 h_1^z/2\pi = 128\,000$ then, for the choice of model parameters (5), $t_2|J|/2\pi = 0.05504$. Hence the phase shifts induced by the spin-spin interaction may no longer be neglected. Although the non-resonant effects due to J can also be reduced by proper choice of the pulse shape, duration and strength [13] and/or at the cost of additional pulses [52], this complication renders the non-linear optimization problem for finding optimal pulse sequences (much) more complicated.

At first sight, many of the problems we have discussed so far may seem to disappear if the physical quantum computer hardware is such that single physical qubits can be addressed individually. This certainly rules out NMR-based quantum computers but on quantum-dots or Josephson-coupling based quantum computers this may well be possible. However, as we now argue, the main problem, namely the accumulation of undesirable phases is likely to be present in these quantum computers too.

To see this it suffices to consider a few, say four, qubits that have their own (free) oscillation frequency. Taking the frequency of the first qubit as a reference the ratio of the remaining frequencies can (approximately) be represented by the ratios N_j/M_j, for $j = 2,\ 3,\ 4$. Let us now assume that we carry out an operation that addresses qubit 1 and 2 only. Let us call τ_{12} the time it takes to carry out the operation of qubits 1 and 2. Furthermore we assume that this operation on these two qubits yields the *exact* result, *i.e.* errors of the type discussed above are *absent*.

On an ideal quantum computer the result of the operation would be exact for the four-qubit state too but on a physical QC this is not necessarily so. In fact, unless $\tau_{12}N_3/M_3$ and $\tau_{12}N_4/M_4$ are an *exact* multiple of 4π and the input state is a linear combination of the 16 possible basis states, the operation will change the phases of each of the basis states and systematic phase errors will be generated. Clearly this conclusion does not depend on the peculiarities of the NMR technique: It holds in general. The only way to control the phase-error problem is to manufacture the qubits with such a precision that their free oscillation frequencies are commensurate with each other.

5 Simulation

5.1 Model parameters

The parameters of model (3) for which $\mathrm{e}^{-\mathrm{i}\tau H}$ implements the elementary operations of the ideal quantum computer are listed in Table 2. On an NMR-like quantum computer, the one-qubit operations can be implemented by applying sinusoidal field pulses, as explained above. The model parameters for the fixed and rotating sinusoidal fields are determined according to the theory outlined above. We use the integer k to compute all free parameters and the subscript $s = 2kMN^2$ to label the results of the quantum computer calculation. For reference we present the set of parameters corresponding to $k = 1$ for quantum computers using fixed and rotating sinusoidal fields in Tables 3 and 4 respectively. Multiplying s (the duration of the sinusoidal field pulse) with the unit of time (2 ns) shows that in our simulations, single-qubit operations are implemented by using short SF pulses that are, in NMR terminology, non-selective and hard. Note that in contrast to the analytical treatment in Section 4.1, in all our simulations the interaction J is non-zero (except when mentioned explicitly).

The two-qubit operation I' can be implemented by letting the system evolve in time according to Hamiltonian H_{NMR}, given by (54). I' is the same for both an ideal or NMR-like quantum computer. Note that the condition $\tau J = -\pi$ yields $\tau/2\pi = 1162790.6977$, a fairly large number (compared to our reference $h_1^z = 1$, see (5)).

Table 3. Model parameters of single-qubit operations on an NMR quantum computer for the case ($k = 1$, $N = 1$, $M = 4$), see (48) and (49). Parameters of model (3) that do not appear in this table are zero, except for the interaction $J = -0.43 \times 10^{-6}$ and the constant magnetic fields $h_1^z = 1$ and $h_2^z = 0.25$. The time-dependent Schrödinger equation is solved using a time step $\delta/2\pi = 0.01$.

	$\tau/2\pi$	ω	$\tilde{h}_1^x$	$\tilde{h}_2^x$	$\tilde{h}_1^y$	$\tilde{h}_2^y$
X_1	8	1.00	0	0	−0.0625000	−0.0156250
X_2	128	0.25	0	0	−0.0156250	−0.0039063
Y_1	8	1.00	0.0625000	0.0156250	0	0
Y_2	128	0.25	0.0156250	0.0039063	0	0
X_1'	8	1.00	0	0	0.1119186	0.0279796
X_2'	128	0.25	0	0	0.0890262	0.0222565
Y_1'	8	1.00	−0.1119186	−0.0279796	0	0
X_1''	8	1.00	0	0	0.1744186	0.0436046
X_2''	128	0.25	0	0	0.1046512	0.0261628

Table 4. Model parameters of single-qubit operations on an NMR quantum computer using rotating sinusoidal fields for the case ($k = 1$, $N = 1$, $M = 4$), see (48) and (49). Parameters of model (3) that do not appear in this table are zero, except for the interaction $J = -0.43 \times 10^{-6}$ and the constant magnetic fields $h_1^z = 1$ and $h_2^z = 0.25$. The time-dependent Schrödinger equation is solved using a time step $\delta/2\pi = 0.01$.

	$\tau/2\pi$	ω	$\tilde{h}_1^x$	$\tilde{h}_2^x$	φ_x	$\tilde{h}_1^y$	$\tilde{h}_2^y$	φ_y
X_1	8	1.00	−0.0312500	−0.0078125	$-\pi/2$	−0.0312500	−0.0078125	0
X_2	128	0.25	−0.0078125	−0.0019531	$-\pi/2$	−0.0078125	−0.0019531	0
Y_1	8	1.00	0.0312500	0.0078125	0	0.0312500	0.0078125	$\pi/2$
Y_2	128	0.25	0.0078125	0.0019531	0	0.0078125	0.0019531	$\pi/2$
X_1'	8	1.00	0.0559593	0.0139898	$-\pi/2$	0.0559593	0.0139898	0
X_2'	128	0.25	0.0445131	0.0111283	$-\pi/2$	0.0445131	0.0111283	0
Y_1'	8	1.00	−0.0559593	−0.0139898	0	−0.0559593	−0.0139898	$\pi/2$
X_1''	8	1.00	0.0872093	0.0218023	$-\pi/2$	0.0872093	0.0218023	0
X_2''	128	0.25	0.0523256	0.0130914	$-\pi/2$	0.0523256	0.0130914	0

5.2 Results

As a standard test we execute all sequences on an implementation of the ideal quantum computer (see Tab. 2). They all give the exact answers (results not shown). It is also necessary to rule out that the numerical results depend on the time step δ used to solve the time-dependent Schrödinger equation. The numerical error of the product formula used by quantum computer is proportional to δ^2 [44–46]. It goes down by a factor of about one hundred if we reduce the time step by a factor of 10. Within the two-digit accuracy used to present our data, there is no difference between the results for $\delta = 0.01$ and $\delta = 0.001$. Hence we can be confident that we are solving the time-dependent Schrödinger equation with a sufficiently high accuracy.

In Table 5 we present simulation results for quantum algorithms, QA_1 and QA_2 defined by (20) and (21) respectively. It is clear that even the least accurate implementation ($s = 8$, $k = 1$) nicely reproduces the correct answers if the input corresponds to one of the four basis states. The corresponding entries for QA_2 seem to suggest that $CNOT_1$ is working well for $s = 8$. However the result for $s = 16$ ($k = 2$) shows that the apparently good result for $s = 8$ is accidental, as we might have expected on the basis of criterion (47) (which in this case reads $24 \gg 1$). In agreement with the theoretical analysis of Section 4 the results converge to the exact ones for sufficiently large k. The pulses used in these simulations are so short that the presence of a non-zero J has a neglegible effect on the single-qubit pulses. For small s, the difference in the accuracy with which QA_1 and QA_2 give the correct answer clearly shows that in order for a quantum algorithm to work properly, it is not sufficient to show that it correctly operates only when the input corresponds to one of the basis states.

In the regime where phase errors are significant the quantum algorithms exhibit the quantum programming problem. This is exemplified in Tables 6 and 7 where we show the results of using $CNOT_2$ and $CNOT_3$ instead of $CNOT_1$. For $k < 32$ there is a clear signature of the quantum programming problem: These logically identical quantum algorithms are sensitive to the order in which the single-qubit operations are carried out.

From the viewpoint of computation it is advantageous to use elementary operations that are as short as possible. If $\gamma = h_2^z/h_1^z = 0.25 = 1/4$ *exactly* (to machine precision) Tables 5–7 show that it is indeed possible to work with relatively short, intense pulses (intense in the

Table 5. Expectation values of the two qubits as obtained by performing a sequence of five $CNOT$ operations on a quantum computer that uses rotating sinusoidal fields to manipulate individual qubits. The initial states $|10\rangle$, $|01\rangle$, $|11\rangle$, and $|singlet\rangle = (|01\rangle - |10\rangle)/\sqrt{2}$ have been prepared by starting from the state $|00\rangle$ and performing exact rotations of the spins. The $CNOT$ operations on the singlet state are followed by a $\pi/2$ rotation of spin 1 to yield a non-zero value of qubit 1. The subscripts in a_s and b_s refer to the time $s = \tau/2\pi = 2kMN^2$ and determine the duration and strength of the sinusoidal field pulses through relations (48) and (49), see Table 4 for the example of the case $s = 8$. The $CNOT$ operation itself was implemented by applying sequence $CNOT_1$ given by (58). On an ideal quantum computer, $CNOT^4$ is the identity operation. The results obtained on an ideal quantum computer are given by a and b.

Operation	a	b	a_8	b_8	a_{16}	b_{16}	a_{32}	b_{32}	a_{64}	b_{64}	a_{256}	b_{256}
$(CNOT_1)^5\|00\rangle$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$(CNOT_1)^5\|10\rangle$	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$(CNOT_1)^5\|01\rangle$	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00
$(CNOT_1)^5\|11\rangle$	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
$Y_1(CNOT_1)^5\|singlet\rangle$	1.00	1.00	0.90	1.00	0.03	1.00	0.58	1.00	0.88	1.00	0.99	1.00

Table 6. Same as Table 5 except that instead of $CNOT_1$ sequence $CNOT_2$ given by (59) was used to perform the quantum computation.

Operation	a	b	a_8	b_8	a_{16}	b_{16}	a_{32}	b_{32}	a_{64}	b_{64}	a_{256}	b_{256}
$(CNOT_2)^5\|00\rangle$	0.00	0.00	0.24	0.76	0.50	0.26	0.20	0.07	0.06	0.02	0.00	0.00
$(CNOT_2)^5\|10\rangle$	1.00	1.00	0.76	0.24	0.50	0.74	0.80	0.93	0.95	0.98	1.00	1.00
$(CNOT_2)^5\|01\rangle$	0.00	1.00	0.24	0.24	0.51	0.74	0.20	0.93	0.06	0.98	0.00	1.00
$(CNOT_2)^5\|11\rangle$	1.00	0.00	0.76	0.76	0.50	0.26	0.80	0.07	0.95	0.02	1.00	0.00
$Y_1(CNOT_2)^5\|singlet\rangle$	1.00	1.00	0.98	0.24	0.95	0.74	0.98	0.93	0.99	0.98	1.00	1.00

Table 7. Same as Table 5 except that instead of $CNOT_1$ sequence $CNOT_3$ given by (60) was used to perform the quantum computation.

Operation	a	b	a_8	b_8	a_{16}	b_{16}	a_{32}	b_{32}	a_{64}	b_{64}	a_{256}	b_{256}
$(CNOT_3)^5\|00\rangle$	0.00	0.00	0.23	0.76	0.50	0.26	0.20	0.07	0.06	0.02	0.00	0.00
$(CNOT_3)^5\|10\rangle$	1.00	1.00	0.77	0.24	0.50	0.74	0.80	0.93	0.95	0.98	1.00	1.00
$(CNOT_3)^5\|01\rangle$	0.00	1.00	0.23	0.24	0.51	0.74	0.20	0.93	0.06	0.98	0.00	1.00
$(CNOT_3)^5\|11\rangle$	1.00	0.00	0.77	0.76	0.50	0.26	0.80	0.07	0.95	0.02	1.00	0.00
$Y_1(CNOT_3)^5\|singlet\rangle$	1.00	1.00	0.79	0.24	0.55	0.74	0.82	0.93	0.95	0.98	1.00	1.00

context of NMR experiments). However, in reality it will not be easy to fabricate qubits such that the ratios of their natural frequencies h_i^z can be written as a ratio of two small integers. As an example we consider the case where $\gamma = h_2^z/h_1^z = 0.275 = 11/40$. For $N = 11$ and $M = 40$, the theory outlined above yields as a condition for accurate operation $25520k \gg 1$. Hence it is sufficient to take $k = 1$. Then $t_1/2\pi = 9680$ and $t_2/2\pi = 128\,000$ and the single-qubit pulses are weak and long compared to those used earlier [53]. Thus we may expect that the single-qubit operations are close to perfect. Table 8 shows the numerical results for the various implementations of the $CNOT$ gate. If the input state is a simple basis state (first four rows of Tab. 8) the output state agrees with the exact result within 20%. If the input state is a singlet (fifth row of Tab. 8) the output is completely wrong. The deviations from the exact results are solely due to the presence of the interaction J (which may be reduced by further optimization of the pulses [13,52]). Turning off this interaction during the single-qubit operations yields the exact results for all cases, also those that operate on the singlet state (results not shown). These results confirm that it is possible to optimize the pulses such that the quantum computer operates correctly. In general, as discussed above, this can only be achieved by a tremendous precision in the fabrication process of the qubits or at the expense of rather slow real-time performance.

The results presented in Tables 5–8 have been obtained using rotating sinusoidal fields. As explained above, in this case a single-qubit operation on qubit j *exactly* rotates qubit j about the specified angle (but perturbs the state of the other spin). In Table 9 we present simulation results obtained by using sinusoidal field in the x or y direction only. Then the single-spin rotation on spin j no longer corresponds to the exact one. Nevertheless, as Table 9 shows, for sufficiently large s the results nicely converge to the correct anwers. Apparently, for a quantum algorithm to compute correctly, it is more important to have the phase errors under control than to perform very accurate single-spin rotations.

The very essence of quantum algorithms is the use of entangled states at some stage of the calculations. It is at this point that the quantum algorithm is most sensitive to

Table 8. Same as Table 5 except that $N = 11$, $M = 40$, $\gamma = N/M = h_2^z/h_1^z = 0.275$, $\tau/2\pi = 9680$ for X_1, $\tau/2\pi = 128\,000$ for X_2 etc. As time is measured in units of 2 ns, weak but rather long sinusoidal field pulses are used to perform the single-qubit operations. For the first four rows, deviations from the exact result are due to the presence of the non-resonant effects of the interaction J. Turning off this interaction during the single-qubit operations yields the exact results for all entries (results not shown).

	$C = CNOT$		$C = CNOT_1$		$C = CNOT_2$		$C = CNOT_3$	
Operation	a	b	a_1	b_1	a_1	b_1	a_1	b_1
$C^5\|00\rangle$	0.00	0.00	0.00	0.19	0.00	0.19	0.00	0.19
$C^5\|10\rangle$	1.00	1.00	1.00	0.89	1.00	0.89	1.00	0.89
$C^5\|01\rangle$	0.00	1.00	0.01	0.81	0.01	0.81	0.01	0.81
$C^5\|11\rangle$	1.00	0.00	1.00	0.11	1.00	0.11	1.00	0.11
$Y_1C^5\|singlet\rangle$	1.00	1.00	0.01	0.85	0.01	0.85	0.01	0.84

Table 9. Same as Table 5 except that instead of rotating sinusoidal fields, sinusoidal fields along either the x or y-axis were used to manipulate individual qubits. See Table 3 for the example of the set of model parameters for $s = 8$.

Operation	a	b	a_8	b_8	a_{16}	b_{16}	a_{32}	b_{32}	a_{64}	b_{64}	a_{256}	b_{256}
$(CNOT_1)^5\|00\rangle$	0.00	0.00	0.00	0.03	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
$(CNOT_1)^5\|10\rangle$	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$(CNOT_1)^5\|01\rangle$	0.00	1.00	0.00	0.97	0.00	0.99	0.00	1.00	0.00	1.00	0.00	1.00
$(CNOT_1)^5\|11\rangle$	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
$Y_1(CNOT_1)^5\|singlet\rangle$	1.00	1.00	0.02	0.98	0.45	1.00	0.17	1.00	0.70	1.00	0.98	1.00

Table 10. Expectation values of the two qubits as obtained by running Grover's database search algorithm on a quantum computer that uses rotating sinusoidal fields to manipulate individual qubits. The subscripts in a_s and b_s refer to the time $s = \tau/2\pi = 2kMN^2$ and determine the duration and strength of the sinusoidal field pulses through relations (48) and (49), see Table 4 for the example of the case $s = 8$. The results obtained on an ideal quantum computer are given by a and b.

Item position	a	b	a_8	b_8	a_{16}	b_{16}	a_{32}	b_{32}	a_{64}	b_{64}	a_{256}	b_{256}
0	0.00	0.00	0.48	0.53	0.15	0.16	0.04	0.04	0.01	0.01	0.00	0.00
1	1.00	0.00	0.52	0.50	0.85	0.15	0.96	0.04	0.99	0.01	1.00	0.00
2	0.00	1.00	0.55	0.48	0.15	0.84	0.04	0.96	0.01	0.99	0.00	1.00
3	1.00	1.00	0.45	0.50	0.85	0.85	0.96	0.96	0.99	0.99	1.00	1.00

Table 11. Same as Table 10 except that instead of rotating sinusoidal fields, sinusoidal fields along either the x or y-axis were used to manipulate individual qubits. See Table 3 for the example of the set of model parameters for $s = 8$.

Item position	a	b	a_8	b_8	a_{16}	b_{16}	a_{32}	b_{32}	a_{64}	b_{64}	a_{256}	b_{256}
0	0.00	0.00	0.92	0.91	0.39	0.35	0.11	0.10	0.03	0.03	0.00	0.00
1	1.00	0.00	0.09	0.91	0.61	0.36	0.89	0.10	0.97	0.03	1.00	0.00
2	0.00	1.00	0.95	0.10	0.36	0.65	0.10	0.90	0.03	0.98	0.00	1.00
3	1.00	1.00	0.05	0.09	0.64	0.64	0.90	0.90	0.97	0.97	1.00	1.00

(accumulated) phase errors. As another illustration of this phenomenon, we present in Tables 10 and 11 some typical results obtained by executing Grover's database search algorithm on the same model quantum computers as those used in the examples discussed above. We find that reasonably good answers are obtained if $s \geq 32$, in concert with the observations based on quantum algorithms QA_1 and QA_2.

The results discussed above show effects of imperfections in the physical implementation of single-qubit operations. Thereby we assumed that J, and the static applied fields h_1^z and h_2^z are fixed in time and known to very high precision. The Ising-model time evolution was used to perform two-qubit operations, leaving only the duration of this operation as a possible source for causing errors. In Table 12 we give examples of the extreme sensitivity of a quantum algorithm to the precision with which the parameters have to be specified. Essentially we repeated the calculation of Table 5 for $s = 256$ but on purpose we made an error in the specification of the duration of I'. As

Table 12. Same as Table 5 except for a change in the duration of the operation I'. $(a_{256}^{(1)},b_{256}^{(1)})$: $\tau/2\pi = 1162790.4977$; $(a_{256}^{(2)},b_{256}^{(2)})$: $\tau/2\pi = 1162790.5977$; $(a_{256}^{(3)},b_{256}^{(3)})$: $\tau/2\pi = 1162790.6977$ (correct value); $(a_{256}^{(4)},b_{256}^{(4)})$: $\tau/2\pi = 1162790.7977$; $(a_{256}^{(5)},b_{256}^{(5)})$: $\tau/2\pi = 1162790.8977$.

Operation	a	b	$a_{256}^{(1)}$	$b_{256}^{(1)}$	$a_{256}^{(2)}$	$b_{256}^{(2)}$	$a_{256}^{(3)}$	$b_{256}^{(3)}$	$a_{256}^{(4)}$	$b_{256}^{(4)}$	$a_{256}^{(5)}$	$b_{256}^{(5)}$
$(CNOT_1)^5\|00\rangle$	0.00	0.00	0.00	0.52	0.00	0.16	0.00	0.00	0.00	0.13	0.00	0.48
$(CNOT_1)^5\|10\rangle$	1.00	1.00	1.00	0.48	1.00	0.87	1.00	1.00	1.00	0.84	1.00	0.48
$(CNOT_1)^5\|01\rangle$	0.00	1.00	0.00	0.48	0.00	0.84	0.00	1.00	0.00	0.87	0.00	0.52
$(CNOT_1)^5\|11\rangle$	1.00	0.00	1.00	0.52	1.00	0.13	1.00	0.00	1.00	0.16	1.00	0.52
$Y_1(CNOT_1)^5\|singlet\rangle$	1.00	1.00	0.99	0.50	0.09	0.85	0.99	1.00	0.01	0.85	0.99	0.50

Table 12 shows, an error in the 8th digit can have a devastating effect on the outcome of the quantum computer calculation. This again is just another manifestation of the quantum programming problem but not really a surprise: During the application of I' the spins rotate around the z-axis with their resonance frequencies h_1^z and h_2^z. A small deviation in $\tau/2\pi$ from its ideal value produces phase errors. Note however that the integer part of $\tau/2\pi$ is also essential to perform the correct conditional phase shift. Therefore, in practice it is necessary to specify the duration of the time evolution I' to at least 8 digits (for the case $|J|/h_1^z \approx 10^{-6}$).

6 Conclusions

On a physically realizable, non-ideal quantum computer, operations that manipulate one particular qubit also affect the state of other qubits. This may cause unwanted deviations from the ideal motion of the total system and lead to practical problems of programming quantum computers: An implementation of a quantum computation that works well on one quantum computer may fail on others.

We have classified the various physical sources that lead to deviations. The most obvious one originates from the fact that other spins cannot be kept still during an operation on one particular spin. If these spins do not return to their original state when this operation is over, the quantum computation is unlikely to give correct answers [14].

Proper optimization of the parameters that control the single-qubit operations can largely eliminate this source of errors. However, even if the operation gives almost exact results for all basis states, the operation is not necessarily perfect. That is, the operation generally yields a phase factor which depends on the input states. Therefore, when such an operation is applied on a linear combination of the basis states, the relative phases of the basis states change, resulting in incorrect quantum computation. This is a second source for deviations from correct quantum computer operation.

We have derived additional conditions on the parameters that control the single-qubit operations and have obtained the conditions for reliable quantum computation. Unfortunately, these conditions cannot be satisfied simultaneously. However they can be satisfied to any precision by increasing the duration of the single-qubit operations. Using the controlled-NOT gate and Grover's search algorithm as examples, we have given concrete demonstrations of how the above mentioned problems arise and how they can be solved.

At this moment, we do not know how to stabilize the quantum computation by controlling the evolution of the state of a closed quantum system. In a classical computer the presence of dissipation enables reliable computation. However, dissipation seems detrimental for quantum computer operation. Therefore, at this moment, the only option is to perform each operation as perfect as possible. The present paper shows how this may be done.

The condition on the commensurability of the precession frequencies of the individual qubits leads to an increase of the execution time of single-qubit operations. Unless the precession frequencies of the qubits are the same to great precision, the execution time will grow rapidly with the number of qubits and substantially limit the speed of quantum computation. Therefore new techniques have to be developed to compensate for this loss in efficiency. Quantum error correction schemes that work well on an ideal quantum computer require many extra qubits and many additional operations to detect and correct errors. On a physical quantum computer however, the error-correction qubits will suffer from the same deficiencies as those exposed in this paper. Possibly, the clever use of dissipation processes may help to perform automatic error correction [54]. All this puts considerable demands on the technology to fabricate qubits.

In recent NMR experiments Vandersypen *et al.* demonstrated that it is possible to carry out order finding and number factoring by controlling the quantum dynamics of upto seven nuclear spins [55,56]. The experimental results do not contradict our findings, on the contrary. Indeed the success of these NMR experiments crucially depends on how well the pulses can be optimized to suppress systematic errors. As a matter of fact, a very large amount of (computational) work in doing these experiments goes into the design of the pulse sequences and into finding tricks to eliminate systematic (not random) errors [57]. Also in real time, these experimental quantum calculations are very slow (order of 100 ms), also in concert with our findings. The observation that there is a surprisingly large degree of cancellations of systematic errors is one of the main results of these experiments [57]. There clearly is a need to understand where these cancellations come from. The exact numerical analysis of the quantum dynamics of

the 2-qubit system presented in this paper provides insight into this remarkable phenomenon. It remains a great challenge to demonstrate that a quantum computer of many qubits can perform a genuine computation in less real time than a conventional computer.

Support from the Dutch "Stichting Nationale Computer Faciliteiten (NCF)" and the Grant-in-Aid from the Japanese Ministry of Education, Science, Sports and Culture is gratefully acknowledged.

References

1. P. Shor, in: *Proc. 35th Annu. Symp. Foundations of Computer Science*, edited by S. Goldwasser (IEEE Computer Soc., Los Alamitos CA, 1994) p. 124.
2. L.K. Grover, Phys. Rev. Lett. **79**, 325 (1997).
3. D.P. DiVincenzo, Science **270**, 255 (1995).
4. A. Ekert, R. Jozsa, Rev. Mod. Phys. **68**, 733 (1996).
5. V. Vedral, M. Plenio, Progr. Quantum Electronics **22**, 1 (1998).
6. P.W. Shor, SIAM Rev. **41**, 303 (1999).
7. R.P. Feynman, Int. J. Theor. Phys. **21**, 467 (1982).
8. M. Nielsen, I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).
9. C. Miquel, J.P. Paz, W.H. Zurek, Phys. Rev. Lett. **78**, 3871 (1997).
10. B. Pablo-Norman, M. Ruiz-Altaba, Phys. Rev. A **61**, 012301 (2000).
11. Gui Lu Long, Yan Song Li, Wei Lin Zhang, Chang Cun Tu, Phys. Rev. A **61**, 042305 (2000).
12. G.P. Berman, G.D. Doolen, G.V. López, V.I. Tsifrinovich, Phys. Rev. A **61**, 062305 (2000).
13. G.P. Berman, G.D. Doolen, G.V. López, V.I. Tsifrinovich, Phys. Rev. A **61**, 042307 (2000).
14. H. De Raedt, A. Hams, K. Michielsen, S. Miyashita, K. Saito, J. Phys. Soc. Jpn (Supp.) **69**, 401 (2000).
15. J.A. Jones, M. Mosca, J. Chem. Phys. **109**, 1648 (1998).
16. J.A. Jones, M. Mosca, R.H. Hansen, Nature **393**, 344 (1998).
17. I.L. Chuang, L.M.K. Vandersypen, Xinlan Zhou, D.W. Leung, S. Lloyd, Nature **393**, 143 (1998).
18. I.L. Chuang, N. Gershenfeld, M. Kubinec, Phys. Rev. Lett. **80**, 3408 (1998).
19. R. Marx, A.F. Fahmy, J.M. Meyers, W. Bernel, S.J. Glaser, Phys. Rev. A **62**, 012310 (2000).
20. E. Knill, R. Laflamme, R. Martinez, C.-H. Tseng, Nature **404**, 368 (2000).
21. D.G. Cory, R. Laflamme, E. Knill, L. Viola, T.F. Havel, N. Boulant, G. Boutis, E. Fortunato, S. Lloyd, R. Martinez, C. Negrevergne, M. Pravia, Y. Sharf, G. Teklemariam, Y.S. Weinstein, W.H. Zurek, Fortschr. Phys. **48**, 875 (2000).
22. J.A. Jones, Fortschr. Phys. **48**, 909 (2000).
23. J.J. Cirac, P. Zoller, Phys. Rev. Lett. **74**, 4091 (1995).
24. C. Monroe, D.M. Meekhof, B.E. King, W.M. Itano, D.J. Wineland, Phys. Rev. Lett. **75**, 4714 (1995).
25. T. Sleator, H. Weinfurther, Phys. Rev. Lett. **74**, 4087 (1995).
26. P. Domokos, J.M. Raimond, M. Brune, S. Haroche, Phys. Rev. A **52**, 3554 (1995).
27. B.E. Kane, Nature **393**, 133 (1998).
28. A. Imamoğlo, D.D. Awschalom, G. Burkard, D.P. DiVincenzo, D. Loss, M. Sherwin, A. Small, Phys. Rev. Lett. **83**, 4204 (1999).
29. Y. Makhlin, G. Schön, A. Shnirman, Nature **398**, 305 (1999).
30. Y. Nakamura, Yu. A. Pashkin, J.S. Tsai, Nature **398**, 786 (1999).
31. G. Nogues, A. Rauschenbeutel, S. Osnaghi, M. Brune, J.M. Raimond, S. Haroche, Nature **400**, 239 (1999).
32. K. Mølmer, A. Sørensen, Phys. Rev. Lett. **82**, 1835 (1999).
33. A. Sørensen, K. Mølmer, Phys. Rev. Lett. **82**, 1971 (1999).
34. R. Fazio, G.M. Palma, J. Siewert, Phys. Rev. Lett. **83**, 5385 (1999).
35. T.P. Orlando, J.E. Mooij, Lin Tian, C.H. van der Wal, L.S. Levitov, S. Lloyd, J.J. Mazo, Phys. Rev. B **60**, 15398 (1999).
36. A. Blias, A. Zagoskin, Phys. Rev. A **61**, 042308 (2000).
37. M.C. de Oliveira, W.J. Munro, Phys. Rev. A **61**, 042309 (2000).
38. A. Barenco, C.H. Bennett, R. Cleve, D.P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J.A. Smolin, H. Weinfurter, Phys. Rev. A **52**, 3457 (1995).
39. S. Lloyd, Science **261**, 1569 (1993).
40. G.P. Berman, G.D. Doolen, D.D. Holm, V.I. Tsifrinovich, Phys. Lett. A **193**, 444 (1994).
41. The Quantum Computer Emulator (QCE) can be downloaded from `http://rugth30.phys.rug.nl/compphys/qce.htm`.
42. M. Suzuki, S. Miyashita, A. Kuroda, Prog. Theor. Phys. **58**, 1377 (1977).
43. M. Suzuki, Proc. Jpn Acad. Ser. B **69**, 161 (1993).
44. H. De Raedt, Comp. Phys. Rep. **7**, 1 (1987).
45. P. de Vries, H. De Raedt, Phys. Rev. B **47**, 7929 (1993).
46. H. De Raedt, K. Michielsen, Comp. Phys. **8**, 600 (1994).
47. H. De Raedt, A.H. Hams, K. Michielsen, K. De Raedt, Comp. Phys. Comm. **132**, 1 (2000).
48. T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Introduction to Algorithms* (MIT Press, Cambridge, 1994).
49. G. Baym, *Lectures on Quantum Mechanics* (W.A. Benjamin, Reading MA, 1974).
50. R. Ernst, G. Bodenhausen, A. Wokaun, *Principles of Nuclear Magnetic Resonance in One and Two Dimensions* (Oxford University Press, New York, 1987).
51. R. Freeman, *Spin Choreography* (Spektrum, Oxford, 1997).
52. N. Linde, B. Hervé, R.J. Carbajo, R. Freeman, Chem. Phys. Lett. **305**, 28 (1999).
53. In NMR language, for the physical system we took as an example, the intensity of the sinusoidal fields is of the order of a 10 kHz or less.
54. J.P. Barnes, W.S. Warren, Phys. Rev. Lett. **85**, 856 (2000).
55. L.M.K. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, R. Cleve, I.L. Chuang, Phys. Rev. Lett. **85**, 5452 (2000).
56. L.M.K. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, M.H. Sherwood, I.L. Chuang, Nature, **414**, 883 (2001).
57. L.M.K. Vandersypen, Ph.D. Thesis, Dept. of Electrical Engineering, Stanford University (2001).